UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
ÁREA DE INGENIERÍA
Postgrado en Sistemas de Información

Trabajo Especial de Grado de Especialista

Software Educativo para la Enseñanza de la Asignatura Estructura de Datos de la Carrera Ingeniería Informática de la Universidad Nacional Experimental del Táchira

Presentado por
David E. Ortiz V.
Para optar al título de
Especialista en Sistemas de Información

Asesor
María Nereyda Carrero

San Cristóbal, Junio 2009

San Cristóbal, Junio de 2009

Señores
Universidad Católica Andrés Bello
Dirección General de los Estudios de Postgrado
Postgrado en Sistemas de Información

Por la presente me permito comunicar que he sido el tutor del Trabajo de Grado de Especialización en Sistemas de Información titulado: “Software Educativo para la Enseñanza de la Asignatura Estructura de Datos de la Carrera Ingeniería Informática de la Universidad Nacional Experimental del Táchira” del estudiante David Eduardo Ortiz Vezga (C.I: 9.237.706) quien opta por el título de Especialista en Sistemas de Información.

Así mismo, hago constar que como tutor estoy conforme con el contenido presentado, por lo que cuenta con mi aprobación para ser inscrito como Trabajo Especial de Grado de Especialización.

Sin otro particular al cual hacer referencia, se despide cordialmente,

Ing. María Nereyda Carrero
C.I. 5.660.319

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
ESTUDIOS DE POSTGRADO
ÁREA DE INGENIERÍA
POSTGRADO EN SISTEMAS DE INFORMACIÓN

Software Educativo para la Enseñanza de la Asignatura Estructura de Datos de la Carrera Ingeniería Informática de la Universidad Nacional Experimental del Táchira

Autor: David E. Ortiz V.
Tutora: María N. Carrero.
Fecha: Junio, 2009.

RESUMEN

La informática y la educación se relacionan creando nuevas ramas de importancia para el desarrollo de los países. Es el caso de la automatización de actividades pedagógicas a través de nuevos recursos para el aprendizaje tales como tutoriales, programas de ejercitación, simuladores, test automatizados, entre otros. En la Universidad Nacional Experimental del Táchira UNET como en otras universidades del país, las políticas académicas apuntan hacia el uso de las TIC. Con base a un diagnóstico se determinó que la enseñanza de la asignatura estructura de datos del tercer semestre de ingeniería informática requiere de recursos didácticos que motiven al estudiante a aprender a su propio ritmo y que refuercen los contenidos vistos en clase presencial. El software educativo modalidad tutorial es una alternativa de solución a la problemática, desarrollado bajo un enfoque sistemático con metodología de *cascada modificada* para avanzar rápidamente entre las fases y obtener un modelo lo más acertado a las necesidades de los estudiantes y profesores de la asignatura. El proyecto abarca las fases de diagnóstico, diseño educativo y producción, hasta las especificaciones del software de diseño en el computador, costos y recursos de implementación. El software educativo está diseñado para exhibir contenidos teóricos, interactuar en animaciones empleadas para explicar procesos, evaluar y proponer ejercicios de práctica de programación.

Palabras Claves: software educativo, tutorial, estructura de datos, diseño educativo, multimedia.

Agradecimientos

A Dios todo poderoso, por iluminar mi camino y bendecirme cada día.

A mi esposa, padres, hermanos, sobrinos, familiares y amigos por estar a mi lado apoyándome cuando más los necesité.

A la Prof. María Nereyda Carrero, a los Profesores de la UCAB, UCAT y UNET por brindarme su apoyo, tiempo, trabajo y experiencia.

Y a todas las personas que de una y otra forma hicieron posible la culminación de este trabajo.

Mis más sinceras Gracias…

David

INDICE

LISTA DE TABLAS

LISTA DE GRÁFICOS

INTRODUCCIÓN

La evolución de las tecnologías de información ha estado condicionada por las necesidades de los usuarios. En este sentido las organizaciones han generado soluciones tecnológicas para mejorar la calidad del servicio orientadas al cliente, utilizando herramientas de planificación, diseño e implantación de sistemas con capacidades informáticas, en pro de controlar y mejorar las actividades que se realizan cotidianamente, así como también, responder a los requerimientos de los usuarios ávidos de novedosos medios para conseguir información y conocimiento.

En el área educativa, la informática, la computación y las comunicaciones en redes abren paso a nuevas formas de aprender. Hoy en día los alumnos y sus profesores están cada vez más interesados en la computación y su aplicación en la educación. Las instituciones de educación formal e informal: escuelas, liceos, universidades, academias, entre otros; hacen uso del computador como instrumento de clase, consulta e investigación.

Para obtener el mayor beneficio del uso del computador y sus capacidades es necesario delinear estrategias que generen nuevas formas de trabajo, que resulten atractivas, sencillas e interesantes a los usuarios, contribuyendo ha lograr más rápidamente el cumplimiento de sus objetivos que de la manera como se venia realizando con los métodos tradicionales.

Una aplicación muy popular del computador en la educación es el diseño y uso de programas de software en el aprendizaje, también denominado software educativo. Este producto está creado en función de las características y patrones de conducta de los aprendices para asimilar una determinada materia. La concepción del software educativo parte de la creación de un modelo instruccional que oriente el aprendizaje (diseñado por un instructor experto en contenido, currículo y en la didáctica de la especialidad); luego la aplicación de recursos tecnológicos para que el

aprendiz ejecute en el computador las actividades que lo conduzcan a asimilar el nuevo conocimiento.

La Universidad Nacional Experimental del Táchira (UNET), como otras universidades del país de educación presencial, dirige sus planes de acción hacia régimen semi presencial y no presencial. En este sentido las unidades curriculares de las diferentes carreras están llamadas a desarrollar proyectos de enseñanza programada y bajo el uso de las tecnologías de la información y comunicación. Considerando que la UNET cuenta con recursos tecnológicos, personal calificado e infraestructura para e-learning y b-learning se presenta la oportunidad para desarrollar un proyecto de este tipo, partiendo del diseño de un software educativo para la enseñanza de la asignatura Estructura de Datos de la carrera ingeniería en informática, dictada desde hace diez años en sistema de clases presenciales de teoría y prácticas de laboratorio.

Bajo este panorama de aprendizaje con el uso del computador, es importante resaltar que en ningún momento se pretende sustituir de manera definitiva el docente por el computador, sino más bien facilitar el trabajo de enseñanza en el aula de clase, incorporando sesiones de autoestudio con el uso de un software educativo. Para obtener buenos resultados en el rendimiento estudiantil, es necesario que el docente se involucre (compromiso), planifique, motive y oriente al educando, que éste se convierta en el líder en busca de mejores estrategias de enseñanza aprendizaje.

En el presente informe del trabajo especial de grado se presentan Cuatro capítulos. El capítulo I, reseña el planteamiento del problema, la justificación e importancia, el alcance, el objetivo general y los objetivos específicos para el desarrollo del proyecto. El capítulo II incluye los antecedentes, el marco organizacional, las bases teóricas y las bases legales que servirán de referencia para el progreso del proyecto. El capítulo III identifica el tipo de investigación, el diseño de la investigación, la población, las técnicas e instrumentos a utilizar en la recolección de datos y como se

van a analizar e interpretar los resultados. El capítulo IV aborda el desarrollo de la propuesta del software educativo bajo los conceptos, teorías y metodologías explicadas en los capítulos anteriores. Finalmente se presentan breves conclusiones y recomendaciones para la ejecución del proyecto, fabricación y uso del software de enseñanza.

CAPÍTULO I
EL PROBLEMA

Planteamiento del Problema

El desarrollo tecnológico ha revolucionado la forma de hacer las actividades en los procesos de las organizaciones, especialmente las del sector educativo. Hoy más que nunca docentes y educandos interactúan por medios que hasta hace pocos años eran inimaginables de concebir; se abre paso a nuevas técnicas y métodos para motivar, instruir, evaluar y orientar al estudiante. En tal sentido, el uso del computador con aplicaciones de software multimedia y la Internet representan en la educación una oportunidad para aumentar la calidad en el proceso enseñanza aprendizaje.

Barajas y Álvarez (Citado por Gross 2003) señalan que la excesiva demanda en las universidades, las presiones de los gobiernos para que aumenten el número de estudiantes admitidos en las carreras y el estancamiento del presupuesto asignado para su funcionamiento ha provocado una crisis. El reto es cada vez mayor, proveer conocimiento especializado a más estudiantes en áreas de constante evolución sin disminuir la calidad de los procesos. Es obvio que la capacidad de las instituciones de educación superior y el uso de métodos convencionales no es suficiente. Incorporar la tecnología de la información y comunicación en modalidades, métodos y técnicas de educación formal conlleva a solucionar el problema, al menos cuantitativamente.

La Universidad Nacional Experimental del Táchira (UNET) en el cumplimiento de los artículos 108 y 110 de la Constitución Nacional y el decreto 825 de la República Bolivariana de Venezuela, ha incluido en su planificación estratégica el uso de las Tecnologías de Información y Comunicación (TIC) para impartir las asignaturas, de carrera larga o

tecnológica, bajo modalidad semi presencial y para fortalecer la docencia directa ó las clases en la modalidad presencial.

La carrera de Ingeniería Informática se imparte bajo régimen presencial, no obstante algunas asignaturas con gran cantidad de alumnos repitientes se dictan bajo régimen semi presencial, para solucionar en gran parte el problema de horario e insuficiencia de espacio físico (aulas de clase). En esta situación es necesaria una efectiva planificación de las actividades de clase, la creación y mantenimiento de herramientas pedagógicas que faciliten el aprendizaje en las sesiones de auto estudio.

La asignatura Estructura de Datos, es dictada en modalidad presencial en aproximadamente cuatro secciones de 30 alumnos, según la demanda del semestre. Actualmente, los recursos pedagógicos utilizados en las clases teóricas son: libros, guías, pizarrón y retroproyector; en las clases prácticas el computador para la ejercitación. Los canales de comunicación más utilizados entre profesor y alumno son: las listas de correo electrónico y la página web (estática) de la asignatura, la cual muestra: los temas del programa, el cronograma de evaluaciones parciales y entregas de proyectos, los enunciados de problemas para descargar y ejercitar, entre otros.

En este contexto de educación tradicional, con un modelo educativo centrado en el profesor, se manifiesta poca atención en dos aspectos importantes: el alumno y como aprende el mismo. Hoy día existe variedad de instrumentos y medios para la enseñanza – aprendizaje, en muchos casos, basados en el uso del computador y las comunicaciones. Si no se incorporan adecuadamente al sistema de educación formal traería consecuencias contraproducentes a la universidad, tales como divorcio con la realidad, desmotivación, pérdida de inversión, altos tiempos y costos de producción, subutilización, entre otros.

Para lograr un mejor aprovechamiento de los recursos tecnológicos y de esta manera mejorar la calidad en la enseñanza aprendizaje de la asignatura Estructura de Datos sería interesante contar con un software que

le permita al estudiante abordar temas nuevos, aclarar dudas de clase y recuperar las sesiones a las cuales no pudo asistir, basado en el paradigma del autoaprendizaje y educación programada, en la cual el estudiante aprende a su propio ritmo, se compromete y es motivado a investigar sin depender de la presencia física del docente. En consecuencia, a futuro será posible dictar la asignatura en modalidad a distancia o bajo régimen semi presencial (disminución de horas de clases presenciales), evitando saturar los espacios físicos (aulas y laboratorios), la deserción escolar y la obsolescencia de los métodos y herramientas tradicionales para impartir clases.

Justificación

Con la aprobación e implantación de la propuesta de este software educativo se eleva la calidad de la enseñanza en la UNET, logrando así el propósito estratégico especificado en las políticas de la universidad y en el ordenamiento jurídico venezolano en materia de Ciencia y Tecnología. De contar con un recurso didáctico que facilite el proceso del auto aprendizaje, el docente no necesariamente deberá abarcar todos los contenidos en las clases presénciales, ya que podrá dejar algunos temas ó contenidos para ser trabajados por autoestudio con el software educativo.

El diseño de este software educativo contribuye a incrementar el grado de interactividad necesario para provocar en el estudiante una actitud más activa que pasiva, fomentada en las clases tradicionales, en función de los objetivos planificados en la asignatura y a competencias básicas de autocontrol, participación dinámica e investigación, entre otras importantes en la formación de un ingeniero.

También esta herramienta sería de gran ayuda para estudiantes del turno nocturno, personas con trabajos tiempo completo, que en algún

momento, por razones de fuerza mayor no pueden asistir a clase y tienen un tiempo muy reducido para recuperar y cumplir con las tareas propuestas.

Es importante señalar que el producto derivado de este trabajo podría beneficiar a cualquier persona interesada o institución de educación superior con disposición al autoaprendizaje a través de las Tics en las carreras: Ingeniería en Informática, Licenciatura en Computación, Ingeniería de Sistemas o carreras afín.

Otro beneficio que conlleva la creación y aplicación del software es la unificación de criterios para impartir clases y la garantía del proceso enseñanza aprendizaje, conducido por docentes en su mayoría ingenieros, de profesión técnica prestados a la educación, facilitando la aplicación de las teorías de aprendizaje y una metodología instruccional bien sustentada.

Finalmente, el presente trabajo constituye un elemento motivacional para otras asignaturas similares que requieran de un proyecto de creación de herramientas pedagógicas basadas en el uso del computador y el autoaprendizaje.

Objetivo General

Diseñar un software educativo para la enseñanza de la asignatura Estructura de Datos de la carrera ingeniería informática.

.

Objetivos Específicos

1. Elaborar un diagnóstico de necesidades en la asignatura con los profesores y alumnos participantes.
2. Hacer un diseño instruccional adecuado a las necesidades y realidad de la UNET.
3. Elaborar documentación de los guiones, ventanas y animaciones que el software debe operar para cada unidad.

4. Elaborar documentación del prototipo modelo de software educativo.
5. Determinar los recursos, costos, para desarrollar el proyecto de implantación del software.

Alcance y Limitaciones

El software educativo de la asignatura Estructura de Datos, de la carrera Ingeniería Informática de la UNET, tiene como objetivo enseñar y proveer al estudiante una forma amena de aprender mediante la lectura de líneas de texto, visualización de videos, animaciones y ejecución de actividades de ejercitación y evaluación al final de cada tema, de manera que provea al estudiante la comprensión de los principios y nociones de la asignatura, las habilidades básicas para aplicar los conceptos y resolver los problemas provenientes del entorno social.

El alcance de la investigación confluye en una propuesta para el diseño lógico, el desarrollo metodológico de las fases de planeación sin la construcción e implementación. Especificaciones del proyecto: diagnóstico, diseño de las unidades de auto instrucción, prototipo modelo, costos, personal y recursos materiales.

Las especificaciones de los módulos ó partes funcionales que comprenderá el software educativo son:

- ✓ Módulo de ayuda: Presenta y explica la forma o como el usuario debe usar la herramienta producto en sus diferentes fases, actividades y escenarios.
- ✓ Módulo de exploración: Presenta al usuario los textos de los contenidos y videos que señalan paso a paso como se gestiona la memoria interna y/o externa (dispositivos de almacenamiento) del computador cuando se implementa una estructura de datos.
- ✓ Módulo de aplicación: Presenta al usuario situaciones donde aplicar el uso de una estructura de datos previamente estudiada en el

módulo de exploración. En este módulo el usuario deberá realizar actividades de planificación, modelado de datos y procesos en función de un problema propuesto.

- ✓ Módulo de programación: En este módulo se presenta el uso de un lenguaje de programación adecuado para manipular direcciones de memoria (recomendado C/C++) empleadas en el procesamiento de información de las estructuras de datos. En este módulo el usuario aprende a conjugar los principios operativos de las estructuras de datos con la estrategia de programación en un lenguaje popular de previo conocimiento como C/C++.
- ✓ Módulo de evaluación: En este módulo el usuario realiza actividades de evaluación teórica con preguntas de selección simple, verdadera ó falsa, entre otras de tipo objetiva. El banco de preguntas será actualizado por el docente periódicamente para evitar la memorización o rutina. También se aplicarán preguntas sobre situaciones de aplicación que soliciten la construcción de gráficos o rutinas de programas en el lenguaje C/C++.

El presente trabajo no presenta limitación alguna, ya que la universidad cuenta con el recurso tecnológico y humano para la realización de la propuesta. El resultado obtenido del proyecto servirá de base para que el jefe de departamento, decano y otros entes directivos puedan analizar y aprobar la construcción del software educativo bajo los criterios definidos.

CAPÍTULO II
MARCO TEÓRICO

Antecedentes

A nivel internacional, *Cataldy Zulma (España)*, en el año 2000 formuló una metodología disciplinada para el diseño, desarrollo y evaluación del software educativo orientada en el modelo de prototipo y espiral incremental, mediante la identificación de los métodos, los procedimientos, y las herramientas, que provee la ingeniería de software para el desarrollo de programas educativos de calidad, siguiendo las pautas de la teoría educativa subyacente. De este trabajo de investigación se tomará la aplicación de las teorías educativas inherentes al desarrollo de software educativo y los criterios de evaluación del producto final para la gestión de calidad.

A nivel nacional, *Zerpa G. Carlos (Universidad Central de Venezuela) y Ramírez L. Jorge (Universidad Simón Bolívar),* en el año 2001 desarrollaron la propuesta de un software educativo multimedia que versa la información ocupacional de las diferentes especialidades que conforman el ámbito de la carrera de Ingeniería. El material aborda el aspecto de tareas y desempeño laboral y formación humana, orientado a estudiantes que cursan fundamentalmente el 1er. Semestre de la carrera de Ingeniería en la Universidad Central de Venezuela y pretende constituirse en un material instruccional complementario en la asignatura Introducción a la Ingeniería que permitirá evaluar características personales y vocacionales, facilitará procesos de compilación de información e incorporará tecnología multimedia en el proceso de enseñanza aprendizaje para la clarificación vocacional con miras a generar en cada estudiante un mejor proceso de toma de decisiones en relación a la especialidad de la Ingeniería que pretende cursar. De esta investigación se tomarán algunos lineamientos concernientes a la descripción

de las plataformas educativas, tecnología, y componente comunicacional que constituyen la estructura del software, así como otras características relevantes al desarrollo del proyecto.

Marco Organizacional

Misión UNET

La Universidad del Táchira UNET es una institución de alto nivel académico, al servicio comprometido y solidario del ser humano integral y concreto, en la permanente búsqueda de la paz; que reúne en un ambiente de respeto mutuo, dedicación responsable, auto disciplina y honestidad intelectual y personal, a hombres y mujeres empeñados en la labor creativa de generar conocimientos humanísticos, científicos y tecnológicos a través de un riguroso esfuerzo de investigación, reflexión y análisis de la realidad en la que actúa y de la información proveniente de otras realidades, asegurando su trascendencia institucional por la pertinencia de su acción y el uso eficiente de los recursos.

Visión UNET

Auténtica Universidad en cuanto a comunidad de verdaderos seres universitarios. Comprometida con la positiva transformación del Táchira, donde todos los universitarios uniremos esfuerzos solidarios con la sociedad para convertirnos en la Región Latinoamericana con los mejores índices de crecimiento humano - social, cultural - educativo y económico; producto cabal del cumplimiento de nuestra misión; en un clima de respeto por el ser humano, el conocimiento y el medio ambiente (UNET, 2009).

Organigrama del Decanato de Docencia

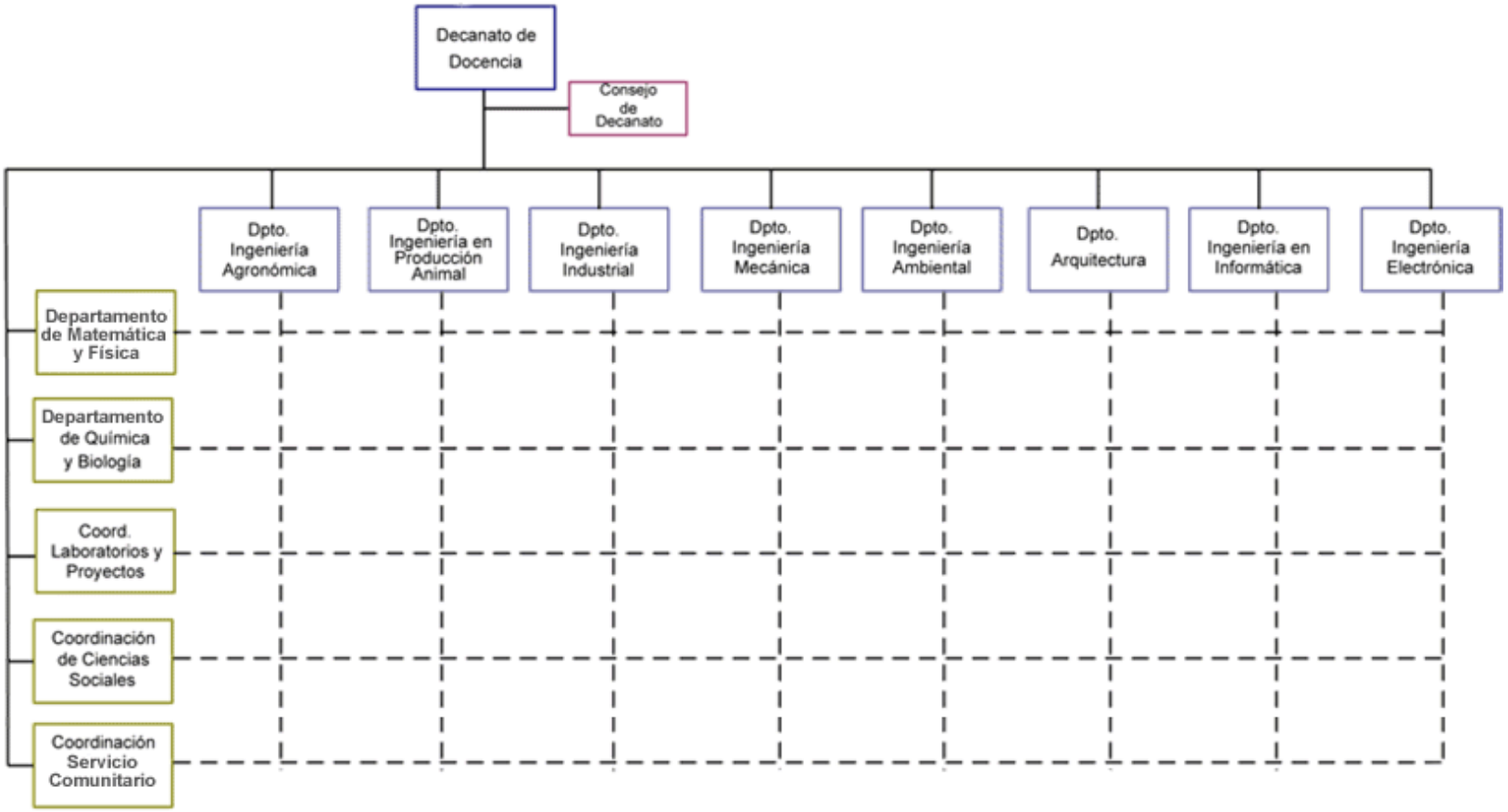


Gráfico Nro. 1. Organigrama del Decanato de Docencia de la UNET.
Fuente: Organización y Sistemas UNET.

Departamento de Ingeniería Informática

El Departamento de Ingeniería en Informática creado en el año 1997, tiene como misión administrar y gerenciar las actividades inherentes a impartir la carrera de ingeniería informática en la UNET. Su visión transmitir conocimientos científicos, en función de generar y desarrollar nuevas tecnologías, delimitando los esfuerzos hacia el área de la sistematización y computación, a través de medios automatizados (UNET, 2009).

Bases Teóricas

La Enseñanza

García y Cañal (1996), señalan que toda intervención docente es una respuesta al problema básico de cómo abordar adecuadamente la enseñanza. Es éste, sin duda, uno de los problemas centrales y más específicos del campo de estudio de la investigación didáctica, constituyendo, por otra parte, una cuestión que no parece estar ni mucho menos resuelta en la actualidad y cuya aclaración resulta en estos momentos insoslayable para superar los obstáculos que dificultan la mejora de la enseñanza en los diferentes niveles educativos.

Además afirman que las estrategias de enseñanza consisten en un sistema peculiar constituido por unos determinados tipos de actividades de enseñanza que se relacionan entre sí mediante unos esquemas organizativos característicos. En una perspectiva, las actividades son los elementos básicos del sistema, de tal manera que cada estrategia de enseñanza quedará definida por los tipos de actividades que incluye y por el esquema organizativo que regule las relaciones entre las actividades.

La enseñanza está dirigida siempre a facilitar la construcción de aprendizajes de todo tipo, desde una perspectiva profesional, se deberá centrar esfuerzos en el establecimiento de canales comunicativos que

posibiliten un flujo de información real para propiciar avances significativos en la dirección de los aprendizajes perseguidos.

El Diseño Instruccional

En el campo de la teoría del aprendizaje, el diseño instruccional se define como la concreción de un método para desarrollar la instrucción considerando: los objetivos educativos que se pretenden, las características generales del alumnado, el contexto en el que se ha de realizar, la estrategia didáctica que se seguirá, la evaluación, entre otros elementos. Un buen diseño instruccional aumenta las posibilidades de que se alcancen los objetivos previstos.

Según Moreno y Bailliére (2002):

> Un modelo de diseño instructivo no se debe reducir a justificar los medios técnicos incorporados, sino que requiere un análisis exhaustivo del escenario y del método en función de una serie de circunstancias. Por lo tanto el método es externo a la tecnología, externo también a las posibilidades didácticas, y consiste más bien en una reflexión sobre el modo (cómo) de conjuntar ambos aspectos, partiendo de unos contenidos (qué), persiguiendo unos objetivos (para qué), explicitando las razones (por qué), la secuencia y la temporalización de la enseñanza (cuándo), teniendo en cuenta los recursos (cuánto) y los agentes que intervienen en el proceso (quiénes). (p. 47)

Aspectos generales a considerar para el diseño de cursos

1. Objetivos de aprendizaje para los alumnos, considerando aspectos cognitivos, emocionales, sociales y físicos.
2. Contenidos que se desean transmitir.
3. Organización de la información en pequeñas unidades (*unidades de aprendizaje* que se integrarán las diversas *unidades didácticas* o lecciones que componen el curso).
4. Guías de aprendizaje.
5. Actividades de aprendizaje que faciliten a los estudiantes la adquisición y reestructuración de conocimientos (problemas, casos, trabajos colaborativos, entre otros).

6. Ejercicios de evaluación para comprobar que se van adquiriendo los conocimientos.
7. Retroalimentación ó Feed-back, refuerzos que se ofrecerán a los estudiantes.
8. Control del ritmo de aprendizaje de cada estudiante.

Recomendaciones para estructurar los contenidos

Esta parte es importante al inicio del diseño instruccional. Las siguientes pautas señalan como deben ser los contenidos:

- ✓ Correctos y actuales.
- ✓ Adecuados (o adaptables) a las características de los destinatarios (edad, capacidades, entre otros).
- ✓ Estructurados y progresivos, de manera que los anteriores faciliten la comprensión de los que siguen.
- ✓ Redactados correctamente, sin faltas de ortografía y con un lenguaje comprensible que facilite la comprensión.
- ✓ Motivadores, atractivos y originales en su presentación.
- ✓ Personalizados y con dinámicas de trabajo colaborativo.
- ✓ Contextualizados en un marco de referencia familiar para los estudiantes y que respondan a los intereses y expectativas de los destinatarios.
- ✓ Secuenciados con las actividades de aprendizaje y demás ejercicios preparados por el docente.

Las unidades de aprendizaje

Pueden considerarse los elementos más pequeños que integran un curso, la parte más pequeña de instrucción o información que por sí sola resulta significante para los estudiantes. Suele dar respuesta a las preguntas: qué, por qué, cómo, cuándo y dónde. Las unidades de aprendizaje se

estructuran en tres elementos básicos: los contenidos, la interactividad (conjunto de actividades) y la evaluación.

Las unidades didácticas

Las unidades de aprendizaje se van agrupando en *unidades didácticas*, cuyos principales elementos estructurales son los siguientes:

- ✓ Presentación de la unidad didáctica: objetivos, índice de objetivos, destinatarios, breve descripción de las actividades y la evaluación.
- ✓ Diversas unidades de aprendizaje (que incluyen los tres elementos: contenidos, actividades, evaluación).
- ✓ Resumen y conexiones entre las unidades de aprendizaje de la unidad didáctica y con otras unidades.
- ✓ Actividades de aplicación relacionadas con las unidades de aprendizaje.
- ✓ Evaluación general de la unidad didáctica.

Teorías del Aprendizaje

Gros (2000) afirma que el desarrollo de software educativo se ha basado, en su mayoría, en dos de las principales teorías psicológicas del aprendizaje: el conductismo y el constructivismo, ambos desarrollan los principios en los que se fundamenta la metodología didáctica de un gran número de programas multimedia educativos. A continuación se presentan las principales teorías que sustentan el diseño de herramientas de autoaprendizaje ó software para la enseñanza de estructura de datos:

1. El Conductivismo y el Condicionamiento Operante

Los principios del *conductivismo* se basan en la creación de una psicología "objetiva" cuyo objeto de estudio sea la conducta observable; su método de estudio, el método experimental y su problema central, así como la predicción y el control de la conducta. La conducta consiste en respuestas,

reacciones del organismo (animal o humano), a ciertos sucesos ambientales, denominados estímulos. La respuesta es cualquier movimiento que el organismo manifiesta y el estímulo es cualquier objeto del medio ambiente capaz de provocar un cambio a nivel fisiológico. Watson considera que la relación estímulo-respuesta es observable y fácil de ser estudiada en laboratorio.

Los resultados obtenidos en el laboratorio obligan necesariamente a suponer la existencia de factores internos de la conducta. Conceptos tales como motivación, impulso, así como sensación y percepción de las imágenes, entre otros, comenzaron a ser utilizados para hacer referencia a las propiedades dinámicas e intrínsecas de la conducta.

Tiempo después Skinner (Citado por Cataldi 2000) formuló su teoría conductista del *condicionamiento operante* en los años treinta y, durante los primeros años de su carrera se interesó por la educación elaborando las “máquinas de enseñanza” y los “sistemas de instrucción programada”. El cambio conductual en el "condicionamiento operante" se da a través del refuerzo diferencial por aproximaciones sucesivas hacia la forma de comportamiento deseada, mediante el proceso de moldeamiento para modificar la conducta.

En esta época, afirma Gross (2000) es cuando comienzan los estudios referidos a la elaboración de lo que se considera una buena “programación didáctica”. La elaboración de una programación se iniciaba con el establecimiento de los objetivos generales en función del curriculum de los alumnos, se construía el programa, elaborando la serie de secuencias a seguir en "cuadros". Luego, se estudiaba el tipo de respuesta más adecuada y la clase de feedback a lograr. El paso siguiente era la evaluación y revisión del programa sobre la base de las respuestas de los alumnos.

En este período, cobran interés los objetivos operacionales y conductuales a partir de un trabajo de Mager (Citado por Gros 2000), que se

usó como un manual para los escritores de enseñanza programada. El objetivo debe describir una conducta observable y sus productos o logros.

Esta teoría aporta principios valiosos en el diseño de la interfaz de usuario del software educativo para la enseñanza de estructura de datos: por ejemplo, una ventana emergente con mensaje de felicitación a respuestas correctas suministradas durante la evaluación puede ser un refuerzo de a una conducta de aprendizaje buscada. A través de la manipulación de estímulos es posible conducir el aprendizaje.

2. El Constructivismo

El planteamiento de este enfoque, así lo afirma Gros (2000), es que el individuo es una construcción propia que se va produciendo como resultado de la interacción de sus disposiciones internas y su medio ambiente, y su conocimiento no es una copia de la realidad sino una construcción de la persona misma. Esta construcción resulta de la representación inicial de la información y de la actividad, externa o interna, que desarrollamos al respecto.

A través de los procesos de aprendizaje el alumno construye estructuras, es decir, maneras de organizar la información, las cuales facilitarán mucho las adquisiciones futuras, y por lo tanto, los psicólogos educativos y los diseñadores de currículos y de materiales didácticos (libros, guías, programas computacionales, manipulables, etcétera) deben hacer todo lo posible para estimular el desarrollo de dichas estructuras. Esto incluye entender que una estructura es algo amplia, complicada, interconectada, y que obviamente no puede ser representada fácilmente por los objetivos conductuales. Comúnmente se dice que las estructuras son compuestas de esquemas.

Continua explicando Gros (2000), un esquema es una representación de una situación concreta o de un concepto que permite manejarlos internamente y enfrentarse a situaciones iguales o parecidas en la realidad.

Según Bruner (citado por Gros 2000), el alumno no descubre el conocimiento, sino que lo construye, en base a su maduración, experiencia física y social. Algunas de las habilidades a adquirir son: la capacidad de identificar la información relevante para un problema dado, de interpretarla, de clasificarla en forma útil, de buscar relaciones entre la información nueva y la adquirida previamente.

Hablar de ambientes de enseñanza constructivistas significa concebir el conocimiento desde la perspectiva de Piaget (citado por Gros 2000) mediante desarrollos cognitivos basados en una fuerte interacción entre sujeto y objeto, donde el objeto trata de llegar al sujeto, mediante cierta perturbación de su equilibrio cognitivo, quien trata de acomodarse a esta nueva situación y producir la asimilación del objeto, con la consecuente adaptación a la nueva situación.

El aporte de esta corriente al proyecto de creación de software educativo para la enseñanza de Estructura de Datos se verá en las estrategias de juego y simulaciones presentadas para que el estudiante ejercite de manera libre y construya su conocimiento a su propio ritmo.

3. El Cognitivismo

Para Gros (2000) el cognitivismo es una teoría de aprendizaje donde la mente es un agente activo en el proceso de aprendizaje, construyendo y adaptando los esquemas mentales o sistemas de conocimiento. En los inicios del modelo cognitivo, señala Bruner (citado por Gros 2000), había una firme intención en la realización de esfuerzos para indagar acerca de los procesos de construcción de los significados y producciones simbólicas, empleados para conocer la realidad circundante. Sin embargo, el papel creciente de la informática y las computadoras incorporó un planteamiento basado en la metáfora de las computadoras.

Rogers (citado por Gros 2000) habla de "la facilitación del aprendizaje que aparece como una potencialidad natural de todo ser humano". Dice que

"el aprendizaje significativo" tendrá lugar cuando el sujeto perciba al tema como importante para sus propios objetivos o satisfaciendo alguna de sus características o necesidades personales sociales. El término significativo también puede ser entendido siguiendo a Ausubel (citado por Gros 2000), como un contenido que tiene una estructuración lógica interna y como aquel material que puede ser aprendido de manera significativa por el sujeto. Rogers afirma que "el aprendizaje social más útil en el mundo es el aprendizaje.

Los principios de esta teoría educativa se verán reflejados en la creación de representaciones simbólicas que hacen explicaciones de forma casi visual, tales como los mapas conceptuales, diagramas de procesos, diagramas de flujo, entre otras.

Uso del Color en el Diseño de una Interfaz Gráfica de Usuario

Para Wright, Mosser y Wooley (2006), una tarea importante en el diseño de software educativo es el diseño de una interfaz gráfica con el usuario (Graphical User Interfaces, GUI) y el color es un componente principal de una GUI. Hay dos componentes presentes en una interfaz a color: el sistema humano visual (percepción) y el sistema de despliegue del color.

El color tiene un impacto principal sobre la interacción humano-computadora: positivo ó negativo. El uso de color apropiado puede ayudar a la memoria del usuario y facilitar la formación de modelos mentales efectivos. El uso efectivo del color puede ser una herramienta poderosa, sin embargo, el uso inefectivo del color puede degradar el desempeño de una aplicación y disminuir la satisfacción del usuario. Debido a estos factores, se puede afirmar que el uso efectivo del color en interfaces de computadora es un importante tópico en sistemas de interacción hombre – máquina que requiere ser examinado metódicamente.

Para entender el potencial del color en las interfaces, hay que examinar algunas características fundamentales de la percepción del color. Las características fundamentales primarias incluyen varios modelos de color, el sistema visual humano, principios fisiológicos del color y efectos del color tales como las ilusiones y las combinaciones del color.

Modelos de color

Según Wright et al. (2006) los modelos del color son:

1. **Modelo basado en la percepción (HSV Matiz / Hue, Saturación / Saturation y Valor / Value)**

En el HSV el Matiz es la composición de la longitud de onda espectral de color que produce los colores que vemos tales como el anaranjado, azul, etc. La Saturación es la pureza relativa del color sobre una escala del gris al tono más vibrante del color particular. El Valor es la fuerza u oscuridad del color. La Claridad también referida como brillantez, se refiere a la cantidad de energía luminosa que crea el color.

2. **Modelo basado en despliegues (RGB Red-Green-Blue)**

El RGB es usado para el despliegue en monitores de computadora. La Comisión Internacional sobre Iluminación es una organización mundial que desarrolló la primera versión del modelo medido espectralmente conocido como CIE en 1931. El CIE es una medida precisa espectral usada para precisar colores y remover la ambigüedad de ellos.

Todos los colores presentados en una computadora deben ser trasladados dentro del espacio del color RGB. Desafortunadamente, no hay un mapeo uno a uno de los modelos basados perceptualmente a los modelos basados en despliegue. Este hecho puede explicar algunas de las dificultades encontradas cuando tratamos de recrear justamente el color

correcto para una interfaz de pantalla. No es siempre posible obtener la sombra exacta. El modelo CIE permite traducciones del HSV al RGB.

El sistema visual humano, fisiología y percepción

El ojo humano contiene una lente y una retina. La retina contiene receptores sensitivos a la luz conocidos como bastones y conos. El propósito primario de los bastones es proporcionar visión de noche, mientras que los conos trabajan en niveles más altos de intensidad de la luz. Los conos contienen fotopigmentos, también conocidos como fotorreceptores, los cuales son sensitivos al rojo, al verde o al azul. Aproximadamente el 64% de los conos contienen fotopigmentos rojos, 32% contiene verdes y solamente alrededor de 2% contienen fotopigmentos azules. Las propiedades fisiológicas del sistema nervioso dictan la sensación del color. Los humanos son sensitivos a un rango de longitudes de onda. Las longitudes de onda no coloreadas, sin embargo el color es el resultado de la interacción de la luz y nuestro sistema nervioso. Las longitudes de onda que producen colores diferentes son enfocadas a distancias diferentes detrás de la lente.

La lente no transmite todas las longitudes de onda de la misma manera, exhibiendo menos sensibilidad a las longitudes de onda más cortas, lo cual tiene el efecto de absorber los azules. Inversamente, somos más sensitivos a las longitudes de onda más largas, lo cual es exhibido por una sensibilidad aumentada a los amarillos y anaranjados. Bastante raro, podemos ver los azules mejor en la periferia que en el frente debido a la distribución física de los fotorreceptores azules.

Consecuente a la organización física del ojo hay efectos interesantes o ilusiones causadas por ciertas organizaciones de color o combinaciones. Debido a la falta de fotorreceptores azules, las líneas azules delgadas (como el texto azul) tienden a verse borrosas, y pequeños objetos azules tienden a desaparecer cuando tratamos de enfocarlos. Los colores que difieren solamente por la cantidad de azul no producen bordes claros. Por ejemplo,

los colores con la misma cantidad de verde y rojo que varían solamente en la cantidad del azul producen orillas borrosas.

El contraste de colores adyacentes puede crear una ilusión observada fácilmente. Dos objetos del mismo color pueden aparecer marcadamente diferentes en color dependiendo del color del fondo. El uso inefectivo de los colores puede causar vibraciones y sombras; imágenes que distraen al usuario y pueden forzar la vista. Wright et al. (2006).

Modelos mentales y uso efectivo del color

Wright et al. (2006) cita que la gente interactúa con su mundo a través de modelos mentales que ellos han desarrollado. Específicamente, las ideas y las habilidades que traen a su trabajo están basadas en modelos mentales que ellos desarrollaron acerca de su propia actividad laboral. Para diseñar interfaces, necesitamos: primero ayudar al usuario a desarrollar modelos mentales del sistema que le permita entender el trabajo, segundo desarrollar las herramientas de la interfaz que le ayudarán a realizar el trabajo. El uso adecuado del color comunica hechos e ideas más rápidamente y más estéticamente al usuario. El color también puede ayudar a desarrollar modelos mentales eficientes y factibles si se siguen las siguientes pautas:

- ✓ *Simplicidad,* La simplicidad es importante en el diseño de interfaces a color. Existe una simplicidad inherente en el color la cual debería ser usada cuando se desarrolla el diseño. Los cuatro colores fisiológicamente primarios son el rojo, el verde, el amarillo y el azul. Estos colores son fáciles de aprender y recordar. Vinculando significados prácticos e intuitivos a estos colores simples cuando se diseña una pantalla, el diseñador de la interfaz enriquece el desarrollo del usuario con un modelo mental efectivo.
- ✓ *Consistencia,* La consistencia es vital al asignar significados a los colores. El orden intuitivo de los colores puede ayudar a establecer consistencia intuitiva en el diseño. El orden espectral y perceptual

rojo, verde, amarillo, azul puede guiar el orden de los conceptos vinculados a los colores. El rojo es primero en el orden espectral y se enfoca en el frente, el verde y el amarillo se enfocan en medio, mientras que el azul se enfoca en el fondo.

- ✓ *Claridad,* La claridad es también una pauta importante para usar color. Experimentos han mostrado que el tiempo de búsqueda para encontrar una pieza de información es disminuido si el color de esta pieza es conocido por anticipado, y si el color sólo se aplica a esa pieza. Los colores de interfaz estandarizados deberán de ser establecidos e usados a través del desarrollo. El uso claro y conciso del color puede ayudar a los usuarios a encontrar piezas de información más rápidamente y más eficientemente. El aprendizaje puede ser grandiosamente aumentado con el color. La estética y lo atractivo de la interfaz son inherentemente aumentados por el uso del color.
- ✓ *Lenguaje del color,* El lenguaje de color es importante en el uso del color. Los individuos desarrollan un lenguaje de color a medida que maduran, basándose en el uso común y cultural. Debido a este hecho, el simbolismo existente y el uso cultural del color deberán de ser considerados al diseñar una interfaz. Por ejemplo, el servicio de correo de Estados Unidos utiliza el azul para los buzones del correo, Inglaterra utiliza un rojo brillante, y Grecia utiliza un amarillo brillante. Al desarrollar un sistema de correo electrónico para estos países, los colores mencionados anteriormente servirían efectivamente para los iconos del correo.

Combinaciones de los colores

Wright et al. (2006) señala que es más difícil utilizar el color efectivamente que usarlo inefectivamente. Para usar el color efectivamente se requiere cuidadosa coordinación con los colores y sus niveles de

intensidad asociados. Usar la combinación equivocada de colores para el fondo y para el frente puede crear ilusiones que forzarán la vista. Para agilizar la combinación de colores se puede utilizar las reglas de Murch:

- ✓ Evite el despliegue simultáneo de colores espectralmente extremos que estén altamente saturados.
- ✓ El color AZUL puro deberá de ser descartado para el texto, líneas delgadas y figuras pequeñas.
- ✓ Evite colores adyacentes que se diferencien solo por la cantidad de azul que contienen.
- ✓ Los operadores de edad avanzada necesitan niveles más altos de brillo para distinguir los colores.
- ✓ Los colores cambian de apariencia a medida que el nivel de luz ambiental cambia.
- ✓ La magnitud de un cambio detectable en el color varía a través del espectro.
- ✓ Es difícil enfocar hacia las orillas creadas solamente por el color.
- ✓ Evite utilizar el ROJO y el VERDE en la periferia de despliegues a gran escala.
- ✓ Los colores opuestos se ven bien juntos.
- ✓ Para los observadores que tienen deficiencias del color (ciegos al color), evita hacer distinciones de un solo color.

Recomendaciones para usar el color en una interfaz

- ✓ Utilizar el color azul para el fondo.
- ✓ Utilizar la secuencia de color espectral (rojo, anaranjado, amarillo, verde, azul, índigo y violeta).
- ✓ Mantener pequeño el número de colores.
- ✓ Evitar usar colores adyacentes que difieren solamente en la cantidad de azules puros.

- ✓ Utilizar colores brillantes para indicar peligro o para llamar la atención del usuario.

Uno de los elementos más importantes de usar el color efectivamente es conocer al usuario, el ambiente del usuario, y la tarea que el usuario está realizando. Esto es igualmente importante para la integración del color que para cualquier otra parte del diseño de la interfaz.

El uso del color en un programa de software educativo afecta de manera considerable el aprendizaje y su impacto social. Es importante que para la construcción del software "enseñanza de Estructura de Datos" se tomen en cuenta todos estos principios de manera que motive al estudiante a aprender utilizando esta herramienta.

Para la fabricación del software educativo es importante considerar las pautas y reglas de diseño previstas en las investigaciones de autores como Murch, Marcus, Pancakes (citados por Wright et al. (2006) y otros, y no imponer criterios sin fundamento basados en la apreciación estética personal que conlleven a perjudicar el éxito de la interfaz operativa del software.

Computación y Multimedia

Norton (2006), afirma que hace mucho tiempo, las personas descubrieron que los mensajes son más efectivos, se entienden y recuerdan con mayor facilidad, cuando se presentan por medio de una combinación de medios diferentes. Esta combinación es lo que significa el término multimedia: utilizar más de un tipo de medio al mismo tiempo; ejemplo la televisión combina imagen y sonido.

La computadora ha llevado el contenido multimedia a un nivel más alto permitiéndonos utilizar medios diferentes de manera simultánea. Por ejemplo, Una enciclopedia en una versión multimedia puede mover imágenes explicativas al mismo tiempo que un narrador proporciona la información que tradicionalmente es leída; el usuario puede elegir a que sección acudir haciendo clic en los vínculos de hipertexto. El resultado es una consulta de

información entretenida y multimedia para sacar mayor provecho de aprendizaje.

Las tecnologías de computación permiten que los productos multimedia basados en PC vayan a un paso más lejos, debido a que la computadora puede aceptar y responder a la entrada de información por parte del usuario, puede llevar a cabo eventos multimedia interactivos, involucrando al usuario de una manera diferente a la empleada por cualquier medio de comunicación.

La interactividad ha sido definida de distintas maneras, pero en el campo de la multimedia, el término significa que el usuario y el programa se responden entre sí: el programa normalmente proporciona al usuario una variedad de opciones que el usuario selecciona para dirigir el flujo del programa.

En el campo educativo, los horizontes y las perspectivas del uso de multimedia son cada día más prometedoras, ya que las técnicas de la inteligencia artificial, del procesamiento distribuido, del multiprocesamiento, del desarrollo de interfases gráficas, entre otras, ofrecen un potencial muy grande y ponen a disposición de los educadores ciertos medios que probablemente en otros tiempos muchos hubieran soñado tener para lograr una formación más amplia e integral.

Software Educativo

Se define Software educativo como cualquier solución tecnológica dinámica y sistemática que interviene en el proceso educativo e instruccional. Dinámica, porque da soporte a la simulación de ambientes y actividades, a las habilidades y destrezas, a la construcción y apropiación del conocimiento y sistemática, porque Integra el contenido (teorías, reglas, escenarios), medios, soporte pedagógico y acciones (eventos, navegaciones) como un conjunto de componentes relacionados que trabajan juntos para alcanzar un fin común.

Según Gros (2000),"Software educativo es cualquier producto de ejecución en la computadora realizado con finalidad educativa". Bajo esta amplia definición podemos encontrar programas de lo más variado. Una clasificación bastante estándar divide el formato de los programas en cinco tipos distintos: tutoriales, práctica y ejercitación, simulación, hipertextos e hipermedia.

Los *programas tutoriales* tienen por objeto enseñar un determinado contenido. Se trata de programas didácticos cuya idea fundamental es llevar el conocimiento al usuario a través de la interacción con el programa. En los programas tutoriales, lo importante es la organización del conocimiento y las estrategias de enseñanza que adopta el programa para conseguir el aprendizaje del usuario.

Los *programas de práctica y ejercitación* se caracterizan por proporcionar al alumno la oportunidad de ejercitarse en una determinada tarea una vez obtenidos los conocimientos necesarios para el dominio de la misma. Este tipo de programa es usado en materias prácticas, las cuales han incorporado en los últimos tiempos el juego para entretener al usuario mientras realiza actividades repetitivas.

Los *programas de simulación* tienen por objeto proporcionar un entorno de aprendizaje abierto basado en modelos reales. Este tipo de programa permite al usuario experimentar, contrastar diversas hipótesis, interactuar y tomar decisiones con base a situaciones "que pasa si…". Al igual que el tipo anterior generalmente se presenta en un escenario de juego.

Los *programas hipertextuales* están basados en modelos no lineales. Organiza textos completos como núcleos de información conectados por diversos enlaces, de manera que no se prescribe el orden de los contenidos presentados. Es el usuario quien decide que información desea activar y en qué orden.

Los *programas hipermediales* son similares a los hipertextuales en el basamento del modelo no lineal, pero se diferencia por el hecho de presentar

la información, conectada al enlace, a través de recursos de audio, animación y video.

Debemos tener presente que la clasificación anterior es teórica, en la actualidad, en un mismo programa podemos encontrar formatos diferentes. Por ejemplo, un programa puede tener una parte tutorial complementada por una simulación y algunos ejercicios que miden el conocimiento adquirido.

Estructura Básica de un Software Educativo

Para Marqués (2000), la mayoría de los programas didácticos tienen tres módulos principales claramente definidos: el módulo que gestiona la comunicación con el usuario (sistema input/output), el módulo que contiene debidamente organizados los contenidos informativos del programa (bases de datos) y el módulo que gestiona las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor).

1. El entorno de comunicación o interfase

La interfase es el entorno a través del cual los programas establecen el diálogo con sus usuarios, y es la que posibilita la interactividad característica de estos materiales. Está integrada por dos sistemas:

- **a.** El sistema de comunicación programa-usuario, que facilita la transmisión de informaciones al usuario por parte del ordenador, incluye: Las pantallas a través de las cuales los programas presentan información a los usuarios, Los informes y las fichas que proporcionen mediante las impresoras y el empleo de otros periféricos: altavoces, sintetizadores de voz, robots, módems, convertidores digitales-analógicos...
- **b.** El sistema de comunicación usuario-programa, que facilita la transmisión de información del usuario hacia el ordenador, incluye: El uso del teclado y el ratón, mediante los cuales los usuarios introducen al ordenador un conjunto de órdenes o respuestas que

los programas reconocen. Y el empleo de otros periféricos: micrófonos, lectores de fichas, teclados conceptuales, pantallas táctiles, lápices ópticos, modems, lectores de tarjetas, convertidores analógico-digitales...

Con la ayuda de las técnicas de la Inteligencia Artificial y del desarrollo de las tecnologías multimedia, se investiga la elaboración de entornos de comunicación cada vez más intuitivos y capaces de proporcionar un diálogo abierto y próximo al lenguaje natural.

2. Las bases de datos

Las bases de datos contienen la información específica que cada programa presentará a los alumnos. Pueden estar constituidas por:

- **a.** Modelos de comportamiento. Representan la dinámica de unos sistemas.
- **b.** Datos de tipo texto. Información alfanumérica.
- **c.** Datos gráficos. Las bases de datos pueden estar constituidas por dibujos, fotografías, secuencias de vídeo, entre otros.
- **d.** Sonido. Como los programas que permiten componer música, escuchar determinadas composiciones musicales y visionar sus partituras.

3. El motor o algoritmo

El algoritmo del programa, en función de las acciones de los usuarios, gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar los alumnos. Distinguimos 4 tipos de algoritmo:

- **a.** Lineal, cuando la secuencia de las actividades es única.
- **b.** Ramificado, cuando están predeterminadas posibles secuencias según las respuestas de los alumnos.

c. Tipo entorno, cuando no hay secuencias predeterminadas para el acceso del usuario a la información principal y a las diferentes actividades. El estudiante elige qué ha de hacer y cuándo lo ha de hacer.

d. Tipo sistema experto, cuando el programa tiene un motor de inferencias y, mediante un diálogo bastante inteligente y libre con el alumno, asesora al estudiante o tutoriza inteligentemente el aprendizaje. Su desarrollo está muy ligado con los avances en el campo de la Inteligencia Artificial.

Características Pedagógicas de un Software Educativo

1. Adaptación al Ritmo de Aprendizaje del Usuario

Debe ser eficaz en el aprendizaje individual, donde el usuario pueda avanzar de acuerdo con sus propias necesidades, estilo y ritmo de aprendizaje. Permite al estudiante acceder, cuantas veces quiera, a la información sin temor al rechazo y la crítica.

2. Libertad de Movimiento dentro del Contenido

Se puede avanzar o retroceder, como profundizar, de acuerdo con los requerimientos y necesidades de información.

3. Administración del Tiempo

El Usuario toma el tiempo necesario para aprender, organiza su tiempo como mejor le parezca.

4. Representación del Contenido

Hace referencia a la utilización de los medios (imagen, Sonido, Texto) para representar un Contenido (teorías, reglas, escenarios), y así obtener y entender en menor tiempo la información.

5. Planeación del Contenido

Presentar la información de una forma clara y contundente, reduce la barrera entre lo que el docente quiere expresar, y lo que el alumno entiende. Se pueden dejar escenarios donde el alumno involucra su propia creatividad e ingenio, haciendo más interesantes, relevantes y útiles algunas temáticas; rescatando la fantasía, retos, juegos, etc.

Modelos para la Construcción del Software Educativo

Gros (2000), reseña que la mayor parte de los modelos de elaboración de software educativo se han basado en modelos didácticos ya existentes. Son de carácter prescriptivo y tienen por objeto guiar en las diversas fases de producción del producto, facilitando el trabajo de los diferentes profesionales implicados en el proyecto. Algunos de estos modelos incorporan explícitamente constructos relativos a determinadas teorías sobre el aprendizaje y la enseñanza. Otros, son de carácter general y suponen que pueden ser adaptados a posiciones teóricas diversas. Entre los principales modelos se encuentran:

1. El modelo sistemático o en cascada

El modelo sistemático tiene su origen en la ingeniería del software y ha sido adaptado a la producción de software educativo a propuesta de autores como Gagné y Dick & Carey. Este modelo considera la elaboración de los productos informáticos como un proceso lineal o en cascada constituido por cinco fases independientes: análisis, diseño, desarrollo, evaluación e implementación.

Fase	Objetivo	Actividades
Análisis	Determinar los resultados esperados y las condiciones de utilización	Identificación de: - Problemas instruccionales. -Características de los futuros usuarios. -Tipo de software a desarrollar - Herramientas informáticas a emplear.
Diseño	Especificar lógicamente el funcionamiento del producto	Elección de: - Tipo (formato) de programa - Tipos de aprendizajes que se persiguen - Tipo de diseño instruccional que se a adoptar en el programa. Elaboración del guión del programa. Diseño de materiales de ayudas y soportes.
Desarrollo	Construir el programa de acuerdo al diseño lógico	- Construcción de Interfaz, animaciones, etc. - Construcción de los guiones. - Enlazar secuencias.
Evaluación	Efectuar una valoración del producto desarrollado	- Diseñar instrumentos de evaluación - Aplicar instrumentos - Analizar resultados y hacer los ajustes.
Implementación	Colocar en funcionamiento el producto final en un contexto real	- Distribuir el programa - Informar las pautas generales para iniciar - Realizar mantenimiento periódico.

Tabla Nro. 1. Fases del modelo sistémico o cascada.
Fuente: El autor.

En la aplicación de un modelo sistemático, se cuenta con la ventaja de tener claras todas las actividades estructuradas en secuencia, al término de una fase se inicia la siguiente, pero en la realidad se observa que es difícil cerrar de manera definitiva las etapas hasta que el producto esté totalmente terminado, por consiguiente, el modelo sistémico podría mejorar si se trata como un ciclo de evaluación continua.

2. Los modelos no lineales

De entre las diversas propuestas de modelaje, se destacan: el modelo en espiral o incremental y el modelo de desarrollo rápido de prototipos.

En el modelo incremental, las etapas son las mismas que en el ciclo de vida en cascada y su realización sigue el mismo orden, pero corrige la problemática de la linealidad del modelo en cascada. Este modelo incremental fue desarrollado por Lehman (Citado por Gross 2000). En cada paso sucesivo se agregan al sistema nuevas funcionalidades o requisitos que permiten el refinado a partir de una versión previa.

Este modelo es útil cuando la definición de los requisitos es ambigua e imprecisa, porque permite el refinamiento, o sea se pueden ampliar los requisitos y las especificaciones derivadas de la etapa anterior.

Uno de los problemas que se puede presentar es la detección de requisitos tardíamente, siendo su corrección tan costosa como en el caso de la cascada.

En el modelo de desarrollo rápido de prototipo, la idea se centra en facilitar la comprensión de los requisitos que plantea el usuario, sobre todo si este no tiene una idea muy acabada de lo que desea.

Esta versión temprana de lo que será el producto, con una funcionalidad reducida, en principio, podrá incrementarse paulatinamente a través de refinamientos sucesivos de las especificaciones del sistema, evolucionando hasta llegar al sistema final.

Bases Legales

El ministerio del Poder Popular para las Telecomunicaciones y la Informática, a través de su portal web titulado Directorio de Gobierno Electrónico Venezuela, presenta entre los artículos mas relevantes de Legislación de las TI, el 108 y 110 de la Constitución Nacional de la República Bolivariana de Venezuela, indicando:

> Nuestra carta magna reconoce el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional, igualmente establece que el Estado garantizará servicios públicos de radio, televisión y redes de bibliotecas y de informática, con el fin de permitir el acceso universal a la información. Los centros educativos deben incorporar el conocimiento y aplicación de las nuevas tecnologías, de sus innovaciones, según los requisitos que establezca la ley. (TIC: ¿Regular o Desregular?, 2008, lo más relevante de la legislación TI, sec. 1)

Estos artículos señalan claramente el deber que tienen las instituciones de educación de incorporar la aplicación de nuevas tecnologías y uso de redes para la formación y desarrollo del país, por consiguiente los procesos académicos deben ser atendidos tomando en cuenta este marco legal y como contribución a este asunto se formuló el proyecto de software educativo para la enseñanza de estructura de datos de la carrera de ingeniería informática de la UNET.

CAPÍTULO III
MARCO METODOLÓGICO

Tipo de Investigación

El presente trabajo “Diseño del Software Educativo para la Enseñanza de Estructura de Datos de la Carrera Ingeniería Informática de la UNET” se corresponde con una *investigación de proyecto factible*, la cual según Barrios (1998) “consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupo sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos” (p. 7). Con la realización del proyecto se pretende dar solución a la problemática de mejoramiento de la calidad de enseñanza y escasez de espacio físico para clases presenciales, aportando una solución alineada con las políticas de la UNET con el mayor provecho de las tecnologías y recursos que posee la institución.

Diseño de la Investigación

Según Martín (citado en Balestrini 1997): “El diseño de la investigación se define como el plan global de investigación que integra de un modo coherente y adecuadamente correcto técnicas de recogida de datos a utilizar, análisis previstos y objetivos.” (p. 118) Y según Balestrini (1997):

> Los diseños de campo permiten establecer una interacción entre los objetivos y la realidad de la situación de campo; observar y recolectar los datos directamente de la realidad, en su situación natural; profundizar en la compresión de los hallazgos encontrados con la aplicación de los instrumentos; y proporcionarle al investigador una lectura de la realidad objeto de este estudio más rica en cuanto al conocimiento de la misma. (p.119)

Con base a lo señalado por Martín y Balestrini, el presente trabajo se corresponde con un diseño de campo, ya que los datos de interés serán recogidos en forma directa de la realidad, de los estudiantes y profesores de la asignatura para formular inicialmente el diagnóstico de la situación actual y determinar las necesidades y requerimientos del software.

La metodología ha usar para el desarrollo de este trabajo de investigación está basada en un modelo derivado de *cascada pura,* es decir, una cascada modificada denominada *"Metodología Dinámica para el Desarrollo de Software Educativo"*. Esta metodología está compuesta por cuatro fases: Diseño Educativo, Producción, Realización e Implementación, y un eje transversal que es la Evaluación. No se requiere la culminación de una fase para pasar a la otra, es posible obtener rápidamente un prototipo que permita hacer validaciones parciales y correcciones de ser requeridas.

Diseño Educativo

1. *Estudio de Necesidades*: Esta necesidad especifica una situación de aprendizaje determinada. Si se habla de una situación de aprendizaje es fácil determinar las necesidades, tales como: tiempo a emplear en una actividad o clase, mucho contenido, poco contenido, muchos alumnos, automatizar procesos que no interesan como contenido, generar actividades de refuerzo, etc.
2. *Descripción del aprendiz*: Es necesario saber cuál es la potencial audiencia para poder seleccionar aspectos relacionados con la cultura, costumbres, edades, estilos de aprendizajes, etc.
3. *Propósito y objetivos referidos al proyecto*: Se refiere a lo que se quiere hacer desde el punto de vista del medio y para qué lo quiero hacer.
4. *Formulación de objetivos terminales de aprendizaje*: En esta parte se redactan los objetivos generales y específicos que se quieran alcanzar con el uso del material.

5. *Análisis estructural*: Se especifican las subhabilidades a desarrollar, se toman en cuenta los atributos básicos de los conceptos que se quieran trabajar.
6. *Especificación de los conocimientos previos:* Las competencias, habilidades y destrezas que debe tener el usuario son los que finalmente van a determinar el éxito o no del material educativo computarizado o en todo caso le hace el camino más fácil o más difícil al mismo.
7. *Formulación de objetivos específicos*: Se procede a formular los objetivos específicos. Los mismos deben estar lo más sencillo posible, es decir, tienen que redactarse en términos operacionales.
8. *Selección de estrategias instruccionales*: Se definen los eventos de aprendizaje que sean considerados necesarios por el diseñador para lograr los objetivos propuestos. Se piensa en cuál es la mejor manera o como un determinado contenido va a ser presentado al usuario. Es necesario hacer una revisión de las teorías educativas (cómo aprenden las personas), para poder prescribir las acciones a seguir.
9. *Contenido (información a presentar)*: Aquí se debe seleccionar y organizar con cuidado el contenido temático que desea. Se hace una lista de temas o puntos de interés.
10. *Selección de estrategias de evaluación*: Se refiere a la selección y/o diseño de estrategias de evaluación de los aprendizajes. Se trata de cómo saber si el usuario ha logrado los objetivos de aprendizaje previstos. También se puede prescribir si se quiere aspectos del desempeño, es decir, llevar un control de actuación del usuario, el tiempo que tarda en un contenido en particular, el número de veces que pide ayuda, el número y el tipo de errores cometidos, etc.
11. *Determinación de variables técnicas*: En este caso se especifican aspectos relacionados con metáforas, principio de orientación, uso de

íconos, botones, fondos, textos, planos, sonidos, videos, animaciones, simulaciones, etc.

Producción

1. *Guión de contenido*: se hace un esquema de la descripción de la audiencia, se anota el propósito, se señala el tema, los objetivos específicos de aprendizaje, se decide cuál es la línea de producción, se establece el esquema de navegación y se realiza el web o diagrama de contenido.
2. *Guión didáctico*: Se redacta con un lenguaje sencillo y claro. Se utiliza un vocabulario familiar a la audiencia. Se presenta el contenido ya desarrollado utilizando como soporte las estrategias instruccionales elaboradas. Puede ser asociado a un guión literario.
3. *Guión técnico (Storyboard)*: es el resultado de la visualización del guión didáctico o libreto. Se nutre de la determinación de las variables técnicas especificadas en la fase anterior. Es importante tomar en cuenta las teorías referidas a la percepción, la importancia del uso del color, sonido, las zonas de comunicación en pantalla, etc.

Realización

1. *Prototipo:* el primer prototipo es el Storyboard, luego, a partir de este, se diseñan cada una de las pantallas que conformarán el material educativo computarizado. Se hace lo equivalente pero en el computador a nivel de pantallas principales, se tendrá una red de pantallas que permitirán verificar si el producto tiene sentido para satisfacer la necesidad educativa.
2. *Corrección del prototipo*: en este tipo de materiales se debe dejar abierta la posibilidad de realizar ajustes y revisiones en pro de ir logrando por aproximaciones sucesivas mejoras hasta obtener lo deseado.

Implementación

Una vez que se dispone de un diseño debidamente documentado se lleva a cabo el diseño computacional. Se especifica el tipo de software y hardware a emplear.

Eje transversal de Evaluación

La evaluación se debe hacer constantemente. Hay una evaluación continua independientemente de la fase, esta evaluación se hace en función de los resultados que se van obteniendo durante todo el proceso. Por ejemplo en la fase de diseño educativo se evalúan a nivel de expertos en contenidos.

Población y Muestra

Para el estudio y análisis, se determinó que el objeto de observación o población será el conjunto de estudiantes y profesores de la asignatura de estructura de datos de la carrera de ingeniería informática de la UNET. Según Ander (citado en Balestrini 1997), una población o universo de estudio es “la totalidad de un conjunto de elementos, seres u objetos que se desea investigar y de la cual se estudiará una fracción (la muestra) que se pretende que reúna las mismas características y en igual proporción”. (p.124).

Debido al pequeño tamaño que tiene la población (120 alumnos aproximadamente y 2 profesores), no es necesario la extracción de una muestra para la observación y análisis, en este caso conviene recopilar información de la totalidad de la población, es decir se aplica un muestreo censal, el cuál según Churchill (2003) lo define como “la investigación que abarca toda la población” (p. 448).

Técnicas e instrumentos de Recolección de Datos

En función de los objetivos definidos en el presente proyecto se empleará un conjunto de instrumentos y técnicas de recolección de datos, orientadas de manera esencial a alcanzar los fines propuestos.

Entre las técnicas a usar en el proyecto, tenemos:

- ✓ *La entrevista estructurada*, aplica a los profesores de la asignatura, con el propósito de obtener toda la información relevante del proceso de enseñanza. Según Bernal, Salavarrieta, Sánchez & Salazar (2006):

> La entrevista es una técnica orientada a establecer contacto directo con las personas que se consideran fuente de información. A diferencia de la encuesta, que se ciñe a un cuestionario, la entrevista, si bien puede soportarse en un cuestionario muy flexible, tiene como propósito obtener información más espontánea y abierta. Durante la misma, puede profundizarse la información de interés para el estudio. (p.177)

- ✓ *El cuestionario*, aplica a los alumnos, con el propósito de obtener toda la información relevante sobre como aprender y que dificultades son más comunes.

Análisis e Interpretación de los resultados

Para la fase de determinación de necesidades o diagnóstico, se aplicarán un conjunto de instrumentos de recolección de datos a estudiantes y profesores de la asignatura. Después se procederá a organizar la información en gráficos histográficos que sirvan de base para la fase de análisis e interpretación de los hallazgos, con el propósito de dar respuesta a las interrogantes planteadas sobre la problemática y situación actual en la enseñanza de la asignatura, evidenciar los aspectos que estaban ocultos en el proceso enseñanza aprendizaje y conocer las pautas sobre los que se basará el diseño lógico y la construcción del nuevo software.

Otro recurso para el análisis de alternativas es la aplicación de matrices de decisión, las cuales con base a ponderaciones efectuadas a diversos aspectos, de distinta importancia, relativos a cada alternativa sirven para seleccionar la alternativa más conveniente para la realización del proyecto.

Considerando que la metodología seleccionada permite trabajar de manera simultánea en fases distintas, el análisis e interpretación se realiza al inicio del proyecto para identificar las necesidades en el diagnóstico, casi de manera paralela se analiza los aspectos técnicos para la construcción del software y al final del proyecto las alternativas técnicas funcionales que garanticen el éxito.

CAPÍTULO IV

DESARROLLO DEL PROYECTO

Diagnóstico

Aspecto	Variables	Interrogantes
Compromiso del Estudiante	Tiempo de autoestudio Asistencia a clases Cumplimiento de tareas	.¿Cuantas horas dedica a autoestudio? .¿Cuantas horas en promedio asiste? .¿Qué porcentaje de tareas en promedio entrega?
Habilidades Iniciales.	Experiencia anterior Acceso a la información	.¿Que utilidad tiene la experiencia adquirida en las asignaturas previas? .¿Cual es el nivel de acceso a la información útil para la asignatura?
Operatividad	Medios de estudio Adecuación del material Completitud del material Complejidad en contenido programático	.¿Cual es el medio más utilizado para estudiar la asignatura? .¿Los materiales de estudio: guías, libros, diapositivas, etc., son suficientes para asimilar la asignatura? .Señale la unidad del programa que le fue más difícil conseguir material útil para el aprendizaje . Señale la unidad del programa que le fue más difícil para asimilar
Interacción	Participación en clase Canales de comunicación	.¿Cual es su nivel de intervención en las clases de la asignatura? .¿Cual es el canal de comunicación mas efectivo entre Usted y el profesor?
Orientación	Actividades remediales Conducción del aprendiz	.Posterior a la evaluación, ¿Ud. recibió orientación sobre los contenidos que debe reforzar? .¿Ud. recibió orientación para estudiar?

Tabla Nro. 2. Cuadro de variables de estudio del diagnóstico
Fuente: El autor.

Diseño del Cuestionario

La encuesta ó cuestionario fue aplicada en el lapso académico 2008-3 a los estudiantes de las secciones 1, 2, 3 y 4 de la asignatura Estructura de Datos. El tiempo de aplicación de esta encuesta fue de 20 minutos aproximadamente. El contenido de la encuesta se detalla en el anexo 1.

Para asegurar el propósito de la encuesta, posterior a la construcción de los ítems, la encuesta fue evaluada y validada por un experto en el área, para hacer los ajustes necesarios. La validación del instrumento se detalla al final del anexo 1.

Diseño de la Entrevista

Dirigida a los docentes que imparten la asignatura Estructura de Datos. Tiempo aproximado de aplicación 25 a 30 minutos.

- ✓ ¿Cuántos semestres o años tiene impartiendo la asignatura?
- ✓ ¿Qué dificultades se presentaron en la enseñanza de la asignatura?
- ✓ ¿Qué recursos usted aplicaría en el futuro para mejorar la enseñanza?
- ✓ De su opinión acerca de los software educativos
- ✓ ¿Cuál(es) unidad(es) del programa de estudio de la asignatura usted considera compleja para los estudiantes?
- ✓ Comentarios adicionales.

Resultados de la aplicación de la Encuesta a los estudiantes

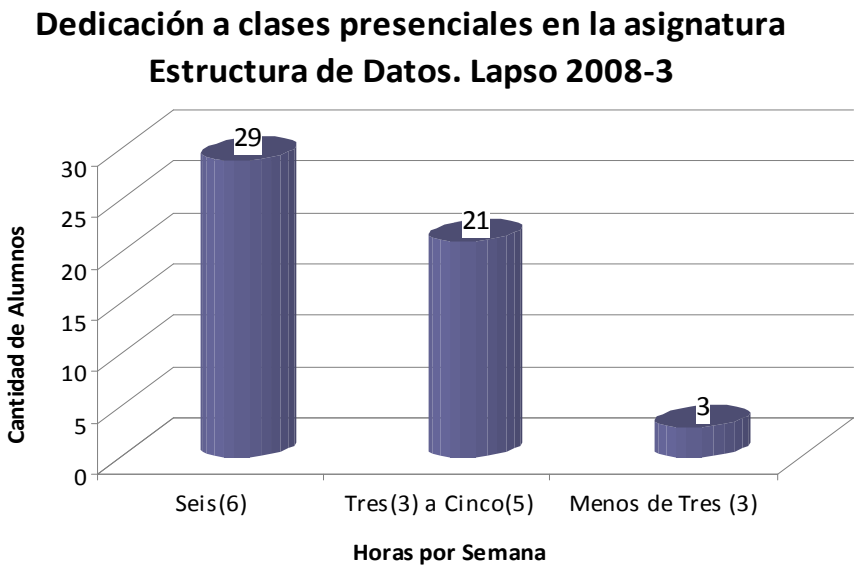


Gráfico Nro. 2. Respuesta a la Pregunta 1 de la Encuesta
Fuente: El Autor

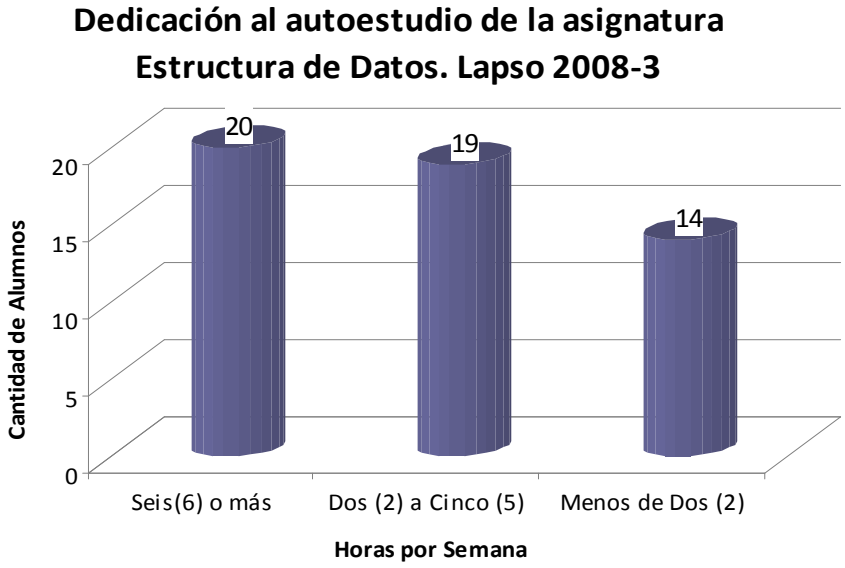


Gráfico Nro. 3. Respuesta a la Pregunta 2 de la Encuesta
Fuente: El Autor

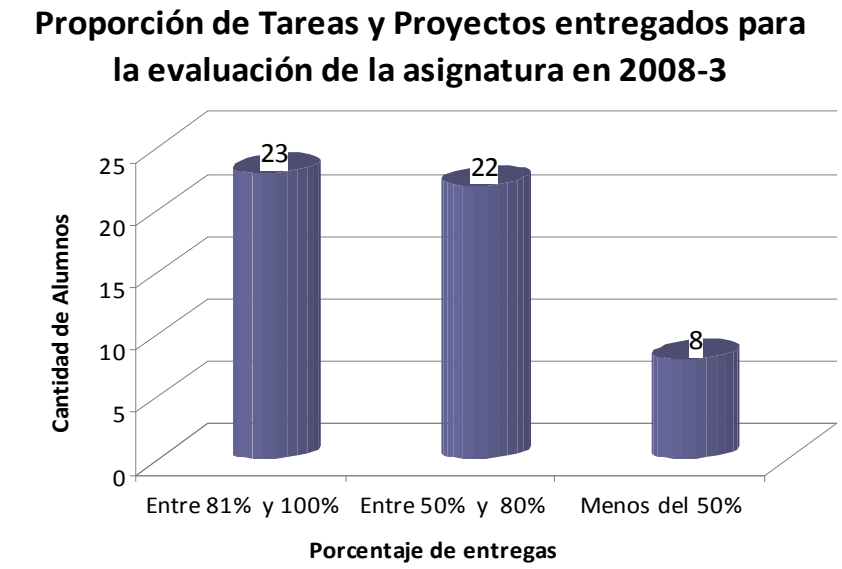


Gráfico Nro. 4. Respuesta a la Pregunta 3 de la Encuesta
Fuente: El Autor

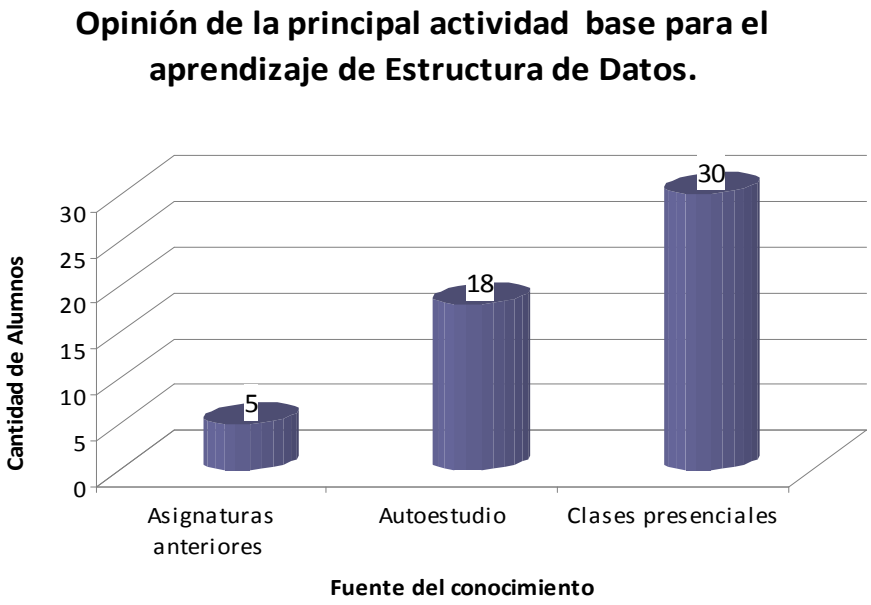


Gráfico Nro. 5. Respuesta a la Pregunta 4 de la Encuesta
Fuente: El Autor

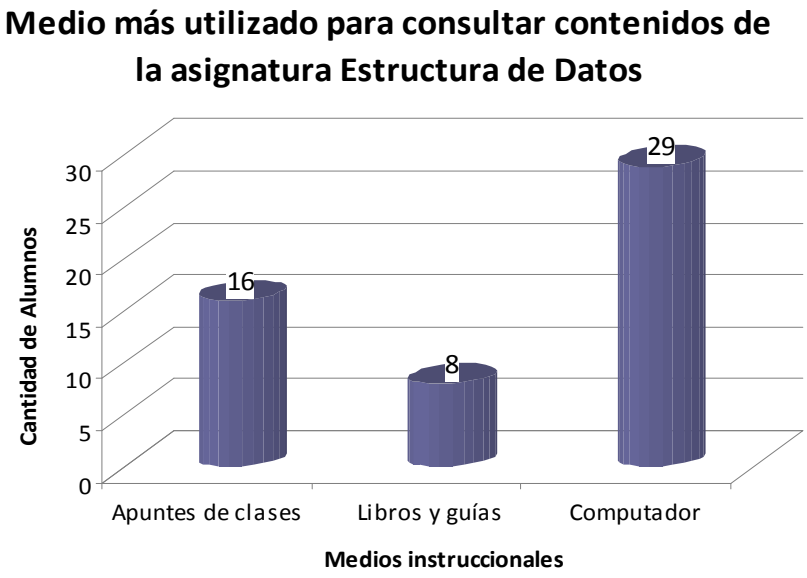


Gráfico Nro. 6. Respuesta a la Pregunta 5 de la Encuesta
Fuente: El Autor

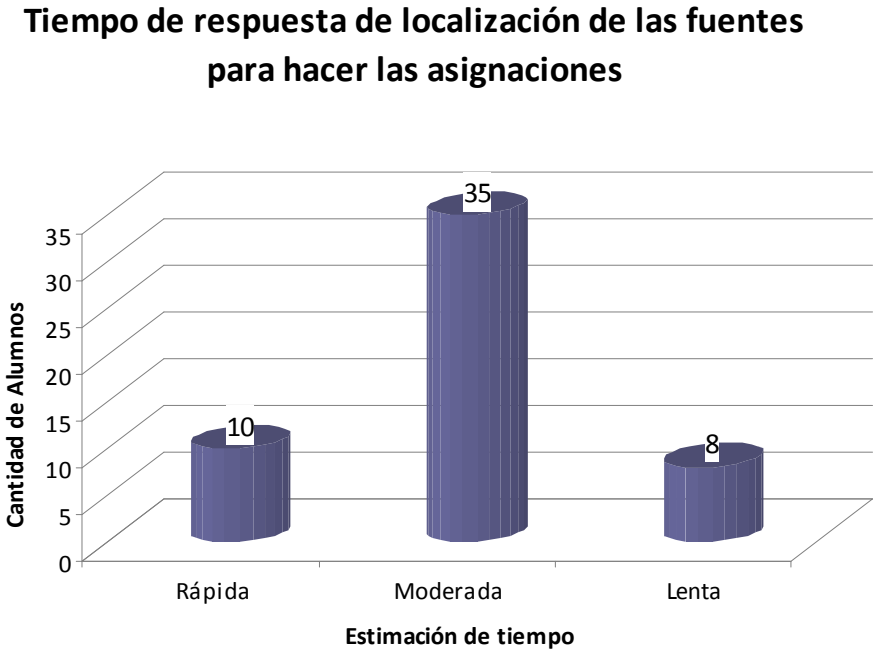


Gráfico Nro. 7. Respuesta a la Pregunta 6 de la Encuesta
Fuente: El Autor

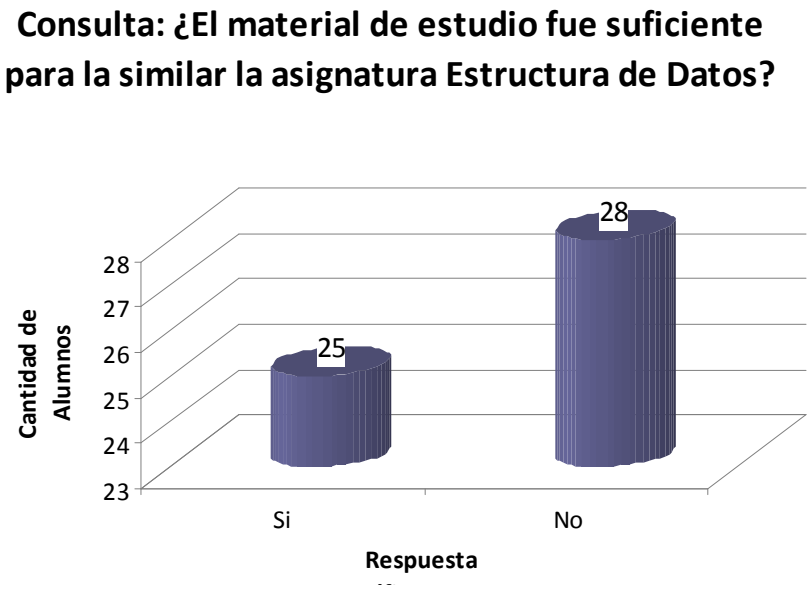


Gráfico Nro. 8. Respuesta a la Pregunta 7 de la Encuesta
Fuente: El Autor

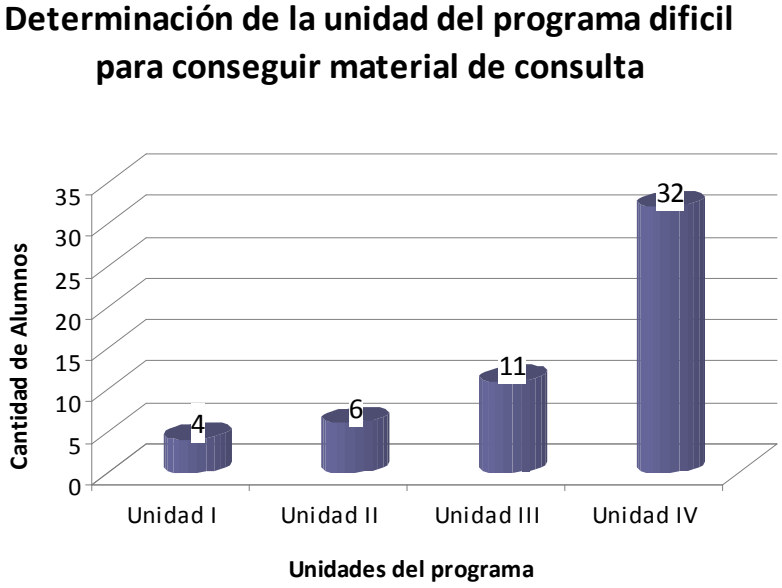


Gráfico Nro. 9. Respuesta a la Pregunta 8 de la Encuesta
Fuente: El Autor

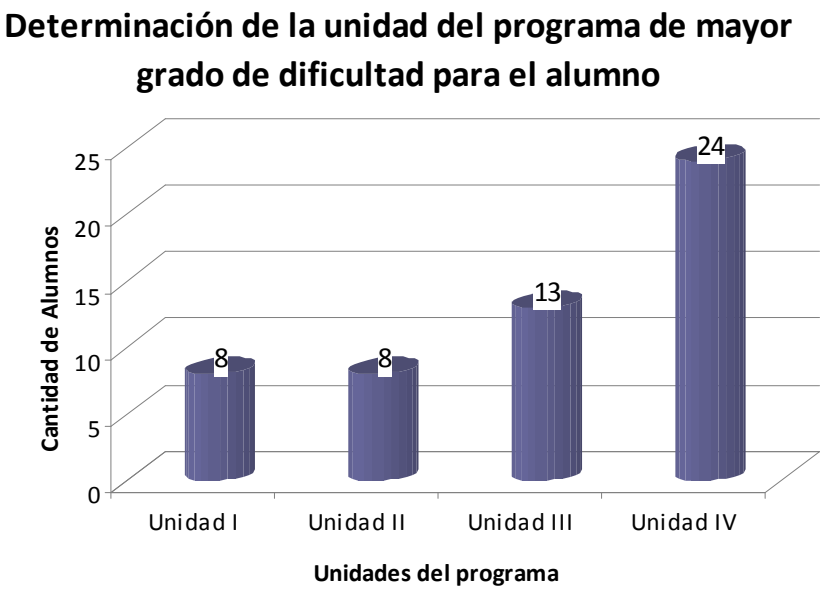


Gráfico Nro. 10. Respuesta a la Pregunta 9 de la Encuesta
Fuente: El Autor

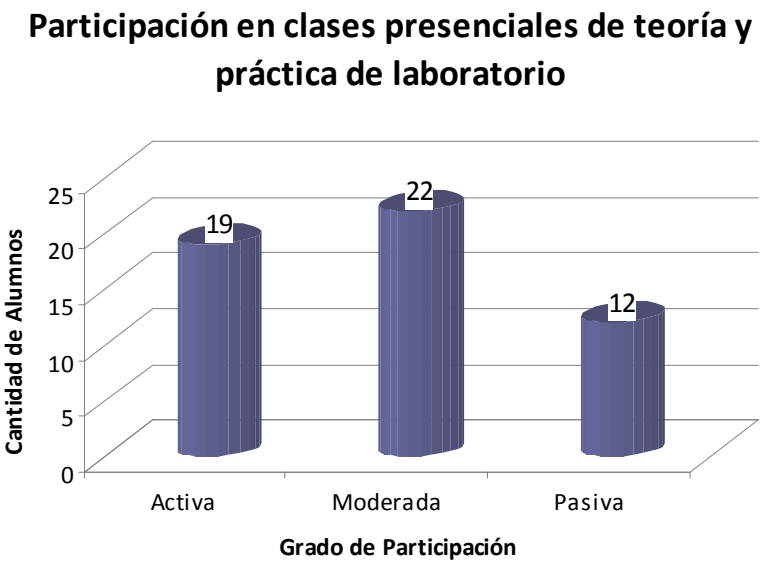


Gráfico Nro. 11. Respuesta a la Pregunta 10 de la Encuesta
Fuente: El Autor

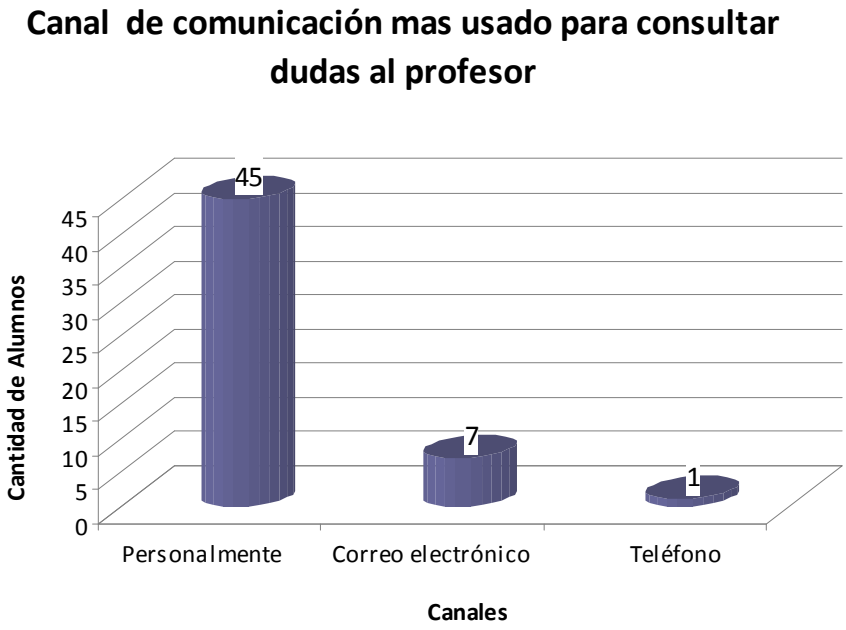


Gráfico Nro. 12. Respuesta a la Pregunta 11 de la Encuesta
Fuente: El Autor

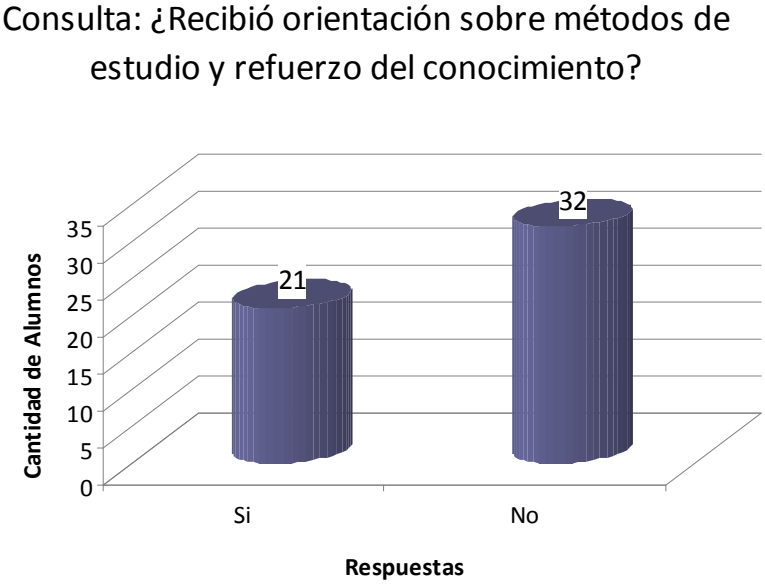


Gráfico Nro. 13. Respuesta a la Pregunta 12 de la Encuesta
Fuente: El Autor

Consulta: ¿ Recibió orientación para superar las debilidades y fallas detectadas en la evaluación?

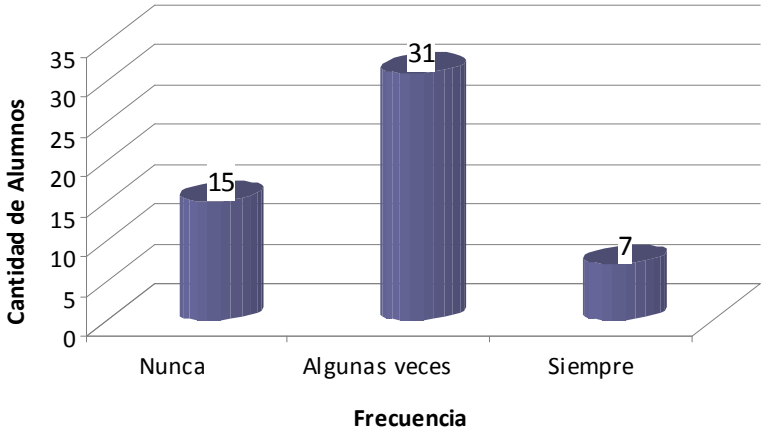


Gráfico Nro. 14. Respuesta a la Pregunta 13 de la Encuesta
Fuente: El Autor

Consulta: ¿El uso de las TIC contribuye a solucionar problemas de enseñanza de Estructura de Datos?

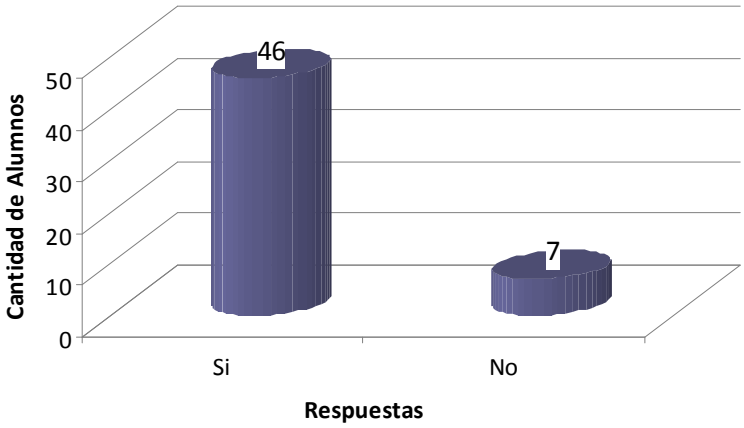


Gráfico Nro. 15. Respuesta a la Pregunta 14 de la Encuesta
Fuente: El Autor

Diagnóstico basado en los resultados de la encuesta y entrevistas

El estudiante de Estructura de Datos no asiste regularmente a clase presencial por diversos motivos. Alrededor del 45% de los estudiantes no asisten a todas las seis horas de clase semanales, a pesar de considerar las clases de teoría y práctica como la principal actividad para lograr el aprendizaje. El estudiante dedica tiempo suficiente para consultar y realizar las asignaciones, pero una la mayor parte consideran que los materiales: libros, guías, entre otros, no son suficientes para la consulta y apoyo en la realización de las tareas. Este problema va aumentando a medida que el estudiante avanza en las unidades programáticas.

Existe una relación directamente proporcional entre la dificultad de conseguir material y la dificultad para asimilar los contenidos de las unidades del programa de estudio de la materia, no obstante los profesores opinan que este último está elaborado en grado ascendente de dificultad. El estudiante por lo general supera las dificultades de acceso y los altos tiempos de respuesta en la búsqueda de información útil para cumplir con las entregas y prepararse para las evaluaciones parciales.

Muy pocos atribuyen su éxito en estructura de datos a los conocimientos adquiridos en asignaturas anteriores. Según los profesores un problema presente es el uso de la memoria a corto plazo de los estudiantes, esto se manifiesta cuando exclaman frases tales como: “No recuerdo el tema…en el semestre pasado “. Por tal motivo los profesores adoptaron medidas como la nivelación al inicio de un tema nuevo y la investigación de aspectos básicos fuera de las horas de clase.

En las clases presenciales la participación de la mayoría de los estudiantes es más activa que pasiva, prefieren aclarar las dudas en frente del profesor más que por correo electrónico, demuestran interés en aprender sobre todo en la práctica de laboratorio de computación, más que en el aula de teoría. La mayoría opina que el computador es el principal medio de consulta para las tareas, más que los libros y las guías impresas. Si se

aplicara las TIC de manera eficiente podría mejorar la problemática de enseñanza en la asignatura, usando como principal herramienta de trabajo el computador “B-Learning”, minimiza las barreras geográficas y refuerza lo aprendido en las clases.

Otro problema es la poca orientación sobre métodos de estudio para la asignatura y como superar las fallas detectadas en una evaluación. La opinión de los profesores al respecto señala que algunos estudiantes no asisten a la revisión de la evaluación parcial después de publicar los resultados, otros que si asisten solo les interesa la calificación y muy pocos preguntan en que fallaron y que actividades deben hacer para remediar y poder avanzar.

De acuerdo al estudio realizado, el panorama actual en la asignatura estructura de datos presenta un cúmulo de problemas con muchas aristas que deben ser atendidos minuciosamente. Entre las propuestas de solución que se pueden adoptar se encuentran:

1. Crear materiales de enseñanza actualizados bajo la modalidad de software educativo tales como: tutorial, simulador, libros electrónicos, entre otros. Sobre todo para los temas finales del programa.
2. Incluir métodos de medición al finalizar un objetivo instruccional del software educativo, que le permitan al estudiante tener una referencia de su avance en la materia.
3. Crear el aula virtual de la asignatura Estructura de Datos (uso de las TIC), aprovechando la plataforma tecnológica que ofrece el CETI para administrar cursos de educación a distancia.
4. Coordinar con la unidad de Orientación de la UNET un estudio de aptitud vocacional de los estudiantes de la carrera ingeniería informática de primer a tercer semestre.
5. Diseñar con la unidad de Orientación de la UNET curso de métodos de estudio para la carrera.

6. Diseñar cursos para los docentes: metodología para enseñar a programar computadores, uso de las TIC, investigación en el aula de clase, entre otros.

Metodología dinámica para el desarrollo de software educativo

Fase 1: Diseño Educativo ó Diseño instruccional

1. **Estudio de Necesidades**: Con base en el diagnóstico realizado con los estudiantes y profesores de la carrera de ingeniería informática de la UNET se pudo evidenciar problemas tales como:
   - **a.** En el lapso académico recién finalizado, cerca del 45% de los estudiantes no asistieron a todas las clases presenciales.
   - **b.** Los materiales de apoyo actuales tales como: libros, guías, diapositivas, entre otros, no son suficientes para asimilar los contenidos de la asignatura estructura de datos.
   - **c.** La unidad del programa titulada “Búsqueda heurística” es considerada como un área de estudio con muy poco material de apoyo adecuado al nivel del estudiante para la consulta.
   - **d.** Cerca del 90% de los estudiantes opinan que el uso efectivo de las TICs puede contribuir a mejorar los problemas de enseñanza de la asignatura Estructura de Datos.
   - **e.** Los profesores de la asignatura opinan que el estudiante de hoy en día se interesa por aprender de manera rápida y cómoda al ritmo de su vida actual. Pero en contraposición, se desmotiva fácilmente cuando no consigue pronta respuesta a sus inquietudes. En conclusión se puede afirmar que para mejorar estas situaciones, es conveniente desarrollar nuevas estrategias de enseñanza aprendizaje dirigidas a la creación de instrumentos de trabajo tales como un software educativo que sirvan de apoyo al estudiante investigador.

2. **Descripción del aprendiz:** el estudiante de Ingeniería Informática de la UNET, en la mayoría de los casos es una persona motivada por el uso y aprendizaje de nuevas tecnologías, con principios y conductas propias de un adulto joven, edad promedio 21 años. Orientado más hacia lo práctico que a lo teórico, predispuesto a trabajar largas jornadas para lograr los objetivos instruccionales que se le imponen.

3. **Propósito:** El propósito del software educativo reseñado en el presente trabajo es proporcionar al estudiante una herramienta de estudio, exploración y consulta de información básica en la asignatura Estructura de Datos, que le permita reforzar el conocimiento adquirido en clase presencial y aplicar con efectividad los algoritmos, métodos y técnicas oportunas para resolver problemas de lógica, programación, computación y automatización de procesos.

4. **Objetivo General:** Facilitar el aprendizaje de la asignatura estructura de datos en la carrera de ingeniería informática de la UNET a través de un patrón preestablecido de enseñanza con el computador.

5. **Objetivos Terminales o de Unidades Instruccionales:**
   - **a.** Enseñar los principales métodos de ordenación para conjuntos de datos en memoria principal: arreglo y en dispositivos de almacenamiento secundario: archivo binario.
   - **b.** Enseñar los principales métodos de búsqueda en conjuntos de datos en memoria principal: arreglo y en dispositivos de almacenamiento secundario: archivo binario.
   - **c.** Enseñar la funcionalidad de la estructura de dato Pila y su aplicación en la resolución de problemas.

d. Enseñar la funcionalidad de la estructura de dato Cola y su aplicación en la resolución de problemas.

e. Enseñar la funcionalidad de la estructura de dato Lista y su aplicación en la resolución de problemas.

f. Enseñar la funcionalidad de la estructura de dato Árbol y su aplicación en la resolución de problemas.

g. Enseñar la funcionalidad de la estructura de dato Grafo y su aplicación en la resolución de problemas.

h. Enseñar como aplicar heurística en la selección de la mejor alternativa del árbol min max para juegos de dos contrincantes.

6. Conocimientos previos y/ó Conducta de Entrada: El estudiante que se inicie en la asignatura estructura de datos empleando como medio de aprendizaje el software educativo deberá ser una persona con habilidades para:

a. Leer e interpretar correctamente un texto escrito en lenguaje claro y preciso del área.

b. Analizar un problema por descomposición en partes.

c. Diseñar un algoritmo básico con variables e instrucciones que resuelvan un problema.

d. Programar usando técnicas estructuradas y orientada a objetos.

Además el nuevo estudiante debe conocer sobre:

a. Pseudocódigo y corrida en frío.

b. Variables lógicas, operadores lógicos y tablas de verdad

c. Lenguaje de programación C/C++.

d. Al menos un entorno de desarrollo de aplicaciones (C++Builder, Dev C++, Code Blocks, Visual Studio ú otro): Edición, almacenamiento, compilación, enlace, construcción y ejecución de un programa.

7. **Objetivos Específicos:**

   a. El alumno estará en condiciones de explicar el algoritmo básico de un método o rutina de la estructura de datos vista en la lección.

   b. Una vez finalizado un tema de la unidad instruccional, el alumno podrá identificar situaciones de la vida real como ejemplo de aplicación de una estructura de dato ó método objeto de estudio.

   c. Una vez finalizada una unidad instruccional el estudiante deberá demostrar los conocimientos adquiridos programando rutinas reutilizables en lenguaje C/C++ sin errores y con eficiencia.

   d. El alumno después de estudiar varias lecciones del software y ejercitar en las clases prácticas presenciales, estará en condiciones de leer, comprender y diseñar programas con plantillas de clases, paradigma de programación genérica.

8. **Estrategias Instruccionales:** La estructura de cada lección del software comprende 3 aspectos:

   a. Conceptualización: introducción, reseña histórica, exposición teórica de conceptos, características, ventajas, desventajas ejemplos de aplicación e importancia.

   b. Explicación funcional: video ó animación de los pasos o algoritmo básico. El estudiante podrá hacer entradas de datos, parar, retroceder o adelantar.

   c. Codificación de un estándar: La instrucción finaliza con la presentación de una plantilla de clase (recurso de programación genérica de C++) para incorporar a una librería personal a ser usada en cualquier programa que la requiera.

9. **Contenidos a desarrollar:**

   1. Plantillas de Función y de clase

   2. Método de ordenación interna:

3. Burbuja Clásica, Burbuja con señal
4. Shaker sort
5. Selección directa
6. Inserción directa.
7. Ordenación Shell
8. Recursividad
9. Divide y vencerás
10. Merge Sort para arreglos
11. Quicksort
12. Métodos de búsqueda interna:
13. Secuencial simple. Con ordenación / Sin ordenación
14. Secuencial por bloques o saltos
15. Búsqueda binaria
16. Métodos de ordenación externa:
17. Selección directa para archivos binarios
18. Mezcla directa
19. Mezcla equilibrada
20. Búsqueda externa:
21. Búsqueda Secuencial para archivos.
22. Búsqueda Binaria para archivos.
23. Búsqueda por Transformación de clave. Dispersión ó Hashing.
24. Estructuras de datos Lineales:
25. Pila LIFO en arreglo
26. Cola FIFO en arreglo
27. Cola de frente fijo
28. Cola de frente variable o en array circular
29. Listas simples en arreglo
30. Listas encadenadas en arreglo
31. Listas en almacenamiento dinámico. Listas enlazadas.
32. Listas simples, dobles, circulares simples y circulares dobles

33. Listas como pilas. Listas como colas.

34. Bicola dinámica.

35. Estructuras de datos anidadas o compuestas:

36. Lista de lista: Multilistas

37. Lista colas: Cola de prioridad

38. Montículos. Heapsort

39. Estructuras de datos no Lineales:

40. Árbol Binario

41. Recorrido de un árbol en anchura

42. Recorridos de un árbol en profundidad: inord, posord preord.

43. Árbol binario de búsqueda

44. Árbol Balanceado AVL

45. Grafos.

46. Recorridos de un grafo: en anchura o en profundidad

47. Algoritmos de ruta mínima.

48. Búsqueda heurística en la resolución de problemas

49. Árbol de decisión min-max

50. Aplicación de heurística min-max en juego usuario vs computador.

10. Estrategias de Evaluación:

a. Al final de cada unidad de aprendizaje, el estudiante deberá responder a un cuestionario con ítems de selección simple de al menos 3 alternativas.

b. Ejercicios de completación de líneas de código fuente escrito en lenguaje C/C++ bajo un contexto o situación previamente enunciada.

c. Dado un programa C/C++ de implementación de un método ó estructura de dato vista en la unidad de aprendizaje, el estudiante indicará el resultado o resultados parciales de la ejecución con los datos de prueba suministrados.

d. Dado un enunciado el estudiante deberá elaborar un programa escrito en C/C++ aplicando el contenido visto en la unidad de aprendizaje. El programa solución se enviará a la dirección de correo electrónica señalada para que el profesor lo corrija.

Fase 2: Producción

1. **Guión Didáctico:** En el anexo 2 se presenta el guión de la unidad instruccional número uno del tutorial titulada “Métodos de Clasificación ú Ordenación”. El Orden de aparición de los contenidos narrados en lenguaje sencillo señala la secuencia o línea de aprendizaje.

2. **Guión Técnico y Prototipo**: El anexo 3 muestra las pantallas del prototipo del tutorial a desarrollar en la fase de construcción.

Costos del proyecto

El proceso de fabricación del software educativo en la UNET según los lineamientos descritos en esta investigación no acarrea gasto ó nueva inversión de recursos económicos, ya que la universidad cuenta con laboratorios de desarrollo de proyectos multimedia dotados con el software y hardware de última generación.

La UNET desde hace 8 años, a través de su Coordinación de Teleinformática CETI, ha incorporado la educación virtual “U Virtual”, dando soporte técnico a todas las asignaturas de las distintas carreras que deseen incorporar un “aula virtual”, interactuando con el estudiante a través de las TIC. Este recurso permitirá no solo manipular el software educativo en un dispositivo como CD, sino también ofrecerá la posibilidad al estudiante de abordar las unidades de aprendizaje a través del portal de educación virtual llamado “UNET Virtual” ó “U Virtual”.

En contra a los costos, este proyecto de software educativo representa un aporte y junto con el uso de otras fuentes electrónicas e-books disminuyen en cierto grado las compras de libros de estructura de datos para consultas en biblioteca. La consecuencia final: disminución del uso del papel y por tanto la preservación del ambiente con sus árboles y plantas, hoy día amenazadas de extinción por la industria del papel.

Recursos del proyecto

Recurso Humano:

Para el desarrollo del proyecto software educativo “estructura de datos” es necesario conformar un equipo de trabajo interdisciplinario integrado por:

1. Un experto de contenido y didáctica. Coordinador de la asignatura.
2. Un diseñador gráfico. Asesor de Imagen e impacto visual.
3. Un programador de multimedia. Encargado de realizar e implantar.

En la UNET hacen vida universitaria personal académico y técnicos comprometidos con la investigación, de experiencia en las áreas de trabajo que requiere el proyecto, por lo tanto no es necesario contratar personal externo a la institución. Los roles de los integrantes 2 y 3 pueden ser asumidos por una persona.

Recursos materiales:

1. Material de oficina: Papel, lápiz, regla, borrador, calculadora, etc.
2. Libros y revistas de informática, área de programación y estructura.
3. Computadoras (Cantidad 3). Descripción del Hardware Básico: Procesador Intel dual core 2.0 Ghz , 1 Gb mínimo de RAM, Disco duro de 160 Gb con un mínimo de 40 Gb de espacio libre, acelerador de medios gráfico Intel, unidades de disco CD/DVD - RW, 2 Puertos USB 2.0, Puertos de Red (Internet) cable y tarjeta de red inalámbrica, Monitor LCD 19’ full color, altavoces y micrófono.

4. Impresora multifuncional (Cantidad 1).
5. Discos CD (Cantidad 1 caja)
6. Memoria USB ó Pen Drive de 4 Gb (Cantidad 3)
7. Internet / Intranet
8. Software básico: Sistema operativo Windows Vista ó XP. Microsoft Office, ActiveX, Adobe Reader, Java, Antivirus actualizado, etc.
9. Software de elaboración multimedia: Para seleccionar el software de fabricación se consideraron las siguientes alternativas entre otras:
   a. Microsotf Power Point
   b. VCL de C++ Builder
   c. Adobe Flash
   d. Macromedia Authorware

A continuación se presenta una matriz para evaluar las alternativas en función de las cualidades del producto si se desarrollara con esa tecnología: Velocidad de desempeño, Facilidad de operación, Capacidad de Interacción con el estudiante y versatilidad.

Aspecto / Alternativa	Velocidad (1-3)	Fácil operar (1-5)	Interactividad (1-7)	Versatilidad (1-5)
Microsoft PowerPoint	3	4	2	3
VCL de C++ Builder	2	3	4	1
Adobe Flash	2	2	5	5
Macromedia Authorware	2	3	6	3

Tabla Nro. 3: Selección del software con base a los aspectos / alternativas
Fuente: El autor.

La suma de las puntuaciones de cada aspecto señalado en la tabla da un total de 20 puntos. Las alternativas c y d obtuvieron la más alta puntuación catorce (14).

Emplear Adobe Flash para la realización del software es una buena elección, pues se logra un producto de buena calidad multimedia, en un formato universal, multiplataforma, empleado en la mayoría de animaciones y videos de páginas web y trabajos multimedia. Una desventaja de esta alternativa es el excesivo tiempo (horas) de trabajo para conseguir buenos resultados.

Otra buena alternativa es Macromedia Authorware, el cual es una herramienta de diseño multimedia de desarrollo rápido de popularidad en ambientes de informática educativa, pero solo corre bajo Windows, no es tan universal y versátil como Adobe Flash.

No se recomienda las opciones a y b, en razón de la poca interactividad y uso limitado de efectos de PowerPoint y la minimizada versatilidad y popularidad de las aplicaciones visuales de Builder.

CAPÍTULO V

CONCLUSIONES

El uso de la tecnología en la educación propicia el avance de los pueblos hacia el desarrollo. La informática educativa es un campo para la investigación aplicada en búsqueda de soluciones a distintos problemas académicos, desde un nivel micro, situaciones diarias del aula de clase, hasta un nivel macro, administración de nuevos curriculum educativos. Las universidades han entendido su papel en este sentido generando estrategias y dedicando recursos para mejorar, innovar y formar profesionales cada día más competentes.

El diseño del software educativo como el diseño de otros medios instruccionales modernos requiere especial atención y dedicación para obtener un producto de calidad, sustentado sobre un modelo pedagógico y un conjunto de novedosos recursos tecnológicos con propiedad comunicacional que faciliten la labor docente, capte la atención y satisfaga a los estudiantes ávidos de conocimiento. La planificación del software educativo para la enseñanza de Estructura de Datos se realizó en función de las necesidades, las personas, los recursos y los costos, para asegurar la factibilidad de su construcción e implementación.

Los estudiantes de la carrera de Ingeniería Informática y en especial los que se inician en la asignatura Estructura de Datos, se enfrenta a la incertidumbre del cómo estudiar para aprender la materia. El software tutorial diseñado representa una solución a esta inquietud, guía la labor de estudio y fortalece las nociones implantadas por el profesor en clases presenciales.

Ninguna herramienta tecnológica suprime totalmente la labor del docente en la enseñanza del programa a sus alumnos. El profesor líder se preocupa por su aprendiz y le provee diversos medios e instrumentos

acordes con su nivel y expectativas. El software educativo para la enseñanza de la asignatura Estructura de Datos no debe ser considerada la única forma de aprender, deber ser una vía que el profesor proporciona, así como otros medios tradicionales.

RECOMENDACIONES

- ✓ Revisar las experiencias de los estudiantes cuando interactúen con el tutorial, a fin de precisar las situaciones de aprendizaje más débiles.

- ✓ Actualizar el contenido del tutorial cada lapso semestral para evitar que el alumno entre en rutina ó mecanización que produzcan una estatización del conocimiento. Sobre todo en las actividades de evaluación.

- ✓ Preparar un plan de ejecución de la fase de construcción para aprovechar al máximo los recursos en función del tiempo.

- ✓ Diseñar un plan para implementar el software tutorial no solo en CD sino también en el espacio de aula virtual que tiene la UNET, denominado U – Virtual.

- ✓ Diseñar un manual del usuario profesor y otro para el usuario alumno.

- ✓ Implementar una prueba piloto del impacto del tutorial con grupo de control y grupo experimental.

REFERENCIAS BIBLIOGRÁFICAS

Arias, López y Honmy (2006). *Metodología dinámica para el desarrollo de software educativo.* [Documento en línea] Disponible: http://www.virtualeduca .org /virtualeduca/virtual/ actas2002/actas02/913.pdf [Consulta: 2008, Mayo 29].

Barrios, Maritza (1998). *Manual de trabajo de grado de especialización y maestría de tesis doctorales*. Caracas: Universidad Pedagógica Experimental Libertador

Balestrini Acuña, Miriam (1997). *Como se elabora el proyecto de investigación.* Caracas: BL Consultores Asociados.

Bernal, César; Salavarrieta, Duván; Sánchez Amaya, Tomás & Salazar Rosalba (2006). *Metodología de la investigación (2ª ed.).* Prentice Hall

Cataldi, Zulma (2000). *Metodología de diseño, desarrollo y evaluación de software educativo.* España. [Documento en línea] Disponible: http://www fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf [Consulta: 2008, Abril 30].

Churchill, Gilbert (2003). *Investigación de mercados (4ª ed.).* México: Thomson Paraninfo, S.A

Duart, Josep; Sangrá, Albert (2003). *Aprender en la virtualidad.* España: Editorial Gedisa.

García, Juan; Cañal, Pedro (1996), *Hacia una definición de las estrategias de enseñanza*. España. [Documento en línea] Disponible: http://www .benavente.edu.mx/archivo/mmixta/lect_opc/LO_EE.doc [Consulta: 2008, Mayo 1]

González Reyes, Fabio (2006) *Diseño de software educativo*. [Documento en línea] Disponible:http://www.mailxmail.com/curso/informatica/disenosoftware/ [Consulta: 2008, Mayo 15].

Gros, Begoña (2003). *El ordenador invisible. Hacia la apropiación del ordenador en la enseñanza.* España: Editorial Gedisa.

Gros, Begoña (2000). *Diseño de software educativo. Pautas pedagógicas para su construcción.* España: Editorial Gedisa.

Marqués, Pere (2000). *El software educativo.* Universidad Autónoma de Barcelona. España: [Documento en línea] Disponible: http://www.karisma.org.co/documentos/softwareredp/clasif_software_educativo_de_pere.doc. [Consulta: 2008, Abril 30].

Marqués, Pere (2000). *Diseño Instructivo de unidades didácticas.* España: [Documento en línea] Disponible: http://dewey.uab.es/pmarques/ud.htm [Consulta: 2008, Mayo 1]

Ministerio del Poder Popular para las Telecomunicaciones y la Informática (2008). *Directorio de Gobierno Electrónico Venezuela- TIC. ¿Regular o Desregular?* [Documento en línea] Disponible: http://www.gobiernoenlinea.ve/directorioestado/aspectos_legales.html#1 [Consulta: 2008, Abril 30].

Moreno, Bailliére (2002). *Diseño instructivo de la formación on-line. España*: Editorial Ariel.

Norton, Peter (2006). *Introducción a la Informática. (6ª ed.).* México: Editorial McGRAW-HILL.

Padrón G., José (1998). *La estructura de los procesos de investigación* [Documento en línea] Disponible: http://www.monografias.com/trabajos/estruprocinv/estruprocinv.shtml. [Consulta: 2008, Abril 30].

Senn, James A. (1992). *Análisis y diseño de sistemas de información (2ª ed.)*. México: McGRAW-HILL.

Universidad Nacional Experimental del Táchira (2009). *Nuestra Organización* [Documento en línea]Disponible:http://www.unet.edu.ve/nuestra_organizacion [Consulta: 2008, Mayo 1].

Universidad Nacional Experimental del Táchira (2009). *Departamento de Ingenieria en Informática* [Documento en línea] Disponible: http //docencia .unet.edu.ve/ Departamentos/DptoInfo.php [Consulta: 2008, Mayo 1].

Wright, Peggy; Mosser, Diane; Wooley, Bruce (2006), *Técnicas y Herramientas para Usar Color en el Diseño de la Interfaz de una Computadora.*[Documento en línea] Disponible: http://www.acm.org/crossroads/espanol/ xrds3-3/color.html [Consulta: 2008, Junio 30].

Whitten, Bentley & Barlow (1997). Análisis y Diseño de Sistemas de Información. (3ª ed.). Colombia: McGraw-HILL.

Zerpa, Ramírez (2001). *Proyecto de Información Ocupacional para Estudiantes de Ingeniería: Una Propuesta de Desarrollo en Ingeniería de Software Educativo.* Caracas [Documento en línea] Disponible: http://www.ucv.ve/edutec/Ponencias/50.doc [Consulta: 2008, Abril 30].

ANEXOS

Anexo 1. Encuesta

Instrucciones: A continuación se presentan una serie de preguntas a las cuales usted responderá voluntariamente seleccionando solo una alternativa con la mayor sinceridad. La finalidad es obtener información importante para el mejoramiento académico en la asignatura Estructura de Datos. Tiempo aproximado de aplicación 10 minutos.

Edad del Estudiante: _____ años

1. Indique en promedio la cantidad de horas semanales que Usted asistió a clases de la asignatura Estructura de Datos:
   Seis (6) ____
   Entre tres (3) y cinco (5) ____
   Menos de tres (3) ____

2. Señale en promedio la cantidad de horas semanales que Usted dedicó al autoestudio de la asignatura Estructura de Datos:
   Seis (6) ó mas ____
   Entre dos (2) y cinco (5) ____
   Menos de dos (2) ____

3. Al término del semestre, que porcentaje de tareas y proyectos Usted entregó para su evaluación en la asignatura:
   Entre 81% y 100% ____
   Entre 50% y 80% ____
   Menos del 50% ____

4. El logro de los objetivos de aprendizaje en Estructura de Datos se debe mayormente al:

Aprendizaje en Programación I ____

Aprendizaje autodidacta ____

Aprendizaje en clases teoría/práctica ____

5. El medio más utilizado por Usted para consultar contenidos de la asignatura Estructura de Datos fue:

Apuntes de clases ____

Libros y guías ____

Computador ____

6. Para cumplir con los compromisos efectivamente y las asignaciones de la asignatura Usted encontró las fuentes de información de forma:

Rápida ____

Moderada ____

Lenta ____

7. Los materiales de estudio utilizados como: guías, libros, programas, diapositivas, etc., ¿fueron suficientes para asimilar la asignatura Estructura de Datos?

Si ____

No ____

8. Señale la unidad del programa que le fue más difícil conseguir material útil para estudiar:

I Métodos de ordenación y búsqueda ____

II Estructuras de datos lineales ____

III Estructura de datos no lineales ____

IV Búsqueda heurística ____

9. Señale la unidad del programa que le fue más difícil asimilar:

I Métodos de ordenación y búsqueda ____

II Estructuras de datos lineales ____

III Estructura de datos no lineales ____

IV Búsqueda heurística ____

10. Indique como fue su participación en las clases presenciales de teoría y práctica de laboratorio:

Activa ____

Moderada ____

Pasiva ____

11. Indique el canal de comunicación mas usado entre Usted y el profesor de la asignatura para aclarar dudas:

Persona a persona ____

Correo electrónico ____

Teléfono ____

12. ¿Usted recibió orientación personalizada sobre métodos de estudio y actividades de complemento de la asignatura?

Si ____

No ____

13. Posterior a la revisión de la evaluación, ¿Usted fue orientado para superar las debilidades y fallas detectadas?

Nunca ____

Algunas veces ____

Siempre ____

14. ¿Considera Ud. que el uso de las nuevas TIC contribuyen a resolver en buena parte los problemas de enseñanza de la asignatura Estructura de Datos?

Si ____

No ____

Porque:

_____

_____

_____

Comentarios adicionales:

_____

_____

¡Gracias por su atención!

San Cristóbal, 2 de Febrero 2009

CERTIFICACIÓN

Yo María Nereyda Carrero Miranda, cédula de identidad 5.660.319, ingeniera de sistemas, especialista en software educativo, doctora en educación, profesora de la Universidad Nacional Experimental del Táchira, una vez examinado el instrumento encuesta al estudiante para recolectar datos en el trabajo especial de grado titulado “Software Educativo para la Enseñanza de la Asignatura Estructura de Datos de la Carrera Ingeniería Informática de la Universidad Nacional Experimental del Táchira“, del autor David E. Ortiz Vezga, Cédula 9.237.706, certifico que el diseño estructural y la redacción del contenido de cada ítem, cumple con los objetivos y normas para ser aplicado en concordancia con los fines propuestos en esta investigación.

Ing. María Nereyda Carrero
C.I. 5.660.319

Anexo 2. Ejemplo de Guión Didáctico

Métodos de Clasificación

Introducción

- La ordenación es una actividad tan fundamental, que todos llevamos a cabo un proceso de ordenación de datos casi a diario.
- Imaginemos por un momento que deseamos buscar nuestra nota definitiva en el listado de notas que publica el profesor a final del semestre, ¿Qué seria más rápido? ¿Buscar en la lista sin ningún tipo de orden o Buscar en la lista ordenada?
- O por ejemplo si quisiéramos localizar una persona en la guía telefónica y los mismos están ordenados por número telefónico, ¿podríamos localizar la información en poco tiempo?

Tipos de ordenamiento

- Ordenamiento interno: Se lleva a cabo completamente en memoria principal. Todos los objetos que se ordenan caben en la memoria principal de la computadora (RAM).
- Ordenamiento externo: No cabe toda la información en memoria principal y es necesario ocupar memoria secundaria. El ordenamiento ocurre transfiriendo bloques de información a memoria principal en donde se ordena el bloque y este es regresado, ya ordenado, a memoria secundaria. Implica el manejo de archivos.

Métodos de Ordenación interna

- Métodos directos o cuadráticos:

- ✓ Intercambio directo (Burbuja Clásica)
- ✓ Burbuja con señal
- ✓ Sacudida (ShakerSort)
- ✓ Selección directa
- ✓ Inserción Directa

- Métodos logarítmicos :
  - ✓ ShellSort
  - ✓ Fusión con divide y vencerás
  - ✓ QuickSort con divide y vencerás

Ordenación por Burbuja

- El método de la Burbuja es el más utilizado por los estudiantes por su facilidad de codificación, pero quizás sea el método más ineficiente.
- Este método puede trabajar de dos maneras diferentes:
  - ✓ Llevando los elementos más pequeños hacia la parte izquierda del arreglo.
  - ✓ Llevando los elementos más grandes hacia la parte derecha del mismo.
- La idea básica de este algoritmo consiste en comparar pares de elementos adyacentes e intercambiarlos entre sí hasta que todos se encuentran ordenados.
- Se realizan (n-1) fases, transportando en cada una de las mismas el menor o mayor elemento (según sea el caso) a su posición ideal. Al final de las (n-1) fases los elementos del arreglo estarán ordenados

Ordenación por Burbuja Clásica (Explicación)

- Animación del método

Algoritmo para Burbuja Menor

```
INICIO
HAGA DESDE I=1 HASTA N-1
  HAGA DESDE J=N-1 HASTA I STEP -1
     SI A[J-1] > A[J] ENTONCES
        AUX   = A[J-1]
        A[J-1] = A[J]
        A[J]    = AUX
    FIN SI
  FIN HAGA DESDE
FIN HAGA DESDE
FIN
```

Codificación en C++ para un vector de números enteros

```
void bubbleSort(int *A, int N){
        int aux;
        for (int i=1;i<N;i++)
            for(int j=N-1;j>=i;j--)
                if(A[j-1]>A[j]){
                    aux   = a[j-1];
                    a[j-1] = a[j];
                    a[j]    = aux;
                }
}
```

Plantilla de función.

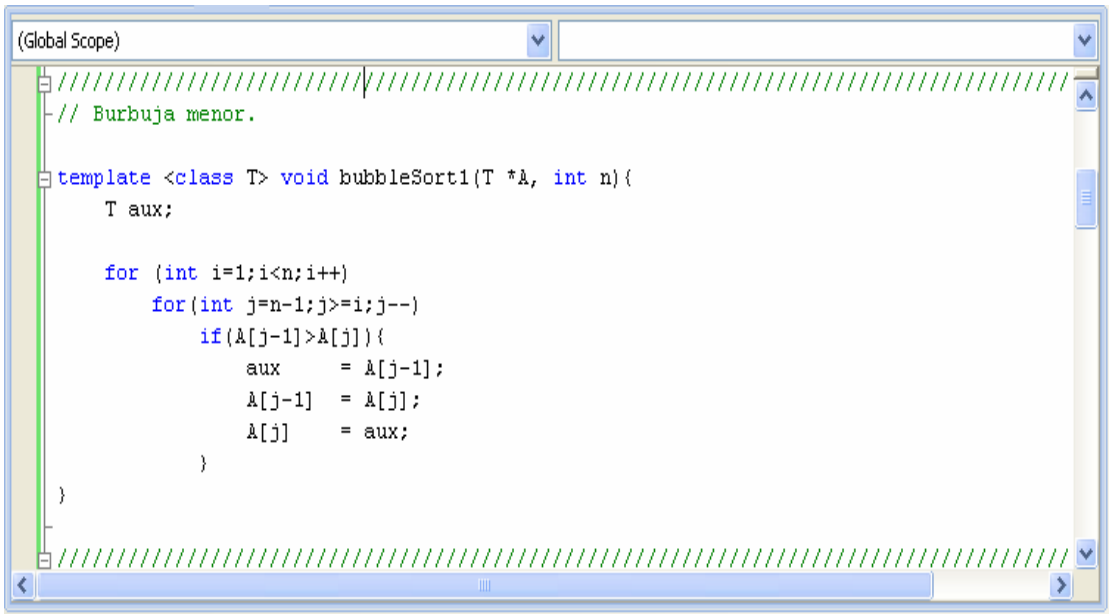
- El algoritmo propuesto anteriormente sirve para ordenar un vector de números enteros en forma ascendente, pero si queremos ordenar datos reales o algún tipo de dato estructurado tendríamos que escribir otra nueva función indicando como parámetros el tipo de datos que deseamos clasificar.
- Por esta razón se creó lo que se conoce en C++ como plantilla de función, la cual es una función parametrizada con un tipo de dato genérico (T).

- Una función genérica define un conjunto de operaciones que se aplicarán a diferentes tipos de datos.
- Una plantilla especifica un conjunto infinito de funciones que pueden ser aplicadas a distintos tipos de datos.
- Una plantilla describe las propiedades genéricas de una función.

Sintaxis

- Las plantillas de funciones o funciones genéricas tienen la sintaxis: template <class T> declaración de la función
- La palabra reservada template indica que se va a declarar una plantilla.
- La *lista de tipos* contiene los tipos genéricos separados por comas.
  Aunque la notación anteponga **class** al tipo, se refiere a cualquier tipo, sea clase o no (enteros, reales, etc.).

Plantilla de función para el algoritmo burbuja menor.

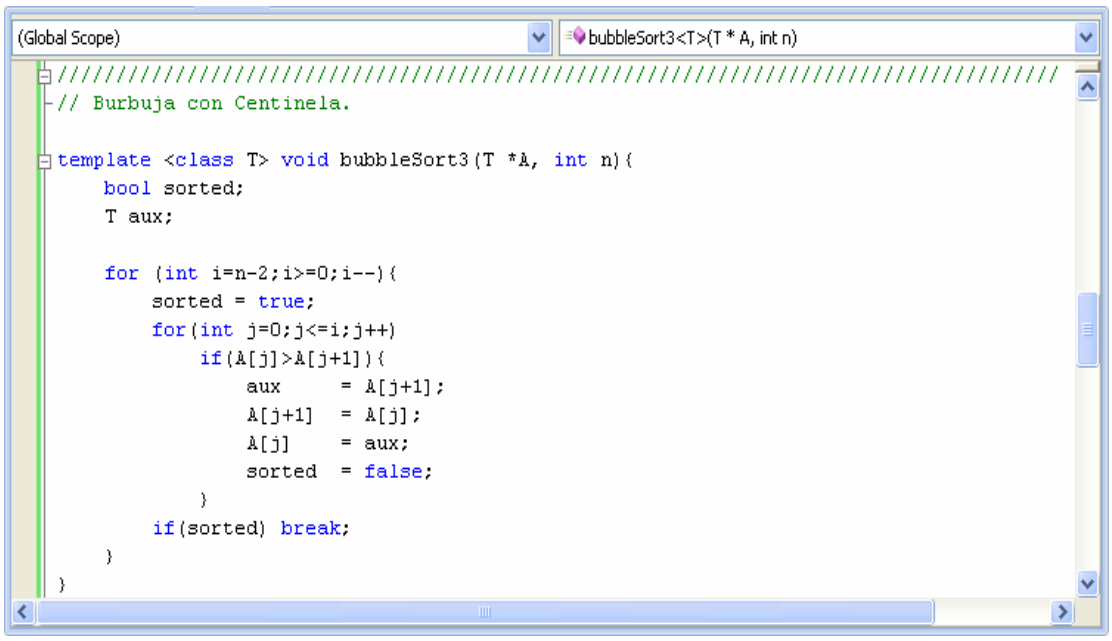


Burbuja con señal

- Este método es una modificación del método de burbuja analizado anteriormente. Se basa fundamentalmente en comprobar después de cada fase si el vector se encuentra ordenado (el arreglo esta ordenado si no hay intercambio de elementos), con lo cual terminara la ejecución.

```
INICIO
HAGA DESDE I=1 HASTA N-1
        ORDENADO = V.
    HAGA DESDE J=N-1 HASTA I
        SI A[J-1] > A[J] ENTONCES
          AUX   = A[J-1];  A[J-1] = A[J];   A[J]  = AUX
          ORDENADO = F.
        FIN SI
    FIN HAGA DESDE
    IF ORDENADO == V. ENTONCES TERMINAR
FIN HAGA DESDE
FIN
```

Codificación en C++ usando plantillas



Ordenación Shaker Sort (Sacudida)

- Este método es una optimización del método de intercambio directo (burbuja). Y consiste en mezclar las dos formas (burbuja menor y burbuja mayor) en que se puede realizar el método de la burbuja.
- El algoritmo tiene 2 etapas por cada fase:
  - ✓ De derecha a izquierda: se trasladan los elementos más pequeños hacia la parte izquierda del arreglo, almacenando en una variable la posición del último elemento intercambiado.
  - ✓ De izquierda a derecha: se trasladan los elementos más grandes hacia la parte derecha del arreglo, almacenando en otra variable la posición del último elemento intercambiado.
- Cada fase trabaja con los componentes del arreglo comprendidos entre las posiciones almacenadas en las variables.
- El algoritmo termina cuando en una fase no se producen intercambios o bien, cuando el contenido de la variable que almacena el extremo izquierdo del arreglo es mayor que el contenido de la variable que almacena el extremo derecho.

Método del Shaker Sort (Explicación)

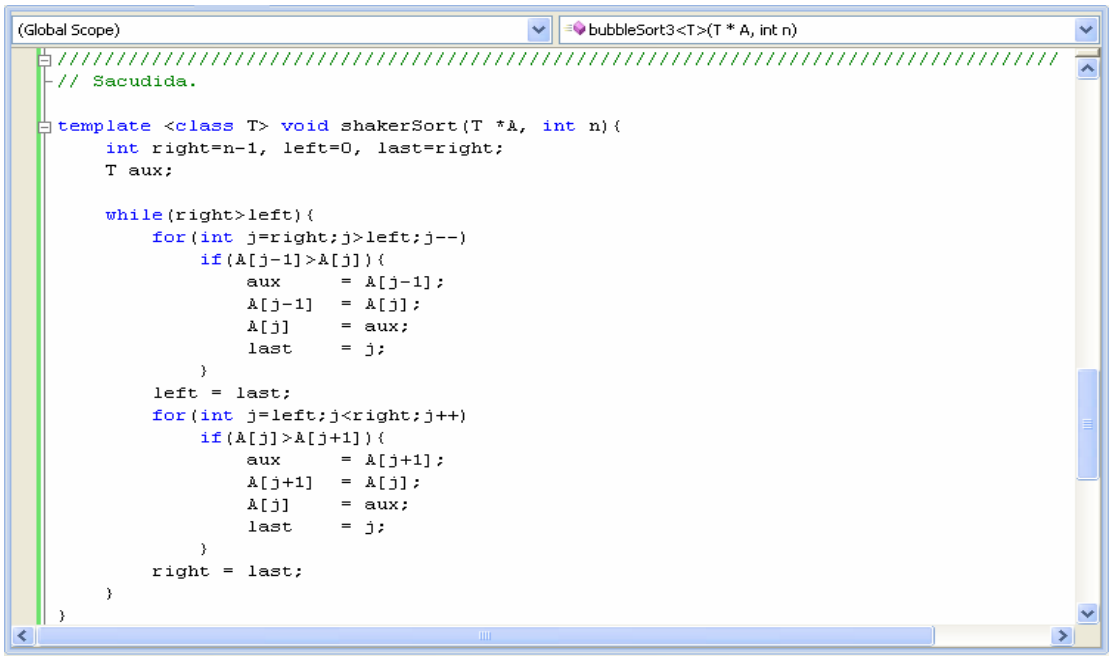
- Animación del método

Algoritmo para Shaker Sort

```
IZQ = 0, DER = N-1, K = DER
HAGA MIENTRAS (DER > IZQ)
  HAGA DESDE J=DER HASTA IZQ+1 STEP -1
    SI(A[J-1] > A[J]) ENTONCES
      AUX = A[J-1], A[J-1] = A[J], A[J] = AUX, K=J
    FIN SI
  FIN HAGA DESDE
```

```
  IZQ = K
  HAGA DESDE J=IZQ HASTA DER-1
   SI(A[J] > A[J+1]) ENTONCES
    AUX = A[J+1], A[J+1] = A[J], A[J] = AUX, K=J
   FIN SI
  FIN HAGA DESDE
  DER = K
FIN HAGA MIENTRAS
FIN
```

Codificación en C++ usando plantilla de función



Ordenación por Selección Directa

- Este algoritmo consiste en buscar el menor elemento en el arreglo y colocarlo en primera posición. Luego se busca el segundo elemento más pequeño del arreglo y se coloca en segunda posición. El proceso continua hasta llegar al final del vector.
- El método se basa en los siguientes principios:
  - ✓ Seleccionar el menor elemento del arreglo.

- ✓ Intercambiar dicho elemento con el primero.
- ✓ Repetir los pasos anteriores con los (n-1), (n-2) elementos y así sucesivamente hasta que sólo quede el elemento mayor.

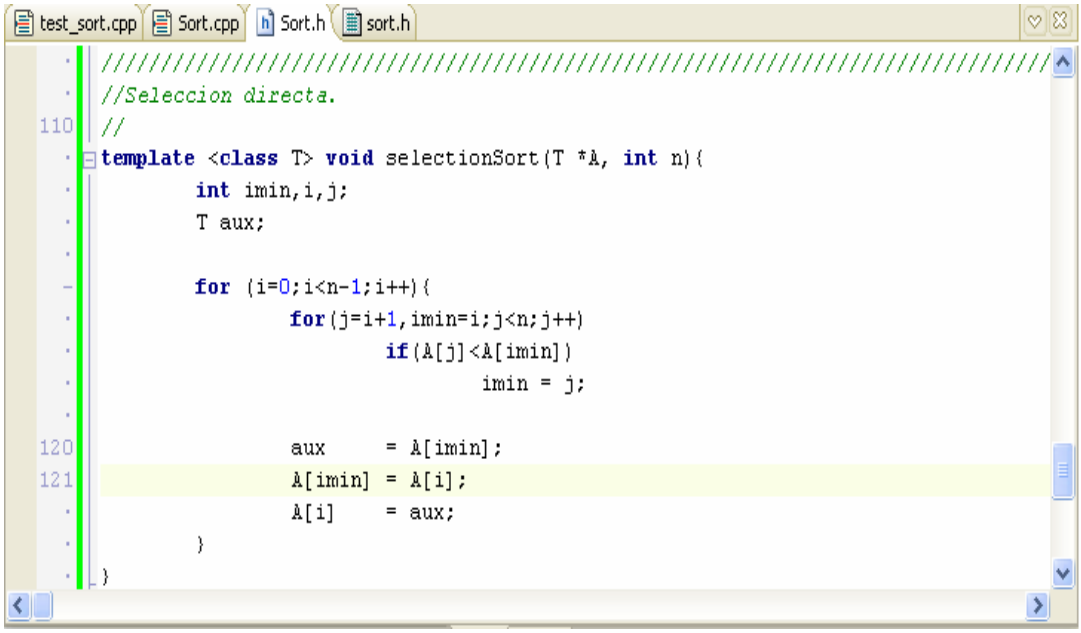
Ordenación por selección directa (Explicación)

- Animación del método

Algoritmo para el método selección directa

```
INICIO
HAGA DESDE I=0 HASTA N-2
    imin = I
    HAGA DESDE J=I+1 HASTA N-1
      SI(A[J]<A[imin]) ENTONCES
        imin = J
      FIN SI
    FIN HAGA DESDE
   aux = A[imin]
   A[imin]=A[i]
   A[i]=aux
FIN HAGA DESDE
FIN
```

Codificación en C++ para el método de selección directa



Ordenación por Inserción Directa

- El método de ordenación por inserción directa también es conocido por el método de la baraja, debido a que generalmente es utilizado por los jugadores de cartas cuando ordenan éstas.
- Este algoritmo consiste en insertar un elemento del arreglo en la parte izquierda del mismo, que ya se encuentra ordenada. Este proceso se repite desde el segundo hasta el n-ésimo elemento.

Inserción directa. Método de la Baraja (Explicación)

- Animación del método. La animación se basa en ordenar un conjunto de barajas o naipes insertando cada una según el método.

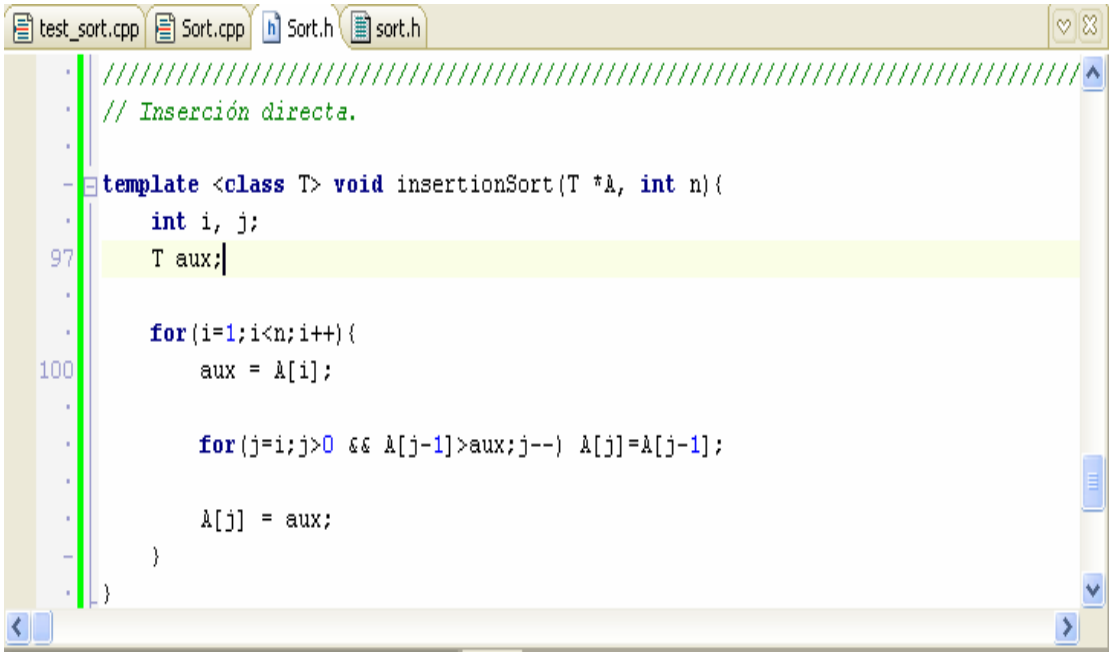
Ordenación por inserción directa (Explicación)

- Animación del método. La animación es similar a la anterior pero sustituyendo las barajes por elementos numéricos y mostrando las variables de control.

Algoritmo para el método de inserción directa

```
INICIO
HAGA DESDE I=1 HASTA N-1
  AUX = A[I], J=I
  HAGA MIENTRAS (J>0 ∧ A[J-1] > AUX)
    A[J]=A[J-1],
    J=J-1
  FIN MIENTRAS
  A[J] = AUX
FIN HAGA DESDE
FIN
```

Codificación del Método de Inserción directa en C++



Ordenación Shell

- Debe su nombre a su creador el Dr. Donald Shell el cual lo propuso en el año 1959. También se le conoce como inserción por incrementos decrecientes, ya que es una modificación del método de inserción directa.
- Este método consiste en subdividir el arreglo original en grupos de elementos más pequeños. Dichos grupos están compuestos por los elementos separados k posiciones.
- En seguida se aplica el método de ordenación por inserción directa en cada subgrupo.
- El proceso se repite empezando con una k = n/2 y se decrementa en cada iteración hasta llegar a k = 1.

Método Shell (Explicación)

- Animación del método

Algoritmo del Método de ordenación Shell

```
INICIO
K=N/2
HAGA MIENTRAS( K>0)
        HAGA DESDE I=K HASTA N-1
          AUX = A[I];  J=I
          HAGA MIENTRAS (J>=K ∧ A[J-K] > AUX)
             A[J]=A[J-K]; J=J-K
          FIN MIENTRAS
          A[J] = AUX
        FIN HAGA DESDE
        K=INT(K/2)
FIN HAGA MIENTRAS
FIN
```

Codificación del Método de Ordenación Shell

Sort.h

```
127 // Método de ordenación shell.

template <class T> void shellSort(T *A, int n){
130     int i, j, k;
        T aux;

        for(k=n/2;k>=1;k/=2)
            for(i=k;i<n;i++){
                aux = A[i];

                for(j=i;j>=k && A[j-k]>aux;j-=k) A[j]=A[j-k];

                A[j] = aux;
140         }
}
```

Recursividad

¿Qué es la recursividad?

- La recursividad es un concepto fundamental en matemáticas y en computación. Es una alternativa diferente para implementar estructuras de repetición (ciclos).
- Algo es recursivo si se define en términos de sí mismo (cuando para definirse hace mención a sí mismo). Para que una definición recursiva sea válida, la referencia a sí misma debe ser relativamente más sencilla que el caso considerado.

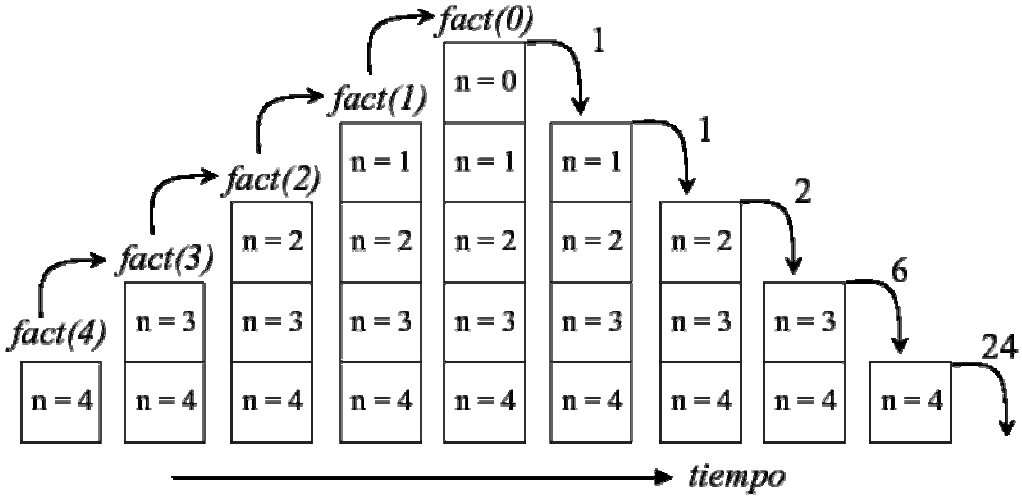
Función recursiva. Las funciones recursivas se componen de:

- Caso base: una solución simple para un caso particular (puede haber más de un caso base). Los casos bases corresponden a situaciones que se pueden resolver con facilidad.
- Llamada recursiva o auto invocación. Siempre se debe avanzar hacia un caso base. Las llamadas recursivas con parámetros que se acercan más al caso base simplifican el problema, y en última instancia los casos bases nos ayudan a encontrar la solución.

Ejemplo: Factorial de un número

Escribe un programa que calcule el factorial (!) de un entero no negativo. He aquí algunos ejemplos de factoriales:

- 0! = 1
- 1! = 1
- 2! = 2 ➔ 2! = 2 * 1!
- 3! = 6 ➔ 3! = 3 * 2!
- 4! = 24 ➔ 4! = 4 * 3!
- 5! = 120 ➔ 5! = 5 * 4!



Codificación en C del factorial recursivo

```
int Factorial ( int  Num   )
{
  if  ( Num==0 )  return 1;                  // Caso Base
  return Num * Factorial ( Num -1 );   //  Autollamado
}
```

Divide y vencerás

- Muchas veces es posible dividir un problema en subproblemas más pequeños, generalmente del mismo tamaño, resolver los subproblemas y entonces combinar sus soluciones para obtener la solución del problema original.
- Dividir para vencer es una técnica natural para las estructuras de datos, ya que por definición están compuestas por piezas. Cuando una estructura de tamaño finito se divide, las últimas piezas ya no podrán ser divididas.
- Cualquier rutina de programación que se diseñe bajo la técnica de divide y vencerás es recursiva. El caso base resuelve subproblemas o partes pequeñas como si fuese el problema general. La rutina debe

contener instrucciones para dividir o partir la estructura de datos cuando no sea pequeña, luego hacer el autollamado con cada parte. Al final se integran las soluciones de cada parte.

Ordenación por Fusión (Mezcla directa)

¿Cómo funciona el método?

- Se basa en la técnica de divide y vencerás.
- Partimos de un vector de elementos A[ ]
  - ✓ Si el número de elementos es pequeño se ordena con alguno de los método lentos y finaliza (caso base)
  - ✓ Si no es así dividimos el vector por la mitad
  - ✓ Ordenamos recursivamente los dos sub-vectores resultantes.
  - ✓ Mezclamos las dos mitades ordenadas en un array ordenado.
- Resulta apropiado cuando la colección de elementos a clasificar está representada mediante una lista enlazada

Ordenación por Fusión (Explicación).

- Animación del método

Algoritmo para MergeSort

```
// Algoritmo de rutina diseñada con divide y vencerás
// Se considera 10 el máximo para ordenar eficiente con Burbuja
// La rutina MEZCLAR integra las soluciones de cada parte y se explica abajo al final de la rutina MERGESORT

PROC MERGESORT(A :Array, IZQ :ENTERO, DER :ENTERO)
TAM = DER-IZQ+1
SI (TAM<=10) ENTONCES
        BURBUJA (A,TAM)
```

```
SINO
        CENTRO = INT((DER+IZQ)/2)
        MERGESORT (A,IZQ,CENTRO)
        MERGESORT (A,CENTRO+1,DER)
        MEZCLAR (A,IZQ,CENTRO,DER)
FIN SI
FIN PROC
```

Algoritmo para Mezcla

```
// Rutina para mezclar o integrar partes del array ya ordenadas
PROC MEZCLAR(A :Array, IZQ, CENTRO, DER : ENTERO)
    iniIzq = IZQ, iniDer = CENTRO+1, iniAux = IZQ

    HAGA MIENTRAS(iniIzq<=CENTRO ^ iniDer<=DER)
        SI A[iniIzq]>A[iniDer] ENTONCES
            VECTORAUX[iniAux] = A[iniDer]
            iniDer = iniDer + 1
        SINO
            VECTORAUX[iniAux] = A[iniIzq]
            iniIzq = iniIzq + 1
        FIN SI
        iniAux = iniAux + 1
    FIN MIENTRAS
    HAGA MIENTRAS(iniIzq<=CENTRO)
        VECTORAUX[iniAux] = A[iniIzq]
        iniIzq = iniIzq + 1
        iniAux = iniAux + 1
    FIN MIENTRAS
    HAGA MIENTRAS(iniDer<=DER)
        VECTORAUX[iniAux] = A[iniDer]
        iniDer = iniDer + 1
        iniAux = iniAux + 1
    FIN MIENTRAS
     HAGA MIENTRAS(IZQ<=DER)
        A[IZQ] = VECTORAUX[IZQ]
        IZQ = IZQ +1
    FIN MIENTRAS
FIN PROC
```

Codificación en C++ (función mezcla)

test_sort.cpp | Sort.h

```
////////////////////////////////////////////////////////////////////////////////
// Método de ordenación por fusión (Mezcla directa).
template <class T> void merge(T *A, T *auxArray, int left, int center, int right)
{
    int leftPos, rightPos, auxPos;

    //Tomar la posición inicial del vector izquierda y el vector derecha
    leftPos=left; rightPos=center+1; auxPos=left;

    //Recorrer el vector [left,center] (izq.) y el vector [center+1,right] (der.)
    //hasta llegar al final de alguno de los dos guardando en auxArray el menor
    while(leftPos<=center && rightPos<=right)
        if(A[leftPos]>A[rightPos])
            auxArray[auxPos++] = A[rightPos++];
        else
            auxArray[auxPos++] = A[leftPos++];

    //Pasar los elementos restantes del vector izquierda hasta auxArray
    while(leftPos<=center)
        auxArray[auxPos++] = A[leftPos++];

    //Pasar los elementos restantes del vector derecha hasta auxArray
    while(rightPos<=right)
        auxArray[auxPos++] = A[rightPos++];

    //Regresar los elementos ordenanos desde auxArray hasta A
    while(left<=right)
        A[left] = auxArray[left++];

}
```

test_sort.cpp | Sort.h

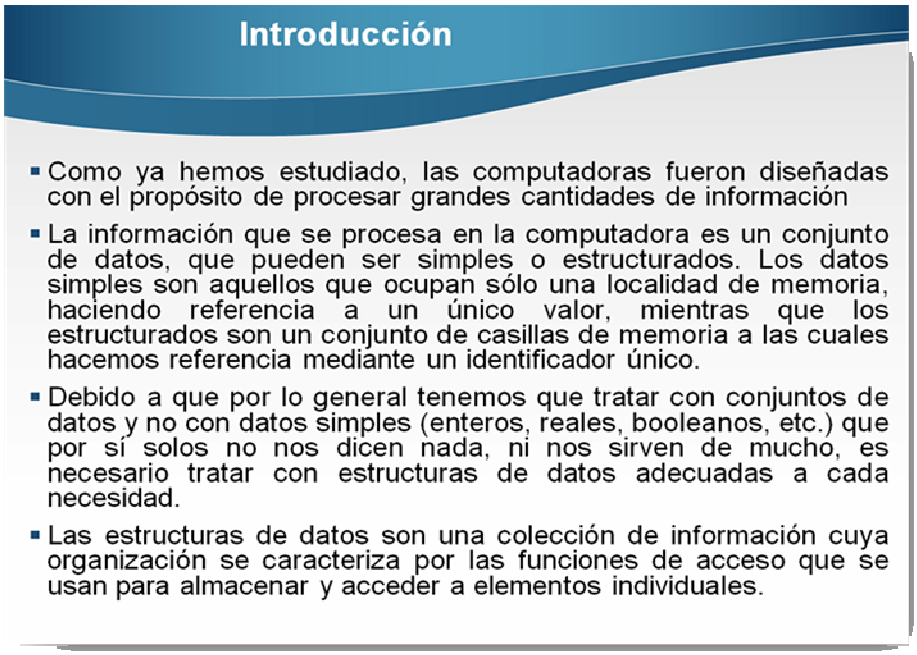
```
template <class T> void mergeSort(T *A, int n)
{
    //Vector auxiliar donde se mezclaran ambos vectores
    T *auxArray = new T[n];

    //Invocar al mergesort
    mergeSort<T>(A,auxArray,0,n-1);

    //Borrar la memoria ocupada por el array auxiliar
    delete [] auxArray;
}

template <class T> void mergeSort(T *A, T *auxArray, int left, int right)
{
    int size = right-left+1, center;
    if(size<=10)
        //Caso Base ordenar por burbuja si son pocos elementos
        bubbleSort1<T>(A+left,size);
    else
    {
        //calcular el centro del arreglo
        center = (right+left)/2;
        //Ordenar la mitad izquierda
        mergeSort<T>(A, auxArray, left, center);
        //Ordenar la mitad derecha
        mergeSort<T>(A, auxArray, center+1, right);
        //mezclar ambas mitades para obtener el vector ordenado
        merge<T>(A, auxArray, left, center, right);
    }
}
```

Anexo 3. Guión Técnico y Prototipo



Introducción (cont.)

- Una estructura de datos se caracteriza por lo siguiente:
  - Pueden descomponerse en los elementos que la forman.
  - La manera en que se colocan los elementos dentro de la estructura afectará la forma en que se realicen los accesos a cada elemento.
  - La colocación de los elementos y la manera en que se accede a ellos puede ser encapsulada.
- El estudio de las estructuras de datos constituye una de las principales actividades para llegar al desarrollo de grandes sistemas de software

Método de Ordenación por Inserción Directa

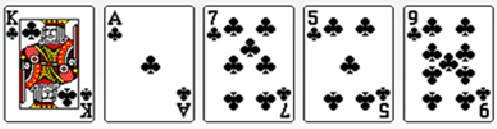
Prof. David Ortiz

Ordenación por inserción directa

- El método de ordenación por inserción directa también es conocido por el método de la baraja, debido a que generalmente es utilizado por los jugadores de cartas cuando ordenan éstas.
- Este algoritmo consiste en insertar un elemento del arreglo en la parte izquierda del mismo, que ya se encuentra ordenada. Este proceso se repite desde el segundo hasta el n-ésimo elemento.

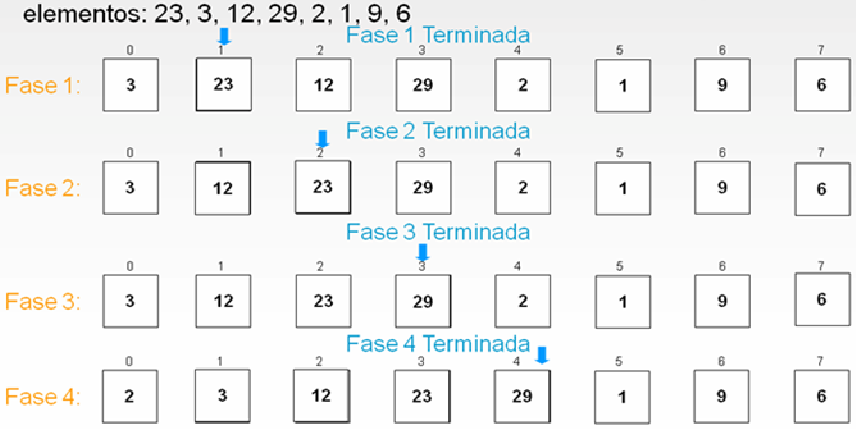
Ordenación por inserción directa (Método de la Baraja)

- Suponga que un jugador tiene el siguiente grupo de cartas, ¿Cómo haría para ordenarlas?



Ordenación por inserción directa (Explicación)

- Suponga que se quiere ordenar en forma ascendente los siguientes elementos: 23, 3, 12, 29, 2, 1, 9, 6



Ordenación por inserción directa (Explicación)

Algoritmo para el método de inserción directa

```
HAGA DESDE I=1 HASTA N-1
    AUX = A[I], J=I
    HAGA MIENTRAS (J>0 ∧ A[J-1] > AUX)
        A[J]=A[J-1],
        J=J-1
    FIN MIENTRAS
    A[J] = AUX
FIN HAGA DESDE
```

Codificación del Método de Inserción directa en C++

test_sort.cpp | Sort.cpp | Sort.h | sort.h

```
////////////////////////////////////////////////////////////////////////////
// Inserción directa.

template <class T> void insertionSort(T *A, int n){
    int i, j;
    T aux;

    for(i=1;i<n;i++){
        aux = A[i];

        for(j=i;j>0 && A[j-1]>aux;j--) A[j]=A[j-1];

        A[j] = aux;
    }
}
```