

**ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS DIDÁCTICAS
UTILIZADAS EN LA ENSEÑANZA DE LA ASIGNATURA ESTRUCTURA DE
DATOS I EN LAS JORNADAS DIURNA Y NOCTURNA DEL PROGRAMA DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD EL BOSQUE DE BOGOTA**

WILSON MAURO ROJAS REALES

**UNIVERSIDAD PILOTO DE COLOMBIA
DIRECCIÓN DE POSTGRADOS
ESPECIALIZACIÓN EN DOCENCIA UNIVERSITARIA
BOGOTÁ D,C. 2012**

**ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS DIDÁCTICAS
UTILIZADAS EN LA ENSEÑANZA DE LA ASIGNATURA ESTRUCTURA DE
DATOS I EN LAS JORNADAS DIURNA Y NOCTURNA DEL PROGRAMA DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD EL BOSQUE DE BOGOTÁ**

WILSON MAURO ROJAS REALES

**Trabajo de investigación para optar al título de Especialista en Docencia
Universitaria.**

**Director:
MARTHA ARENAS MOLINA**

**UNIVERSIDAD PILOTO DE COLOMBIA
DIRECCIÓN DE POSTGRADOS
ESPECIALIZACIÓN EN DOCENCIA UNIVERSITARIA
BOGOTÁ D,C. 2012.**

TABLA DE CONTENIDO

1.	PLANTEAMIENTO DEL PROBLEMA _____	10
2.	JUSTIFICACIÓN _____	11
3.	OBJETIVO GENERAL _____	12
4.	OBJETIVOS ESPECÍFICOS _____	12
5.	MARCO REFERENCIAL _____	12
6.	MARCO DE ANTECEDENTES _____	12
7.	NIVEL GLOBAL _____	13
8.	NIVEL NACIONAL _____	25
9.	NIVEL LOCAL _____	36
10.	REFLEXIONES DEL AUTOR _____	37
11.	MARCO HISTÓRICO _____	38
	11.1. El rol del docente en el proceso de enseñanza de la asignatura Estructuras de datos _____	39
	11.2. El proceso de enseñanza _____	40
	11.3. Recurso didáctico-pedagógico _____	41
12.	MARCO LEGAL _____	42
13.	FUNDAMENTO LEGAL DE LA UNIVERSIDAD EL BOSQUE _____	42
14.	MARCO TEÓRICO _____	43
15.	MARCO CONCEPTUAL _____	48
	15.1. Software educativo _____	48
	15.2. Algoritmo _____	49
	15.3. Estructuras de datos _____	49
	15.4. Herramientas didácticas _____	50
	15.5. Mapas conceptuales _____	51
	15.6. Enseñanza _____	53
16.	DISEÑO METODOLÓGICO _____	53
17.	TIPO DE INVESTIGACIÓN _____	53
	16.1. Estudio Co-relacional _____	54

18. POBLACION _____	54
19. INSTRUMENTOS DE RECOLECCION DE INFORMACION _____	55
20. ANALISIS DE RESULTADOS _____	55
21. CONCLUSIONES _____	64
21.1. DESDE LA ASIGNATURA _____	64
21.2. DESDE LAS HERRAMIENTAS DIDACTICAS _____	64
21.3. Herramientas propuestas _____	65
21.4. DESDE LA FORMACION DOCENTE _____	66
22. BIBLIOGRAFIA _____	68

INDICE DE FIGURAS

Figura 1. Ventana de Mapa Principal_____	18
Figura 2. Herramientas Didácticas docente A_____	57
Figura 3. Herramientas didácticas docente B _____	59
Figura 4. Herramientas didácticas docentes A y B _____	60
Figura 5. Herramientas didácticas utilizadas por el profesor A y B por Temas.____	61
Figura 6. Uso De Herramientas Didácticas En La Enseñanza_____	63
Figura 7. Red de conceptos_____	66

INDICE DE TABLAS

Tabla 1: Grupos experimentales _____	28
Tabla 2: Grupos de control _____	29
Tabla 3: Resultados pre evaluación grupos piloto _____	29
Tabla 4: Resultados pre evaluación grupos de control _____	30
Tabla 5: Herramientas didácticas utilizadas por el docente "A". _____	56
Tabla 6: Herramientas didácticas utilizadas por el docente "B". _____	58

INTRODUCCIÓN

La educación ha sido la manera como la humanidad ha pasado de generación en generación los conocimientos, mediante diferentes formas: oral, en piedra, telas, papiros, papel y recientemente en formato digital. Precisamente, la enseñanza es una actividad básica del ser humano. Es la forma cómo nuestra cultura ha perpetuado y transmitido su conocimiento, que es tan vasto, que aún el ser humano no ha podido asimilarlo por completo.

Es pues la magnitud del conocimiento que el ser humano en la actualidad ha decidido especializarse en un área determinada para sobre ella ejercer una actividad profesional.

Ahora bien, con la aparición de las tecnologías de información como bien podemos citar: la red de redes internet, los sistemas operativos, simuladores, las redes sociales, entre otras, que han pasado a ser parte de nuestro diario vivir convirtiéndose en un gran aporte utilizado como recursos didácticos utilizados en el proceso de enseñanza por parte de los docentes. La influencia de las TIC's ha sido tal, que las técnicas y métodos de enseñanza han sido cambiantes precisamente porque las tecnologías de información han obligado a la educación a ponerse en la vanguardia de las tecnologías en el transcurrir del tiempo.

No cabe duda que las técnicas de enseñanza han evolucionado de una manera muy rápida en los últimos años. Hasta hace muy pocos años, la docencia de cualquier especialidad científica o técnica era mayoritariamente transmitida mediante la presentación oral o escrita en una pizarra por parte del profesor. Las clases conocidas como teóricas consistían en clases magistrales, donde sólo en raras ocasiones se veían transparencias o diapositivas.

En la actualidad cada vez es más común el uso de otros medios en la docencia: más y mejor utilización de las transparencias, proyectores de ordenador, software para simular prácticas de laboratorio, intercambio de documentación mediante correo electrónico, presentación de material en internet, etc. Estas nuevas tecnologías están cambiando radicalmente el concepto de enseñanza, no sólo en la Universidad sino en cualquier nivel educativo.

Es por esta razón que la enseñanza de los estudios en ingeniería de la informática no debe, por razones obvias, quedar al margen de estas nuevas metodologías. Es decir, que los futuros profesionales de la Informática deben estar formados haciendo uso –por parte de los docentes- de estas herramientas didácticas.

Sin embargo, a pesar de la existencia de estrategias pedagógicas y didácticas que pueden ser aplicadas a la enseñanza de la asignatura *Estructuras de datos* el

común denominador de los docentes en los procesos de enseñanza siguen utilizando métodos complejos y tradicionales para la enseñanza de esta.

La asignatura denominada *Estructuras de datos* hace referencia al almacenamiento, manipulación y organización de la información con la que trabajan los lenguajes de programación. Las *estructuras de datos* determinan la conexión lógica de los datos y los programas utilizados para su manipulación. Hace parte de cualquier programa referente a las Ciencias de la computación y es uno de los pilares fundamentales para la formación de profesionales en esa área.

Actualmente, existen algunos estudios que tratan sobre la problemática referente a los medios utilizados para la enseñanza de la asignatura *Estructuras de datos*; sin embargo, en su gran mayoría convergen hacia el uso de un programa de computador como apoyo a la labor de la enseñanza docente y se limitan al uso de la herramienta misma. La intencionalidad de este trabajo no es seguir esa misma línea sino mostrar las diversas herramientas didácticas como un recurso o apoyo en el proceso de enseñanza utilizado por los docentes de la asignatura.

Para ello, esta investigación realiza un estudio correlacional de las diversas herramientas didácticas utilizadas por los docentes que imparten la asignatura *Estructuras de datos* en la Universidad El Bosque de la ciudad de Bogotá – Colombia tanto en la jornada diurna como la nocturna. Este estudio mostrará las herramientas didácticas utilizadas en el proceso de enseñanza de parte de dos docentes dedicados a la enseñanza de la asignatura *Estructuras de datos*. Dada la dificultad encontrada en los alumnos en la adquisición del aprendizaje en cada uno de los temas que hacen parte de la asignatura, se hace necesario buscar diversas estrategias para la enseñanza de la misma. Es claro que las metodologías tradicionales poco pueden aportar al estudiante para el aprendizaje eficaz de la asignatura, y el hecho de realizar este estudio brindará una mejor visión para reforzar y/o descartar algunas de ellas.

La investigación está organizada en cuatro partes: 1.- Planteamiento del problema, objetivos y estado del arte del objeto de estudio, 2.- Construcción de marcos de referencia, 3.- Metodología de investigación, 4.- Resultados, análisis y conclusiones.

En la primera parte; se describe la problemática de estudio, su importancia, el nivel de conocimiento deseado sobre el objeto de investigación.

La segunda, trata sobre los hallazgos de otros investigadores a nivel mundial, nacional y local y además, se hace una breve reflexión sobre las investigaciones encontradas.

En la tercera parte se plantea la estrategia metodológica. En ella se tiene en cuenta el tipo de diseño que se adopta para realizar la investigación, se define la

población sobre la cual se van a inferir los datos encontrados, se explica los instrumentos para la recolección de información.

En la cuarta parte, se realiza el plan de análisis con base en los resultados encontrados y describen una serie de conclusiones finales.

1. PLANTEAMIENTO DEL PROBLEMA

En cualquier programa de ingeniería informática, se sabe de la complejidad existente en la enseñanza de la asignatura *Estructuras de datos*, esto debido a la no existencia de un único método para la resolución de los problemas que allí se presentan así como tampoco un enfoque didáctico acorde para abordar su enseñanza. Uno de los principales objetivos de la enseñanza de las *estructuras de datos* es la construcción de algoritmos más eficientes. Sin embargo, en el transcurrir de los tiempos una de las estrategias didácticas más extendidas y utilizadas para la enseñanza de la asignatura Estructura de Datos I ha sido la clase magistral o expositiva. Esto es, el docente se dedica –o dedicaba- a impartir una serie de conocimientos y a resolver una serie de ejercicios modelos de acuerdo a la temática de turno. Como consecuencia, la enseñanza de la asignatura *Estructuras de datos* no refleja con objetividad el grado de apropiación del conocimiento adquirido por parte del estudiante ni la transferencia de conocimiento (aplicación), dada la complejidad de la enseñanza de la asignatura en el aula de clases mientras no se haga uso de recursos o medios didácticos totalmente diferentes y que involucre al estudiante en hechos reales que ilustren el nivel de abstracción conceptual que conforma la asignatura.

Es evidente, que de seguir enseñando con las mismas estrategias y herramientas didácticas, se seguirán reflejando las deficiencias por parte de los alumnos en el uso y manejo de los algoritmos implementados con las estructuras de datos adecuadas; esto, indudablemente repercutirá en la no obtención de las bases sólidas para la comprensión de la asignatura y no solo de ella, sino de las otras asignaturas precedentes. Es por ello, que surge la necesidad de buscar otros mecanismos o herramientas didácticas que permitan al docente fortalecer el proceso de enseñanza de la asignatura. El hecho de realizar un análisis comparativo de las herramientas didácticas utilizadas por los dos docentes que imparten la asignatura *Estructuras de datos* en el programa de Ingeniería de Sistemas al interior de la Universidad el Bosque; tanto en la jornada diurna como la nocturna, es un primer avance y aporte en la búsqueda de mejores alternativas para potencializar y fortalecer el proceso de enseñanza de la asignatura en mención.

La búsqueda de herramientas didácticas de enseñanza para la asignatura de *Estructuras de datos* invita a resolver la siguiente pregunta problema:

¿Qué herramientas didácticas son utilizadas por los docentes que lideran el proceso de enseñanza de la asignatura *Estructuras de Datos* en la jornada diurna y nocturna en el programa de Ingeniería de Sistemas de la Universidad El Bosque de Bogotá?

2. JUSTIFICACIÓN

Estructuras de datos es una de las asignaturas fundamentales en las carreras de Ingeniería de Sistemas, su estudio es antiguo y se ha convertido en un estudio obligatorio.

Uno de los primeros obstáculos con los que se encuentra un estudiante es el cómo aprender la asignatura *estructuras de datos*, y no solo el estudiante sino también el docente. No es fácil encontrar una metodología de enseñanza para la asignatura, así como tampoco es fácil poder digerir la lectura de un libro, puesto que se trata de información abstracta lo que complejiza el método de enseñanza.

No es fácil encontrar metodologías y herramientas didácticas de enseñanza para las *Estructuras de datos*, en ocasiones no se ajustan a las necesidades específicas del estudiante y en contra de lo que opinan algunos, la idea de que la tecnología desplaza a los docentes está superada. El papel del docente no solo no pierde importancia sino que se amplía y se hace imprescindible. EL docente tiene la obligación y el reto de buscar alternativas y/o estrategias para llevar el conocimiento al estudiante, la labor del docente se hace más profesional, más creativa, exigente.

El estudio correlacional de las herramientas didácticas utilizadas para la enseñanza de la asignatura de estructuras de datos en la jornada diurna y nocturna de la Universidad el Bosque, es necesaria para que no solo los docentes y estudiantes, sino también a nivel directivo del programa de Ingeniería de Sistemas se busquen los espacios adecuados y se preparen propuestas metodológicas con base en los resultados obtenidos en pro de una mejor enseñanza de la asignatura y de una mejor utilización de las herramientas didácticas.

Uno de los grandes beneficios que se logra con el desarrollo de este trabajo, es el poder potencializar el proceso de enseñanza de la asignatura *Estructuras de datos* por parte de los docentes constituyéndose en un gran aporte no solo para el programa de Ingeniería de Sistemas sino para la Universidad el Bosque puesto que la sociedad espera de los futuros profesionales (formados por esos docentes universitarios) la solución a problemas complejos relacionados con las tecnologías de la información. Además, la Universidad desea formar egresados competentes en el área de Ingeniería de Sistemas capaces de aportar todo su conocimiento a la industria Colombiana. Así mismo, el hecho de realizar un análisis comparativo de las herramientas didácticas utilizadas para la enseñanza de la asignatura *Estructuras de Datos* brinda al docente un camino para pensar en la innovación didáctica, en el mejoramiento continuo, en reforzar y mejorar sus técnicas de enseñanza y como consecuencia de ello potencializar en el estudiante el conocimiento que se quiere transmitir.

3. OBJETIVO GENERAL

Conocer qué herramientas didácticas son utilizadas por los docentes para la enseñanza de la asignatura *Estructuras de datos I* en la jornada diurna y nocturna del programa de Ingeniería de Sistemas en la Universidad el Bosque en Bogotá.

4. OBJETIVOS ESPECÍFICOS

- Identificar las herramientas didácticas utilizadas por el docente de la jornada diurna para la enseñanza de la asignatura *estructuras de datos I* en el programa de Ingeniería de sistemas en la Universidad el Bosque en Bogotá
- Identificar las herramientas didácticas utilizadas por el docente de la jornada nocturna para la enseñanza de la asignatura *estructuras de datos I* en el programa de Ingeniería de sistemas en la Universidad el Bosque en Bogotá.
- Cotejar las herramientas didácticas utilizadas por los docentes de la jornada diurna y nocturna para la enseñanza de la asignatura *estructuras de datos I* en el programa de Ingeniería de sistemas en la Universidad el Bosque en Bogotá.

5. MARCO REFERENCIAL

6. MARCO DE ANTECEDENTES

El presente apartado está constituido por tres partes interrelacionadas. La primera transita por un relato a nivel global de los estudios que se han realizado sobre herramientas didácticas utilizadas en la enseñanza de la asignatura estructuras de datos.

El propósito de la segunda parte; es discurrir sobre la enseñanza de la asignatura estructuras de datos a nivel local en Colombia.

En la tercera parte, se hace un recuento de lo que sucede al interior de la Universidad el Bosque en el proceso de enseñanza de la asignatura Estructuras de datos.

Por último, se realiza una reflexión crítica sobre el estado del arte del objeto de estudio.

7. NIVEL GLOBAL

La asignatura *Estructuras de Datos* hace parte del programa de Ingeniería de Sistemas o ingeniería Informática (muy probablemente también se incluya en cualquier carrera relacionada con las Ciencias de la computación), su contenido temático como tal se basa en el estudio de las diferentes estructuras de datos (listas, pilas, colas, TAD's, árboles, grafos, entre otros) que al ser utilizadas por los lenguajes de programación se obtiene como resultado un programa de computador que solucionará algún problema específico.

La enseñanza de la asignatura es soportada mediante algún lenguaje de programación que debe ser conocido por parte del docente para poder llevar a cabo su labor al interior del aula. En general; el docente para la enseñanza de dicha asignatura, se apoya en la transmisión verbal y el desarrollo de ejercicios modelos previamente elaborados.

En el caso de España, los actuales planes de estudios y los futuros — contemplados en la Declaración de Bolonia (EEES, Espacio Europeo de Educación Superior)—, de Ingeniería Técnica en Informática e Ingeniería Informática, contemplan materias troncales relativas tanto a Algoritmos como a *Estructuras de Datos*. Igual sucede en los países iberoamericanos donde también es común incluir estas disciplinas en los currículum de carreras de Ingeniería de Sistemas y Licenciaturas en Informática. ACM, la organización profesional norteamericana más prestigiosa a nivel mundial, incluye en las recomendaciones de sus diferentes currículos y carreras relacionadas con informática el estudio de materias de algoritmos y estructuras de datos. En todas estas instituciones, la metodología aplicada por parte del docente para la enseñanza de la asignatura de *Estructuras de datos* no ha variado mucho. Se sigue con la transmisión verbal y el desarrollo de ejercicios modelos elaborados previamente por el docente, se trabajan un número de horas teóricas y algunas clases prácticas y como complemento —algunas veces- se utiliza un aula virtual disponible en la web para brindar a los alumnos material adicional.

El enfoque general presentado anteriormente, se viene aplicando desde hace muchos años y muchas Universidades e instituciones de educación superior en especial los programas de ciencias informáticas e ingenierías de sistemas, se han dado cuenta de la imperiosa necesidad de realizar cambios pedagógicos, metodológicos, didácticos y organizacionales producto de los magros resultados obtenidos por los alumnos en cuanto a la calidad de sus aprendizajes, la alta deserción y el bajo rendimiento académico.

La Universidad Nacional de Río cuarto (Argentina) ha impulsado una serie de proyectos de innovación e investigación para el mejoramiento de la enseñanza de la asignatura *Estructuras de datos*. Durante los años 2004 y 2005 realizaron un análisis minucioso de los aciertos y errores y concluyeron que efectivamente la

necesidad de cambios en la enseñanza de la asignatura era inminente. Organizaron de una mejor forma los contenidos temáticos de la asignatura y potencializaron la solución de ejercicios prácticos desde el laboratorio. El proyecto aún continúa en una nueva fase en la búsqueda de mejorar las estrategias de enseñanza.

Los docentes que imparten la cátedra de *Estructuras de datos* en dicha universidad; no se han cruzado de brazos, de manera permanente están en la búsqueda de nuevas estrategias y didácticas para la enseñanza de la asignatura.

Uno de los proyectos en los cuales ha trabajado la Universidad Nacional de Río Cuarto (Argentina) está titulado “Enseñanza de la programación y estructuras de Datos” bajo la dirección de *Ariel Ferreira Szpiniak* y *Guillermo A. Rojo*. Coinciden ellos en la existencia de diversos enfoques y tendencias pero sin consenso sobre cuál es la mejor manera de enseñar los conceptos básicos de la asignatura Estructura de Datos. En el estudio realizado durante el año 2004 observaron que algunos docentes utilizaban métodos de enseñanza cuya fundamentación estaba enfocado a un paradigma de programación en particular como lo son el paradigma funcional, el imperativo o el imperativo con el aporte de orientación a objetos. Además, dentro de un paradigma determinado se visualizan varios enfoques para enseñar: algunos enseñan la asignatura *Estructuras de Datos* soportada en un lenguaje de programación particular, utilizando su sintaxis y su semántica, y otros emplean un lenguaje algorítmico lo bastante general cómo para permitir su traducción posterior a cualquier lenguaje de programación. El primero de estos enfoques tiene el inconveniente de ligar los conceptos básicos de la programación a un lenguaje determinado, el cual tiene sus propias características y especificidades, perdiendo de vista los conceptos generales. Lo que pudieron observar en términos generales es que la técnica de enseñanza aplicada por los docentes no son más que la enseñanza de los conceptos básicos pero en otros tantos lenguajes, con diferentes características y especificidades.

La metodología empleada por los docentes *Ferreira* y *Rojo* consiste en la ejecución de distintas fases que implican la elaboración de un algoritmo que expresa la solución del problema planteado en un lenguaje algorítmico. Una vez obtenido el mismo se traduce a un lenguaje de alto nivel y se implementa en computadora. Sobre ella se realizan los test de prueba. La obtención del algoritmo involucra un proceso de análisis descendente que puede determinar el diseño de subprogramas, módulos, acciones o funciones.

Una de las características que consideran ellos poco habitual es el hincapié depositado en las estructuras iterativas y las buenas prácticas para resolver problemas de involucren tratamiento de secuencias de objetos. En ello trabajaron arduamente en pos de encontrar estrategias didácticas adecuadas que reduzcan la sobrecarga cognitiva a la hora de afrontar los primeros problemas de este tipo y

posibiliten comprenderlos fácilmente sin la necesidad de incluir desde el comienzo las estructuras de representación como arreglos, listas enlazadas, etc.

La metodología aplicada en el desarrollo de la investigación fue la siguiente: En la asignatura trabajaron con clases teóricas y clases prácticas. Para las clases prácticas los alumnos son distribuidos en comisiones de 30 alumnos. Las clases teóricas son semanales y de tres (3) horas de duración. Las clases prácticas también son semanales y de cuatro (4) horas de duración. En ambos casos se cuenta con clases de consulta de dos (2) horas de duración por semana. Para la implementación de algoritmos se dispone de una clase semanal dos (2) horas en el laboratorio. Además se dispone de cuatro (4) horas por semana extraclasses reservadas exclusivamente para el acceso de los alumnos al laboratorio. Como complemento se utiliza un aula virtual disponible en la web para brindar a los alumnos novedades, información, materiales teóricos y prácticos, consultas, etc.

Los resultados arrojados, fueron vitales para la toma de decisiones y de hecho realizaron cambios pedagógicos, metodológicos, didácticos y organizaciones. Han convertido el proceso de enseñanza de la asignatura *Estructuras de datos* en una investigación activa para la cátedra buscando así la posibilidad de mirar nuevas estrategias didácticas para la enseñanza de la asignatura conscientes de que dichas estrategias no siempre están relacionadas con el rendimiento académico por parte de los alumnos. El proyecto ha continuado bajo el nombre de “Estrategias para mejorar la calidad de la enseñanza en las Carreras de Computación”.

Prosigamos con el tema, “Desarrollo de una herramienta de apoyo para los docentes de cursos básicos de Estructuras de Datos”, es el título recibido el trabajo realizado por *J. Andrés Sandoval Bringas, Mónica A. Carreño León e Italia Estrada Cota* docentes de la Universidad Autónoma de Baja California Sur, en México. Los autores, llevaron a cabo el desarrollo de una herramienta informática como apoyo a la labor docente y que le permita enseñar la asignatura *Estructuras de Datos* bajo un nuevo enfoque. Con el uso de la herramienta el docente experimentará de una manera vivencial el desarrollo de la asignatura, por ejemplo: para el manejo del tema “Listas” podrá comprobar su funcionamiento y desempeño de manera visual. De esta forma, las explicaciones brindadas por el docente a nivel teórico, será reforzado de manera inmediata con el uso de la herramienta. Los autores concluyen que existe la necesidad de investigar sobre nuevas estrategias de enseñanza de la asignatura pero consideran que el uso de herramientas informáticas para tal fin constituye un aporte valioso para el proceso de enseñanza.

De otro lado, el trabajo presentado por la Universidad de Guadalajara – México, titulado “Tres modalidades de enseñanza aplicadas a la asignatura de *Estructuras de datos*”, bajo la dirección de *María Obdulia González Fernández, Nadia Natasha Reus González y Claudia Islas Torres*, tiene como objetivo mostrar las diferencias

observadas en el rendimiento en el aprendizaje de un grupo de alumnos de la carrera de Ingeniería de Computación, al cursar la asignatura *Estructuras de Datos* bajo las modalidades: presencial, on-line y blended learning. Los resultados obtenidos por medio del análisis del rendimiento escolar de los grupos pilotos sometidos a estudios ayudó a demostrar que las herramientas didácticas utilizadas por los docentes en cada uno de estos escenarios, no representan diferencias significativas puesto que las calificaciones logradas al final del semestre fueron similares en los tres grupos piloto.

Para los autores, las distintas modalidades de formación derivadas del uso de las TIC's (Tecnologías de la comunicación e información) como son los ambientes virtuales y los ambientes mixtos, han servido de apoyo a la educación presencial en especial a la enseñanza tradicional. El estudio realizado por los docentes fue aplicado a los alumnos de Licenciatura en Ingeniería de Computación de cuarto semestre, específicamente en la asignatura de *Estructuras de datos* en el centro Universitario de los Altos de la Universidad de Guadalajara. La finalidad del estudio era demostrar la pertinencia de cómo pueden migrarse distintas asignaturas dependiendo de su naturaleza, a las modalidades antes mencionadas, todo esto haciendo el análisis correspondiente y el diseño de estrategias de enseñanza necesarias para obtener el éxito deseado. El estudio realizado fue comparativo donde los sujetos de análisis pertenecen al mismo grupo de la asignatura *Estructuras de datos* del cuarto semestre de la licenciatura en Ingeniería de en Computación calendario A durante el año 2009. Para ello, se tomaron las técnicas de enseñanza utilizadas en la modalidad presencial, virtual y b-learning estas dos últimas utilizando la plataforma Moodle como apoyo a la asignatura cuya carga horaria es de cuatro horas semanales de carácter teórico y el grupo estuvo conformado por un total de 23 alumnos. La elección de la muestra fue totalmente al azar, al inicio del semestre se le informó a los alumnos sobre el proyecto en el que el grupo participaría, siempre y cuando estuviesen de acuerdo en formar parte de la muestra que serviría para dicho estudio.

Las variables consideradas para dicho estudio con las que se midió el rendimiento de aprendizaje fueron:

- ✓ Resultados de la evaluación del primer examen parcial.
- ✓ Resultados de la evaluación del segundo examen parcial.
- ✓ Resultados de la evaluación del tercer examen parcial
- ✓ Calificación final.

Con base en los resultados obtenidos en la investigación, los autores concluyeron que no existe una diferencia significativa en las tres modalidades en lo referente a rendimientos escolares de los alumnos para un curso de naturaleza teórico, tomando en cuenta que dichos ambientes difieren en las formas de trabajo dependiendo de las características del mismo. Sin embargo, los alumnos del grupo virtual no obtuvieron mejores resultados y esto debido a las condiciones en que se desarrolló el curso. Observaron que las técnicas de enseñanza aplicadas

de manera presencial deben ser totalmente diferentes a las estrategias de enseñanza a utilizar de manera virtual.

Aún más, Paolo Rocchi, miembro de la empresa IBM, en su artículo “Un método deductivo para enseñar *Estructuras de Datos*” publicado en la “Revista de enseñanza y tecnología” (Diciembre 2000) considera que uno de los mayores obstáculos en la enseñanza no solo de la asignatura de *Estructuras de datos* sino en cualquier área de la informática está relacionado al manejo de la terminología, a la definición de los conceptos, al significado de las soluciones. Considera además, que la enseñanza aplicada, se focaliza más en la habilidad operativa y muy poco a la formación. Las lecciones se concentran más en “cómo” hacer y no en “por qué” se hace. El alumno trabaja más en función de memoria que en términos de razonamiento. Manifiesta que las dificultades didácticas no nacen por razones psicológicas o sociales y no se originan tampoco en cuestiones lingüísticas ni expresivas. Nacen más bien del hecho que la Informática es una ciencia joven, que sus fundamentos son en gran parte oscuros y que la didáctica resulta deficitaria. En mayor o menor medida se causan daños al estudiante, al analista, al dirigente, al usuario, en pocas palabras, a quienes de cualquier manera se acercan a la Informática. La propuesta del autor se basa en la enseñanza deductiva y formativa, en la cual consideran que no solo basta con utilizar herramientas didácticas para la enseñanza de la asignatura sino que es primordial que el docente acerque al estudiante a “deducir” su propio aprendizaje.

Para llevar a cabo su trabajo investigativo, organizaron dos cursos (en la ciudad de Roma – Italia, durante el año 1998) intensivos por cuatro días: uno de ellos siguiendo un método de enseñanza deductivo y el otro siguiendo una metodología de enseñanza tradicional con una aplicación de seis horas diarias. Al terminar el curso, los estudiantes expresaron sus opiniones en un módulo preparado a tal efecto.

Finalmente, el autor esboza el método de enseñanza deductivo y formativo de la asignatura *Estructuras de datos* sin efectuar una discusión amplia de los argumentos expuestos en su teoría.

Veamos ahora, Patricia Ruth Uviña, Mabel Angélica Bertolami, Maria Elena Centeno y otros docentes de la Universidad Nacional de la Patagonia – San Juan Bosco – Argentina, mediante un aplicativo de Software hicieron uso de los Mapas Conceptuales como herramienta de enseñanza. Ellos consideran que la utilización de Mapas Conceptuales fomenta el pensamiento reflexivo, la creatividad y el espíritu crítico, conductas imprescindibles en la formación y el desenvolvimiento profesional de los alumnos de las carreras de informática y favorecen su reorganización cognoscitiva en el dominio de las estructuras de datos.

Entre los objetivos del proyecto desarrollado por los autores se encuentran:

- ✓ Mejorar la calidad de enseñanza de las estructuras de datos
- ✓ Integrar la herramienta educativa “Mapas Conceptuales” como recurso pedagógico para la enseñanza de distintos contenidos en las asignaturas “Algorítmica y programación” y “Estructuras de datos y algoritmos”

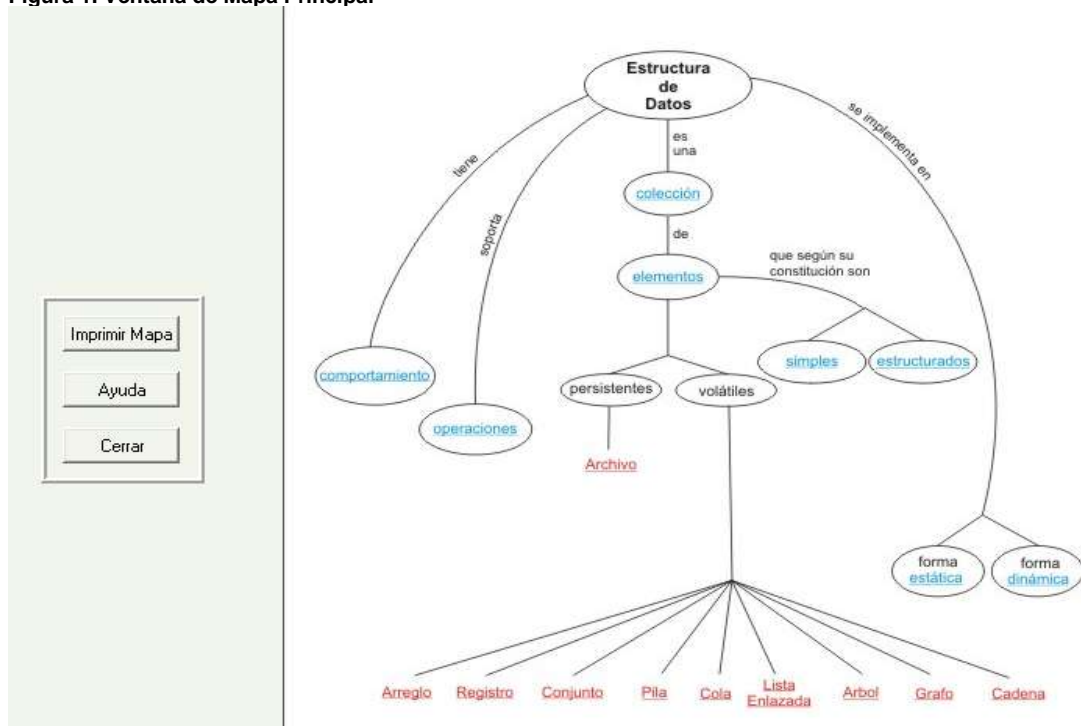
La herramienta utilizada se denomina **Kellu** y por medio de ella, presentan distintos Mapas Conceptuales –cada mapa representa una estructura de datos - que les permiten representar en diversos niveles de estudio la enseñanza de las *Estructuras de datos*.

Durante un mes, dedicaron dos horas semanales de clases teóricas para la presentación de la herramienta Kellu y su aplicación para la fijación de conceptos de algunas de las estructuras de datos. Los resultados fueron bastante aceptables y posteriores a la presentación del aplicativo de software como recurso pedagógico, desarrollaron una serie de encuestas y entre varias de las conclusiones se pueden resaltar las siguientes:

- ✓ La herramienta Kellu fue de gran aceptación por parte de los estudiantes.
- ✓ La herramienta Kellu como recurso didáctico fue un gran apoyo y de gran utilidad para los docentes

A continuación podemos observar la presentación inicial del Software Kellu:

Figura 1. Ventana de Mapa Principal



Fuente: Elaboración del autor

Otro de los estudios desarrollados en la búsqueda de herramientas didácticas como apoyo a la enseñanza de la asignatura *Estructuras de datos*, es el realizado por la Universidad Complutense de Madrid (2005 – 2007) bajo la dirección de Clara Segura e Isabel Pita. Titularon su estudio como: ***Una herramienta para el estudio de Estructuras de datos y Algoritmos***. Consideran las autoras que el “alto nivel de abstracción necesario para la enseñanza de *Estructuras de datos* y *métodos algorítmicos* dificulta su comprensión por parte del alumnado. En los últimos años, la Universidad Complutense de Madrid ha desarrollado e implantado un sistema de enseñanza interactiva –en la facultad de informática- acorde con las nuevas enseñanzas del *Espacio Europeo de Educación Superior (EEES)*. En su estudio, las autoras se apoyaron en un software desarrollado por estudiantes de la misma Universidad bajo el lenguaje de programación Java el cual lleva por nombre *Vedya*.

La herramienta *Vedya* integra todo el contenido temático de la asignatura Estructuras de datos permitiendo al docente crear simulaciones que se ejecutan automáticamente, resolver test dentro del mismo entorno, animaciones de algoritmos que muestran el uso de determinada estructura de datos para resolver un problema, entre otras.

La herramienta es considerada material complementario a las explicaciones teóricas llevadas a cabo en el aula de clases mediante transparencias. A continuación se explica tanto el diseño del estudio realizado como los resultados obtenidos:

La herramienta construida, se puso a disposición de 486 estudiantes y realizaron dos estudios distintos: una experiencia no controlada (los estudiantes tienen acceso a la herramienta y participan de manera voluntaria haciendo uso del software y dando respuestas a diferentes tipos de tests) y una experiencia controlada (dos grupos de estudiantes han contestado a un test en un entorno controlado desde el punto de vista del tiempo y del acceso al material educativo).

Concluyeron que el uso de la herramienta es un apoyo fundamental como complemento a las clases teóricas desarrolladas al interior del aula de clases mediante las transparencias.

Otro de los trabajos realizados al interior de la Universidad Complutense de Madrid, es el realizado por *Pablo Fernández Poblaciones, Salvador Muñoz Sánchez y Antonio Murillo Melero* titulado “Un tutor inteligente para la visualización de Métodos Algorítmicos y Estructuras de Datos” (2008-2009). El proyecto desarrollado –y aún sigue en desarrollo- por estos autores se basa en la construcción de una herramienta de software denominada *Vedya-Profesor* tomando como base el proyecto *Vedya* desarrollado en la misma Universidad.

Vedya-profesor está basado en la aplicación de diversas técnicas de representación de conocimiento de la inteligencia artificial. Con este tipo de desarrollo se trata de mostrar que, utilizando métodos tradicionales de enseñanza se consigue mejorar significativamente la velocidad y la calidad de aprendizaje por parte de los alumnos. Pretenden los autores, “capturar” un método de enseñanza ejemplificado por una interacción humana de tutorización individualizada. Este proyecto aún está en desarrollo y se han encontrado una serie de limitaciones que inciden en la puesta en marcha del plan piloto de pruebas.

Veamos ahora, EDApplets: Una herramienta Web para la enseñanza de las Estructuras de Datos y Técnicas Algorítmicas. Bajo este título, los docentes *Almeida F., Blanco V., y Moreno L* adscritos al Dpto de estadística y computación de la Universidad de la Laguna – Tenerife – España (2003) presentan una herramienta basada en la tecnología de Applets para la enseñanza de la asignatura *Estructuras de datos*. La herramienta permite cubrir diversos aspectos en una enseñanza que puede ser dirigida a distintos tipos de estilo de aprendizaje: activo / reflexivo, metódico / intuitivo, visual / oral, etc.

Los autores consideran que la enseñanza de las asignaturas Estructuras de datos y Programación de algoritmos presenta quizás uno de los niveles de dificultad más elevado a raíz de muchos conceptos algo abstractos.

Siendo así, uno de los grandes retos en la enseñanza en particular de la asignatura *Estructuras de datos* es el de ser capaz de ajustarse a las necesidades particulares de los individuos y una de las formas es a través de software multimedia en especial en este tipo de asignaturas específicas dentro de la carrera de ciencias informáticas.

El desarrollo de la herramienta por parte de los autores, es un complemento a las tareas de enseñanza de la asignatura *Estructuras de datos*. La herramienta hace uso de animaciones sencillas e incorpora tutoriales en línea que el estudiante puede leer de manera inmediata. Concluyen los autores que la herramienta ha sido un aporte complementario de inmenso valor para el desarrollo de la enseñanza de la asignatura *Estructuras de datos* puesto que facilita una enseñanza activa, visual, metodológica y secuenciada, sin descuidar las explicaciones teóricas propias del docente.

Una de las investigaciones realizadas al interior de la Universidad Nacional del Sur – Bahía Blanca (Argentina), se titula “Estrategias para la enseñanza de la programación y estructuras de datos”. En dicha investigación, las autoras *Norma Moroni – Perla Señas* durante el año 2005 determinaron como estrategia para la enseñanza de la asignatura *Estructuras de datos* una herramienta informática constituida por “un editor interactivo de algoritmos”, “un constructor automático de trazas” y un “traductor de algoritmos”. Las autoras mediante un trabajo de campo

buscaban comprobar la efectividad de la herramienta informática como herramienta didáctica utilizada en la enseñanza de la asignatura objeto de estudio.

Para comprobar la efectividad, realizaron una experiencia en cursos de carreras informáticas. Se cotejaron dos cursos similares, uno de ellos funcionó como testigo empleándose en él una metodología tradicional (docente expositor y solucionador de ejercicios modelos) mientras que el otro se desarrolló haciendo uso de la herramienta informática. Además, consideraron tres tendencias de evaluación diferenciadas: diagnóstica, formativa y sumativa, a través de la aplicación de procedimientos cuali y cuantitativos. Esto les permitió obtener información sobre el estado de los alumnos al iniciar la experiencia, sobre su progreso durante la implementación de la misma y sobre la valoración de los logros alcanzados al finalizar el proceso. Los cursos seleccionados estuvieron bajo la responsabilidad del mismo docente y la experiencia se llevó a cabo en la misma institución educativa.

Con el propósito de analizar el estado inicial de los alumnos, de acuerdo a las variables evaluativas explicitadas, se instrumentó una encuesta de diagnóstico, indagando acerca de sus conocimientos previos en la técnica de programación, uso de la computadora y su aplicación. A partir de la información obtenida en esta evaluación, el grupo de investigación decidió instrumentar dos sesiones de clases en las que se enfatizaba la estrategia de resolución de problemas, para lograr la construcción de una base común, en lo relativo a la comprensión de la técnica y a sus posibilidades de aplicación.

Durante el transcurso de la experiencia el grupo de investigación implementó una evaluación de tipo formativa, realizando observaciones de diversas instancias áulicas, en las cuales se consideró: interés de los alumnos, clima del aula, impacto de la novedad en el uso del editor, actitud de los docentes, grado de correlación entre la comprensión de la técnica y su aplicación.

En la evaluación sumativa, en la cual se trabajó con los productos elaborados por los alumnos (algoritmos tradicionales y algoritmos editados) se tuvieron en cuenta: el modelo de resolución, la corrección de las relaciones establecidas, la correctitud del algoritmo, el tiempo de elaboración y el proceso de refinamiento.

Como resultados finales, las autoras concluyeron que es necesario incluir nuevas estrategias para la enseñanza de la asignatura Estructuras de datos y una de ellas es el uso de herramientas informáticas. Observaron que inicialmente hay cierta lentitud en el desarrollo de los ejercicios impuestos por el docente debido a la necesidad de aprender a usar la herramienta pero este tiempo es recuperado posteriormente.

Ricardo Pérez Calderón (Universidad Politécnica del Valle de México), en su artículo "Una herramienta y técnica para la enseñanza de la Programación y

Estructuras de datos” plantea que este tipo de enseñanza es muy particular y se debe sustentar en estrategias didácticas usando la mayoría de los medios que existen en la tecnología de la información. Considera que el uso del Software en la enseñanza es un medio que facilita el logro de los objetivos, ayuda al profesor en su proceso de enseñanza como facilitador de conocimiento. Considera el autor que la enseñanza de este tipo de asignaturas es una actividad compleja y difícil y el uso de una herramienta automatizada ayudaría en gran medida al proceso de enseñanza.

El estudio realizado por el autor, se basa en la recopilación de experiencias docentes en la enseñanza de las asignaturas de programación y estructuras de datos. Con base a la información recolectada, construye un software y realiza un experimento con dos grupos de trabajo (experimental y control), a ambos se les enseñó la teoría y una metodología para la solución de los ejercicios propuestos. Al grupo de control se les exhortó a que realizaran los ejercicios –previa enseñanza sobre la solución de los mismos- sin el uso de la herramienta construida y al grupo experimental haciendo uso de dicha herramienta. A cada grupo se le hizo entrega de cuatro ejercicios y como resultado se obtuvo que el grupo de control tardó menor tiempo en la solución de dichos ejercicios en comparación con el grupo experimental. El autor concluye que el uso de la herramienta es fundamental para la enseñanza de la asignatura y además existe un avance considerable en relación al aprendizaje de los estudiantes.

Otro de los artículos encontrados es el titulado “Enseñar programación y estructuras de datos en las Ingenierías Informáticas” elaborado por *Rosana Satorre Cuerda, Faraón Llorens Largo y Juan Antonio Puchol García*, docentes de la Universidad de Alicante (España) quienes proponen la utilización de “un programa de ordenador” no para programar sino para enseñar a programar aprovechando las ventajas que ofrece el computador como recurso didáctico. Consideran los autores que el uso de un “programa de computador” en las clases prácticas de las asignaturas reforzará los conceptos teóricos explicados por el docente al interior del aula de clases.

Los autores describen la asignatura Estructuras de datos como el estudio y análisis de las metodologías y técnicas para la resolución de problemas mediante el uso de una adecuada estructura de datos a través del computador. Y por tanto, debemos hacer una pequeña reflexión general sobre la importancia de la resolución de problemas. Los problemas deben ser una ocasión privilegiada de construir y profundizar los conocimientos. Se debe evitar la orientación habitual dada a la resolución de problemas, que la limita al operativismo mecánico (convirtiéndolos en “recetas de cocina”) con una falta de reflexión cualitativa previa, y al tratamiento superficial sin detenerse en la clarificación de los conceptos. Uno de los objetivos generales que se pretenden alcanzar con la asignatura de programación y estructuras de datos es “*Que el alumno sea capaz de especificar y diseñar una correcta estructura de datos y un algoritmo eficiente y*

correcto a partir de un problema dado". La consecución de este objetivo, utilizando los medios y métodos tradicionales (pizarra y papel), suele convertirse en una tarea tediosa para el profesor.

Básicamente los autores le dan gran importancia al uso del computador en el aula como recurso didáctico y se trata en "enseñar con el ordenador" siendo una poderosa ayuda para el docente.

Otro estudio es, "Métodos de enseñanza para comprender y aplicar estructuras de datos" es el título del trabajo de investigación elaborado por *Xochitl Clemente, Guillermo Espinosa y Pablo Quintero* docentes de la Universidad Tecnológica de Huejotzingo, Chiapas, México. El trabajo desarrollado por los autores, se centra en la elaboración de material didáctico que ayude a los estudiantes a comprender las bases teóricas de la asignatura y su aplicación. Consideran ellos que el 95% de los libros elaborados por diversos autores no incluyen una propuesta pedagógica que facilite el aprendizaje de la asignatura de Estructuras de datos por parte del estudiante. Además; no constituyen un verdadero apoyo al proceso de enseñanza que imparte el docente, puesto que en la práctica están fuera de contexto debido a que no evidencian la realidad que viven los alumnos y por ende, no les permiten comprender la aplicación de las estructuras a falta de conocimientos previos.

El propósito de los autores es elaborar un texto académico que sirva como apoyo al proceso de enseñanza de la asignatura de Estructuras de datos, de orientación constructivista y que facilite el aprendizaje de las mismas.

El tipo de estudio realizado por los autores fue cualitativo, comenzando como exploratorio y descriptivo analítico, con un enfoque transversal. Debido a la carencia de conocimientos previos sobre las razones que limitan el aprendizaje de las estructuras de datos, se organizó una investigación en diversas fases, la cual inició con una exploración general de los problemas observables empíricos y las descripciones de los teóricos que lo abordan, además de sus causas y de las propuestas teóricas pedagógicas existentes. La exploración incluyó una revisión general con base en una rúbrica de los libros disponibles sobre estructuras de datos, tanto en inglés como en español, para analizar sus características y distinguir aquellas que no están apoyando efectivamente el proceso de enseñanza.

Una vez identificadas las causas, se buscó dentro de las teorías constructivistas existentes, las que proporcionaran un sustento para la comprensión e interpretación de las causas encontradas. Posteriormente se desarrolló el modelo de libro y el libro de estructura de datos aplicando el modelo desarrollado.

El modelo pedagógico del libro, tiene la estructura que se presenta a continuación:

- Planteamiento de una situación cotidiana que involucre las partes básicas del tema que se desea plantear, con el objetivo de desarrollar las habilidades de abstracción simple.
- Cuestionario a contestar del lector, con el objetivo de orientarlo en la identificación de las características que debe notar y en los conceptos que deberá manejar posteriormente.
- Actividad Complementaria al cuestionario, que permitirá al lector aplicar los conceptos extraídos por él, y aplicarlos en una situación que requiera de su aplicación, probando así la comprensión de la información.
- Explicación teórica, una vez preconcebidas las ideas en el alumno es necesario agregar la terminología y reforzar los conceptos extraídos y apoyarlo en la organización científica de la información, esto se realizará mediante una revisión teórica del objeto estudiado.
- Planteamiento de un problema resuelto donde se apliquen los conocimientos recién adquiridos con el objetivo de mostrar la forma en que se aplicarán en el área de la programación.
- Revisión de sus respuestas anteriores para realizar los ajustes necesarios de acuerdo con los nuevos conceptos adquiridos.
- Propuesta de un problema a resolver, orientándolo respecto de la solución del problema mediante pistas que lo orienten en la resolución del mismo.
- Solicitud de diseño y resolución de un problema por parte del lector donde realice la aplicación de los conceptos revisados y se genere en el alumno las habilidades de abstracción reflexiva necesarias para la aplicación de los conceptos.

Finalmente, se desarrolló el libro de texto de estructuras de datos basándose en el modelo y la estructura construida a partir de la investigación realizada.

Otra consideración importante en el desarrollo del libro, fue el diseño del material y el manejo de lenguaje empleado para la explicación de los conceptos; para eso se emplearon algunas las directrices del ILSMH para la redacción de documentos de fácil lectura (Freyhoff, Hess, Kerr, Menzel, Tronbacke, & Van Der Veken, 1998).

Adicionalmente, los autores consideran que la propuesta realizada en la investigación puede contribuir al proceso de enseñanza de las estructuras de datos, porque toma en consideración el nivel de desarrollo actual de los alumnos que ingresan a la institución y está construido con una estructura que favorezca el anclaje de la información y la comprensión mediante ejemplos y prácticas contextualizadas, además del reforzamiento de los conceptos para favorecer la meta cognición.

Otro trabajo interesante es el que actualmente se desarrolla en el Instituto Politécnico Nacional en Culhuacan – México por *Beatriz Dolores Guardián Soto*. El trabajo titulado “La tutoría a través de técnicas virtuales de enseñanza en la asignatura de estructuras de datos” tiene como propósito proponer una

metodología para fomentar el uso de técnicas de enseñanza por parte de los docentes de la asignatura Estructuras de Datos. Para lograr su cometido, la autora realizó una serie de investigaciones sobre las estrategias impartidas por los docentes de dicha asignatura.

El fundamento teórico estuvo basado en el constructivismo, en particular la teoría de asimilación de *David Ausubel* y de los investigadores difusores de su teoría como son el Dr. *Novak* y del Dr. *Gowin* con el diseño de herramientas metodológicas para el educador.

La autora está de acuerdo con el uso de herramientas informáticas como apoyo a la labor de enseñanza por parte del docente y no coloca en tela de juicio los resultados obtenidos mediante ese tipo de técnicas. Sin embargo, considera que la “tutoría” va más allá de una enseñanza y por consiguiente es necesario utilizar una “enseñanza estratégica” puesto que no solo basta con enseñar los contenidos temáticos de la asignatura sino también estrategias que le ayuden al alumno a asimilar esos conocimientos.

Propone entonces la autora, el uso de herramientas informáticas como apoyo a la labor del docente pero también el uso de material adicional como videos, recursos multimedia, entre otros de tal forma que el estudiante también sea parte activa del proceso.

Actualmente, el trabajo se encuentra en su fase de recolección de evidencias y se esperan buenos resultados.

8. NIVEL NACIONAL

Los docentes de educación superior en Colombia, quienes imparten la cátedra de *Estructuras de datos* en los programas de Ingeniería de Sistemas; no se han quedado atrás, han dado muestras de la necesidad de cambiar las técnicas de enseñanza en la asignatura *Estructuras de datos*.

Un aporte importante al tema de estudio, es el trabajo desarrollado por *Gloria Inés Giraldo Echeverri* titulado “Nuevo esquema metodológico en la enseñanza de la asignatura *Estructuras de datos y programación*”. Desde el primer semestre del año 2000, la autora ha trabajado en una nueva metodología para toda la facultad de ingeniería en la Universidad Nacional de Colombia, sede Bogotá. Esta nueva metodología se basa en una estandarización en la forma de desarrollar el curso de *Estructuras de datos y programación*. Como antecedente a la problemática encontrada por la docente, cabe resaltar que los docentes que impartían la asignatura utilizaban didácticas diferentes y a criterio propio de cada uno de ellos por consiguiente no existía una uniformidad en lo referente al desarrollo de la

asignatura *Estructuras de datos y programación* trayendo como consecuencia inconformismo por parte de los estudiantes al sentir que el nivel de conocimientos no era el mismo comparado con el impartido en otros grupos y consecuentemente inconformismo por parte de los docentes que impartían los cursos que tenían como pre-requisito esta asignatura. Conscientes de la problemática, los profesores de *Estructuras de datos y programación* conformaron un grupo de trabajo que busca mejorar el servicio que ofrece el programa de Ingeniería de Sistemas, siendo el compromiso aplicar un nuevo esquema metodológico de enseñanza, con los siguientes objetivos, entre otros:

- Unificación del contenido, didácticas de enseñanza y evaluación.
- Optimización del uso de los recursos didácticos utilizados por los docentes y existentes en la facultad de ingeniería de sistemas.
- Trazar los lineamientos para la futura incorporación de apoyos virtuales como apoyo y fortalecimiento en los recursos para el proceso de enseñanza por parte de los docentes.

De un estudio realizado en el primer semestre de 1999 en cuanto a metodología y enseñanza de la asignatura se observó que durante ese semestre existieron 17 cursos de *Estructuras de datos y programación* impartidos por 14 profesores de los cuales seis eran de planta y ocho de cátedra. Se recopilaron los programas desarrollados por once de ellos y se entrevistaron personalmente a cada uno de los docentes para conocer acerca de los objetivos, técnicas y métodos de enseñanza, estrategias didácticas aplicadas, contenidos de la asignatura, bibliografía, formas de evaluación y lenguajes de programación utilizados. De dicho estudio se pudo concluir lo siguiente:

- Todos los docentes utilizaron el esquema de clases magistrales y prácticas de laboratorio.
- Los conceptos básicos fueron enseñados con diferentes enfoques: programación estructurada, programación orientada a objetos.
- Como apoyo al estudiante, si el profesor era de planta existía un horario para brindar asesorías a estudiantes mientras que si el profesor era de cátedra este espacio no existía.

Con base en los resultados obtenidos en el estudio, se realizaron ajustes al programa de la asignatura siendo los principales la unificación de los contenidos temáticos, la utilización de los recursos didácticos, la elaboración de material de soporte y apoyo al desarrollo de la asignatura, la creación de una página web del curso para intercambiar opiniones e ideas, la elaboración de las evaluaciones fue unificada, preparación de conferencias haciendo uso de diapositivas.

El trabajo realizado consistió además, en el fortalecimiento de la elaboración de recursos y material didáctico para la enseñanza de la asignatura. De esta manera, no se da importancia al hecho de que existan cambios en la planta docente puesto

que ya existe una estructura del desarrollo de la asignatura y el docente que se incorpora por primera vez debe seguir la metodología de enseñanza que fue definida y organizada.

Por otro lado, *Marlene Lucila Guerrero Julio y Sergio Arturo Medina Castillo* docentes del programa de Ingeniería de Sistemas de la Universidad Cooperativa de Colombia (Bucaramanga – Santander) realizaron un estudio titulado: *Una estrategia para el apoyo de los procesos de enseñanza – aprendizaje de la programación en Ingeniería de Sistemas utilizando objetos virtuales de aprendizaje*. Un objeto de aprendizaje es definido por el Ministerio de Educación Colombiano como: “Todo material estructurado de una forma significativa, asociado a un propósito educativo y que corresponda a un recurso de carácter digital que pueda ser distribuido y consultado a través de la internet. EL objeto de aprendizaje debe contar además con una ficha de registro o metadato consistente en un listado de atributos que además de describir el uso posible del objeto, permita la catalogación y el intercambio del mismo”. De esta manera el OVA se convierte en una herramienta de tecnología de información cuyo principal objetivo es el de convertirse en un apoyo del proceso de enseñanza por parte del docente.

La asignatura *Estructuras de Datos* es un componente de programación y los autores diseñaron en primera instancia un modelo OVA para dicha asignatura el cual se puede apreciar en su página de presentación en la siguiente página.

Los autores concluyen que los Objetos Virtuales de Aprendizaje pueden ser utilizados para apoyar los procesos de enseñanza de la asignatura *Estructuras de Datos* siempre y cuando se acompañen de un diseño instruccional claro y detallado y se analicen las competencias que se desean formar en los estudiantes con el fin de generar las estrategias de enseñanza propicias para la vinculación de las actividades y los temas de estudio. Además, los OVA no deben centrarse únicamente en la sistematización de contenidos sino que también deben contener asociaciones con diferentes herramientas de tecnología de información que permitan garantizar el proceso de enseñanza a través de soluciones informáticas.

Otro artículo que aporta aspectos importantes al tema de estudio, es el publicado en la revista “Educación en Ingeniería” en Junio de 2009 titulado “Un cambio de paradigma en la enseñanza de *Estructuras de datos y fundamentos de programación* en Ingeniería de Sistemas” elaborado por *Ricardo Timarán Pereira (Universidad de Nariño, san Juan de Pasto), Anívar Chaves Torres y Javier Jiménez Toledo (Institución Universitaria CESMAG, San Juan de Pasto), Juan Carlos Checa Mora (Universidad Cooperativa de Colombia, San Juan de Pasto), Hugo Ordoñez Erazo (Universidad Mariana, San Juan de Pasto), Constanza Colunge (Corporación Universitaria Autónoma de Nariño, San Juan de Pasto)*. En el artículo, los autores presentan los resultados del proyecto de investigación cuyo objetivo fue aplicar el modelo de programación funcional con el lenguaje Scheme en la enseñanza de la asignatura *Estructuras de datos y programación* en el

programa de Ingeniería de Sistemas. La investigación fue desarrollada en cinco instituciones de educación superior de la ciudad de Pasto (Colombia), que forman parte de la red universitaria de investigación en Sistemas de Nariño – RUISNAR.

Los autores; a partir del debate existente sobre cuál es el modelo de enseñanza más apropiado para la asignatura *Estructuras de datos y programación* y cuál es el modelo de programación a utilizar como soporte tecnológico, propusieron la utilización del lenguaje Scheme como soporte didáctico en la enseñanza de la asignatura. El propósito de su investigación era comprobar que la utilización del modelo de programación funcional con lenguaje Scheme en la enseñanza de la asignatura *Estructuras de datos y programación* contribuye más que el modelo imperativo, en el desarrollo de habilidades en la formulación de algoritmos y desarrollo de programas.

La investigación se desarrolló bajo el enfoque cuantitativo, aplicando el método empírico-analítico. Se propuso un diseño experimental con pre-prueba, posprueba y grupos experimental y de control. El diseño corresponde a un cuasiexperimento, se estableció como grupo experimental, los estudiantes de un determinado semestre que cursan una asignatura de *Estructuras de datos y programación* (ver cuadro 1) y como grupo de control los estudiantes que ya cursaron la asignatura en un semestre anterior (ver cuadro 2). La técnica de recolección de datos fue la encuesta. Se utilizaron cuestionarios para determinar los conocimientos sobre los fundamentos de programación y ejercicios de programación para establecer la aplicación de dichos fundamentos.

Tabla 1: Grupos experimentales

Institución	Sem.	Asignatura	I.H.S.	Número estudiantes
Universidad de Nariño	3	Taller de programación II	6	74
Institución Universitaria CESMAG	2	Modelos de Programación	6	79
Universidad Mariana	8	Electiva Programación Funcional	4	32
Universidad Cooperativa	9	Ingeniería del Conocimiento	4	14
Corporación Universitaria Autónoma de Nariño	2	Programación Funcional	4	22
Total estudiantes				216

Elaborado por los investigadores

Tabla 2: Grupos de control

Institución	Sem.	Asignatura	I.H.S.	Número estudiantes
Universidad de Nariño	5	Estructuras de información	4	38
Institución Universitaria CESMAG	4	Programación orientada a objetos	6	61
Universidad Mariana	9	Desarrollo de aplicaciones multiplataforma	4	12
Universidad Cooperativa	10	Electiva Profesional	4	7
Corporación Universitaria Autónoma de Nariño	5	Programación O-O	4	6
Total estudiantes				124

Elaborado por los investigadores

Al inicio del curso se aplicó una evaluación para determinar los conocimientos de los fundamentos de programación que ya poseían los estudiantes. Los resultados se presentan en el siguiente cuadro:

Tabla 3: Resultados pre evaluación grupos piloto

No.	Pregunta	% No		
		Correctas	Incorrectas	Contestadas
1	Un paradigma de programación es...	43,9	37,5	18,6
2	El lenguaje de programación Scheme está asociado al paradigma...	29,2	29,8	41,0
3	Una serie finita de pasos lógicos para solucionar un problema recibe el nombre de...	63,0	26,0	11,0
4	Una variable es...	89,9	9,6	0,5
5	En programación, una función es...	63,8	29,8	6,4
6	Cuál de las siguientes expresiones esta en notación prefija	48,6	25,3	26,0
7	Dentro de un programa una estructura condicional permite	77,6	14,7	7,7
8	Dentro de un programa un ciclo se utiliza para	86,0	11,8	2,2
9	Un arreglo es	51,5	28,2	20,3
10	Se entiende por recursividad ...	33,7	45,5	20,8
Promedio		58,7	25,8	15,5

Elaborado por los investigadores

Se evidencia que todos los temas son conocidos por algunos de los estudiantes, algunos por la mayoría como el concepto de variable y ciclo, otros son conocidos solo por una minoría como es el caso de lenguaje Scheme y recursividad. El

mismo instrumento se aplicó a los estudiantes que en cada institución habían cursado la asignatura en el semestre anterior bajo el modelo imperativo (grupos de control). Los resultados se presentan en siguiente cuadro:

Tabla 4: Resultados pre evaluación grupos de control

Pregunta	% Correctas	% Incorrectas	% No contestadas
1	56,6	32,6	10,8
2	31,0	41,6	27,4
3	73,2	23,6	3,2
4	94,5	5,5	0,0
5	72,1	24,6	3,3
6	53,8	23,5	12,7
7	77,8	13,5	8,8
8	93,6	6,4	0,0
9	68,6	21,0	10,4
10	43,6	36,5	19,9
Promedio	66,5	23,9	9,7

Elaborado por los investigadores

De igual manera que los grupos experimentales, los temas más conocidos fueron variables y ciclos y los menos conocidos, lenguaje Scheme y recursividad. Este resultado se presenta por cuando estos temas son propios del modelo funcional y no se estudian en los cursos diseñados desde el modelo imperativo.

Comparando los resultados de la evaluación diagnóstica con los grupos piloto y de control, se puede decir que el conocimiento del grupo piloto, por sus condiciones académicas, es inferior en 6,8% en los temas evaluados, al grupo de control. Este resultado era el esperado al iniciar la investigación.

Al finalizar el semestre se realizó la post-evaluación con el mismo instrumento de la pre-evaluación para determinar el avance de los grupos piloto en el aprendizaje de los fundamentos de programación.

Los resultados obtenidos demostraron que la estrategia de enseñanza del modelo de programación funcional con el lenguaje Scheme, en los primeros cursos de *Estructuras de datos y programación*, facilita a los estudiantes de ingeniería de sistemas, el desarrollo de habilidades en la formulación de algoritmos, elaboración de programas y la comprensión de las *Estructuras de datos* fundamentales de un lenguaje de programación, con respecto al modelo imperativo. Mejora la capacidad de entender y utilizar la recursividad como una herramienta para

optimizar su desempeño en la construcción de programas que brinden una solución efectiva a problemas específicos. Con base en esta experiencia y considerando sus beneficios, se recomendó incluir dentro de los planes de estudio de ingeniería de sistemas de las instituciones participantes, la enseñanza de este modelo de programación como herramienta didáctica.

Como trabajos futuros están el documentar los resultados del rendimiento de los estudiantes que participaron en este proyecto, en las asignaturas posteriores del componente de programación y formalizar en los planes de estudio del programa de Ingeniería de Sistemas, de las universidades del departamento de Nariño (Colombia), la programación funcional con Scheme, como el primer curso de enseñanza de la asignatura *Estructuras de datos y programación*.

El estudio más reciente en Colombia, es el realizado por *Alejandra Zuleta Medina (Docente universitaria, centro de estudios superiores María Goretti CESMAG – Pasto, Colombia)* y *Anivar Chaves Torres (Universidad Nacional Abierta y a Distancia, UNAD, Pasto, Colombia)*. En la revista “Unimar” No. 57 publicada en Julio de 2011 aparece el artículo titulado “Uso de herramientas informáticas como estrategia para la enseñanza de *Estructuras de datos y programación de computadores*”. Los autores, en su artículo presentan los resultados de una revisión crítica de varias investigaciones que tratan la problemática existente sobre la enseñanza de las asignaturas *Estructuras de datos y programación de computadores* en particular aquellas investigaciones que se basan en la utilización de herramientas informáticas para intentar solucionar algunos de los múltiples problemas que se presentan en el proceso de enseñanza de la asignatura tema de estudio.

Para los autores, la enseñanza de la asignatura *Estructuras de datos y programación* no se limita a la explicación de conceptos y teorías, sino que busca desarrollar habilidades para trabajar con conceptos abstractos, como por ejemplo la manipulación de datos por medio de algoritmos, la sintaxis y la semántica de los lenguajes de programación y el metalenguaje utilizado para escribir algoritmos y programas. Las investigaciones de Moreno (2003), Moroni (2005), Soler y Lezcano (2009), coinciden en afirmar que la enseñanza de la asignatura *Estructuras de datos* es compleja, por cuanto implica el manejo de abstracciones, la aplicación de una lógica propia del paradigma de programación y la construcción de expresiones, atendiendo a una sintaxis y a una semántica propias de un lenguaje que no siempre es fácil de asimilar por parte de los estudiantes.

Según Oviedo y Ortiz (2002) y Hernández (2006), los docentes no tienen en cuenta algunas debilidades de los educandos, que requieren ser tenidas en cuenta en el proceso de enseñanza como son: el desconocimiento de la materia, pues muchos estudiantes de Ingeniería de Sistemas ingresan al programa sin experiencia previa en programación y por ende sin ninguna habilidad relacionada con esta materia; a algunos otros les falta disciplina para aprender a programar,

pues el desarrollo de habilidades exige trabajo constante y perseverancia; cuando estas dos situaciones hacen presencia, puede ocurrir que pierdan el interés por la asignatura, lo que repercute directamente en los resultados que obtienen.

Por su parte, Trejo (2003) considera que el problema radica no solo en las estrategias de enseñanza de la asignatura sino el desfase que existe entre los temas de la asignatura con respecto a los de otras asignaturas del currículo a los que se espera que apoye, como son las Matemáticas y la Física. Una muestra de esta situación es que los conceptos estudiados en los primeros cursos de la asignatura no permiten su aplicación para desarrollar modelos y gráficas matemáticas, ya que esto exige conocimientos más avanzados de programación

También existe el problema de la diferencia de estilos de enseñanza, especialmente cuando no se hace un diagnóstico para determinar los conocimientos previos del estudiante y no se utiliza material educativo apropiado para dar soporte a la enseñanza; además, hace falta una cuidadosa planificación, organización y evaluación del proceso de mejoramiento del curso, como lo menciona Llamosa (2003), quien investigó las dificultades existentes en el proceso de enseñanza de la asignatura *Estructuras de datos* bajo el paradigma de la programación orientada a objetos.

Además de las dificultades anteriores, los autores de este documento consideran que parte del problema de la enseñanza de la asignatura *Estructuras de datos*, es que los ejercicios desarrollados en clase por parte del profesor no presentan un nivel de complejidad similar a los problemas reales; por esta razón no permiten desarrollar destrezas en diseño, modelado y programación, como tampoco, aprender todos los conceptos de la asignatura, motivo por el cual, cuando los ingenieros de sistemas tienen que solucionar problemas reales, no cuentan con el conocimiento y la destreza suficientes para hacer un trabajo de calidad.

En consecuencia y con base en De la Cruz y Gamboa (2007), es necesario renovar las estrategias utilizadas para la enseñanza de la asignatura tema de estudio, pues muchas de ellas fueron diseñadas para personas con un perfil diferente al de los programadores actuales, particularmente en cursos de introducción, por la brevedad de los cursos y lo extenso del temario.

Muchos investigadores en su afán de mejorar la calidad de la enseñanza de la asignatura han desarrollado y experimentado con diferentes recursos educativos; algunos ejemplos de estas investigaciones son:

- Satorre (1996) utilizó el computador y un programa educativo (programmin en el laberinto) para enseñar la asignatura *Estructuras de datos*. El autor asegura que con esta estrategia didáctica, los estudiantes logran comprender y aplicar conceptos fundamentales como abstracción, modularidad, recursividad y complejidad algorítmica.

- En el Departamento de Ciencias e Ingeniería de la computación de la Universidad Nacional del Sur (Bahía Blanca), Moroni y Señas (1996) desarrollaron un editor para reducir los procesos no cruciales en la resolución de problemas, el diseño y la formulación de programas, que además permite comprobar errores mediante la confección automática de trazas. Como resultado de esta experiencia, observaron que los estudiantes que utilizaban el editor, necesitaban más tiempo para diseñar los algoritmos que quienes lo diseñaban en papel, pero este tiempo estaba compensado por algoritmos más ordenados, con posibilidad de refinamiento progresivo y corrección. El tiempo adicional utilizado en el diseño lo recuperaban en la fase de codificación, actividad que hacían en menor tiempo, debido a que sus algoritmos ya habían sido verificados en el editor y podía generarse el código automáticamente; por otra parte, los estudiantes habían aprendido paulatinamente el lenguaje de programación.
- Papert (1999) incursionó en el uso de software educativo como complemento didáctico con enfoque constructivista; además implementó una línea correspondiente a los lenguajes para el aprendizaje, dando como resultado que el lenguaje LOGO, desarrollado en el Instituto Tecnológico de Massachussets (MIT), fue y es utilizado en numerosas escuelas y universidades en un sentido constructivista del aprendizaje, enfoque desde el cual se ha propuesto diferentes estrategias de enseñanza basadas en recursos tecnológicos, como: ambientes de modelación, ambientes hipermedia, simulación por computador, estrategias basadas en mapas conceptuales, ambientes colaborativos.
- Almeida y sus compañeros (2003) desarrollaron una herramienta llamada EDApplets, aplicación web orientada a la enseñanza de la asignatura *Estructuras de datos*, orientada principalmente a la animación y visualización mediante trazas de algoritmos y estructuras de datos, que permite descubrir diversos aspectos en una enseñanza, que puede ser dirigida a diversos tipos de aprendizaje: activo / reflexivo, metódico / intuitivo, visual / oral. Los autores de esta investigación no presentan resultados respecto a la contribución de esta herramienta en el proceso de aprendizaje pero se deduce que su uso facilita la enseñanza y permite a los estudiantes disponer de una representación visual del comportamiento de los algoritmos de *Estructuras de datos*, lo que incide favorablemente en la comprensión de los mismos.
- Trejo (2003) integró los temas de la asignatura *Estructuras de datos* con aquellos de otras asignaturas, cambiaron el tradicional lenguaje **C** por **Scheme**, que es más sencillo y fácil de aprender, y al mismo tiempo más apropiado para la programación matemática por ser un lenguaje funcional híbrido, obteniendo como resultado que los estudiantes que comenzaron el aprendizaje de la asignatura con el modelo funcional, mostraron facilidad para comprender y asimilar conceptos avanzados de programación. No obstante, tuvo que volver a lenguajes más conocidos como Java, dado que

pocos docentes conocen Scheme, pero mantuvieron el enfoque integrado al currículo.

- Llamosa (2003) presentó un sistema Hipermedia Adaptativo para la enseñanza de los conceptos básicos de la programación orientada a objetos, el cual aplica y ajusta la presentación del contenido multimedia conforme al estilo de aprendizaje y nivel de conocimiento alcanzado por un estudiante, para que éste acceda a los contenidos y a las actividades de aprendizaje, de acuerdo con sus características personales. Este estudio permitió a los docentes conocer las tendencias en los estilos de aprendizaje y planificar sus estrategias de enseñanza con dicho conocimiento. Los estudiantes reciben instrucción más personalizada, lo que incide en su desempeño.
- Jiménez (2005) propuso integrar la visualización y el role-play para reducir la complejidad de seguir la traza de los modelos de interacción entre objetos, cuando se genera muchos mensajes. La visualización consiste en representar las características y el comportamiento de los sistemas o de los objetos, utilizando herramientas gráficas, animaciones y otros medios visuales (Price. 1993), como es el caso de la representación de las diferentes vistas de un sistema mediante diagramas de Lenguaje Unificado de Modelado, UML. Jiménez (2005) no hace referencia a la incidencia del role-play en el aprendizaje, pero resalta el aporte a la metodología de enseñanza en la medida en que la herramienta permite mayor posibilidad de interacción del estudiante.
- La fundación Gabriel Piedrahita Uribe ha desarrollado un trabajo de campo sobre la enseñanza de la asignatura *Estructuras de datos* en el Instituto Nuestra Señora de la Asunción, INSA, de Cali y en el Instituto Colombiano de Estudios Superiores de Incolda, ICESI, universidad de la misma ciudad, con el fin de elaborar y validar la herramienta Micromundos y el entorno de programación alterno, Scratch, experiencia en la cual participa un grupo de docentes pertenecientes a tres instituciones educativas. Esta investigación ha mostrado que al resolver problemas mediante programas de computador, los estudiantes comprenden mejor los conceptos matemáticos asociados a los problemas que tratan y mejoran su capacidad para interpretarlos y resolverlos. Con base en los resultados obtenidos recomiendan que la enseñanza de la asignatura *Estructuras de datos* debe ser incorporada desde la educación primaria (López, 2008).
- Pérez (2008) por su parte, propuso una estrategia para mejorar la enseñanza de la asignatura *Estructuras de datos* basada en un ambiente de aprendizaje con un editor interactivo de algoritmos, un constructor automático de trazas y un traductor de pseudocódigo a seis lenguajes de programación. Entre los resultados menciona que diseñar la solución de un problema y generar el código en varios lenguajes de programación mejora el aprendizaje de los estudiantes y aunque ellos no conozcan previamente esos lenguajes, al ver el código poco a poco lo comprenden y lo aprenden

- También en Pasto se investigó la aplicación del lenguaje Scheme como sustituto de los utilizados tradicionalmente (C, Java, Visual Basic) en cinco programas de ingeniería de las instituciones de educación superior. La investigación permitió concluir que el uso de esta herramienta como estrategia didáctica facilita abordar más contenidos debido a su facilidad, y por ser un lenguaje diseñado con base en el modelo de programación funcional, favorece el aprendizaje del manejo de funciones y la aplicación de la recursividad. (Timarán, 2009) y Chaves (2009).

Como conclusiones a su investigación, los autores plantean las siguientes reflexiones:

- La enseñanza de la asignatura Estructuras de datos es compleja y por ende es necesario utilizar herramientas que fortalezcan la labor de la enseñanza docente.
- Es necesario contar con la dedicación del estudiante, herramientas precisas, disposición del docente, técnicas y estrategias apropiadas para la enseñanza.
- Han sido desarrolladas varias investigaciones basadas en la utilización de herramientas para apoyar la enseñanza. En todos los casos, el uso de herramientas bajo una estrategia de enseñanza ha presentado resultados favorables, ya se trate de despertar el interés de los estudiantes, como de facilitar las interacciones, mejorar la identificación y corrección de errores o mejorar las habilidades de análisis y solución de problemas.
- Desde la perspectiva curricular se ha encontrado que en muchos programas falta articulación entre las asignaturas de Estructuras de datos y otras asignaturas del plan de estudios, que deben aportar conceptos y técnicas que ayuden al desarrollo de programas, como son: Matemáticas, Física, álgebra y estadística. Los ejercicios y proyectos que se desarrolla al interior del aula de clases no mantienen relación con el desempeño profesional.
- Se ha desarrollado y probado diferentes herramientas para la enseñanza de la asignatura y aunque han dado buenos resultados, muchas de ellas no son conocidas y por ello no son aplicadas fuera del contexto donde fueron desarrolladas. Es necesario que los docentes quienes imparten la asignatura tengan una actitud investigativa y estén explorando continuamente los avances en el desarrollo de software educativo, que puedan incorporar en sus estrategias de enseñanza.
- Muchas estrategias didácticas propuestas por los investigadores pueden ser combinadas para conseguir mejores resultados, como por ejemplo: el uso de mapas conceptuales para el estudio de los aspectos teóricos, con herramientas para el diseño y evaluación de algoritmos o programas para el seguimiento de trazas o simulación, según convenga a los temas del curso.
- La asignatura Estructuras de datos, al igual que cualquier otra disciplina, no puede ser enseñada a todos los estudiantes y en todos los contextos por

igual. Cada grupo es diferente y requiere una estrategia distinta; por ello es necesario evaluar y renovar las estrategias de enseñanza y los recursos que éstas incluyen.

- La enseñanza de la asignatura Estructuras de datos no es un problema aislado, sino que está estrechamente vinculado al currículo del programa y al micro-curriculum de la asignatura.

9. NIVEL LOCAL

La universidad el Bosque en aras de mejorar sus procesos de enseñanza y en particular en la asignatura de *Estructuras de datos* ha realizado diversos cambios al interior de su programa de Ingeniería de Sistemas con ese propósito. En primer lugar, es muy importante resaltar las estrategias metodológicas seguidas por los profesores de la asignatura:

- Método expositivo e interrogativo: consiste en el sistema clásico de enseñanza en que se imparten nuevos conocimientos, con el añadido de preguntar directamente a los alumnos para indagar sus conocimientos adquiridos.
- Método de aprendizaje por descubrimiento: consiste en proponer un problema a los alumnos y que a través de unas indicaciones básicas sobre referencias bibliográficas y de consulta, sean capaces de encontrar la solución.
- Método de proyectos: consiste en asignar proyectos de trabajo individuales o en grupo, en que los alumnos deben alcanzar unos objetivos.

En los últimos dos años, los docentes del programa de Ingeniería de Sistemas de la Universidad el Bosque se han preocupado por la situación presentada y es por ello que se han dado a la tarea de realizar un trabajo más colaborativo, en donde de manera permanente existe una retroalimentación al proceso que se lleva a cabo desde los primeros semestres por parte de los docentes. Además, se ha seguido una línea denominada "línea de programación" de la cual hace parte la asignatura *Estructuras de datos*. Se busca con ello, que los docentes de la asignatura *Estructuras de datos* (y no solo ellos) expongan sus necesidades con respecto a los conocimientos básicos que debe tener un estudiante antes de abordar la asignatura tema de estudio para que dichos conocimientos sean estudiados en las otras asignaturas.

Otra de las dificultades existentes y que ha sido superada si bien no en un cien por ciento se ha logrado un gran avance, es la elaboración de un contenido temático acorde a las exigencias del mercado. Anteriormente, se cubrían temas que no tenían ningún valor agregado al futuro profesional y por consiguiente se perdía un tiempo muy valioso en el estudio de casos que no eran muy significativos.

Una de las estrategias didácticas impuesta desde este primer semestre de 2012 es la de seguir un único micro-curriculum, diseñado por los docentes que imparten dicha asignatura tanto en la jornada diurna como la nocturna y durante el transcurso del semestre se han hecho reuniones a manera de seguimiento en aras de realizar los ajustes respectivos en caso de ser necesario. El programa de Ingeniería de Sistemas, espera recoger estos frutos en un mediano plazo.

10. REFLEXIONES DEL AUTOR

A raíz de la dificultad mostrada por los alumnos en el aprendizaje de los temas que constituyen la asignatura *Estructuras de datos* ha sido necesario utilizar diversas estrategias para su enseñanza y han sido innumerables los trabajos realizados en la búsqueda de mejorar y de innovar el proceso de su enseñanza. La preocupación existente es el común denominador por parte de los docentes y se sigue trabajando de manera incesante en buscar alternativas tendientes a mejorar la enseñanza de la asignatura.

A nivel mundial, se evidencia el tiempo dedicado y trabajos de investigación realizados en diferentes escenarios. Algunos investigadores se han centrado en la importancia de utilizar recursos informáticos como apoyo a la labor docente; en otros casos, no solo ha bastado el utilizar una herramienta informática sino también en cambios en los temas tratados en la asignatura *Estructuras de datos*. Otros investigadores, por el contrario han trabajado en el uso de Mapas conceptuales como estrategia metodológica. Independientemente del método, de la técnica, de la metodología, es notorio el interés centrado en la búsqueda de alternativas y la preocupación existente en mejorar la calidad de la enseñanza. No cabe duda que los temas tratados al interior de la asignatura son complejos pero necesarios y por ende la enseñanza de manera tradicional (explicaciones por parte del docente y desarrollo de ejercicios modelos) no ha dado los frutos requeridos y ello, por diversas razones: la transmisión del conocimiento no ha sido fácil de esa manera, la preparación académica e intelectual por parte del docente, la falta de motivación por parte del estudiante, entre otros.

A nivel nacional, también existen trabajos interesantes sobre el tema. Aunque no son muchos, las investigaciones existentes aportan un gran ingrediente a la problemática de la enseñanza de la estructuras de datos. Al igual que los trabajos encontrados a nivel mundial, en nuestro país se ha trabajado en el uso de herramientas informáticas adicionales como apoyo al proceso de enseñanza, se evidencia la preocupación existente sobre los métodos de enseñanza y las conclusiones que han arrojado tales investigaciones permiten realizar un análisis sobre lo que se ha hecho e invita a seguir trabajando en futuras investigaciones.

La Universidad el Bosque, no se ha quedado atrás. Aunque no existen a la fecha investigaciones profundas en el tema, ha sido y sigue siendo una tarea ardua por parte de los docentes el buscar alternativas de enseñanza, se han realizado reuniones permanentes y se han analizado los temas tratados en la asignatura *Estructuras de datos*, en ocasiones se han incluido algunos temas considerados importantes para la formación del profesional en Ingeniería de Sistemas y otras veces se han suprimido otros. No se utiliza un lenguaje de programación específico como soporte a las prácticas de la asignatura y se trata que el estudiante comprenda teóricamente y vivencialmente los conceptos estudiados. Si esto es así, el hecho de utilizar un lenguaje de programación será trivial para el estudiante si tiene claridad en los conceptos teóricos aprendidos.

Si bien es cierto que existen muchas investigaciones sobre el tema, es muy poco el material encontrado específicamente para la asignatura *Estructuras de datos*. No es común encontrar investigaciones extensas sobre el tema pero lo hecho hasta el momento es muy importante y por consiguiente se debe trabajar mucho más en este tipo de problemática. Este trabajo, es un aporte y constituye un grano de arena más en la búsqueda de esas herramientas didácticas efectivas para lograr mejorar la enseñanza de la asignatura *Estructuras de datos*.

11. MARCO HISTÓRICO

Con respecto a la enseñanza de las *Estructuras de Datos* son muchos los expertos que han tratado esta problemática en diferentes escenarios entre ellos tenemos a Brassard G., Bratley P. Con diferentes artículos y libros sobre el tema, Sin embargo "Fundamentos de algoritmia".(1999), es una obra en la que se ofrece un estudio detallado sobre el análisis y diseños de algoritmos así como las *estructuras de datos* adecuadas para obtener algoritmos óptimos. Según los autores, la enseñanza de los algoritmos y las estructuras de datos, tienen sus raíces en la antigua Grecia con los filósofos clásicos. El aprendizaje con Platón y Aristóteles, y los algoritmos con Euclides (300 a.c.). Este último, fue quien desarrolló el primer algoritmo del que se tiene noticia, el concepto de algoritmo es una de las ideas básicas de las matemáticas y en otras tareas de la vida diaria, también uno de los más antiguos, esta palabra hace su aparición en el siglo IX d.c., se tiene conocimiento de que este término "algoritmo" proviene del nombre de un matemático medieval de Uzbekistán, Al-Khowárizmí, quien dió tales reglas en este siglo.

Así mismo, la enseñanza de la asignatura "*Estructuras de datos*" es tan antigua como el nacimiento de la programación de computadoras y se ha convertido en estudio obligatorio en todos los currículos de Ingeniería de Sistemas desde finales

de los años setenta y sobre todo en la década de los setenta cuando apareció el lenguaje PASCAL.

Por otro lado, la evolución de las tecnologías de información, el auge de la red internet, las redes sociales, entre otras ha influido en muchos cambios de nuestra sociedad a lo largo del tiempo; esto indudablemente hace eco en la enseñanza de la asignatura *Estructura de datos* en donde también se han generado cambios en el transcurso del tiempo, los docentes han buscado diversidad de enfoques y tendencias. A fecha de hoy, se puede verificar que no hay un consenso en los métodos y herramientas didácticas utilizadas. Algunas de las tantas razones obedecen a que no existe un método único para la resolución de problemas propios de las *Estructuras de datos*; además, la inexistencia de docentes altamente capacitados en los temas que hacen parte de la asignatura, la utilización inadecuada de las técnicas y metodologías de enseñanza, la falta de creatividad para transmitir el conocimiento de conceptos abstractos sin olvidar la parte motivacional por parte de los estudiantes que sin lugar a dudas influye de manera directa en el proceso de enseñanza.

11.1. El rol del docente en el proceso de enseñanza de la asignatura *Estructuras de datos*

A diferencia de épocas anteriores, los docentes que imparten la asignatura *Estructuras de datos* se encuentran hoy día, ante un volumen creciente de materiales curriculares y elementos auxiliares de enseñanza: de esta gran multiplicación de libros, objetos concretos, mapas, películas, libros de texto, computadoras, software educativo, cd-roms, programas de televisión, medios audiovisuales y tantas otras cosas, ellos deben de alguna manera seleccionar los materiales que han de ser empleados para la enseñanza de la asignatura *Estructuras de datos*. En realidad, disponen de pocas referencias de utilidad general a manera de principios que pudieran ayudarlos a hacer sus selecciones; algunas de ellas, significan decisiones sobre lo que se va a enseñar; otras encierran selecciones de medios en los cuales el contenido ya elegido ha de ser presentado. Muchas de estas ideas modernas, son difíciles de entender, de aceptar y de armonizar con los antiguos conceptos de educación adquiridos por los docentes. Un particular criterio a desarrollar en los docentes ha de ser el de elegir adecuadamente los diferentes recursos didácticos a emplear en el proceso de enseñanza de la asignatura *Estructuras de datos*, considerando el nivel de los alumnos, el currículo de estudios, la metodología y los requerimientos técnicos para su correcta utilización como apoyo a la enseñanza. En muchos casos, representan un riesgo y producen ansiedad del docente dentro del sistema actual, en el que tiene que realizar la selección de dichos materiales; entonces el educador no sólo se encuentra confundido ante una enorme cantidad de productos, sino también desprovisto de principios confiables para tomar una decisión. Necesita de una preparación complementaria en los procesos mediante los cuales los nuevos recursos pedagógicos son desarrollados, perfeccionados y

evaluados para llegar así a apreciar con seguridad su importancia en cuanto a los objetivos educacionales que convengan a cada grupo de alumnos.

11.2. El proceso de enseñanza

La enseñanza es una actividad intencional, diseñada para dar lugar al aprendizaje de los alumnos. Es cierto que hablar de enseñanza requiere hablar de aprendizaje, pero ello no significa que para poder hablar de enseñanza tenga que ocurrir necesariamente el aprendizaje, lo mismo que puedo participar en una carrera y no ganar, o no encontrar algo y realmente haberlo buscado. Existe, por tanto una relación de dependencia entre enseñanza y aprendizaje, pero no es del tipo de relación que supone que no puede haber enseñanza sin aprendizaje. Es decir existe una relación pero no es causal, sino de dependencia ontológica.

La tarea de la enseñanza es lograr el resultado del aprendizaje, aun cuando en realidad tiene más sentido decir que "la tarea central de la enseñanza es posibilitar que el alumno realice las tareas del aprendizaje".

Las tareas de enseñanza tienen que ver, más que con la transmisión de contenidos, con proporcionar instrucciones al alumno sobre cómo realizar las tareas de aprendizaje. En el caso de la asignatura *Estructuras de datos*, el docente tiene una misión bastante complicada con base a la percepción que por parte de los estudiantes se tiene sobre ella. Entre varias de ellas, tenemos:

- Para el estudiante, la asignatura *Estructuras de datos* tiene poca o ninguna utilidad práctica y que se encuentra alejada de la realidad
- La enseñanza de la asignatura es un mecanismo de tortura, constituyéndose el docente en un verdugo.
- Es una asignatura aburrida y muy difícil.

La enseñanza no es un fenómeno de provocación de aprendizaje, sino una situación social que como tal se encuentra sometida a las variaciones de las interacciones entre los aspirantes, así como a las presiones exteriores y a las definiciones institucionales de los roles. Entenderemos, pues, por proceso de enseñanza-aprendizaje, el sistema de comunicación intencional que se produce en un marco institucional y en el que se generan estrategias encaminadas a provocar el aprendizaje.

Con esta definición se resaltan los tres aspectos que mejor caracterizan la realidad de la enseñanza:

- Los procesos de enseñanza ocurren en un contexto institucional, transmitiéndole así unas características que trascienden a la significación interna de los procesos, al conferirle un sentido social.
- Los procesos de enseñanza pueden interpretarse bajo las claves de los sistemas de comunicación humana, teniendo en cuenta las peculiaridades

específicas de aquéllos, una de las cuales es su carácter de comunicación intencional. La intencionalidad nos remite tanto a su funcionalidad social como a su pretensión de hacer posible el aprendizaje.

- El sentido interno de los procesos de enseñanza está en hacer posible el aprendizaje. No hay por qué entender que la expresión "hacer posible el aprendizaje" significa atender a determinados logros de aprendizaje. Como se ha visto, aprendizaje puede entenderse como el proceso de aprender y como el resultado de dicho proceso. Para evitar posibles confusiones convenga decir que el sentido interno de los procesos de enseñanza-aprendizaje está en hacer posibles determinados procesos de aprendizaje, o en proporcionar oportunidades apropiadas para el aprendizaje.

11.3. Recurso didáctico-pedagógico

Los recursos didáctico-pedagógicos son los elementos empleados por el docente para facilitar y conducir el aprendizaje del estudiante (fotos, láminas, videos, software, juego de roles, estudio de casos, entre otros).

Deben ser seleccionados adecuadamente, para que contribuyan a lograr un mejor aprendizaje y se deben tener en cuenta algunos criterios, por ejemplo:

- Deben ser pertinentes respecto de los objetivos que se pretenden lograr.
- Deben estar disponibles en el momento en que se los necesita.
- Deben ser adecuados a las características de los alumnos
- Deben seleccionarse los recursos que permitan obtener los mejores resultados al más bajo costo, que impliquen la mínima pérdida de tiempo y puedan ser utilizados en distintas oportunidades.
- El docente debe prever, seleccionar y organizar los recursos didáctico-pedagógicos que integrarán cada situación de aprendizaje, con la finalidad de crear las mejores condiciones para lograr los objetivos previstos.

Paralelo a ello, los docentes responsables de la asignatura *Estructuras de datos* han buscado formas diferentes para la enseñanza de dicha asignatura, son muy frecuentes los cambios que el docente experimenta y que afecta a un estilo de enseñanza en el que ha primado la transmisión de conocimientos conceptuales, estilo que se ha caracterizado por la exigencia de tener los contenidos claros mentalmente, saber comunicarlos a través de la clase magistral principalmente, tener la capacidad de responder favorablemente a la totalidad de las preguntas que les hiciesen los alumnos y algunas cosas más que afectaban sobre todo a los aspectos formales de su metodología. Estos son los recursos que con el paso del tiempo se van acumulando en forma de una experiencia que, en el mejor de los

casos, produce un dominio de la situación y una cualificación implícita del profesor y a veces, una cierta satisfacción en los alumnos.

12. MARCO LEGAL

En Colombia la Ley General de Educación, define la ingeniería así: “Ingeniería es la profesión que se fundamenta en los conocimientos de las ciencias naturales y matemáticas en la conceptualización, diseño, experimentación y práctica de las ciencias propias de cada especialidad, buscando la optimización de los materiales y recursos, para el crecimiento, desarrollo sostenible y bienestar de la humanidad”. La ingeniería es la aplicación del conocimiento científico basado en las matemáticas, física, química y técnicas para la solución más económica de los problemas que afectan a la sociedad.

13. FUNDAMENTO LEGAL DE LA UNIVERSIDAD EL BOSQUE

La **Universidad El Bosque** es una Universidad privada ubicada en Bogotá, Colombia. Comenzó labores en el año de 1977 y su campus se encuentra ubicado al nororiente de ciudad siendo este un sector privilegiado para sus estudiantes. En la actualidad cuenta con un número aproximado de 9.100 alumnos (2011) y ofrece un total de 20 programas de pregrado, 70 programas de posgrados, 3 cursos preuniversitarios, programas de educación continuada (cursos, diplomados, seminarios, congresos) y un colegio bilingüe.

En medio de las convulsionadas condiciones que el país y el sector de la salud vivían a finales de la década de los setenta, un grupo de 25 profesionales de la salud se unieron para conformar una entidad que atendiera las necesidades en el área de la salud que tenía el país.

Ese esfuerzo se materializó en la creación de la Clínica El Bosque, la cual es el origen de lo que hoy conocemos como Universidad El Bosque. Antes de ser conocida y reconocida como Universidad, esta Institución de Educación Superior fue la Escuela Colombiana de Medicina.

Desde 1977 se iniciaron las labores de la Clínica y de la Escuela facilitando de esta forma la unión entre la práctica y la teoría, haciendo especial énfasis en la humanidad, tanto del paciente como del profesional.

El 27 de julio de 1978, el ICFES reconoce la personería jurídica de la Escuela Colombiana de Medicina. En 1997, gracias al crecimiento y evolución de la institución, el ICFES reconoce la personería jurídica de la Universidad El Bosque.

Ante el Instituto Colombiano para el Fomento de la Educación Superior (ICFES); el programa de Ingeniería de Sistemas, presenta la siguiente cronología:

- Acuerdo 3007 de 1997 (Acta 042 ley 30/ 1992). Se concede licencia para iniciar labores.
- Enero de 1998. Se inicia el primer semestre
- La primera promoción de ingenieros de Sistemas se graduó en el mes de Diciembre del año 2002.
- Septiembre 01 de 1997. Mediante la resolución 378 del 17 de Febrero de 2004, se obtuvo el registro calificado del programa.
- Noviembre 17 de 2010, se obtiene la renovación del registro calificado mediante resolución 10019.

El programa de Ingeniería de Sistemas que ofrece la Universidad el Bosque, posee su sustento legal ante el sistema nacional de Educación Superior, de esta manera.

- Denominación: Ingeniero de Sistemas
- Título expedido: Profesional en Ingeniería de Sistemas
- Duración: 10 semestres (Jornada Diurna), 11 semestres (Jornada Nocturna)
- Nivel: formación pregradual
- Modalidad: Presencial, soportada con una plataforma tecnológica como apoyo a la labor docente.
- Registro: 4952
- Número de créditos: 174

Desde el nacimiento del programa, la asignatura *Estructuras de Datos* ha hecho parte de ella; en sus inicios, la asignatura *Estructuras de datos* se denominaba *Estructuras de información* y solo se dictaba en un semestre del programa. Con la experiencia adquirida en el año 1998 en lo que concierne a la enseñanza de la asignatura *Estructuras de Información* se determina en el año 1999 que es necesario dos semestres para la enseñanza de la asignatura lo cual es aprobado y adicionalmente se estipula que la asignatura *Estructuras de información* debe llamarse *Estructuras de Datos*. En el año 2000, se implementa entonces la enseñanza de *Estructuras de Datos I* y *Estructuras de datos II* para los semestres III y IV respectivamente.

14. MARCO TEÓRICO

Desde una mirada epistemológica la investigación se fundamentará en los teóricos Yolanda Soler Pellicer, docente de la Universidad de Granma-Cuba, Mateo Gerónimo Lezcano Brito, docente de la Universidad Central de Cuba y Jorge A. Villalobos, docente de la Universidad de los Andes de Colombia.

Estos teóricos han coincidido en ser estudiosos de los algoritmos y las estructuras de datos ofreciendo estudios de profundidad. A continuación se presentará la teoría que han expuesto los teóricos frente al objeto de estudio:

Yolanda Soler Pellicer (Universidad de Granma – Cuba) y Mateo Gerónimo Lezcano Brito (Universidad Central “Marta Abreu” – Cuba) durante el año 2008 trabajaron en una investigación titulada “Organización del conocimiento de la asignatura Estructuras de datos y algoritmos para Ingeniería Informática basada en mapas conceptuales” (Artículo publicado en la Revista “Avances en Sistemas e Informática, Vol 5 – No. 3, Diciembre de 2008, Medellín, ISSN 1657-7663). Proponen ellos un sistema de enseñanza que integre diferentes técnicas computacionales y ambientes de autoaprendizaje en búsqueda de nuevas formas didácticas para la enseñanza de la asignatura Estructuras de datos. Para ello, consideran que la utilización de Mapas Conceptuales permitiría llevar a cabo un proceso de enseñanza eficiente y eficaz acorde al contenido temático de la asignatura dada su complejidad en algunos temas. Más allá del uso de un “programa de computador”, de herramientas de software desarrolladas como apoyo a la labor de la enseñanza por parte del docente, los autores consideran que es menester organizar los contenidos de la asignatura y desarrollar una enseñanza “significativa” que pretende que el aprendiz llene los vacíos existentes en su estructura de memoria para lo cual toma en cuenta que los estudiantes no son receptores pasivos de conocimiento, sino por el contrario, participantes activos en la interpretación de los modelos que ellos mismos o el profesor les propone para que intenten aprender aquello que aún no saben.

Escribir un programa de computadora utilizando un lenguaje de programación requiere del alumno varias competencias y habilidades, que involucran básicamente la capacidad de manipular un conjunto de abstracciones interrelacionadas para la resolución de problemas. En tal sentido, consideran los autores que el proceso de enseñanza de un lenguaje de programación es extremadamente complejo y por ello, se propone el uso de mapas conceptuales para la enseñanza de conceptos básicos de programación y desarrollo de algoritmos.

Consideran los autores como conclusiones de la investigación realizada que el uso de Mapas Conceptuales integra recursos que apoyan la enseñanza significativa puesto que contribuye a organizar el sistema de conocimientos como base y lo enriquece con aplicaciones y ejercicios, permite realizar un proceso de enseñanza por parte del docente de una manera más efectiva.

En Colombia, vale la pena resaltar el trabajo realizado por Jorge A. Villalobos, profesor asociado del Departamento de Ingeniería de Sistemas de la Universidad de los Andes, en su obra “Introducción a las Estructuras de Datos” se basa en un modelo pedagógico novedoso y llamativo para la enseñanza de la asignatura. Villalobos se enfoca en una enseñanza basada en problemas con ayudas de

diversas herramientas tecnológicas que giran alrededor de la asignatura y que hacen parte de un programa de ingeniería de sistemas.

A raíz de la preocupación existente en la comunidad académica de la Universidad de los Andes sobre el bajo nivel de motivación de los estudiantes para encarar la asignatura *Estructuras de datos y programación*, Villalobos ha liderado desde hace varios años un proyecto denominado “Cupi2”. Inicialmente, la propuesta estuvo enfocada a utilizar algunas herramientas como apoyo al proceso de enseñanza, buscar nuevos enfoques como objetivos de la asignatura, entre otros. Pero ninguno de ellos ha sido solución integral a la problemática. Las soluciones simples (cambiar de libro, utilizar otro lenguaje de programación, cambiar el orden de los temas, etc) fueron intentadas en infinidad de ocasiones sin lograr mejoras efectivas.

El proyecto Cupi2, se inició en el año 2004 con el propósito de buscar nuevas maneras de enseñar la asignatura *Estructuras de datos y programación* haciendo énfasis en la motivación de los estudiantes y teniendo en cuenta todos los aspectos involucrados en el problema. Cupi2 se planteó desde un comienzo como un proyecto multidisciplinario de investigación y ha pasado por siete etapas principales:

- i. Diagnóstico
- ii. Definición del marco conceptual, las hipótesis, las variables, los indicadores y los instrumentos.
- iii. Diseño de la propuesta
- iv. Validación de las hipótesis por medio de pilotos controlados
- v. Construcción de los materiales y herramientas de apoyo
- vi. Proceso de formación y acompañamiento de profesores
- vii. Creación de una comunidad de aprendizaje y difusión de resultados.

Para lograr una solución integral al problema planteado, se considera que se deben tener en cuenta los siguientes aspectos:

- Un modelo pedagógico adaptado al perfil del estudiante actual. Es necesario volver a pensar a fondo la manera de enseñar la asignatura *Estructuras de datos y programación*.
- Un conjunto de materiales y herramientas de soporte al proceso de enseñanza impartido por el docente.
- Un esquema de formación de profesores, que les dé una comprensión global de la problemática y que les permita participar de manera efectiva en el proceso de aprendizaje de los estudiantes.
- Un conjunto de herramientas de acompañamiento para los profesores, que faciliten la planeación de las clases, la comunicación con los estudiantes, el registro de los resultados obtenidos, el reporte de las actividades y logros, seguimiento, etc.

El proyecto en mención es actualmente eje de referencia para el cambio curricular de muchos programas de Ingeniería de Sistemas en Colombia.

Villalobos, admite que existen diversas investigaciones sobre el proceso de enseñanza de la asignatura *Estructuras de datos y programación* y muchas de ellas tratan sobre el uso de herramientas y estrategias de soporte para afrontar este problema; considera además, que ninguna de ellas ha presentado resultados suficientes en cuanto a la articulación de objetivos pedagógicos de los cursos y experiencias positivas por parte de estudiantes y profesores.

Por su misma naturaleza, un curso *Estructuras de datos y programación* está orientado a la generación de habilidades, más que a la simple exposición de un conjunto de conceptos. Si uno piensa en la cantidad de teoría que hay detrás de la programación se sorprende del poco número de conceptos distintos y de su simplicidad. Este hecho en sí mismo no es un problema, a menos que el énfasis del curso se haga en la exposición de las definiciones y todo lo demás se vuelva accesorio.

En un curso típico de *Estructuras de datos y programación*, el profesor, después de presentar en clase los conceptos (o pedir que los estudiantes lean de un libro), responde las preguntas que puedan surgir de dichas definiciones, y luego comienza a hacer ejercicios en el tablero. Dicha manera de enseñar la asignatura se basa en la esperanza de que el estudiante sea capaz de detectar patrones en los problemas planteados y los logre asociar con las técnicas de enseñanza que utiliza el profesor cuando desarrolla un ejemplo en el tablero, y que, luego, el estudiante sea capaz de hacer la generalización adecuada en su cabeza, para poder aplicar esa asociación patrón-técnica para resolver otros problemas.

La única técnica de programación que se enseña en los cursos de *Estructuras de datos y programación* es “dividir y conquistar” y se hace a un nivel de informalidad y vaguedad que no pasa de ser una buena intención.

Este enfoque, que se puede llamar aprendizaje por imitación, se basa en la hipótesis de que el estudiante es capaz de generar las habilidades necesarias tratando de imitar lo que el profesor hace. Esto tiene muchos problemas implícitos, y explica una buena parte de las dificultades de enseñar la asignatura *Estructuras de datos y programación*: Se da la sensación de que el “cómo hacer las cosas” no es enseñable, y que es algo que depende de la inspiración y del ingenio del programador. Esto genera ansiedad en los estudiantes, puesto que al llegar a un examen no saben si aquello que mágicamente se generó en su cabeza (y que no pueden hacerlo explícito para validarlo) es suficiente para resolver el problema que les van a plantear.

La calidad del curso es muy dependiente de la capacidad del profesor para inducir a los estudiantes a aprender por imitación. Claramente algunos profesores son

más exitosos que otros en esta labor, pero no son capaces de explicar dicho éxito en términos que puedan ser materializados en una guía para profesores. Desde este punto de vista, se tiene la impresión de que la habilidad de enseñar con éxito programación tampoco es enseñable.

En la búsqueda de propuestas actuales frente al tema de innovar la enseñanza de la asignatura *Estructuras de datos y programación*, algunos autores concuerdan con la necesidad de integrar la generación de habilidades más que la transmisión de conocimientos planos, especialmente en el nivel de educación superior. La educación en ingeniería es un campo con fuertes tradiciones, pero diversos ejemplos de transición de currículos que utilizan estrategias pedagógicas que involucran a los estudiantes como agentes de generación de su propio conocimiento, han evidenciado la necesidad de cambio en las didácticas utilizadas y la fuente de posibles soluciones frente a dificultades en la enseñanza de ingeniería.

De otro lado, como proyecto de investigación, la metodología utilizada en Cupí2 se encuentra soportada por cuatro componentes (Aprendizaje activo, Aprendizaje basado en problemas y proyectos, Aprendizaje incremental, Aprendizaje basado en ejemplos) que permiten la construcción de una solución balanceada frente al problema de la enseñanza de la asignatura *Estructuras de datos y programación*. Cada uno de estos componentes ha sido validado en experiencias mundiales de enseñanza de diferentes campos de la ciencia, teniendo importante impacto en la educación en ingeniería. Tal como se ha explicado anteriormente, la existencia de una variedad de dificultades en la enseñanza de la asignatura *Estructuras de datos y programación* ha promovido el uso de novedosas estrategias pedagógicas.

La propuesta de CUpí2, es la integración de una comunidad de enseñanza que permita el intercambio de estrategias didácticas, centralice el conocimiento grupal y ofrezca soporte a las necesidades de comunicación que esto conlleva. Actualmente la comunidad Cupí2 integra más de 25 universidades en Colombia y algunos institutos en el resto del mundo, en donde la enseñanza de la asignatura *Estructuras de datos y programación* se realiza a través de la metodología Cupí2.

Los resultados que se han obtenido han sido satisfactorios, tal como lo reflejaron las encuestas realizadas a los estudiantes en donde se evaluaron once preguntas con respecto a las estrategias didácticas utilizadas por el docente, seis preguntas con referencia al desarrollo de la asignatura. Además, se tuvo en cuenta el promedio de notas obtenida por los estudiantes y el porcentaje de estudiantes que pierde el curso. Concluye el autor manifestando que este nuevo tipo enfoque que le han dado al desarrollo de la asignatura ha sido satisfactorio y un gran aporte no solo a la comunidad académica de la Universidad de los Andes sino también a nivel nacional e internacional.

15. MARCO CONCEPTUAL

15.1. Software educativo

Esta definición engloba todos los programas que han estado elaborados con fin didáctico, desde los tradicionales programas basados en los modelos conductistas de la enseñanza, los programas de Enseñanza Asistida por Ordenador (EAO), hasta los aun programas experimentales de Enseñanza Inteligente Asistida por Ordenador (EIAO), que, utilizando técnicas propias del campo de los Sistemas Expertos y de la Inteligencia Artificial en general, pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos.

No obstante según esta definición, más basada en un criterio de finalidad que de funcionalidad, se excluyen del software educativo todos los programas de uso general en el mundo empresarial que también se utilizan en los centros educativos con funciones didácticas o instrumentales como por ejemplo: procesadores de textos, gestores de bases de datos, hojas de cálculo, editores gráficos. Estos programas, aunque puedan desarrollar una función didáctica, no han estado elaborados específicamente con esta finalidad.

Los programas educativos pueden tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo, diagramas de flujo, estructuras de datos, lenguajes de programación, entre otros), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- i. Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.
- ii. Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- iii. Son interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- iv. Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.

- v. Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

15.2. 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 cuatro tipos de algoritmos:

- i. Lineal, cuando la secuencia de las actividades es única.
- ii. Ramificado, cuando están predeterminadas posibles secuencias según las respuestas de los alumnos.
- iii. 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. Este entorno puede ser: Estático, si el usuario sólo puede consultar (y en algunos casos aumentar o disminuir) la información que proporciona el entorno, pero no puede modificar su estructura. Dinámico, si el usuario, además de consultar la información, también puede modificar el estado de los elementos que configuran el entorno. Programable, si a partir de una serie de elementos el usuario puede construir diversos entornos.
- iv. Instrumental, si ofrece a los usuarios diversos instrumentos para realizar determinados trabajos.

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

15.3. Estructuras de datos

En programación, una estructura de datos es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación. Un dato elemental es la mínima información que se tiene en un sistema.

Una estructura de datos define la organización e interrelación de estos y un conjunto de operaciones que se pueden realizar sobre ellos. Las operaciones básicas son:

- Adicionar un nuevo valor a la estructura.
- Borrar un valor de la estructura

- Búsqueda, encontrar un determinado valor en la estructura para realizar una operación con este valor, en forma secuencial o binario (siempre y cuando los datos estén ordenados).

Otras operaciones que se pueden realizar:

- Ordenamiento, de los elementos pertenecientes a la estructura.
- Apareo, dadas dos estructuras originar una nueva ordenada y que contenga las apareadas.

Cada estructura ofrece ventajas y desventajas en relación a la simplicidad y eficiencia para la realización de cada operación. De esta forma, la elección de la estructura de datos apropiada para cada problema depende de factores como la frecuencia y el orden en que se realiza cada operación sobre los datos.

El estudio de las estructuras de datos constituye una de las principales actividades para llegar al desarrollo de grandes sistemas de información. Cada estructura de datos tiene sus costes y sus beneficios. Los programadores y diseñadores necesitan una comprensión rigurosa y completa de cómo evaluar los costes y beneficios para adaptarse a los nuevos retos que afronta la construcción de la aplicación. Los datos estructurados siguen a las necesidades. Los estudiantes deben aprender a evaluar primero las necesidades de la aplicación, a continuación, encontrar una estructura de datos en correspondencia con sus funcionalidades.

15.4. Herramientas didácticas

Las herramientas didácticas, también denominadas materiales didácticos o medios didácticos, pueden ser cualquier tipo de dispositivo diseñado y elaborado con la intención de facilitar un proceso de enseñanza y aprendizaje.

Según Cabero (2001), existe una diversidad de términos para definir el concepto de herramientas didácticas o materiales didácticos, tales como los que se presentan a continuación:

1. Medio (Saettler, 1991; Zabalza, 1994) 2. Medios auxiliares (Gartner, 1970; Spencer-Giudice, 1964) 3. Recursos didácticos (Mattos, 1973) 4. Medio audiovisual (Mallas, 1977 y 1979) 5. Materiales (Gimeno, 1991; Ogalde y Bardavid, 1991)

“Esta diversidad de términos conduce a un problema de indefinición del concepto, así como también al de la amplitud con que éstos son considerados”. (Cabero, 2001:290) Es decir, cada autor da un significado específico al concepto, lo que conduce a tener un panorama mucho más amplio en cuanto a herramientas o materiales didácticos se refiere.

La terminología utilizada para nombrar a las herramientas o materiales didácticos da lugar a considerarlos, según Cebrián (Citado en Cabero, 2001:290) como “Todos los objetos, equipos y aparatos tecnológicos, espacios y lugares de interés cultural, programas o itinerarios medioambientales, materiales educativos que, en unos casos utilizan diferentes formas de representación simbólica, y en otros, son referentes directos de la realidad. Estando siempre sujetos al análisis de los contextos y principios didácticos o introducidos en un programa de enseñanza, favorecen la reconstrucción del conocimiento y de los significados culturales del currículum”.

Son empleados por los docentes e instructores en la planeación didáctica de sus cursos, como vehículos y soportes para la transmisión de mensajes educativos. Los contenidos de la materia son presentados a los alumnos en diferentes formatos, en forma atractiva en ciertos momentos clave de la instrucción. Estos materiales didácticos (impresos, audiovisuales, digitales, multimedia) se diseñan siempre tomando en cuenta el público al que van dirigidos, y tienen fundamentos psicológicos, pedagógicos y comunicacionales.

Según se usen, pueden tener diversas funciones:

- Proporcionar información.
- Guiar los aprendizajes.
- Ejercitar habilidades.
- Motivar.
- Evaluar.
- Proporcionar simulaciones.
- Proporcionar entornos para la expresión y creación.

Una clasificación de las herramientas o materiales didácticos que conviene indistintamente a cualquier disciplina es la siguiente (Nérici, p.284):

- i. Material permanente de trabajo: Tales como el tablero y los elementos para escribir en él, video-proyectores, cuadernos, reglas, compases, computadores personales.
- ii. Material informativo: Mapas, libros, diccionarios, enciclopedias, revistas, periódicos, etc.
- iii. Material ilustrativo audiovisual: Posters, videos, discos, etc.
- iv. Material experimental: Aparatos y materiales variados, que se presten para la realización de pruebas o experimentos que deriven en aprendizajes.

15.5. Mapas conceptuales

Es una técnica usada para la representación gráfica del conocimiento. Como se ve, un mapa conceptual es una red de conceptos. En la red, los nodos representan los conceptos, y los enlaces las relaciones entre los conceptos

Los mapas conceptuales fueron desarrollados en la década de 1960 por Joseph D. Novak, profesor emérito en la Universidad de Cornell, basándose en la teoría de David Ausubel del Aprendizaje Significativo. Según Ausubel "el factor más importante en el aprendizaje es lo que el sujeto ya conoce". Por lo tanto, el aprendizaje significativo ocurre cuando una persona consciente y explícitamente vincula esos nuevos conceptos a otros que ya posee. Cuando se produce ese aprendizaje significativo, se produce una serie de cambios en nuestra estructura cognitiva, modificando los conceptos existentes, y formando nuevos enlaces entre ellos. Esto es porque dicho aprendizaje dura más y es mejor que la simple memorización: los nuevos conceptos tardan más tiempo en olvidarse, y se aplican más fácilmente en la resolución de problemas.

Según Novak, los nuevos conceptos son adquiridos por descubrimiento, que es la forma en que los niños adquieren sus primeros conceptos y lenguaje, o por aprendizaje receptivo, que es la forma en que aprenden los niños en la escuela y los adultos. El problema de la mayor parte del aprendizaje receptivo en las escuelas, es que los estudiantes memorizan definiciones de conceptos, o algoritmos para resolver sus problemas, pero fallan en adquirir el significado de los conceptos en las definiciones o fórmulas.

Cuando se realiza un mapa conceptual, se obliga al estudiante a relacionarse, a jugar con los conceptos, a que se empape con el contenido. No es una simple memorización; se debe prestar atención a la relación entre los conceptos. Es un proceso activo.

El mapa conceptual puede tener varios propósitos:

- Generar ideas (*brain storming*, etc.);
- Diseñar una estructura compleja (textos largos, hipermedia, páginas web grandes, etc.);
- Comunicar ideas complejas;
- Contribuir al aprendizaje integrando explícitamente conocimientos nuevos y antiguos;
- Evaluar la comprensión o diagnosticar la incompreensión;
- Explorar el conocimiento previo y los errores de concepto;
- Fomentar el aprendizaje significativo para mejorar el éxito de los estudiantes;
- Medir la comprensión de conceptos.
- Generar conceptos o ideas sobre algo o un tema.

15.6. Enseñanza

La **enseñanza** es la **acción y efecto de enseñar** (instruir, adoctrinar y amaestrar con reglas o preceptos). Se trata del sistema y **método de dar instrucción**, formado por el conjunto de conocimientos, principios e ideas que se enseñan a alguien.

La enseñanza implica la interacción de tres elementos: el profesor, docente o maestro; el alumno o estudiante; y el objeto de conocimiento. La tradición enciclopedista supone que el profesor es la fuente del conocimiento y el alumno, un simple receptor ilimitado del mismo. Bajo esta concepción, el proceso de enseñanza es la transmisión de conocimientos del docente hacia el estudiante, a través de diversos medios y técnicas.

Sin embargo, para las corrientes actuales como la cognitiva, el docente es un facilitador del conocimiento, actúa como nexo entre éste y el estudiante por medio de un proceso de interacción. Por lo tanto, el alumno se compromete con su aprendizaje y toma la iniciativa en la búsqueda del saber.

La enseñanza como transmisión de conocimientos se basa en la percepción, principalmente a través de la oratoria y la escritura. La exposición del docente, el apoyo en textos y las técnicas de participación y debate entre los estudiantes son algunas de las formas en que se concreta el proceso de enseñanza.

Con el avance científico, la enseñanza ha incorporado las nuevas tecnologías y hace uso de otros canales para transmitir el conocimiento, como el video e internet.

16. DISEÑO METODOLÓGICO

17. TIPO DE INVESTIGACIÓN

Dada las características del problema de investigación y de los objetivos planteados, esta investigación es de tipo cualitativa y su enfoque está centrado en un estudio explicativo - correlacional.

Cualitativo puesto que se refiere a las herramientas didácticas utilizadas por el docente para la enseñanza de la asignatura *Estructuras de datos*, y es necesario describir la experiencia del docente en el uso de esas herramientas así como también generar una teoría a partir de los datos recogidos.

Explicativo porque se hace necesario estudiar las técnicas de enseñanza utilizadas por el docente eliminando factores –la relación enseñanza-aprendizaje-

que pueden ser causa de un resultado particular y probar solamente los que se quieren medir de manera directa. En este caso, las herramientas didácticas utilizadas por el docente en el proceso de enseñanza.

16.1. Estudio Co-relacional

Una correlación es una medida del grado en que dos variables se encuentran relacionadas.

El estudio correlacional se ocupa de determinar la variación en unos aspectos en relación con otros. Este estudio es el indicado para organizar las relaciones estadísticas entre las características y la concentración de las causas del fenómeno estudiado. En una situación creada, explica por qué se presenta, en qué grado dos o más de sus variables están relacionadas y en qué circunstancias se produce este estado. También, un estudio correlacional puede intentar determinar si individuos con una puntuación alta en una variable también tiene puntuación alta en una segunda variable y si individuos con una baja puntuación en una variable también tienen baja puntuación en la segunda. Estos resultados indican una relación positiva.

Con respecto al estudio correlacional, el proceso de definición del problema se inicia al considerar un amplio entendimiento del marco de referencia, el desenvolvimiento en el área del fenómeno estudiado, el conocimiento de las variables pertinentes y de la forma en que se afectan mutuamente.

Siguiendo a Roberto Hernández Sampieri nos dice que los estudios correlacionales son aquellos que “tienen como propósito medir el grado de relación que exista entre dos o más conceptos variables”. Los estudios correlacionales abordan el comportamiento de las variables en cuestión. Si alguno de ellos se modifica es previsible que la otra lo haga también.

Para el caso particular de la investigación la correlación se da al determinar el grado y el tipo de fuerza de la relación entre dos variables, en este caso las variables hacen referencia a las herramientas didácticas utilizadas por los docentes para la enseñanza de las *Estructuras de datos*.

18. POBLACION

La Universidad el Bosque cuenta con dos docentes encargados de la enseñanza de la asignatura *Estructuras de datos* en el programa de Ingeniería de Sistemas. Cada uno de ellos, posee dos grupos de estudiantes uno en la jornada Diurna y otro en la Jornada nocturna.

El docente “A”, es profesional en Ingeniería de Sistemas, especialista en Construcción de Software. Es la primera vez que labora como docente a nivel universitario, en anteriores ocasiones tuvo la oportunidad de laborar en instituciones técnicas. A nivel profesional, posee una experiencia de más de diez

años en el sector empresarial. En la actualidad, trabaja para una empresa del sector privado en el área de Ingeniería.

EL docente “B”, es profesional en Ingeniería de Sistemas, especialista en Seguridad de la información. Actualmente se encuentra realizando estudios de Maestría en Tecnologías de la Información y es candidato a especialista en Docencia Universitaria. En la rama educativa, posee más de diez años de experiencia y a nivel profesional, ha estado vinculado al sector empresarial por más de diez años. Actualmente se desempeña como consultor / asesor de empresas.

19. INSTRUMENTOS DE RECOLECCION DE INFORMACION

La recolección de información se realizará a través de la implementación de los siguientes instrumentos de recolección de información:

- i. Registros fílmicos: se hizo uso del video, que por su propia naturaleza, resulta un medio apropiado que permite recoger las ideas y experiencia del proceso de enseñanza al interior del aula. Además, nos permite grabar las distintas situaciones que tengan que ver con la enseñanza y aporta la posibilidad de observar algunos aspectos que pueden pasar desapercibidos.
- ii. Transcripciones: Esta actividad consistió en escribir literalmente en un documento todo lo observado y escuchado en cada uno de los registros fílmicos.

20. ANALISIS DE RESULTADOS

A continuación, se presenta los resultados obtenidos de las transcripciones realizadas de los registros fílmicos. En total, se realizaron ocho (8) sesiones tanto para el docente A como el docente B en la jornada diurna y nocturna. Los resultados arrojan información valiosa que describe el desempeño docente en la clase de Estructuras de Datos. Mediante los registros fílmicos, podemos identificar las diferentes estrategias didácticas utilizadas por los docentes.

Seguidamente, se presentan los resultados categorizados de la siguiente manera:

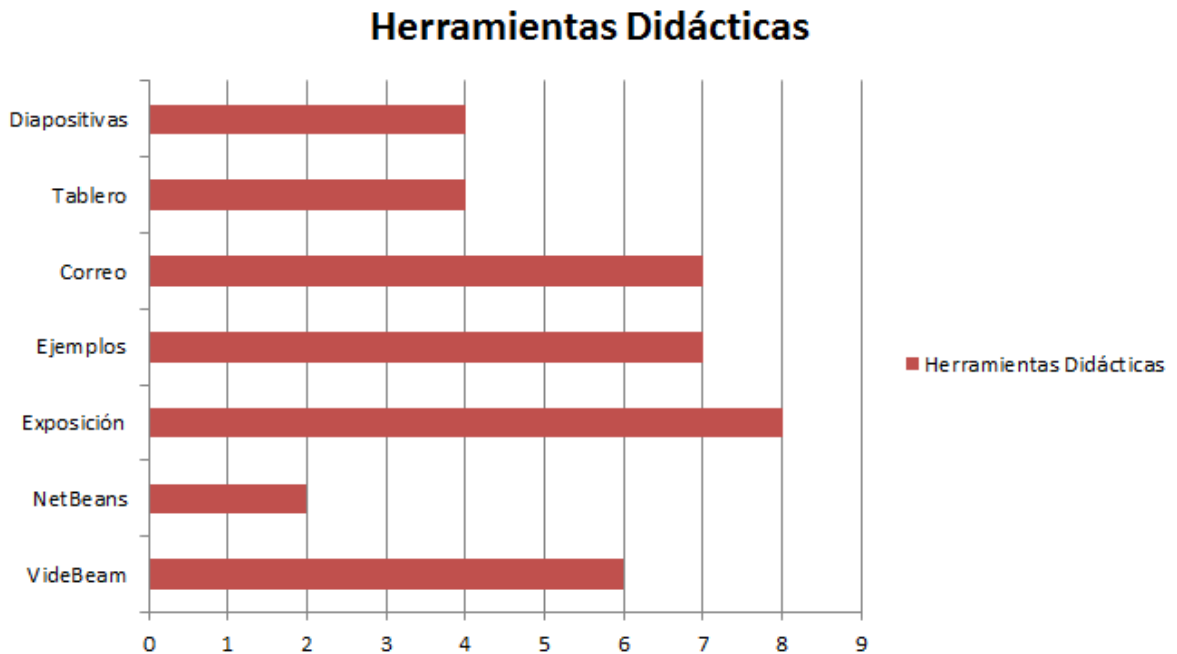
Categoría 1: Herramientas didácticas utilizadas por el docente “A”.

Tabla 5: Herramientas didácticas utilizadas por el docente “A”.

Sesión	Tema	Jornada	Herramientas didácticas
1	Manejo de Archivos	Diurna	VideoBeam, Software NetBeans, Exposición docente, solución de ejemplos modelos, correo electrónico.
2	Memoria dinámica	Diurna	Tablero, exposición docente, solución de ejemplos modelos, correo electrónico.
3	Fundamentos de Estructuras dinámicas	Nocturna	VideoBeam, exposición docente, solución de ejemplos modelos, correo electrónico, diapositivas.
4	Listas Enlazadas	Nocturna	Tablero, exposición docente, solución de ejemplos modelos, correo electrónico.
5	Stacks	Diurna	VideoBeam, Software NetBeans, Exposición docente, solución de ejemplos modelos, correo electrónico.
6	Tablas de Hash	Nocturna	VideoBeam, Diapositivas, exposición docente, tablero, solución de ejemplos modelos, correo electrónico.
7	Tablas de Hash (Cont)	Nocturna	VideoBeam, Diapositivas, exposición docente, tablero, solución de ejemplos modelos, correo electrónico.
8	Exposiciones diversas	Diurna	VideoBeam, Diapositivas

Elaborada por Wilson Mauro Rojas Reales

Figura 2. Herramientas Didácticas docente A



Elaborado por Wilson Mauro Rojas Reales

Se evidencia claramente que el docente “A” utiliza como recurso didáctico frecuente la *exposición* y generalmente hace uso de un componente tecnológico (VideoBeam) como apoyo a su labor docente. Al analizar cada uno de los registros fílmicos (ver anexo A – Transcripciones docente A); durante las sesiones filmadas, se pudo evidenciar que el docente después de realizar su labor expositora del tema, desarrolla algunos ejemplos modelos y posteriormente el estudiante debe solucionar un ejercicio de mayor complejidad. El resultado, debe ser enviado al correo electrónico del docente. En dos de las sesiones, el docente utilizó un Software denominado **NetBeans** (*NetBeans* es una herramienta de software muy utilizada para el desarrollo de Software en diversos lenguajes de programación. Es una herramienta para crear programas en el lenguaje Java el cual es muy útil para el estudio de las diversas Estructuras de Datos) para realizar explicaciones sobre ejemplos de mediana complejidad.

Es importante resaltar que en cada una de las sesiones, se evidencia que el docente hace uso de diversas herramientas didácticas, combinando las denominadas herramientas tradicionales con las tecnológicas. Por otro lado, durante todas las sesiones se pudo observar que la participación del estudiante es poca, siendo el máximo protagonista el docente. Los ejemplos explicados por parte del docente no incitan a una discusión puesto que son ejemplos encontrados en los textos bibliográficos referenciados para la asignatura Estructuras de Datos.

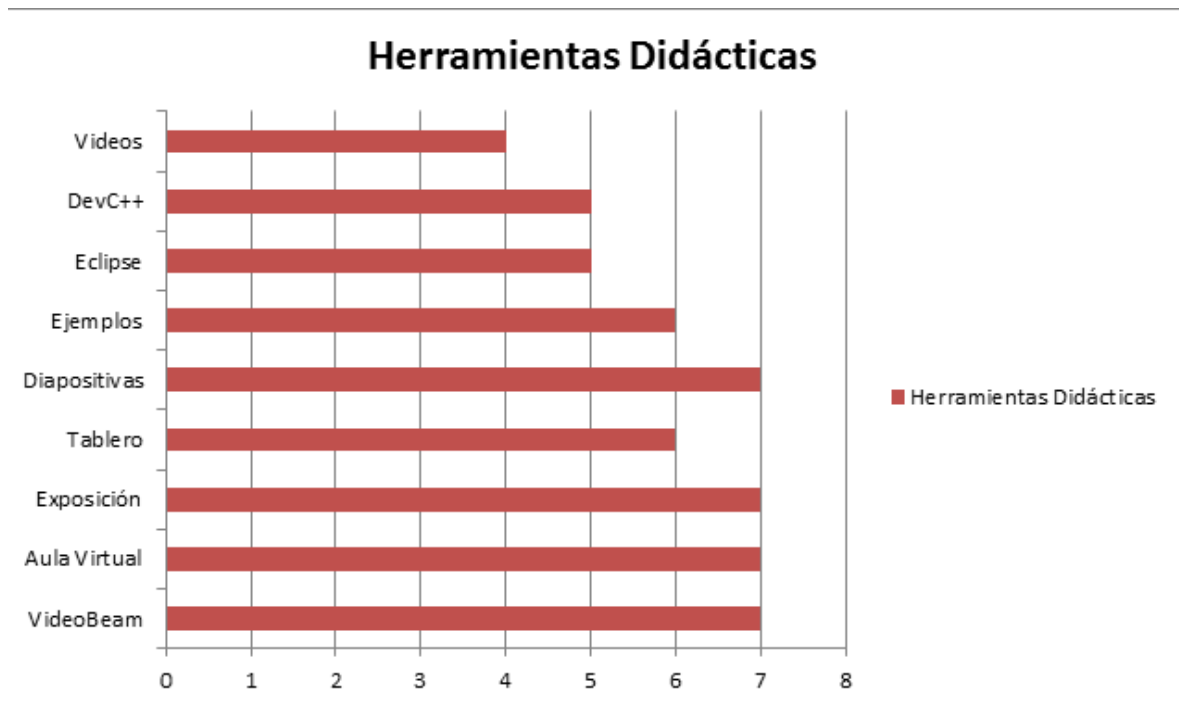
El tema denominado **Tablas de Hash** resultó ser muy complejo para los estudiantes; tanto es así, que solo un grupo de tres estudiantes pudo resolver el problema propuesto por el docente y la clase no pudo ser terminada en una sola sesión. En la última sesión; los protagonistas fueron los estudiantes, quienes tuvieron a cargo el desarrollo de diversas exposiciones para dar a conocer el uso de las aplicaciones basadas en las diversas *Estructuras de datos*, por esta razón el docente no hizo uso del correo electrónico recurso didáctico muy utilizado por el docente para recibir la solución a los ejercicios propuestos en cada una de las sesiones.

Categoría 2: Herramientas didácticas utilizadas por el docente “B”.

Tabla 6: Herramientas didácticas utilizadas por el docente “B”.

Sesión	Tema	Jornada	Herramientas didácticas
1	Manejo de Archivos	Diurna	VideoBeam, Exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos, Software Eclipse, DevC++.
2	Memoria dinámica	Diurna	VideoBeam, Exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos.
3	Fundamentos de Estructuras dinámicas	Nocturna	VideoBeam, videos, exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos.
4	Listas Enlazadas	Nocturna	VideoBeam, videos, exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos, Software Eclipse, DevC++.
5	Stacks	Diurna	VideoBeam, videos, exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos, Software Eclipse, DevC++.
6	Tablas de Hash	Nocturna	VideoBeam, videos, exposición docente, tablero, Aula Virtual, diapositivas, solución de ejemplos modelos, Software Eclipse, DevC++.
7	Exposiciones diversas	Nocturna	VideoBeam, Diapositivas, Aula Virtual

Figura 3. Herramientas didácticas docente B



Elaborado Wilson Mauro Rojas Reales

Al observar los resultados obtenidos para el docente “B”, se evidencia que la combinación de los recursos: VideoBeam, Aula Virtual, Exposición y Diapositivas son los más utilizados en cada una de las sesiones. Al analizar cada una de las filmaciones se observa también que el docente propicia la discusión en clases y eso es producto de los interrogantes planteados durante la exposición y explicación de las diapositivas utilizadas en cada tema.

El docente “B”, utiliza el tablero para explicar gráficamente interrogantes propuestos por los estudiantes y además, utiliza no solo una herramienta de Software sino dos: Eclipse y DevC++. Estas herramientas, permiten desarrollar programas bajo los lenguajes de programación Java y C++ respectivamente. Históricamente, uno de los mejores lenguajes de programación utilizados para representar las diversas *Estructuras de Datos* es “C” y en la actualidad se utiliza “C++”; el docente propone a los estudiantes el uso de las herramientas Eclipse y DevC++ para que ellos observen las diferencias al representar las estructuras de datos tanto en la una como en la otra.

El docente “B” también desarrolla y resuelve ejemplos modelos haciendo uso de las herramientas de software y también recurre al tablero para explicar con mayores detalles el ejemplo propuesto. El docente; finalizado el estudio de un tema, propone una serie de ejercicios que deben ser resueltos por los estudiantes

en un plazo de ocho días y el informe a presentar con las solución de tales ejercicios debe ser enviado al docente por medio de la plataforma virtual para la asignatura. Para ello, cada estudiante debe ingresar al portal web de la Universidad (www.unbosque.edu.co) e ingresar por el link correspondiente (Aulas Virtuales).

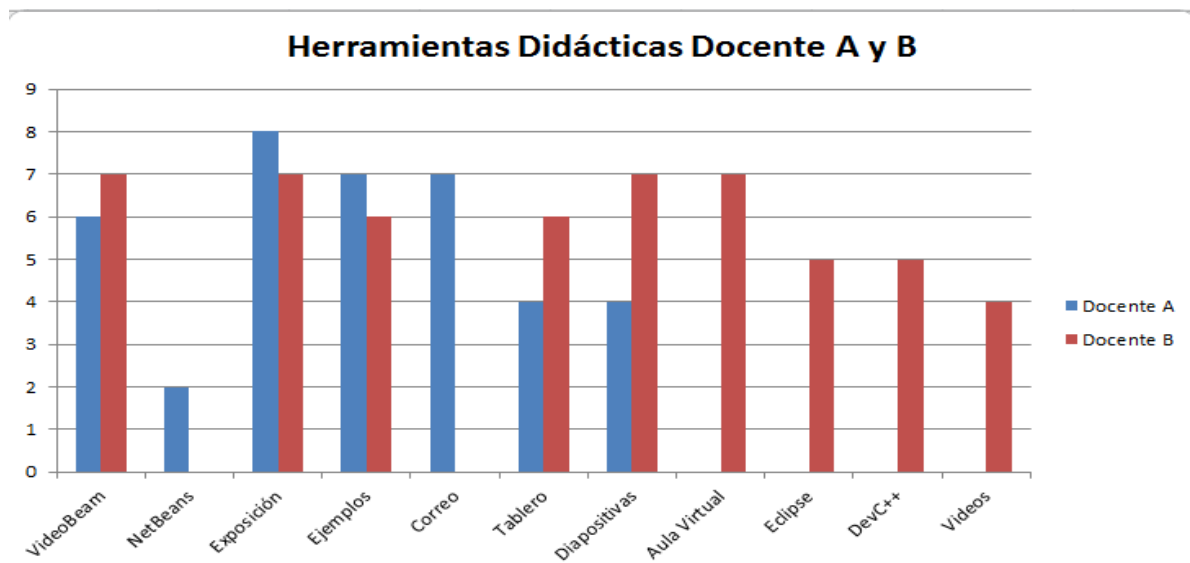
En la última sesión de clases, las exposiciones estuvieron a cargo de los estudiantes pero se trató de la solución de ejercicios en clases. El docente, hizo de entrega de diversos ejercicios y a nivel grupal fueron solucionados. Cualquier grupo al resolver alguno de los ejercicios propuestos, presentaba su solución a los demás haciendo uso del VideoBeam y el tablero de ser necesario para ampliar la explicación.

Otro de los recursos utilizados por el docente “B” es el Video. En algunas sesiones, el docente presentó un video alusivo al tema a desarrollar en ocasiones antes de realizar la presentación de las diapositivas y en otras al finalizar la exposición del tema. Los videos presentados se utilizaron como complemento a la explicación docente.

Aún más, el docente “B” utilizó permanentemente el espacio virtual de la asignatura en donde coloca a disposición de los estudiantes lecturas complementarias de cada tema tratado. Las transcripciones del docente B, se evidencian en el Anexo B.

Categoría 3: Herramientas didácticas utilizadas por el profesor A y B.

Figura 4. Herramientas didácticas docentes A y B



Elaborado Wilson Mauro Rojas Reales

Al realizar un análisis de las herramientas didácticas utilizadas por los docentes que imparten la asignatura Estructuras de Datos I en el programa de Ingeniería de Sistemas en la Universidad el Bosque en la ciudad de Bogotá, se puede notar claramente que ambos utilizan con mayor frecuencia el VideoBeam, la solución de ejemplos y la exposición docente. El docente “B” utiliza con mayor frecuencia el tablero y las diapositivas. Aunque se persigue el mismo resultado, el docente “A” hace uso del correo electrónico y el docente “B” recurre al Aula Virtual. En ambos casos, lo que se busca es un medio para que los estudiantes puedan remitir las soluciones de los ejercicios propuestos por los docentes. En lo referente al uso de herramientas de Software, el docente “B” utiliza dos tipos de ellas mientras que el docente “A” solo una. Y un aspecto para destacar, es el uso del Video por el docente “B”. En general; con base en las filmaciones realizadas, el docente “B” se apoya en muchas más herramientas didácticas que el docente “A”.

Categoría 4: Herramientas didácticas utilizadas por el profesor A y B por Temas.

Figura 5. Herramientas didácticas utilizadas por el profesor A y B por Temas.

Tema - Docentes	VideoBeam		Exp.Docente		Tablero		Aula Virtual		NetBeans		Sol.Ejemplos		Correo		Diapositivas		Eclipse		DevC++		Videos	
	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B
Manejo de Archivos	X	X	X	X	X	X		X	X		X	X	X			X		X		X		
Memoria Dinámica		X	X	X	X	X		X			X	X	X			X						
Fundamentos ED	X	X	X	X		X		X			X	X	X		X	X						X
Listas Enlazadas		X	X	X	X	X		X			X	X	X			X		X		X		X
Stacks	X	X	X	X		X		X	X		X	X	X			X		X		X		X
Tablas de Hash	X	X	X	X	X	X		X			X	X	X		X	X		X		X		X
Exposiciones	X	X	X					X							X	X		X		X		

Elaborado Wilson Mauro Rojas Reales

En el gráfico anterior, se puede observar algunas similitudes y al mismo tiempo algunas diferencias en el uso de las herramientas didácticas por parte de los docentes “A” y “B” por cada uno de los temas tratados. Las dos diferencias marcadas a lo largo de todas las sesiones filmadas, fue el mecanismo utilizado por los estudiantes para el envío de las soluciones a los ejercicios propuestos por los docentes (el docente “A” siempre utilizó el correo electrónico y el docente “B” utilizó el Aula Virtual de la asignatura) y la herramienta de Software utilizada en ocasiones para la presentación de algunos ejemplos o ejercicios. Mientras el docente “A” siempre utilizó “NetBeans”, el docente “B” utilizó “Eclipse” y “DevC++”. A continuación, un análisis por cada tema:

- **Manejo de Archivos:** Las herramientas didácticas utilizadas fueron similares, con la diferencia del uso del Software para la presentación de algunos ejemplos y el mecanismo para recibir de parte de los estudiantes los informes referentes a la solución de algunos ejercicios propuestos por el docente.

- **Memoria Dinámica:** Para el docente “A”, fue necesario utilizar el tablero y explicar el tema recurriendo a componentes gráficos y dar soluciones a problemas que se presentan por la mala gestión de la administración de la Memoria del computador. El docente “B” utilizó el VideoBeam para presentar una serie de diapositivas y en ellas explicar todo el proceso de gestión de memoria al interior del computador y sus problemas derivados.
- **Fundamentos de Estructuras de Datos:** Las herramientas didácticas utilizadas por ambos docentes fueron similares. La única diferencia radicó en el video utilizado por el docente “B”. El video fue un presentado como una introducción al tema pretendiendo involucrar a los estudiantes en la importancia de conocer las diversas estructuras de datos. El docente “A” en cambio, brindó una explicación verbal cuando se iniciaba en este tema.
- **Listas Enlazadas:** En este tema, el docente “B” utilizó gran cantidad de recursos en comparación con el docente “A”. El docente “A”, de manera simultánea expuso a los estudiantes el tema y utilizando el tablero presentaba gráficos con el fin de complementar y soportar los comentarios expuestos. El docente “B” consideró que este tema era de vital importancia para los estudiantes y dada su complejidad presentó una serie de diapositivas, acompañadas de un video introductorio así como también la explicación de ejemplos mediante las herramientas de Software Eclipse y DevC++.
- **Stacks:** Son más las diferencias encontradas en la forma de enseñar este tema por parte de los docentes, que las similitudes. El docente “B”, nuevamente hace uso de diversos recursos en comparación con el docente “A”. Al ser un tema de mediana complejidad, el docente “B” busca la forma de poder interpretar cada una de las explicaciones mediante variedad de alternativas. Por el contrario, el docente “A” utiliza sus tradicionales recursos didácticos tal como lo hizo en las anteriores sesiones.
- **Tablas de Hash:** Este es uno de los temas de alta complejidad de la asignatura. Así lo entienden los docentes y por ende emplean múltiples recursos. Sin embargo, nuevamente se nota el uso del Video y de herramientas de Software adicionales por parte del docente “B”. El docente “A”, utilizó diapositivas y en ocasiones se apoyó en el uso del tablero para brindar algunas explicaciones complementarias. Cabe resaltar que en este tema, el docente “A” solucionó diversos ejemplos durante la sesión de clases, y por consiguiente fue necesario trabajar en dos sesiones de clases el mismo tema. La metodología seguida por el docente “B” fue la misma de sus sesiones anteriores, uso de diapositivas y video y la solución de ejercicios fue muy escasa.
- **Exposiciones:** El uso de este recurso didáctico estuvo orientado a los estudiantes. El docente “A” introdujo a los estudiantes a indagar sobre el uso y aplicabilidad de las Estructuras de datos en la industria. Se crearon diversos grupos y cada uno de ellos presentó una variedad de ejemplos que incitaron a ciertas discusiones. Por el contrario, el docente “B” fue quien explicó la importancia de utilizar las estructuras de datos en el sector

empresarial y los grupos expositores se limitaron al desarrollo de ejercicios con el propósito de aclarar dudas de manera inmediata con la ayuda del docente “B”.

Vale la pena destacar también, que las clases propiciadas por el docente “A” son pasivas. Es decir, la incitación a la participación del estudiante es escasa e inclusive en algunas ocasiones se pudo detectar a ciertos estudiantes –frente al computador- dedicados a otras labores al interior del aula de clases. El docente “B” en cambio, de manera permanente cuestiona a los estudiantes con interrogantes presentados en diversos escenarios a los cuales se enfrentaran ellos como futuros profesionales.

Categoría 5: Propuesta de uso de herramientas didácticas en la enseñanza de Estructuras de Datos de acuerdo al tema.

Figura 6. Uso De Herramientas Didácticas En La Enseñanza

	VideBeam	Exp.Docente	Tablero	AulaVirtual	Software	Sol.Ejemplos	Foro	Diapositivas	Video
Tema - Docentes	B	B	B	B	B	B	B	B	B
Manejo de Archivos	X	X		X	X	X		X	
Memoria Dinámica	X	X		X		X	X	X	
Fundamentos ED	X	X		X		X		X	
Listas Enlazadas	X	X		X	X	X	X	X	X
Stacks	X	X		X	X	X	X	X	
Tablas de Hash	X	X		X	X	X	X	X	X
Exposiciones									

Amarillo: Baja complejidad, Rojo: Media complejidad, Verde: Alta complejidad

Elaborado Wilson Mauro Rojas Reales

Dada la complejidad de la mayoría de temas que conforman la asignatura, el autor propone el uso de múltiples recursos en las sesiones de clases dependiendo de la complejidad del tema. En el gráfico anterior, se presenta una escala de complejidad basada en colores en donde el color amarillo representa los temas de baja complejidad, el color rojo representa los temas de mediana complejidad y el color verde representa los temas de alta complejidad.

Cabe destacar que el autor propone la “no” utilización del tablero puesto que al tratarse de temas complejos (sin importar el grado) es mejor hacer uso de diapositivas, de elementos de software que ayuden a comprenderlos de una mejor manera. En ocasiones, la terminología también puede causar mayor confusión en los estudiantes y es mejor hacer uso de gráficos llamativos y muy explicativos. Adicionalmente, aparece como una nueva propuesta el uso de “foro”. Con ello se pretende que el estudiante formule sus inquietudes y ocurra un debate con respecto a un tema específico en donde es posible que entre los mismos estudiantes aclaren sus dudas dejando para el docente la aclaración de situaciones más complejas.

Sin lugar a dudas, es necesario hacer uso de alguna herramienta de Software. El autor no pretende colocar y condicionar al uso de alguna en especial pero en definitiva es necesario puesto que ayuda a comprender mejor la explicación docente.

Al no utilizar el tablero, es muy importante recalcar que el diseño y elaboración de las diapositivas debe ser de mucho cuidado teniendo en cuenta la complejidad del tema. Es importante incluir aquellos aspectos fundamentales que ayudan a la explicación docente y hacer uso de mapas mentales que incluidos en las diapositivas ayudaran sin lugar a dudas la comprensión del tema a tratar.

21. CONCLUSIONES

21.1. DESDE LA ASIGNATURA

La asignatura *Estructuras de Datos* es fundamental en la carrera de Ingeniería de Sistemas, por tanto merece importancia buscar estrategias didácticas que permitan a los docentes disponer de variados recursos como mecanismos de enseñanza. Desde hace muchos años, diversos autores se han dado a la tarea de realizar investigaciones en búsqueda de esas herramientas didácticas que ayuden a mitigar esas falencias de enseñanza dada la complejidad de la asignatura, se han utilizado incluso herramientas de software. Se ha avanzado pero no lo suficiente de tal forma que las investigaciones continúan y cada vez el aporte de las investigaciones es trascendental.

Por otro lado, la clasificación de los temas dependiendo del grado de complejidad es un factor interesante que bien vale la pena analizar. Esto es, el hecho de buscar los recursos didácticos apropiados para el tema a tratar dependiendo de su complejidad es fundamental.

Aún más, el tratamiento de ciertos temas complejos también dependerá en gran medida de la estrategia utilizada por el docente para la preparación de los temas. Es decir, utilizar la didáctica apropiada como soporte al tema de estudio.

21.2. DESDE LAS HERRAMIENTAS DIDACTICAS

Con base en la investigación realizada, el autor brinda un aporte más en lo concerniente a las herramientas didácticas, en donde no solo basta con tenerlas a la mano sino también inciden una serie de factores, como:

- El contenido temático de la asignatura.
- Los materiales y recursos con que cuenta el docente
- Mecanismos de seguimiento y mejora continua
- Formación del docente.

Por otro lado; de acuerdo a los resultados encontrados en la investigación, se evidencia que el uso de “videos” como herramienta didáctica es poco utilizada. Sin embargo, es una herramienta que ayuda notoriamente a las explicaciones brindadas por los docentes. Es necesario potencializar el uso de este tipo de herramienta didáctica.

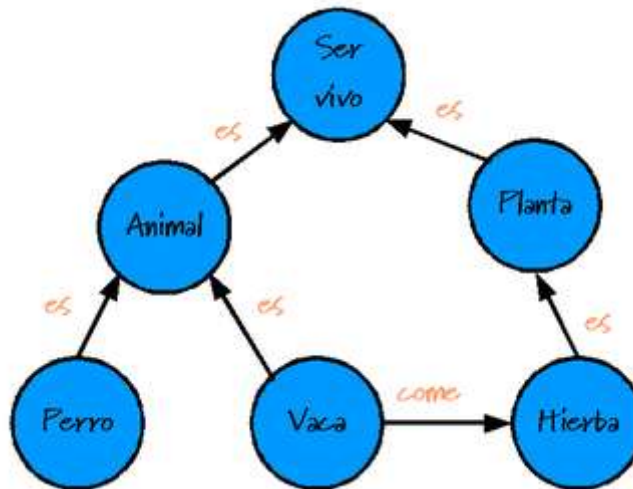
Las herramientas didácticas más utilizadas tanto por el docente “A” como el docente “B” fueron: VideoBeam, solución de ejemplos modelos, explicación docente y el uso de una herramienta de Software. Vale la pena destacar la similitud en el uso de ellas y sin lugar a dudas es fundamental utilizar una herramienta de Software como soporte y/o apoyo a la explicación docente. Es necesario además, que el docente no solo presente soluciones sencillas sino también lo haga con soluciones a ejercicios complejos puesto que en la mayoría de las ocasiones se evidenció que las soluciones de ejemplos modelos por parte del docente fueron sencillas dejando la labor compleja para los estudiantes.

21.3. Herramientas propuestas

El fortalecimiento de la enseñanza de la asignatura Estructuras de Datos, se logra potencializando aquellas herramientas didácticas más utilizadas (VideoBeam, Solución de ejemplos modelos, Explicación docente y software de soporte) – algunas más que otras- e incluyendo otras que como fruto de las investigaciones realizadas por *Yolanda Soler* (docente de la Universidad de Granma-Cuba) y *Jorge Villalobos* (Docente Universidad de los Andes-Colombia) sus propuestas bien valen la pena destacar y tenerlas en cuenta. La primera propone el uso de Mapas Conceptuales y el segundo propone una diversidad de criterios que sin lugar a dudas influye en la enseñanza de la asignatura. Esto es, el Modelo pedagógico, los materiales y herramientas utilizadas y la formación de docentes.

Un mapa conceptual es una técnica usada para la representación gráfica del conocimiento. Es una red de conceptos. En la red, los nodos representan los conceptos y los enlaces las relaciones entre los conceptos.

Figura 7. Red de conceptos



Tomado de: http://es.wikipedia.org/wiki/Mapa_conceptual

Este tipo de herramienta didáctica en conjunto con las utilizadas por los docentes “A” y “B” sumadas todas ellas a los criterios mencionados e investigados por Villalobos ofrecen un nuevo lineamiento que bien vale la pena analizar y poner práctica.

21.4. DESDE LA FORMACION DOCENTE

Si bien es cierta la complejidad de los temas tratados, también es bien cierto que el docente necesita preparación y formación, debe ser una persona recursiva y muy proactiva en aras de buscar diversos mecanismos que le ayuden a soportar sus explicaciones al interior del aula. No solo basta utilizar una herramienta de software, también es necesario conocer el tema a tratar, conocer a sus estudiantes, manejar adecuadamente los recursos didácticos que se tiene a mano, incitar a la participación y a debatir los temas tratados. Como bien lo menciona Villalobos, la formación docente es primordial para la preparación de los temas a sabiendas de la complejidad de los mismos; así mismo, la utilización de las herramientas didácticas acorde con el tema a estudiar y los ejemplos modelos seleccionados de tal forma que todo un conjunto de estrategias fortalecería el proceso de la enseñanza por parte del docente.

La Universidad el Bosque, como Institución comprometida con la educación, exige docentes conscientes de este compromiso. Por ende, el docente no solo debe cumplir con la tarea de enseñar sino comprometerse con el verdadero sentido de la docencia, más allá de la transmisión de conocimientos. Por ello, trabajar para mejorar el proceso de enseñanza de la asignatura Estructuras de Datos en el programa de Ingeniería de Sistemas, debe ser una preocupación permanente del docente Universitario vinculado a la Universidad. Para la Universidad, este tipo de

investigación realizada es simplemente una base sobre la cual se han de construir muchas más no solo en la línea de las Estructuras de Datos sino también en otras disciplinas en aras de buscar cada día las estrategias didácticas eficientes y efectivas acordes al material de enseñanza, sin dejar de lado obviamente la preparación del docente puesto que es pieza fundamental en este rompecabezas. Por otro lado, el aporte que otorga este tipo de estudios para el programa de Ingeniería de Sistemas es un aporte de mucha importancia y marca por supuesto, las pautas que han de tenerse en cuenta para mejorar el proceso de enseñanza de la asignatura Estructuras de Datos, sin descuidar la utilización de las estrategias acordes a cada uno de los temas que componen la asignatura, la formación del docente que como cualquier otra profesión requiere cualidades, calidades y competencias. En ello, el componente disciplinar (nadie enseña lo que no sabe), el pedagógico (como saber científico y como saber reflexivo) y el didáctico (como saber sobre la enseñanza) son fundamentales para el fortalecimiento del proceso de enseñanza de la asignatura de Estructuras de datos en el programa de Ingeniería de Sistemas en la Universidad el Bosque.

22. BIBLIOGRAFIA

CAIRO, Oswaldo y GUARDATI, Silvia. Estructuras de datos. México: Editorial McGraw-Hill, 2006.

GARCIA MOLINA, J y MENARGUEZ TORTOSA, M. Una propuesta para organizar la enseñanza de la orientación a objetos. España, 2002. 6p.

GAYO AVELLO, Daniel, CERNUDA DEL RIO, Agustín, CUEVA LOVELLE, Juan Manuel, DIAZ FONDON, Marián, ALMUDENA GARCIA FUENTE, María Pilar y REDONDO LOPEZ, José Manuel. Reflexiones y experiencias sobre la enseñanza de la programación orientada a objetos como único paradigma. España, 2002. 8p.

GIRALDO ECHEVERRI, Gloria Inés. Nuevo esquema metodológico en la enseñanza de la asignatura programación de computadores. Bogotá, D.E, 2000. 6p.

GUERRERO JULIO, Marlene Lucila y MEDINA CASTILLO, Sergio Arturo. Una estrategia para el apoyo de los procesos de enseñanza de la programación en Ingeniería de Sistemas utilizando objetos virtuales de aprendizaje. Bucaramanga, 2009. 6p.

HERNANDEZ BERLINCHES, R. Problemas de estructuras de datos y algoritmos. Madrid: Editorial Universitaria Ramón Areces, 2006.

MARTINEZ, R y QUIROGA, E. Estructuras de datos. México D.F., Thompson, 2002. 269p.

SAMA VILLANUEVA, Sergio, GONZALEZ RODRIGUEZ, Bernardo Martín y PEREZ PEREZ, Juan Ramón. DSTool: Prototipo para la enseñanza, evaluación y depuración de estructuras de datos basado en mecanismos de reflexión estructurada. España: Universidad de Oviedo, 2002. 7p.

SOLER PELLICER, Yolanda. Consideraciones sobre la tecnología educativa en el proceso de enseñanza-aprendizaje. Una experiencia en la asignatura Estructura de datos. Cuba, 2009. 9p.

TIMARAN PEREIRA, Ricardo, CHAVES TORRES, Anívar, JIMENEZ TOLEDO, Javier, CHECA MORA, Juan Carlos, ORDOÑEZ ERAZO, Hugo y COLUNGE, Constanza. Un cambio de paradigma en la enseñanza de fundamentos de programación en Ingeniería de Sistemas. Bogotá, D.E, Revista Educación en Ingeniería, 2009. 120p.

UVIÑA, Patricia Ruth, BERTOLAMI, Mabel Angélica, CENTENO, María Elena y ORIANA, Gabriela Carmen. Mapas Conceptuales: Una herramienta para el aprendizaje de estructuras de datos. Argentina, 2005. 4p.

VILLALOBOS SALCEDO, Jorge Alberto, PEREZ, Danilo, CASTRO, Juan y JIMENEZ, Camilo. Construcción de un laboratorio flexible de Estructuras de Datos. Bogotá, D.E, 2008. 12p

VILLALOBOS SALCEDO, Jorge Alberto. Una solución integral al problema de enseñar a aprender a programar. Bogotá, 2009. 37p.

VILLALOBOS SALCEDO, Jorge Alberto. Introducción a las estructuras de datos: Aprendizaje activo basado en casos. Bogotá, Colombia: Pearson Educación, 2008.

ANEXO A

Transcripciones – docente A

Tema 1: Manejo de Archivos

El docente hace uso de una herramienta de Software denominada NetBeans. Explica a sus estudiantes que procederá a la creación de un proyecto llamado “Manejo de Archivos” el cual tendrá como clase ejecutora la principal o “main”.

Después dentro de la carpeta de NetBeans Projects encontramos un archivo llamado Console.jar que permite ejecutar una aplicación en donde se almacene la información dentro del archivo y no tener que estar digitando cada vez que se ejecute el programa.

La información tiene que poblarse automáticamente para que pueda facilitarse la lectura de datos dentro del compilador, la manera más sencilla de hacerlo es por medio de archivos.

Se manejan archivos de texto plano así que se manejan archivos .txt. Dentro del método ejecutor de la clase Main se pasará como parámetro la ruta del archivo dentro de un objeto tipo Archivo llamado a1 que corresponde a un archivo de texto que se encuentra dentro de la aplicación Console.jar

Para abrir el archivo debe colocarse una condición de verificación de tal forma que si no se cumple el programa debe imprimir un mensaje que diga “Problema con el archivo” indicando que el archivo no se puede abrir, de lo contrario (else) se puede abrir el archivo y sigue con el programa.

Afuera de la condición if cuando continua el programa se utiliza el objeto a1 y el método .close() me permite cerrar el archivo apenas se es leído; el programa permite la apertura de cualquier archivo siempre y cuando sea en formato texto. Al ejecutar el programa, observamos que aparece el mensaje “Problema con el archivo” puesto que no se ha creado tal archivo. O es posible que el archivo sí exista pero no se encuentra en la ruta que fue pasada como parámetro.

El docente procede a crear el archivo en la ruta correcta y nuevamente realiza la ejecución del programa bajo el entorno de desarrollo de NetBeans. Al crear el archivo, el docente escribe unos cuantos datos al interior para que al ejecutarse el programa se pueden leer dichos datos.

El docente nuevamente ejecuta el programa y ahora, para leer cada línea del archivo se utiliza el objeto a1 y se llama el método readline() (a1.readline());, después almacena ese comando en un String llamado línea, ahora en un ciclo while pregunta si línea es diferente a nulo, como es cierto entonces imprime el dato que se encuentra en esa línea, después dentro de una variable entera

número guardamos la conversión de la línea anterior leída a un número entero con `Integer.parseInt()`; la finalidad del programa es tomar los datos numéricos del archivo para sumarlos y calcular su promedio, así que por consiguiente declara otra variable tipo `int` llamada `suma` y en ella guarda la suma de cada número cada vez que se repita el ciclo. Al final cuando sale del ciclo captura la suma de todos los números y los divide por la variable `contador` que en este caso corresponde al número de veces que repite el ciclo, emulando la cantidad de números que hay en el archivo, al finalizar se imprime el resultado de la suma y el promedio.

El docente indica que la finalidad del programa es explicar la lectura de datos desde un archivo de texto. Explica que el programa puede ser más eficiente si se valida la conversión de `String` a entero con `try/catch` para que no haya ningún error de conversión.

El docente, quiere explicar algo más complejo, modificar los datos del archivo anterior para probar la validación con el `try/catch`, ¿Cómo hace esto? Modificando un dato numérico y coloca una cadena de caracteres. Ejecuta el programa y al leer la línea donde están las letras se produce un error debido a la conversión incorrecta de un valor numérico a `String`. Ahora el docente crea otro archivo de solo caracteres en los que se pueda ver la variedad de datos como son nombre y nota, para que el programa desarrollado no sufra inconvenientes es necesario eliminar la conversión de la variable, las operaciones de suma y promedio, todo esto para que pueda leerse una línea completa, no obstante es menos eficiente. Así que ahora, utilizamos el comando `split` para crear un token que permita buscar el token en el archivo y separe el texto, ahora se crea un arreglo el cual se pueda guardar toda la línea y que pueda leerse sin ningún problema.

El docente explica que para separar el nombre y la nota es necesario colocar en la posición 0 del arreglo solo el nombre, ahora con las notas las convierte en enteros y las almacena en las posiciones 1,2 y 3. Después explica que el token " " (espacio) no es aconsejable utilizarlo y es mejor un símbolo como el `#` para cuando se presenten casos como: nombre (espacio) apellido, seguido de la nota, (símbolo) nota.

Luego, es necesario crear una clase `alumno` en donde hacemos un constructor reinicializando las variables y reasignándolas, después en el `main` creamos un objeto de esa clase y automáticamente se coloca como parámetro las variables que se reasignaron, todo esto para pasar los datos que se almacenaron en el arreglo en las variables de la clase `Alumno`.

Finalmente, el docente explica el caso en que queramos escribir dentro del archivo desde la consola, ¿Cómo se hace esto? Es necesario ir a la primera condición donde preguntamos si el archivo no se puede abrir (`!a1.reset();`) y el `.reset()` lo cambiamos por `.rewrite();` que permite reescribir en el archivo mientras se ejecuta el programa.

Concluyendo, el docente indica cómo mantener el orden de los archivos cuando se crean dentro del proyecto, creando una carpeta exclusiva para los archivos y en el código especificar la ruta de esa carpeta.

Para finalizar – anota el docente – el docente explica una serie de ejercicios que deben ser resueltos por los estudiantes y las soluciones deben ser enviadas a la cuenta de correo del docente.

Tema 2: Memoria Dinámica

El docente se ubica frente al grupo, tiene en sus manos un marcador y da inicio a la clase, luego de dar un corto saludo. Inicia haciendo una pregunta:

¿Saben la diferencia entre memoria estática y memoria dinámica? Uno de los estudiantes responde: La dinámica es la de punteros.

El docente responde “SI” y pregunta por la memoria estática a lo cual responden que es la que no es la de punteros. Todo el grupo se ríe al igual que el profesor, al parecerle una respuesta poco adecuada.

Nuevamente, el docente pregunta: Cuál es el típico ejemplo de memoria estática utilizada en programación? A la cual, uno de los estudiantes responde: Los Arrays.

Uno de los estudiantes se levanta del puesto y pasa frente a la cámara de grabación, el docente hace un comentario al respecto al igual que otros estudiantes del grupo y se ríen. Retoma el tema y pregunta: Porque un Array es estático? Uno de los estudiantes responde diciendo que es estático porque hay que definirle cuánto va a ocupar previamente. El docente toma la respuesta como válida y la reafirma diciendo que efectivamente se le da un tamaño el cual no puede cambiar dinámicamente durante la ejecución del programa.

Hace una nueva pregunta: ¿Por qué un puntero sirve para manejar una memoria dinámica? Un estudiante responde: porque se podrá cambiar durante el programa. Nuevamente el docente toma como correcta la respuesta y retoma las palabras del estudiante.

Empieza a hacer una explicación acerca de punteros para ello escribe en el tablero explicando que primero se define el tipo. Antes de continuar pregunta que es un puntero y cuál es su uso. Un estudiante responde: “Para apuntar”. El docente a su vez, afirma que es “cierto” (todos los estudiantes se ríen)

El docente explica la definición de puntero: “variable que apunta a una dirección de memoria, es decir almacena números enteros. Almacena direcciones de memoria”.

Y describe un ejemplo de un puntero explicando la sintaxis que este debe tener y la anota en el tablero. La sintaxis está compuesta por el tipo, se pone el identificador correspondiente y a la derecha circunflejo y el tipo de aquello a lo que apunte el puntero. Dice que esta es la forma general con todos.

En otra línea del tablero explica cómo se declara una variable de tipo puntero, antes de comenzar a escribir pregunta la forma como se debe hacer la declaración y entre todos van dando la respuesta mientras que él va anotando.

Continúa explicando que pasa cuando el compilador llega a dicha instrucción. Alguien responde que se crea una cajita (se reserva una parte de la memoria), el docente la dibuja y anota que tipo de números almacena. El docente pregunta: Inicialmente qué valor tiene?

Uno de los estudiantes responde “basura”. A lo cual, el docente indica que es correcto puesto que se trata de un valor desconocido que se puede considerar como basura al no conocer dicha información la cual hasta este momento no interesa.

El docente continúa anotando debajo de la caja; para explicar que es como una flecha que no apunta a ningún sitio. Está claro hasta ahí? (pregunta el docente).

Sí (responden en su mayoría, los estudiantes).

El docente, procede a dar un ejemplo con valores numéricos y pregunta si se puede hacer que un puntero que inicialmente no apuntaba a nada termine apuntando a una de las cajas dibujadas. La respuesta de los estudiantes es un rotundo “Sí”.

EL docente, pregunta la forma como se debe hacer. Hay una respuesta que toma como válida, anota el nombre de la variable que inicializa el puntero y explica que se utiliza para apuntar a tierra, es decir que lo único que hace es inicializar el puntero. (No hace mayor referencia al tema y lo deja para explicarlo a mayor profundidad más adelante, cuando se éste profundizando al respecto).

Retoma todas las respuestas que dieron con respecto a cómo hacer que un puntero apunte a una parte específica. Explica cómo funciona el símbolo arroba anotándolo en el tablero la sintaxis. (Devuelve la dirección de memoria).

Explica que todo esto está en la memoria RAM y como al anotar esa expresión el puntero queda dirigido a la otra caja que previamente había dibujado.

El docente hace un comentario: “Lo de punteros es sencillo aunque las personas piensan que es algo complicado”. Se puede a través de un puntero modificar el número que hay en la caja a la cual apunta? (Pregunta nuevamente el docente).

Responden “Sí”, nuevamente la mayoría de los estudiantes. El docente, explica la forma como se hace (anota nuevamente la sintaxis explicando cada una de sus partes) y la secuencia de como el dato que se anota llega a ubicarse en la otra caja. Explica que de esta forma se pueden hacer los cambios el número de veces que se quiera. Explica que así funciona el puntero pero que no se utiliza para eso porque sería una tontería hacerlo para eso, sin embargo dice que luego entran en juego otras operaciones que hacen el tema importante.

También explica la importancia de poder crear cajas de memoria cuando se esté programando de forma dinámica, cosa que no pasa por ejemplo con un Array.

Ahora, el docente pregunta: Cómo se puede crear una memoria dinámica?

Uno de los estudiantes responde que se debe utilizar la instrucción **new**. El docente pregunta si puede borrar el tablero y procede a hacerlo, borra una parte y deja las primeras líneas con la sintaxis. Explica la sintaxis de la instrucción **new** y pregunta: **new** es una función o un procedimiento?

Un estudiante responde "Procedimiento". El docente, explica que sucede al ejecutarse la instrucción anotada. Dibujando las cajitas y el diagrama de como el puntero se engancha a la misma. Hace énfasis que la caja no tiene un nombre pero explica que no es un detalle importante porque a través del puntero se puede acceder a la caja que es lo importante. Ahora le da valores numéricos al ejemplo y anota la sintaxis para hacerlo, (los estudiantes ayudan con la construcción de la sintaxis, mientras el docente la anota). El docente lanza una nueva pregunta: Cómo se haría para imprimir por pantalla el valor de la caja ?.

Anota la sintaxis para hacerlo, al final dice que se genera un error y pregunta si es de ejecución o compilación?

El error es de compilación, explica porque se daría el error y lo corrige.

Luego explica algunos casos en los cuales no se puede imprimir directamente por pantalla y el porqué.

(Hace un paréntesis para informar que quienes no hayan visto el curso anterior de Programación con él les enviará apuntes por correo de los temas vistos).

Se retoma la clase con una pregunta de un estudiante relacionada con el ejemplo que se venía dando y la sintaxis que se maneja para hacerlo así como la acción.

(Una estudiante ingresa tarde a clase, saluda y se ubica en su puesto).

El profesor escribe una nueva variable y una sintaxis para una acción y pregunta si eso se puede hacer. A lo cual responde que sí, porque son del mismo tipo que apuntan a variables de tipo entero, es decir se hace una asignación directa. Pregunta del docente: Con cualquiera de los punteros se puede cambiar la información de la caja?. La respuesta de los estudiantes en gran parte es "Sí".

Ahora, el docente explica porque la respuesta es válida y escribe una nueva línea de código y pregunta que ocurre. Hay varias respuestas que las toma como válidas y explica que hace esa nueva sentencia. Nuevamente anota otra y explica que esta nueva sirve para lo contrario del anterior, borra o libera la caja, mientras que la anterior insertaba.

Anota una nueva sentencia explicando que no es necesario o indispensable hacerlo, pero recomienda tener en cuenta que hay lenguajes de programación en los que si es necesario hacerlo. Un estudiante pregunta por otra sentencia y si es lo mismo que la que está explicando. El docente responde que no es igual y le recalca que se debe tener cuidado y explica en que radica la diferencia. Porque esa instrucción realmente lo que hace es que se desengancha de la caja y a esa dirección de memoria no habría nada que le esté apuntando, mientras que la

anterior lo que hace es que borra el contenido de la misma. Es decir que con la instrucción por la cual pregunta el estudiante se generaría un error en ejecución. Y enfatiza en que eso no se puede hacer jamás porque no estaría liberando la memoria, sino dejando esa dirección de memoria suelta a la deriva. Un estudiante pregunta que si de esa forma no se podría recuperar la dirección de memoria, quedaría entonces perdida o ocupada?

El docente responde que los sistemas operativos existentes tienen algo así como un recolector de basura que se encarga de hacer el trabajo de recuperar la memoria y liberarla.

Pregunta si está claro cómo funcionan las instrucciones que se acabaron de explicar y continua al obtener una respuesta positiva.

El profesor aclara que al tratarse de una introducción al tema de listas, la explicación se hará muy rápidamente...como se generan y como se programan y en clases posteriores se profundizará el tema enfocado a la estructura de datos.

El docente informa que tema sigue en la próxima clase, el cual también corresponde a punteros (como borrar nodos de diferentes posiciones) y a la gestión de Memoria dinámica. Recomienda repasar el tema e informa que enviará material al correo de cada uno de los estudiantes.

Se levantan los estudiantes y se termina la clase.

Tema 3: Fundamentos de Estructuras de datos

Bienvenidos señores estudiantes, en primer lugar en esta presentación vamos a ver principalmente las estructuras dinámicas, (el docente: está preparando las diapositivas en el computador y está realizando la lectura directamente de pantalla), bien vamos a ver lo que es el objetivo de la asignatura, se debe tener presente que el objetivo de la asignatura es: En primer lugar conocer cómo están estructurados los datos principalmente los datos compuestos y en segundo lugar conocer los algoritmos y procedimientos básicos utilizados para manipular la información de estas estructuras, entendiéndose por manipular a los procesos de inserción, modificación, eliminación, ordenación y búsqueda de datos, bien cuál es el contenido que trataremos el día de hoy?, en primer lugar analizar, conocer los datos de tipo puntero, en clase posterior conoceremos las estructuras dinámicas concretamente las listas enlazadas, como están estructuradas como trabajan, saber cuáles son las operaciones que se realizan sobre estas listas sobre estas estructuras conocer cuáles son las pilas y colas su estructura los conceptos.

Bien datos de tipo puntero, un dato de tipo puntero es una estructura simple que almacena un solo valor en este caso dicho valor es un valor especial, un valor hexadecimal en otras palabras se almacena el valor de una dirección de memoria, aquí es importante puntualizar no que los punteros no son en sí estructuras complejas eso hay que tenerlo bien claro si no hay que alarmarse por así decir que vamos a trabajar con punteros, una variable de este tipo de datos almacena un dato simple si así como los valores de tipo integer como las variables de tipo

char las variables de tipo float almacenan un solo dato es lo mismo con los punteros pero en este caso las variable no almacenan información numérica lo que almacenan en sí son direcciones de memoria un valor concreto una dirección de memoria, vamos a ver posteriormente como se utiliza y como se aplican estas variables en el uso de estructuras dinámicas estructuras más complejas de estructuras compuestas.

Es importante recalcar que la información en la computadora se almacena en casilleros de memoria cada uno de estos casilleros tiene una dirección propia o una dirección específica, es decir cada uno de estos casilleros de memoria en los cuales se encuentra almacenada la información tiene su propia dirección de memoria si es un valor hexadecimal que es justamente a través de esta dirección que se accede a los datos que están almacenados en estos casilleros, estos son algunos aspectos que estamos viendo antes de entrar a ver en sí lo que son las estructuras tipo puntero. Para entender lo que son los datos de este tipo vamos a analizar en primer lugar un programa sencillo, si ustedes ven en la diapositiva, hemos hecho un pequeño programa en C en el cual se hace algo tan sencillo como declarar una variable número de tipo entero, no hay nada nuevo en esa parte posteriormente a la variable número se le asigna el valor: el dato 43 un dato de tipo entero seguidamente presentamos este valor si el valor que está en la variable número en la siguiente línea le actualizamos, le cambiamos el valor a 100 si se actualiza el valor de la variable número finalmente presentamos este nuevo valor el 100 que se cargó en la línea anterior, se presenta el último valor de la variable, ustedes pueden ver al ejecutar este programa el cual no tiene nada de nuevo, pero es una base que estamos tomando para las posteriores explicaciones de los datos de tipo puntero.

En esta diapositiva ustedes se podrán dar cuenta, hemos tratado más o menos de especificar o indicar como estaría almacenada la variable en los casilleros de memoria, ustedes ven que el 43, en la columna donde está el valor 43 justamente corresponde a la variable número y todo el resto de casilleros están vacíos y en la parte izquierda cada uno de los casilleros está acompañado de una dirección de memoria de un valor hexadecimal, ahora antes de entrar a más detalles voy a puntualizar algo, las variables que corresponden a los diferentes tipos de datos sean char entero float o double cada una de estas variables utilizan una cantidad de casilleros específica para almacenar su información que quiero decir con esto la variable número recordemos que era una variable de tipo entero y se podrán dar cuenta que para almacenar el 43 que es un valor de tipo entero se ha necesitado cuatro casilleros, por ejemplo los datos de tipo char necesitan un solo casillero para almacenar la información y los datos de tipo float, double, long que tienen mayor capacidad de almacenamiento obviamente utilizaran mayor cantidad de casilleros podrían ser ocho casilleros doce casilleros o inclusive dieciséis casilleros de memoria es decir para almacenar los datos no siempre se trata de un solo casillero que se utiliza sino dependiendo del tipo de dato podrán ser cuatro, ocho o más casilleros de memoria entonces, en este caso miren que el 43 como lo decía

esta en cuatro casilleros consecuentemente hay cuatro direcciones de memoria, pero en este caso cuando un dato se almacena en más de un casillero, la dirección de memoria que corresponde a ese bloque de cuatro casilleros está dada por la primera dirección, la primera dirección de esos cuatro casilleros corresponde a la dirección de esta variable en la cual se encuentra almacenado el valor 43, entonces no quiere decir, que para un dato una variable de tipo entero están asociados cuatro direcciones de memoria siempre va a ser una sola, es decir en este caso la primera dirección de memoria es la que se le asigna por defecto a todo el bloque en el cual se encuentra almacenado el dato, entonces es más o menos como se almacenarían los datos en memoria.

En el siguiente programa vamos a hacer exactamente lo mismo que se hizo en el programa anterior, pero en este caso lo haremos utilizando ya no directamente la variable de tipo número sino en este caso lo haremos utilizando direcciones de memoria y concretamente utilizando una variable de tipo puntero, entonces vamos a analizar línea por línea este programa, igualmente en la primera línea se declara una variable de tipo entero, igual que en el programa anterior en la siguiente línea miren que declaramos una variable **ptrnum**, se darán cuenta que esta variable es de un tipo `int *` lo cual indica que es una variable puntero `*`, cuando declaramos variables puntero, se hace a través del operador `*` (asterisco), `int *`, `char *`, `double *`, lo cual indica que es un puntero a entero, puntero a char o puntero a un dato de tipo double, siempre hay que especificar el tipo y adicionalmente con el `*` indicamos que es un puntero la variable que se está declarando, un puntero es una variable que va a almacenar direcciones de memoria, entonces hemos declarado la variable **ptrnum**, antes de continuar voy a puntualizar algo, cuando trabajamos con punteros es muy importante o normalmente se utilizan dos operadores en primer lugar lo que es el `&` (ampersand) y en segundo lugar lo que es el `*` (asterisco), son los dos operadores importantes o principales o únicos que se utilizan cuando trabajamos con programas que utilizan punteros, entonces vamos a ver en esta línea que lo que estamos haciendo dice `ptrnum = &numero`, entonces en este caso el `&` lo que hace es extraer la dirección de memoria en la cual se encuentra almacenada una determinada variable por ejemplo en este caso si se ha declarado previamente la variable número y con el operador `&` de número lo que se está haciendo es extraer la dirección en la que se encuentra almacenada la variable número, si `&` de número me extrae la dirección de memoria y esa dirección de memoria obviamente ustedes se podrán dar cuenta que se almacena en la variable `ptrnum`, recordemos que las variables de este tipo almacenan direcciones de memoria entonces hemos extraído la dirección de la variable número y la hemos almacenado, la hemos cargado en la variable de tipo puntero, entonces ese es el uso del operador `&` cuando hablamos de punteros, extraer las direcciones de memoria de una determinada variable o de una determinada estructura, ahora vamos a analizar en la siguiente línea lo que es el operador `*`, si el operador `&` extraía la dirección de memoria el operador `*` va a hacer exactamente todo lo contrario va a extraer el valor, el valor el dato la información que se encuentra almacenada en una determinada dirección es decir todo lo

contrario, por ejemplo en este caso miren **ptrnum**, recordemos que allí esta almacenada una dirección de memoria, entonces con * de ptrnum lo que estoy haciendo es accediendo al dato, a la información ya sea accediendo para leer o escribir datos en ese casillero, entonces * de ptrnum lo que hace es acceder al dato que está almacenando el valor 43 en esa dirección de memoria, seguidamente lo que se hace es presentar el valor que está en la variable número, de hecho va a ser el valor 43 puesto que cargamos el 43 en la dirección de memoria de la variable número, presentamos el valor de la variable número seguidamente utilizando el mismo principio de dirección de memoria actualizamos a un valor 100, finalmente presentamos o a continuación presentamos la variable número con el valor 100 que se ha cargado y en este programa le he aumentado una línea adicional, en la cual se está presentando la dirección de memoria de la variable número, es decir el valor que tiene la variable ptrnum en la cual se encuentra almacenada una dirección de memoria que corresponde a la variable número, este es un valor hexadecimal, ustedes pueden ver en la parte inferior la corrida del programa, presentamos el valor 43 presentamos el valor 100 y finalmente presentamos la dirección de la variable número, es tal como ustedes se podrán dar cuenta un valor hexadecimal 0012ff7c que es un dato hexadecimal, entonces esa es una breve explicación de cómo trabajarían o como trabajan las variables de tipo puntero, es una explicación previa para analizar lo que son las listas enlazadas las cuales utilizan punteros ya de una forma un poco más avanzada.

Tema 4: Listas enlazadas

Docente: Buenos días, ahora vamos a continuar con el tema de las listas enlazadas, conceptualmente vamos ver el siguiente gráfico el cual nos indica cómo está estructurada o que es una lista enlazada, la cual corresponde a una serie de nodos si, ustedes se podrán dar cuenta que aquí tengo el primer nodo que tiene un valor de 9 el siguiente nodo que tiene un valor de 8 el siguiente que tiene un valor de 10 y el ultimo nodo que tiene un valor de 3, cada uno de estos nodos está compuesto de dos partes, el primer campo en el cual se almacena la información propiamente dicha del nodo, en este caso son valores enteros 9 – 8 – 10 y 3 en el primer campo y en el segundo campo lo que se almacena es una referencia, un enlace una forma de enlazar cada nodo al siguiente nodo. Es decir, en el campo dos si en la segunda parte del nodo se almacena alguna referencia para indicar cuál es el nodo que está a continuación de él en este caso del 9, entonces en el primer campo el valor y en el segundo una referencia o una dirección al próximo nodo, entonces en este caso que es lo que se almacenaría en la segunda parte si, lo que se va a almacenar allí es una dirección de memoria un dato de tipo puntero la dirección de memoria que corresponde al siguiente nodo en este caso al nodo 8, cual es la dirección del nodo 8 esa dirección de memoria de ese nodo se encuentra almacenada en la segunda parte del nodo, en este caso del nodo 9, para hacer el enlace se podrán dar cuenta que el último nodo en este

caso el nodo que tiene el valor 3 en su segunda parte lo que tiene hay es un valor null, o sea no tiene ninguna dirección de memoria puesto que hay no hay ningún enlace a ningún nodo siguiente simplemente lo que se almacena ahí es un valor un dato de tipo null, esto es de una manera conceptual no vamos a ver estos nodos como estarían organizados más o menos en la memoria del computador esta misma lista como estaría organizada, miren esta diapositiva más o menos sería una forma de como de indicar como estaría la lista en memoria en primer lugar fijémonos que cada uno de los nodos no se encuentra en posiciones consecutivas, se encuentran separadas por diferentes cantidades de casilleros, es decir los datos los nodos se encuentran dispersos en la memoria, entonces miren el primer nodo en su primera parte tiene el valor 9 tal como habíamos indicado si, adicionalmente recordemos que los datos de tipo entero utilizaban cuatro casilleros para almacenarse si, ustedes se podrán dar cuenta que hay el valor nueve esta justamente en cuatro casilleros igualmente los datos de tipo puntero utilizan cuatro casilleros para almacenar este dato este valor hexadecimal utilizan cuatro casilleros para almacenarse es decir todo el nodo utilizaría en total ocho casilleros de memoria, entonces miren para cada uno de los nodos el nodo 9 el nodo que tiene el valor 8 el nodo que tiene el valor 10 y el nodo que tiene el valor 3 no, cada uno de ellos en su segunda parte lo que tiene es una dirección de memoria pero cuál es la dirección de memoria si, ustedes podrán darse cuenta que he indicado hay con unas flechas se ha almacenado se ha cargado hay el valor de memoria que corresponde al siguiente nodo es decir si se fijan en el nodo 8 el primer casillero es el d2614, bueno son valores hexadecimales ya es el d2614 que justamente que es el primer casillero de la estructura que es el primer casillero del nodo ese valor es el que lo almaceno en el campo del hilo en el campo de enlace del nodo anterior, es la forma como se van a ir enlazando cada uno de los diferentes nodos si y es la manera como utiliza la computadora como utilizan las aplicaciones para ir recorriendo los diferentes valores de la lista enlazada, repito nuevamente los nodos de una lista enlazada se encuentran dispersos en la memoria no se encuentran de forma consecutiva si como era en los arreglos si recordemos que en los arreglos las estructuras de datos estáticas la información se encontraba de manera consecutiva si, declarábamos un arreglo de 10 de 20 de 30 elementos los cuales se encontraban a continuación y simplemente accedía a ellos a través del índice del arreglo en este caso las estructuras dinámicas no es así los nodos se encuentran dispersos y para acceder de un nodo a otro lo hago a través del campo de liga o del campo de enlace, es muy importante trabajar con estructuras dinámicas puesto que estas fíjense que yo puedo ir creando nodos conforme yo vaya necesitando almacenar más información yo voy creando nodos y voy haciendo simplemente los enlaces correspondientes, esto es en cuanto para ver como estaría una lista enlazada organizada distribuida almacenada en memoria (cambia de diapositiva), bien a continuación vamos a ver un programa una aplicación en el cual primero o básicamente se crean nodos no justamente se crean los nodos del ejemplo anterior si, 9 , 8 , 10 ,3 y en este caso la lógica de la aplicación me indica que cuando digite un valor de -1 simplemente el proceso de ingreso de nodos finalice.

Miren, he ingresado al final el -1 y el -1 obviamente no se almacena en la lista simplemente es una manera de controlar que no se siga o indicar hemos llegado al fin del ingreso de nodos a la lista enlazada, entonces en este caso vamos a analizar como estaría este programa, como estaría trabajando este programa para lo cual me voy a apoyar aquí en la pizarra para ir graficando como se van a ir creando los nodos miren en este programa tengo (cambio a la pizarra), numeradas las líneas así que voy a ir haciendo referencia a cada una de las líneas de la aplicación, voy a ir haciendo referencia a las líneas de la aplicación para ir graficando como lo trabajaría este programa bueno, la primera línea se declara la librería estándar de entrada y salida seguidamente creo una estructura una estructura elemento que tiene los dos campos que les había mencionado no el campo información y el campo siguiente, miren el campo información es un dato de tipo entero y el campo siguiente es un dato de tipo puntero, fíjese que tengo elemento * lo cual me indica que es un puntero a una estructura no, habíamos hablado que los punteros pueden ser punteros a enteros punteros a char punteros a cualquier tipo de dato no en este caso lo que tengo es un puntero una variable, la variable siguiente que va a almacenar una dirección de memoria un puntero a una estructura de tipo elemento, bueno posteriormente defino un nuevo tipo de dato el tipo de dato nodo justamente que es un tipo de dato de la estructura antes indicada, bien entramos al programa principal (cambia a las diapositivas), en el cual se declaran las variables que vamos a utilizar, en primer lugar una variable de tipo entero de tipo dato la cual me va a servir para ir recogiendo los valores que voy a ingresar, bien a continuación tengo tres variables de tipo puntero, una variable pcabeza una variable p y una variable auxiliar, las tres de tipo puntero ya en este aspecto es importante comentar, que cuando trabajamos con listas enlazadas la puerta de entrada el punto de partida para acceder a los datos de la lista enlazada es (empieza a escribir en la pizarra) el primer nodo de la lista enlazada en este caso vamos a utilizar una variable pcabeza para almacenar la referencia o la dirección de memoria de dicho nodo del nodo inicial de la lista enlazada, entonces vamos a ver como trabajaría este programa bien, (dibuja el nodo en la pizarra y va a hacer la explicación en ella) , bien en la línea 15, que es justamente la línea donde empieza a ejecutarse ya lógica del programa en este caso tengo la línea pcabeza=malloc(sizeof de nodo que significa esto, la función malloc me sirve para asignar o crear un espacio de memoria para el nodo que voy a crear es decir en este caso se crea (cambia a las diapositivas para mostrar el código), se crea un nodo vacío no, un nodo cuyo nombre es justamente pcabeza si, en la línea 15 repito que es lo que hace la función malloc, me reserva memoria y me asigna memoria justamente para una estructura de tipo nodo que es la que ustedes pueden ver hay sizeof me devuelve la cantidad de casilleros en este caso ocho casilleros que utiliza la estructura nodo recordemos que nodo utiliza cuatro casilleros para el campo información y cuatro para el campo puntero para el campo siguiente, entonces me reserva ocho casilleros en definitiva que están divididos en dos partes una parte para la información y otra parte para el dato de enlace para el dato de tipo puntero, entonces en esta línea lo que se ha hecho reservar memoria y sobre todo asignar ya un espacio de memoria para para este

nodo, bueno en la línea 16 simplemente se presenta un mensaje que dice ingrese la información del nodo si no hay nada nuevo hay en la línea 17 leo el dato ingreso el dato a la computadora el dato que quiero ingresar si un valor de tipo entero en la variable dato si la variable dato que es de tipo entero si ingreso un valor, ahora nos interesa si en este caso vamos a suponer que ingresamos el valor 9 tal como está en la corrida del programa, el número 9 que es el primer elemento de la lista bien lo que nos interesa ver es que es lo que se hace en las líneas 18 y la línea 19 miren en la línea 18 el dato el valor que lo había leído el que lo había ingresado previamente es decir el valor 9 ese dato ese valor lo cargo en el puntero en el nodo pcabeza que ya está creado y en el campo información almaceno ese valor almacena esa información es decir en este caso si aquí se almacena el valor 9 (regresa a la pizarra para dibujar el nodo y seguir con la explicación), es lo que se hace en esa línea, cargar el dato leído el número nueve cargarlo en el campo información y a continuación en el campo siguiente pcabeza en el del nodo pcabeza que lo acabo de crear en el campo siguiente cargo un valor null, es decir este prácticamente quedaría algo así un valor en null es decir no apunta esto a nada si es una, hasta el momento tengo una lista enlazada de un solo elemento, es decir en este caso tengo el elemento pcabeza simplemente bien, hasta esta parte lo único que he hecho es crear el nodo inicial de la lista enlazada ahora vamos a entrar a un ciclo repetitivo si el cual como les decía me permite ir creando nodos hasta que yo presione ingrese el valor -1, entonces en este caso utilizo la variable p que es igualmente de tipo puntero (cambia a las diapositivas para mostrar el código), utilizo la variable p para irme moviendo por la lista enlazada, recordemos que la variable pcabeza no debe cambiar de valor puesto que es el dato que yo necesito posteriormente para hacer el recorrido de la lista enlazada yo necesito saber cuál es la dirección del primer nodo de la lista enlazada esta dirección se encuentra almacenada o va a quedar almacenada en la variable pcabeza por lo tanto esta variable no debe cambiar de valor debe quedarse guardando el valor correspondiente al primer nodo de la lista, para irme moviendo en la lista para el proceso de crear los siguientes nodos voy a utilizar una variable p y como ustedes se podrán dar cuenta si digo que pcabeza es igual a p es decir en este momento p también tiene el mismo valor el mismo nodo.

Bien, vamos (regresa a la pizarra para dibujar el nodo y seguir con la explicación), a ver ahora como sería el proceso ya repetitivo que prácticamente sería muy similar al proceso anterior que revisamos, tengo la línea, ingresa la información del nodo es decir para ingresar el nuevo dato que voy a guardar en la lista enlazada si ingrese la información del nodo scanf para leer el dato para ingresar el dato al sistema en este caso voy a ingresar el valor 8, le corresponde al segundo elemento si ingreso el valor 8 a continuación verifico que sea un valor diferente de -1, recordemos que con -1 salíamos del proceso repetitivo no, si es que el valor es diferente de -1 entonces hago lo mismo creo un nuevo nodo, asigno creo un nuevo nodo en memoria es decir hago más o menos esto miren (regresa a la pizarra para explicar la creación del nodo), un nodo cuyo valor es auxiliar, así de esa manera no hay enlaces no hay valores no hay absolutamente nada lo único que se ha

hecho es reservar memoria y asignar una dirección, ocho casilleros repito divididos en dos partes para el nodo en cuestión, entonces nodo auxiliar ya está creado si ya está creado en memoria bueno en este caso está vacío no tiene información pero ya está creado en memoria eso lo hemos hecho en la línea 26 igualmente en la línea 27 y en la línea 28 le asignamos la información que va a tener este nodo, en este caso cuál era el dato?, el dato era el número 8, y en el campo liga en el campo enlace miren que tiene un valor null en este caso estaríamos hasta de esta manera estaría nuestra estructura, fíjense que hasta este momento no hay ningún enlace entre los dos nodos creados están completamente separados no están enlazados de ninguna manera no en todo caso el nodo auxiliar ya tiene información en el campo de datos tiene el valor 8 y en el campo de enlace tiene un valor null ahora es muy importante la siguiente línea para que nuestra lista quede consistente hacer enlazar el nodo anterior que en este caso es el nodo inicial o el nodo cabeza enlazarlo con el nodo recién creado para esto que es lo que hago, cargo o asigno el valor de la variable auxiliar es decir el valor la dirección de memoria y recordemos que esta variable auxiliar es un dato de tipo puntero, esta dirección de memoria que está en la variable auxiliar esta dirección de memoria tengo que copiarla en este campo de aquí, en este campo de aquí tengo que copiar el valor de la dirección de memoria que esta en este campo auxiliar en el momento en que yo copio esta dirección de memoria automáticamente me indica que esto ya está enlazándose justamente a este nodo si, se actualiza el valor null que había aquí en este campo se actualiza y en vez de ese null copio el valor de la dirección de memoria que está en la variable auxiliar con lo cual fíjense que ya ha quedado enlazado si ya quedaron enlazados el primer nodo con el siguiente con el segundo en este caso y así es como se van a ir creando los siguientes nodos si para eso es lo que me sirve la variable p para irme moviendo por los nodos que estoy creando ahora vamos a ver lo que pasa en la línea 30 miren que es lo que hago actualizo el valor de la variable p si, la variable p en este momento está apuntando al primer nodo de la lista lo actualizo al siguiente nodo es decir al segundo tacho esto y ahora la variable p se ubica en esta posición sí, eso me va a permitir irme moviendo e ir creando más elementos llego al fin del ciclo repetitivo bueno pregunto si dato es diferente de -1 obviamente tengo que continuar creando elementos vamos a crear en este momento el dato 10 no, regresamos a la primera línea es decir a la primera línea del proceso repetitivo a la línea 23 presentamos un mensaje que dice indique ingrese la información del nodo en la línea 24 ingresamos el dato que dato lo vamos a ingresar en este momento va a ser el dato 10 si el dato 10 validamos que 10 sea diferente de -1 obviamente lo cual vamos a ingresar a esa estructura a esa estructura condicional con el dato leído con el dato ingresado el valor 10 bien entonces ahora vamos a hacer el mismo proceso no nuevamente en la variable auxiliar vuelvo a utilizar vuelvo a reservar memoria a través de la función malloc vuelvo a reservar una cierta cantidad de bytes para un nuevo dato es decir en este caso ejecuto nuevamente la función y creo un nuevo nodo miren se ha ejecutado en la línea 26 se ha ejecutado solamente esto se ha creado un nodo vacío obviamente como estoy utilizando la variable auxiliar esto ya se borra se

elimina no existe ya y ahora la variable auxiliar esta acá puesto que lo ejecutado nuevamente esa línea y se ha cargado un nuevo se ha reservado una nueva porción de memoria para este dato para este nodo y hago lo mismo no en la línea 27 y 28 en primer lugar cargo el dato leído el dato ingresado el valor 10 y en la línea 28 cargo aquí un valor null miren que se ha creado ya el nodo el nodo tiene información pero aún no está enlazado al nodo anterior mejor dicho el nodo anterior no está enlazado con el nodo auxiliar que acabamos de crear lo cual se hace en la línea 29 no el valor de memoria que está en la variable auxiliar el valor de la dirección de memoria que está en la variable auxiliar lo actualizo lo copio en el campo p si, si es el campo p perdón el nodo p en el campo siguiente es decir esta dirección de memoria la copio aquí una vez que ha sido copiado esto me indica que esto ya está apuntando al siguiente nodo de la lista si, para seguir creando más nodos para seguir creando más nodos muevo la variable p al último nodo creado no es decir aquí esto se elimina se borra se actualiza la variable p la variable p se va moviendo con el fin de permitirme irme movilizándolo por la lista y creando más nodos vamos a llegamos a la línea 30 en la línea 32 recordamos dato diferente de -1 si el 10 fue diferente de -1 y vamos a continuar en el proceso repetitivo regresamos a la línea 23 en la línea 23 presento un mensaje en la línea 24 ingreso el nuevo dato que quiero almacenar en la lista en este caso el número 3 si en la línea 25 evaluó si este dato es diferente de -1 si lo es no cierto por lo tanto ingreso a ese bloque de líneas de código que corresponden a la estructura condicional if que es lo que hago en la primera línea de este bloque de código de este bloque de código en la línea 26 que es lo que hago nuevamente ejecuto la función malloc que les decía me sirve para reservar memoria y lo que se crea es más o menos esto no un nodo vacío en la variable auxiliar esto ya se borró si en estos momentos se pierde esa referencia ahora la variable auxiliar está apuntando a un nuevo nodo, entonces en la línea 26 he creado ese espacio de memoria en las líneas 27 y 28 actualizo los datos si cargo el valor leído el valor 3 el valor ingresado al sistema y el valor null en el campo de enlace seguidamente que es lo que hago enlazo el último nodo de la lista que está hasta este momento en este caso el 10 el valor de memoria la dirección de memoria almacenada en la variable auxiliar esta dirección la copio en este campo de aquí con lo cual me indica que esto prácticamente queda enlazado si es muy similar como estaría en memoria no los repito los nodos no se encuentran en posiciones consecutivas si se enlazan a través de las referencias a las direcciones de memoria en las que se encuentran los diferentes nodos no ya, se ha hecho un enlace y la variable p igualmente se mueve no se elimina de aquí y se mueve a la siguiente posición finaliza el ciclo repetitivo dato diferente de -1 bueno regresamos a la línea 23 presentamos el mensaje en la línea 24 vamos a leer el dato en este caso voy a ingresar un valor -1 en la línea 24 ingreso un valor -1 pregunto dato es diferente de -1, no lo es por lo tanto no ingreso dato es el valor de la variable dato es igual a -1 no ingreso al bloque de código y simplemente salgo de la estructura condicional también salgo del proceso while sí y con lo cual la lista enlazada quedaría de esa manera no los valores 9, 8, 10 y 3 fíjese que hay un proceso que lo pueden verificar en la guía didáctica hay un proceso que se llama presentar no, al cual se le envía como

parámetro la variable o el dato la referencia del primer nodo de la lista no, la dirección de memoria del primer nodo de la lista en que variable está almacenado esa dirección de memoria si está almacenado en la variable pcabeza si en listas enlazadas simples debe haber una variable que almacene la referencia del primer elemento de la lista para poder hacer el recorrido para poder visitar cada uno de los nodos, entonces en este caso el método presentar hace un recorrido de los nodos de la lista para lo cual recibe como parámetro la referencia al nodo en cuestión si al primer nodo de la lista ya , entonces es la manera como se crearían las o como se crean los nodos en una lista enlazada si lo hemos visto hemos ido analizando cada una de las líneas de código del programa y lo hemos ido especificando como trabajaría este programa si ustedes pueden ver tienen hay la corrida si ustedes tienen hay la corrida del programa para los datos con los cuales hemos trabajado bien vamos a continuar con el siguiente tema que corresponde justamente a las operaciones que se van a hacer sobre listas no, cuales son las operaciones crear la lista sí que sería crear el primer nodo de la lista recorrido es decir moverse por los elementos visitar cada uno de los elementos buscar un elemento en la lista, son las operaciones son muy similares a las que había en arreglos si recorrido buscar insertar elementos yo puedo insertar un elemento intermedio ya vamos a ver posteriormente una como sería este proceso insertar elementos eliminar elementos modificar el valor de un nodo si bien la operación más trascendental que podría existir es la de insertar elementos en una lista no que se refiere el proceso de insertar de alguna manera puede resultar inclusive más sencillo de lo que es el proceso de inserción en los arreglos no porque los arreglos cuando yo voy a insertar un elemento tenía que mover los elementos siguientes tenía que moverlos una posición a la derecha en este caso no tengo que mover absolutamente nada lo único que tengo que hacer es actualizar los enlaces entre los nodos no, entonces miren yo tenía la lista originalmente la lista 9 , 8 , 10 y 3 ya vamos a suponer que quiero insertar el elemento -55 después del nodo cuyo valor es 8, entonces que es lo que se tiene que hacer en este caso borrar el enlace que o que unía al nodo 8 con el nodo 10 borrar el enlace y actualizar los valores o los enlaces de los nodos que ustedes ven si es decir el enlace del nodo 8 en vez de que apunte al nodo 10 tendríamos que actualizarlo y hacerlo que apunte al nuevo nodo creado en este caso un nuevo nodo con un valor -55 y este nuevo nodo creado cuyo valor es -55 si el enlace de este nodo tendría que estar apuntando a su siguiente nodo que en este caso sería el nodo 10 con lo cual la lista quedaría 9 , 8 , -55 , 10 y 3 si, es el proceso de insertar elementos simplemente actualizar eliminar las referencias o los enlaces que se tengan que hacer y actualizar como en este caso con el nuevo nodo creado si igualmente sería el proceso si tuviera que eliminar un elemento de una lista si volvamos al ya a la lista original vamos a suponer que quiero eliminar el valor 8 que es lo que tendría que hacer en este caso si la lista es 9 , 8 , 10 y 3 repito vamos a suponer que quiero eliminar el nodo 8 que es lo que se tendría que hacer es actualizar el enlace del nodo 9 actualizar ese enlace que actualmente está apuntando al nodo 8 en vez de que apunte al nodo 8 hacerlo que se actualice y apunte directamente al nodo 10 es decir prácticamente saltarnos ese elemento

con lo cual simplemente quedaría eliminado de la lista si quedaría la lista 9 , 10 y 3 si actualizando repito la referencia del enlace del nodo 9 hacerlo que apunte al nodo 10 esto es todo lo que tendría que hacer para eliminar un elemento de la lista un nodo de la lista entonces esas son las operaciones que como se trabajaría para las operaciones ya vimos lo que es la inserción la eliminación si buscado sería cuestión de recorrer ir comparando si el elemento que estoy visitando justamente corresponde con el dato que estoy buscando si en la guía didáctica ustedes tienen los programas o la mayoría de programas que corresponden a estas operaciones bien una vez que hemos visto lo que son listas enlazadas vamos a indicar lo que son pilas y colas no una pila es una colección ordenada de elementos a los que solo se puede acceder por un único lugar o extremo de la pila. Los elementos de la pila se añaden o se quitan de la lista de la misma por un solo lugar solamente por la parte superior de la pila si hablar de pilas y colas no es hablar de estructuras técnicamente hablando estructuras nuevas no una pila una cola es simplemente una lista enlazada con algunas restricciones conceptuales si en las listas enlazadas habíamos visto que podíamos insertar elementos en cualquier parte de la misma ya en cambio en las pilas y colas hay algunas restricciones es decir solamente se pueden insertar elementos ya sea solamente por el inicio o solamente por el final si es decir una pila y una cola es una lista enlazada con unos comportamientos específicos nada más técnicamente no hay nada nuevo en cuanto a pilas y colas una cola igualmente es una estructura de dato que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista no un elemento se inserta a la cola por la parte final y se suprime o elimina por el frente o parte inicial de la lista no una cola es muy similar a cuando ustedes van al banco hacen cola para acceder a un servicio el comportamiento es el mismo no entonces eso es lo relacionado con pilas y colas si técnicamente repito no hay nada nuevo son simplemente listas enlazadas con unos conceptos propios de estas estructuras bien y finalmente vamos a ver lo que corresponde a arboles binarios concretamente lo que son arboles binarios de búsqueda ya un árbol binario de búsqueda como ustedes se podrán dar cuenta es una estructura jerárquica no que tiene una raíz y en su parte inferior tiene nodos hijos o tiene nodos descendientes no un árbol binario de búsqueda es una cosa más específica no dada una lista de elementos como ustedes se podrán dar cuenta 50, 10, 70, 30 , 40 , 80, 60 , 20 y 90 dada una lista de números yo puedo construir un árbol binario de búsqueda. Como se construye este árbol binario de búsqueda siempre el primer elemento de la lista dada en este caso el 50 siempre ese elemento va a ser la raíz de la estructura la raíz del árbol y para cada uno de los elementos yo voy a ir analizando si el elemento que voy a insertar es mayor o menor que la raíz por ejemplo en este caso miren vamos a irlo haciendo aquí en una para ir viendo todo el proceso dada la lista que ustedes tienen el primer elemento es el 50 entonces siempre el 50 o el primer elemento va a ser la raíz de la estructura para el siguiente elemento el dato 10 en este caso el 10 es menor que la raíz es menor que el valor 50 lo cual me indica que se va a crear a la parte izquierda de la raíz es decir de esta manera el siguiente elemento a crear es el 70 entonces para el 70 hago el mismo análisis 70 es mayor que el 50 en este caso lo

cual me indicaría que este valor se va a crear a la parte derecha de la raíz es decir de esta manera sí, siempre para cada dato que voy insertando en la estructura tengo que analizar si es mayor o menor que la raíz seguidamente tengo el 30 hago el mismo análisis 30 es menor que 50 no luego me ubico en la posición 10, 30 es mayor o menor que 10 si 30 es mayor por lo cual me indica que se crea hacia la derecha es decir aquí si el valor que lo voy ingresando lo voy a ir analizando en cada uno de los nodos que vaya visitando en el proceso de inserción ya posteriormente que valor tengo el valor 40 si, hago el análisis del 40 con cada uno de los nodos empezando por la raíz 40 es menor que 50 entonces me indica que se va a crear por su parte izquierda llego aquí 40 es mayor o menor que 10 obviamente mayor no cierto me voy a la parte derecha llego al 30, 40 es mayor o menor que 30 mayor con lo cual me indica que este dato se crea en su parte derecha seguidamente tengo el valor 80 si hago el análisis 80 mayor que 50 por su parte derecha 80 mayor que 70 por su parte derecha entonces tendríamos algo así creado de esa manera no posteriormente tengo el valor 60 ya 60 mayor que 50 vamos por la parte derecha 60 menor que 70 lo cual me indica que el 60 se crearía en esta parte no que valor tengo posteriormente el 20 no el 20 menor que 50 por la parte izquierda el 20 mayor que 10 por su parte derecha el 20 menor que 30 por su parte izquierda entonces el 20 se crearía justamente aquí finalmente tengo el valor 90 , 90 mayor que 50 mayor que 70 y mayor que 80 es decir el 90 se crearía algo así no, es como quedaría la estructura. Muchas gracias por su atención.

Tema 5: Stacks

La metodología que implementa el docente es que a medida que va explicando los conceptos va creando las clases y los métodos directamente en netbeans, va desarrollando la aplicación y finalmente explica nuevamente paso a paso cómo funciona el programa.

Vamos a trabajar con el proyecto del cachipun “El profesor está trabajando con Netbeans 6.8 y crea un nuevo proyecto java llamado cachipun2” vamos a hacer una aplicación para el efecto que de poder realizar el ingreso de la información vamos a incluir la librería, la consola ahora necesitamos para poder trabajar en la solución del problema una clase que va a ser la clase cola y en la clase cola van a estar incluidos todos lo que son los jugadores del torneo, por lo tanto vamos a partir creando una clase jugador “Continua con la programación en netbeans ” que van a ser los jugadores que van a participar en el torneo y le vamos a agregar los atributos en este caso necesitamos el nombre el país y el puntaje esos son los tres atributos que vamos a trabajar de la clase jugador, necesitamos hacer el constructor, parte con puntaje 0 porque no parte con ningún puntaje y recibe por parámetro el nombre y el país y además necesitamos hacer lo que son los get y los set encapsulamos y hay tenemos la clase jugador eso es lo que necesitamos por el momento para la clase Jugador, luego viene la clase Cola, la clase cola

ustedes no tienen que implementarla ahora nosotros vamos a tener que implementarla para que nos funcione el problema, pero en la prueba ustedes no necesitan implementar la clase cola y la clase Cola tanto como la clase Pila funcionan con un Nodo que el nodo es el que contiene la información, entonces el nodo contiene "Continua con la programación en netbeans" Jugador que será el dato que va a contener y tiene un puntero al siguiente nodo al fondo un link que tiene el nodo al siguiente igual que generamos el constructor. El constructor recibe el dato que quiero almacenar que en este caso es un jugador y los get y los set, también es una clase bastante sencilla lo único es que es un contenedor de información y luego viene la clase Cola que nosotros vamos a tener que implementar. La cola trabaja con dos nodos, un nodo que marca donde está la entrada y donde está la salida porque en la cola el ingreso es por un lado y la salida es por el otro por lo tanto "Continua con la programación en netbeans" vamos a declarar dos nodos uno que va a ser first y el otro que va a ser last, el primero y el último hay vamos a tener los dos extremos de la cola igual vamos a tener un constructor y el constructor lo único que hace es inicializar los nodos vacíos ahora en la Cola nosotros dijimos que trabajábamos por lo menos con tres métodos "Continua con la programación en netbeans" el pop el push y el isEmpty, para nosotros saber public boolean isEmpty, como sabemos si la cola está vacía porque no hay ningún elemento en ella por lo tanto si el primer elemento es null retornamos verdadero si es que no retornamos falso porque no está vacía, entonces ese es el primer método que normalmente uno implementa ver si es que la cola está vacía o no luego tenemos que implementar el pop y el push entonces para ingresar un elemento implementamos push y el push recibe un jugador que es lo que yo voy a almacenar como la cola trabaja con nodos lo primero que yo tengo que hacer es guardar ese jugador dentro de un nodo, entonces tengo un jugador lo guardo dentro de un nodo y ese nodo es el que nosotros utilizamos dentro de la cola para ponerlo digamos en la fila correspondiente por lo tanto nosotros tenemos que asegurarnos de que primero si es que no tengo datos si está vacía significa que la persona que ingreso va a ser el primero y también va a ser el último porque no había nadie, se ingresa una sola persona esa persona va a ser el primero y el último por lo tanto el first va a apuntar al n y el last va a apuntar al n que sería el nodo que acabamos de crear, entonces si es que es el único elemento va a ser el primero y el último si es que no es el único elemento ósea si es que ya hay más elementos yo tengo que ingresarlo al final por lo tanto lo que tengo que hacer es directamente el nodo punto set siguiente apunte al último nodo que estaba antes en el fondo yo tengo que colocarme detrás del nodo que yo voy a sacar es al revés el this.last que es el último .setget va a apuntar al nodo el último nodo que estaba va a apuntar ahora al nodo que acabo de ingresar y ese va a pasar a ser el último nodo, entonces si yo ingreso un nodo nuevo y la lista está vacía el nodo que acabo de ingresar va a ser el primero y el último a la vez si la lista no estaba vacía el nodo que acabo de ingresar va a quedar al final al principio queda tal cual simplemente va a ingresar al final y va a ser marcado como que ese es mi último porque es el último que ingreso al listado, con eso deberíamos tener el ingreso de elementos, si yo quiero sacar elementos voy a

retornar un jugador se va a llamar método pop y para poder sacar un elemento tengo que asegurarme de que tenga elementos para poder sacar cierto, entonces nuevamente consultamos si no está vacía, que es lo que debo hacer si no está vacía retornar el que esta primero en la fila cierto el ultimo no lo toco solo el que está al principio, entonces dejamos un nodo auxiliar que va a ser igual al primero una vez que yo sé cual es el que tengo que retornar ese ya no va a ser el primero por lo tanto el primero va a ser el siguiente el primero va a ser el que esta segundo en la fila y retorno el que estaba originalmente como primero como el auxiliar es un nodo y yo tengo que retornar un jugador tengo que sacar la información que está dentro del usuario porque dijimos que el jugador lo guardábamos en un nodo y el nodo lo guardamos en la cola, nosotros hay sacamos el nodo y dentro del nodo tenemos que sacar el jugador y ese objeto retornarlo y si es que no solamente retornamos null, si es que no hay ningún elemento en la en la cola, esa es la implementación básica que vamos a tener de una cola.

Ahora bien, si nos vamos al main tenemos que implementar el método por lo tanto lo que primero tenemos que hacer es ingresar los jugadores nosotros podemos trabajar con un controlador o trabajar directo en el main da lo mismo como quieran hacerlo vamos a trabajar ahora todo en el main para que sea más sencillo lo primero que tengo que hacer es cargar todos los jugadores al torneo, por lo tanto tengo que preguntar, ingrese la cantidad de jugadores tengo que agregar la consola y esta es para poder mostrar torneo cachipun, ingresa la cantidad de jugadores int cantidadjugadores igual a readint() hay tengo la cantidad de jugadores que van a participar y ahora tengo que pedir los nombres de todos los jugadores, entonces hacemos un ciclo desde cero hasta cantidadjugadores para no tener que ingresar todos los nombres les voy a poner J1 , J2, J3, J4, solamente para no tener que digitar todo por lo tanto voy a partir desde 1 hasta la cantidad de jugadores igual y voy a generar el nombre pueden generarlo ustedes automático o pueden pedirlo en pantalla da lo mismo y el país van a ser todos de Colombia solamente para evitarme tener que estar digitando todos los países entonces tenemos el nombre y tenemos el país una vez que tenemos los datos yo tengo que crear el jugador le voy a poner igual jugador para que sepamos que estamos, igual a new jugadorrecibo el nombre recibo el país el puntaje no tengo que enviarlo por lo tanto tengo que asegurarme de sacar el puntaje del parámetro porque todos van a partir con cero puntos no tiene sentido que partan con algún puntaje, entonces hay tengo el jugador creado fíjense que la clase parte con mayúscula y el Objeto con minúscula no tienen que llamarse igual necesariamente pero igual uno normalmente los llama igual, tenemos el jugador una vez que tengo el jugador tengo que ingresarlo al torneo que en el fondo es a la cola, por lo tanto voy a crear una cola la voy a llamar lista de jugadores voy a escribir new cola hay tengo la lista y para ingresarlo a la lista lista.push y le paso el jugador hay estoy ingresando el jugador a la lista, voy a pasar otro jugador y lo voy a ingresar a la lista con eso deberíamos tener el ingreso pero yo no puedo comprobarlo porque no puedo ver los jugadores que están en la lista por lo tanto podríamos utilizar la cola implementar otro método que me retorne lo vamos a llamar imprimir, para ver

el contenido en el fondo esto va a recorrer la cola generando un auxiliar mientras no esté vacía, mientras no esté vacía voy a sacar un jugador, jugador j igual a `this.pop` saco un jugador y voy a juntar todos los nombres en un String, voy a concatenar jugador `.getnombre()` y un salto de línea para que me quede más ordenado, entonces voy a juntar el nombre del jugador y voy a guardarlo en la cola auxiliar `aux.push(j)` entonces saco un jugador imprimo el nombre y lo guardo en el auxiliar, pues una vez que termino devolverlos mientras que el auxiliar no este vacío `this.push(aux.pop())` y hay devuelvo todos al auxiliar finalmente retorno el String , entonces hay deberíamos tener la impresión y ese String que me retorne lo voy a imprimir acá que sería `lista.imprimir` veamos si nos está funcionando el ingreso, ingrese la cantidad de jugadores nueve y hay tenemos del J1 al J9 entonces hay ya tenemos el ingreso de los jugadores bien, una buena forma de asegurarse que eso esté funcionando bien es imprimir varias veces porque puede que yo imprima uno pero la cola no quede bien si yo lo imprimo varias veces significa que quedo bien la cola después de realizar la operación entonces imprimimos del uno al nueve, del uno al nueve y del uno al nueve con eso nos aseguramos que el proceso de impresión está bien está imprimiendo y está dejando la cola como estaba originalmente, entonces ya tenemos los participantes ingresados y ahora lo que tiene que pasar es empezar el torneo, como empieza cómo funciona el torneo dijimos que salían dos personas , y cada persona hacia cachipun el que ganaba se devolvía a la cola y sumaba un punto y el que perdía quedaba eliminado por lo tanto yo tengo que seguir mientras tenga elementos mientras `lista.isEmpty()` mientras la lista no está vacía tengo que sacar un jugador `j1` igual a `lista.pop()` tengo un jugador tengo que sacar un segundo jugador `j2` `lista.pop()` y compilan pero en algún momento va a quedar un solo jugador que va a ser el ganador, entonces antes de sacar el segundo jugador tengo que asegurarme que puedo sacar un segundo jugador porque puede que haya una sola persona en la lista, en el caso ese sería el que va a ganar, por lo tanto una vez que saco un jugador pregunto si la lista quedo vacía si luego de sacar un jugador la lista quedo vacía significa que el que saque era el último que quedaba cierto, por lo tanto voy a generar otro objeto jugador le voy a poner ganador lo voy a declarar como null al principio y si la lista está vacía el ganador va a ser `j1`, que fue el que alcance a sacar, por lo tanto si es que la lista quedo vacía después de sacar un jugador el que saque era el único que quedaba si es que la lista no estaba vacía si quedan más jugadores puedo sacar el segundo, entonces puedo sacar el segundo jugador y hay vamos a tener ya los dos jugadores para que compitan dijimos que la clase de jugador necesitaba un método que se llamaba cachipun que era el que me entregaba en el fondo el que iba a jugar, entonces a la clase jugador le vamos le vamos a implementar el método `public` retorna un integer cachipun y acá yo tengo que retornar un número, para retornar un numero tengo que retornar un entero el comando `Math.random()`, eso me genera un numero entre 0 y 1 sin llegar hasta el 1 entre 0 y 0.9999 entonces con eso lo multiplico por la cantidad de números que quiero generar tres números y le sumo uno para que parta desde uno y no desde cero, entonces me genera un numero entre 0 y 0.9, eso yo lo multiplico por 3 y me ha generado un numero entre 0 y 2.7

como va a estar en enteros me va a generar entre 0 y 2 como a mí me sirve entre 1 y 3 le sumo 1, él va a generar un numero entre 1 y 3 ustedes basta con que en la prueba pongan por ejemplo random 1 , 3, y con eso se da a entender que quieren generar un numero aleatorio entre 1 y 3 no me interesa que sepan el código del random, entonces yo voy a retornar un número aleatorio entre 1 y 3 de vuelta al torneo voy a sacar el cachipun 1 que va a ser jugador 1 . cachipun y voy a sacar el cahipun 2 que va a ser el jugador 2 punto cachipun hay voy a tener las opciones de los jugadores, 1 piedra 2 papel 3 tijera entonces yo tengo que seguir jugando mientras sigan sacando lo mismo si sacan lo mismo tienen que volver a sacar cierto, entonces hacemos un ciclo mientras que c1 sea igual que c2, que pasa mientras c1 sea igual que c2 tengo que volver a jugar entonces si es que son iguales los dos hacen cachipun de nuevo si es que vuelven a ser iguales los dos hacen cachipun de nuevo y hay los voy a tener hasta que sean distintos cuando sean distintos uno gano, entonces tengo que ver quien gano, voy a asumir que el ganador es el 1, ganador ya tengo la variable ganador, gano 1 entonces yo voy a decir que el 1 gano y tengo que cambiar las condiciones en que en realidad no puede ganar y cuando no gana si el 1 hizo piedra y el dos hizo, si el uno hizo piedra y el dos hizo papel gano el 2, si es que el 1 hizo papel y el 2 hizo tijera gano el 2 si es que el 1 hizo tijera y el 2 hizo piedra gano el 2, entonces hay tengo todas las combinaciones para ver quién gana, parto asumiendo que gano el 1 y si es que no es así es porque el 1 hizo piedra y el otro papel gano el 2 si es que el 1 hizo papel y el otro tijera gano el 2 y si 1 tiene tijera el 2 hizo piedra por lo tanto cambio las condiciones para ganar, otra alternativa es que hagan todos los if entonces si el 1 es papel y el 2 es tijera gano el 1, si el 1 es papel y el otro es piedra gano el 2, y hay van haciendo todas las combinaciones hay voy a tener cual gano, si gano el 1 que tengo que hacer con el ganador, como! Tengo que devolverlo a la pila cierto, antes de devolverlo a la pila tengo que sumarle un punto entonces vamos a decir j1.sumarPuntaje, ya voy a hacer ese método, jugador public void sumarpuntaje y sumar puntaje lo que hace es this.puntaje++ suma 1 al punto entonces hay suma un punto lego una vez que ya sumo un punto tengo que devolverlo a la lista para que pueda seguir participando lista.push jugador 1 y luego tengo que mostrar que gano el jugador 1 entonces imprimir por pantalla j1.getNombre WIN, hay lo que podemos hacer una vez que tenemos los dos jugadores es mostrar quien está jugando j1.getNombre VS j2.getNombre, entonces una vez que tengo los dos jugadores imprimo quienes están compitiendo voy a sacar eso para que quede en la misma línea y después imprimo quien es el que gano, que pasa si no gano el 1 hay que hacer lo mismo para el 2, al jugador 2 le subo el puntaje a la lista devuelvo el jugador 2 e imprimo que el jugador 2 gano y hay deberíamos tener este ciclo hasta que en algún momento la lista va a quedar vacía porque yo voy a, va a quedar un solo jugador voy a sacar ese jugador, va a quedar como el ganador me voy a saltar todo este else me voy a devolver y hay la lista va a quedar vacía por lo tanto al final mostramos el campeón es ganador.getNombre() y voy a imprimir el puntaje ganador.getPuntaje(), entonces hay vamos a tener los datos del ganador probémoslo ingrese la cantidad de jugadores nueve y hay tenemos jugador 1 vs

jugador 2 gano el 1 el 3 vs el 4 gano el 3, 5 vs 6 gano 5, 7 vs 8 gano 8, 9 vs 1, hay se fijan que el 1 que había ganado se devolvió a la lista volvió a competir con el 9 gano el 9, el 3 con el 5 que son los que habían ganado hay gano el 3, el 8 con el 9 son los que habían ganado hay gano el 9 el 3 con el 9 gano el 3el campeón es el 3 con 3 puntos quien gano hay gano hay y gano la final, entonces hay tenemos el proceso completo, tenemos que implementar la clase jugador con los atributos que vayamos a utilizar de acá había que implementar el método cachipun, una vez que tenemos la clase jugado con los get y los set en el main generamos la lista que va a ser la cola ingresamos la cantidad de jugadores, pueden pedir los nombres por pantalla o pueden generarlos secuencialmente, como quieran se carga se crea un jugador y el jugador se guarda en la lista que sería la cola de jugadores una vez que cargamos todos los jugadores defino un jugador que va a ser el ganador que parte como null que va a ser el ganador del torneo saco un jugador y pregunto si la lista está vacía si al sacar un jugador y la lista está vacía es porque queda un solo jugador en el listado por lo tanto ese jugador va a ser el ganador si es que la lista no está vacía saco un segundo jugador para que compita, una vez que compiten mostramos quienes están compitiendo, los dos hacen cachipun, siguen haciendo cachipun mientras sigan repitiendo los mismos resultados si los dos hacen piedra tienen que repetir si los dos hacen papel tienen que repetir, si los dos hacen tijera tienen que repetir, una vez que sean distintos veo quien gano, define ganador de cachipun, una alternativa es asumir que gano el 1 y después hacer las condiciones para ver cuando gana el 2 otra alternativa es hacer todos los if con las combinaciones o hacer un swith case, con eso miro quien gano después pregunto si gano el 1 al uno le sumo el puntaje que es agregarle un punto por haber ganado devuelvo el jugador 1 a la lista he imprimo quien gano, gano el 1 lo mismo si es que gano el 2 al 2 le sumo un puntaje al 2 lo devuelvo a la lista he imprimo que gano el 2 y eso va a seguir hasta que solo quede 1 entonces de cada pareja uno queda eliminado y el otro se devuelve a la lista y una vez que termino voy a tener el ganador con el dato de la persona que quedo al final y hay muestro los datos el nombre el puntaje y el país si quiero, aquí son todos de Colombia, se entiende el problema ¿alguna duda muchachos?

Tema 6: Tablas de Hash

El profesor: ya se pasó el tema de casos, vimos caminos, recorridos y algoritmos. Nuevo tema Tablas de hash (Profesor presenta tema actual en diapositivas): Almacenamiento de estructura de datos, Representación a través vector, Cuando cambia (valores grandes – tablas hash).

Estudiante hace comentario que aporta a la clase.

Profesor: Se revisan definiciones y conceptos de tablas de Hash.

Profesor hace pregunta a estudiantes sobre usos anteriores en primer corte.

Estudiantes realizan comentarios.

Profesor continúa explicando conceptos de hash (llaves, mundo, problemas, tablas, orden). Realiza una nueva pregunta, un estudiante responde pero la respuesta no es convincente. El docente explica la pregunta realizada previamente apoyándose en las diapositivas.

Estudiante comenta con respecto a explicación modo de pregunta. El docente, empieza explicando nueva diapositiva de tema (función hash). Profesor compara tema de explicación con sucesos de trabajos anteriores y finaliza explicación.

Profesor explica función modulo (actual diapositiva). Profesor pregunta "qué es un módulo".

Estudiantes emiten sus comentarios.

Profesor; continua explicando el tema, propone ejercicio para todos los estudiantes respecto a la diapositiva de explicación.

Estudiantes se agrupan intentan resolver ejercicio contra diapositiva.

Profesor frente a portátil llama a lista. Posteriormente, asesora a cada grupo conformado explicando la forma de solucionar el ejercicio.

Profesor explica en voz alta guía para resolver ejercicio. Algunos estudiantes realizan interrogantes, y el profesor brinda la explicación de manera puntual.

Profesor resuelve pregunta a estudiante mostrando diapositiva, sube telón y explica con marcador, da ejemplos y datos para resolver. Profesor explica generalidades del tema y sobre ejercicios.

Profesor pregunta respecto a la solución del ejercicio (ya todos terminaron)?

Profesor utiliza marcador para anotar en el tablero, como se debe enviar la solución (correo)

Profesor informa fechas de programación.

Transcripciones – docente B

Tema 1: Manejo de Archivos

El docente presenta mediante una serie de Diapositivas el tema a desarrollar durante la sesión de clases.

Inicialmente, explica a nivel conceptual los elementos que hacen parte de un archivo: Registro, campo, dato, etc. Explica sobre los diferentes tipos de archivos y las clases utilizadas en Java para el manejo a nivel de programación de Archivos. Menciona la importancia del uso del bloque **try/catch** para la captura de las excepciones que se pueden presentar durante el tratamiento de archivos. Además, hace énfasis en lo referente a archivos de objetos y las clases utilizadas para el tratamiento de dichos archivos. Explica también sobre el concepto de Serialización de objetos y mediante ejemplos (consignados en las diapositivas) explica el uso de las clases y la forma de escribir / leer sobre los archivos de objetos.

El docente indica que en la plataforma se encuentra una carpeta denominada 07 SourceCode que contiene una serie de ejemplos en el lenguaje de programación Java y que corresponden a los ejemplos explicados plasmados en las diapositivas.

El docente procede a explicar cada línea de código de los ejemplos entregados. Durante el transcurso de la explicación se suscitan algunas preguntas por parte de los estudiantes que son aclaradas una a una.

Posterior a las explicaciones ofrecidas por el docente, se explica el taller correspondiente al tema estudiado. El docente brinda algunas sugerencias para la solución de cada uno de los ítems que conforman el taller propuesto.

Tema 2: Memoria Dinámica

El docente prepara el material y el equipo de trabajo para proyectar las diapositivas correspondientes al tema: Gestión y Administración de Memoria (Dinámica – Estática).

Con base en la primera diapositiva, el docente explica la clasificación existente de Memorias, ofrece algunos ejemplos de cómo es utilizada la memoria del PC cuando se graban archivos o se ejecutan programas. Explica que la memoria del PC es similar a un conjunto de casillas y que cada una de ellas posee información fija y variable. Explica que la información fija, hace referencia a la posición de memoria o dirección y otra información que es variable y que corresponde al dato almacenado en esa posición de memoria.

Explica que ocurre al interior de la celda de memoria cuando se declaran variables, cuando se ejecutan operaciones y qué ocurre en el *heap*. El docente, mediante un gráfico explica paso a paso lo que ocurre en memoria cuando se ejecuta un programa de computador.

EL docente, hace un análisis entre el lenguaje de programación Java y el lenguaje C. Indica que el tratamiento de la memoria por parte de cada uno de estos lenguajes es totalmente diferente. El lenguaje C, hace uso de punteros mientras que en el lenguaje Java ese concepto no existe. Es reemplazado por un concepto denominado Referencias y que implícitamente ocurre el tratamiento de punteros. El docente continua con la explicación del tratamiento de apuntadores basado en el lenguaje C.

El docente pregunta si algún estudiante tiene alguna duda o interrogante, se resuelven algunas inquietudes y continua explicando las operaciones que se pueden realizar mediante punteros. Explica mediante ejemplos consignados en las diapositivas como mediante el lenguaje C, se puede trabajar con punteros y con direcciones de memoria. Además, hace una breve explicación de la sintaxis del lenguaje.

Continúa con la explicación de la administración de Memoria bajo Java, específicamente explica como la JVM hace uso de la memoria del PC. EL docente solicita a los estudiantes descargar del espacio virtual un código de ejemplo para observar algunas características de la memoria del PC. Es un algoritmo desarrollado en Java y al ejecutarse presenta información general del uso de la memoria del PC. El docente explica cada una de las líneas del código y resuelve interrogantes en el transcurso de la explicación.

El docente se pasea en el aula de clases, visitando cada uno de los puestos de trabajo ocupado por cada estudiante. Verifica que efectivamente se esté cumpliendo el objetivo al ejecutarse el programa de computador.

Ahora, explica en qué consiste el GC desde el lenguaje de programación Java. Pregunta con base en algunas diapositivas sobre algunas líneas de código. Para finalizar la clase, ofrece las indicaciones correspondientes al taller propuesto sobre el tema estudiado.

Tema 3: Fundamentos de Estructuras de datos

El docente ingresa al aula de clases y se dispone a preparar el material de estudio con el cual se desarrollará la clase del día.

El docente, pregunta a los estudiantes si existen dudas o inquietudes con respecto al taller propuesto de la semana anterior. Algunos estudiantes lanzan interrogantes

que son resueltos por el docente. El docente hace uso del tablero anotando algunas precisiones que se deben tener en cuenta para el desarrollo del taller.

El docente, inicia la sesión de clases haciendo una introducción sobre el buen uso y las buenas prácticas que deben tenerse en cuenta al momento de diseñar un programa de computador. Explica la importancia que ha de tenerse en cuenta al momento de seleccionar la estructura acorde e ideal para un mejor desempeño del programa a desarrollar o construir.

EL docente, explica a nivel general el tema de las Estructuras de datos. Explica las diferentes estructuras existentes y como son utilizadas en los lenguajes de programación Java y C. Explica que hoy día las estructuras están implementadas por los lenguajes de programación e inicia la explicación de los tipos de datos utilizados por el Lenguaje de programación Java.

Docente: En java, cada uno de los tipos de datos definidos por el lenguaje posee una estructura única. Por ejemplo: el tipo de dato **Integer** posee un tamaño definido. En caso de que alguna operación matemática arroje un resultado cuyo valor no se encuentre en el rango soportado por este tipo de dato, automáticamente se presenta un error. Por consiguiente, es muy importante tener en cuenta el tamaño que ocupa cada uno de los tipos de datos definidos en el lenguaje. Este tipo de estructuras se denominan: Estáticas. Tenemos entonces por definición, que los arrays (vectores y matrices) se encuentran categorizados como estructuras estáticas.

Por otro lado, existe además un tipo de estructura dinámica y en esta categoría corresponden todas aquellas estructuras en las cuales NO existe un tamaño predefinido. En ocasiones, no se tiene un tamaño definido y por tanto se hace necesario utilizar estructuras de tipo dinámicos. Por ejemplo: el uso de pilas, colas, listas enlazadas simples, listas enlazadas dobles, entre otros. El tamaño de una estructura dinámica dependerá específicamente del tamaño de la memoria del PC. En la medida que crece el tamaño de la estructura, dependerá en gran medida de la capacidad de memoria disponible por el PC. Además, las estructuras de datos dinámicas tienen la característica que la información NO necesariamente se almacena de forma continua. Es allí, en donde Java utiliza el tipo de dato **Referencia**.

El docente, continúa la explicación introductoria apoyándose en las diapositivas en las cuales se presentan gráficos alusivos al tema. Informa que explicará brevemente cada una de las estructuras estáticas en Java a manera de reforzar cada uno de los conceptos puesto que son temas pertenecientes al primer y segundo curso de informática.

Docente: En java, existen estructuras definidas como los tipos de datos. Cada tipo de dato posee un tamaño definido y por consiguiente NO se pueden violar las

restricciones impuestas por el lenguaje. Por ejemplo: cuando se define un array unidimensional, cada una de las posiciones de memoria se encuentran de manera continua y se reserva en memoria el tamaño predefinido en el array. Lo mismo ocurre para los arrays bidimensionales.

Cuando se trabajan estructuras de datos dinámicas, las posiciones de memoria NO necesariamente son continuas. Además, cada posición de memoria posee un identificador único y es por medio de este identificador que se accede al dato en memoria. El identificador –llamado puntero en el lenguaje de programación C, C++- en el caso de Java, es un valor sobre el cual no se permite realizar operación alguna. En cambio, en el lenguaje C por ejemplo uno podría sumar un valor numérico a un apuntador.

Las estructuras de datos como por ejemplo; las pilas, son muy utilizadas para el uso de los navegadores web. Cuando un usuario accede a varias url's, el PC guarda dicha información en una estructura de Pila. De tal forma, que el usuario bien puede recuperar una url anterior a la que actualmente está visitando.

Lo mismo ocurre con la Recursividad. El llamado recursivo, internamente lo que genera es una pila de llamadas de tipo LIFO. Otro ejemplo del uso de las estructuras dinámicas es el utilizado por los Servidores de Impresión. Internamente mantienen una cola de archivos que van siendo impresos en la medida que se van eliminando los primeros. Esta es una estructura de tipo FIFO.

EL docente, termina su sesión de clases indicando la existencia de un taller referente al tema expuesto. Dicho taller tiene un plazo de desarrollo de ocho (8) días y el resultado o informe final debe subirse a la plataforma. El docente presenta los enunciados del taller y explica brevemente en que consiste la solución del mismo.

Por último, el docente atiende una serie de preguntas sobre el desarrollo del taller.

Tema 4: Listas enlazadas

El docente prepara el material a utilizar para la sesión de clases referente al tema de Listas Enlazadas. Antes de iniciar, pregunta a nivel general sobre el desarrollo de taller anterior. Resuelve algunas inquietudes.

El docente, explica la solución del taller de quince días (15) atrás. Realiza comentarios de las situaciones que NO se deben presentar y vuelve a reiterar la forma de entrega de la solución del taller.

Una vez colocadas las diapositivas, el docente inicia la sesión de clases haciendo un breve resumen de la clase anterior. Explica con base en una diapositiva, la

clasificación existente en las Listas Enlazadas. Explica conceptualmente sobre Listas enlazadas Simples, listas enlazadas dobles, listas circulares simples, listas circulares dobles.

El docente, explica que en cada una de estas estructuras se presentan varias situaciones:

- Inserción de un nodo (se puede insertar por el inicio de la lista, después o antes de otro nodo dado como referencia, o por el final).
- Eliminación de un elemento (se puede eliminar por el inicio de la lista, después o antes de un nodo o al final de la lista)
- Modificar la información contenida en un nodo (el nodo puede estar al inicio de la lista, al final de la lista o antes o después de un nodo dado como referencia).
- Recorrido de la lista (dependiendo del tipo de lista, así se implementa el recorrido de la misma).

El docente inicia en primer lugar con el tema de **listas enlazadas simples**. Mediante una diapositiva presenta gráficamente el tema, explicando inicialmente el concepto de nodo. Un nodo, está compuesto por dos partes: una referencia y una parte denominada dato o información. Cada nodo, tiene una referencia que indica la posición de memoria en donde se encuentra ubicado el siguiente nodo. Además, el nodo tiene también información que puede ser de cualquier tipo. Las listas enlazadas simples tienen como característica que el recorrido solo se hace en una dirección. En una lista enlazada simple, el último nodo se identifica porque su referencia, NO apunta a ningún otro nodo. Se dice entonces que apunta a **NULL**.

A continuación, el docente explica los diferentes algoritmos utilizados para el manejo de listas enlazadas simples. Los algoritmos son explicados a nivel de pseudocódigo. Y en lo referente al recorrido, explica que solo se hace en una dirección y siempre se realiza por el inicio de la lista.

En el caso de las **listas enlazadas dobles**, cada nodo posee dos referencias: una a la izquierda y otra a la derecha. La referencia de la izquierda, siempre apunta al nodo anterior. En el caso del primer nodo, apuntará a NULL. La referencia ubicada a la derecha, apunta al nodo de la derecha. Si las referencias apuntaran a NULL, indica que la lista solo posee un nodo.

El recorrido en este tipo de estructura, se puede hacer en ambas direcciones. En ambos casos, se notará que el último nodo siempre apuntará a NULL.

Existe además de las anteriores estructuras, las **listas circulares simples**. Este tipo de estructura es similar a las primeras, pero con la diferencia de que el último nodo apunta o tiene una referencia al primer nodo. Hay que tener en cuenta los casos de inserción, borrado, modificación y recorrido de este tipo de lista. Siempre hay que tomar un nodo como referencia para poder llevar a cabo las diferentes operaciones. Por ejemplo: para recorrer la lista circular simple, se toma un nodo como referencia y cada vez que el apuntador se mueva al siguiente nodo se debe verificar si es o no el nodo referencia. El recorrido en esta estructura es en una sola dirección.

Las **listas circulares dobles**; de forma similar a las listas enlazadas dobles, poseen dos apuntadores o referencias. De tal forma que se puede llevar a cabo el recorrido en ambos sentidos, se deberá tener en cuenta un nodo referencia para poder ejecutar las diferentes operaciones en este tipo de estructura.

Para cualquiera de las estructuras estudiadas, hay que tener en cuenta antes de llevar a cabo cualquier operación, el estado de la lista. Es decir; no se podría intentar eliminar un elemento de la lista si antes NO se verifica el estado de la lista, por ejemplo que no esté vacía. Y además, dependiendo si la lista tiene uno o más nodos así se llevará a cabo la operación sobre la lista.

Al final de la sección de clases, el docente explica el taller correspondiente y adicionalmente, presenta las indicaciones sobre el código fuente asociado a cada una de las estructuras estudiadas.

Tema 5: Stacks

Después de realizar la preparación del material de apoyo, el docente determina quince minutos para aclaraciones sobre el taller de la semana anterior sobre el tema de listas enlazadas. Surgen varios interrogantes por los estudiantes los cuales son resueltos uno a uno. Sin embargo; siguen algunas dudas y el docente decide explicar en el tablero haciendo uso de marcadores. Elabora unas gráficas y explica las partes en las cuales se compone el nodo. Explica el concepto de listas enlazadas simples, dobles, listas circulares simples, dobles.

El docente resuelve un ejercicio para cada uno de los casos estudiados y con base en dicho ejercicio resuelve algunos interrogantes.

Terminada la explicación a manera de resumen de la clase anterior, procede a explicar la nueva estructura de datos denominada **Pilas o Stacks**.

El docente inicia explicando que son estructuras tipo FIFO (First in First Out). En donde el primer elemento que ingresa a la Pila, será el último en salir.

El docente manifiesta que este tipo de estructuras es utilizada en los navegadores web, cuando el usuario ingresa a una serie de URL's, la información va quedando almacenada en una estructura de Pila, de tal forma que el usuario puede volver a seleccionar una url que haya visitado anteriormente.

En este tipo de estructura, se debe tener en cuenta el manejo de la variable **Tope** y la variable **Max**. La primera, indica la posición en donde se encuentra almacenado el primer elemento de la Pila (es el elemento que ingresó último). Y la segunda hace referencia al tamaño máximo de la pila. Este tipo de estructura NO gestiona la memoria del PC de manera óptima puesto que es posible que al definir un tamaño máximo de la Pila, no toda la estructura sea utilizada. Por ejemplo: si se define un tamaño máximo de 10 elementos y solo se almacenan dos o tres elementos, el resto de memoria se “desperdicia”.

El docente continua su explicación basado en las diapositivas en donde por medio de gráficas bastante llamativas explica los algoritmos utilizados para la administración de este tipo de estructura. Explica que es muy importante tener en cuenta el estado de la Pila cuando se intenta eliminar un elemento o ingresar. Por ejemplo, no se podría ingresar un elemento a la Pila cuando esta, esté totalmente llena. Es decir; la variable **tope** es igual a la variable **max**.

De igual forma, no se podría eliminar un elemento de la Pila si esta, se encuentra vacía. Adicionalmente, el docente manifiesta que en el Aula virtual se encuentran algunos algoritmos desarrollados en el lenguaje de programación Java que hacen referencia al tema tratado. Se trata de ejercicios que permiten el desarrollo de las diferentes operaciones utilizadas para el manejo de este tipo de estructuras.

Además, explica el nuevo taller referente al tema tratado. Explica cada uno de los enunciados del taller y brinda algunos tips para su correcto desarrollo.