UAEM | Universidad Autónoma del Estado de México

Estructura de Datos

Centro Universitario UAEM Valle de México

Centro Universitario UAEM Valle de México

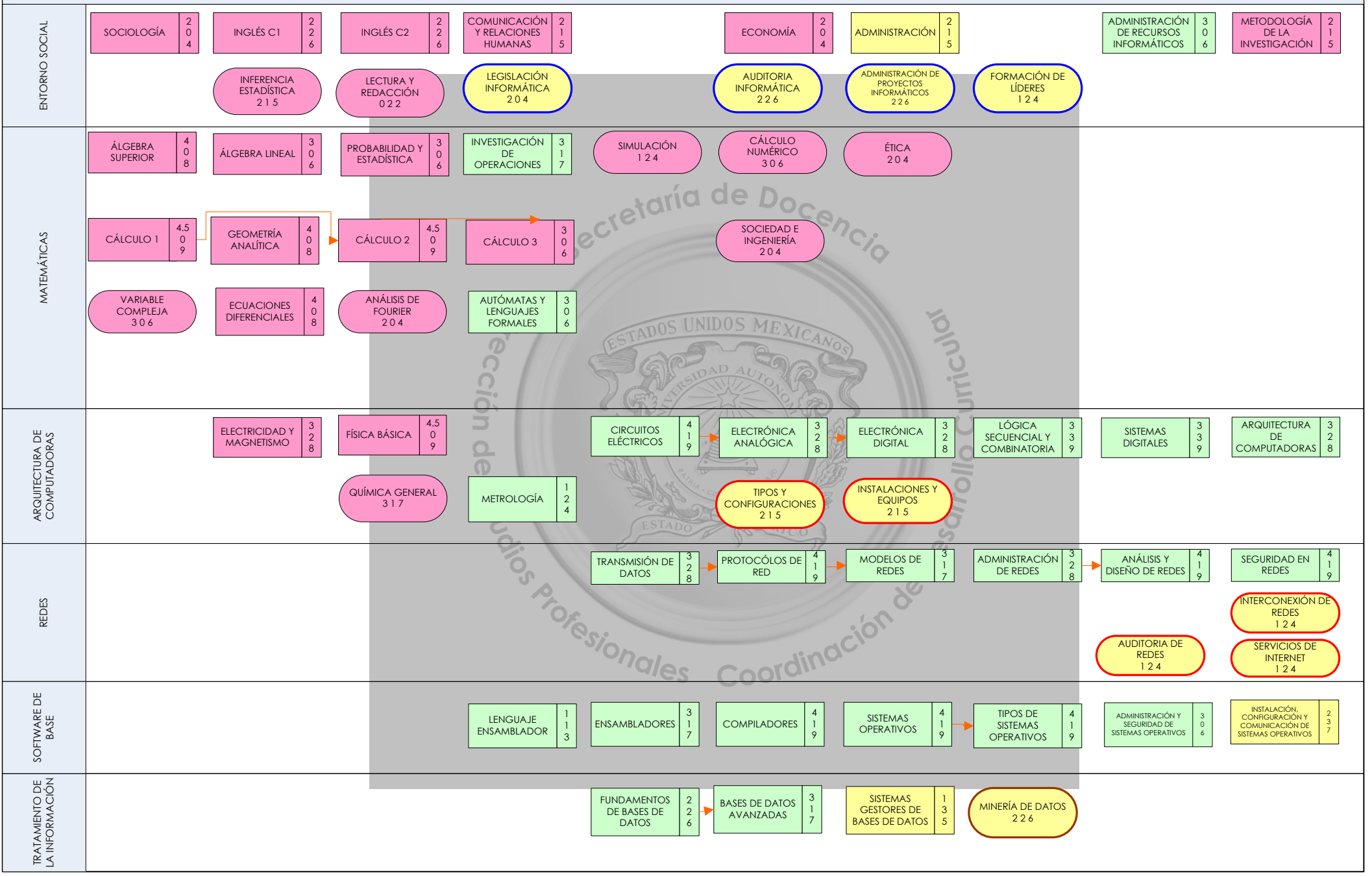
Unidad II. Estructuras de Datos Lineales - Pilas

Licenciatura en Ingeniería en Computación
Fecha de cración – julio 2016 – semestre impartido 2016B

Ph. D. Victor Manuel Landassuri Moreno
vmlandassurim@uaemex.mx
landassuri@gmail.com

MAPA CURRICULAR DE LA LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN 2004

CRÉDITOS TOTALES: 430-450



MAPA CURRICULAR DE LA LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN 2004

CRÉDITOS TOTALES: 430 - 450

PROGRAMACIÓN E INGENIERÍA DE SOFTWARE

- PROGRAMACIÓN ESTRUCTURADA 3 3 9 → PROGRAMACIÓN AVANZADA 3 3 9
- ESTRUCTURAS DE DATOS 3 2 8 → ORGANIZACIÓN DE ARCHIVOS 3 2 8
- PROGRAMACIÓN ORIENTADA A OBJETOS 3 2 8
- PROGRAMACIÓN PARALELA Y DISTRIBUIDA 3 2 8
- TEORÍA DE SISTEMAS 3 0 6
- ANÁLISIS DE SISTEMAS 3 2 8 → DISEÑO DE SISTEMAS 3 2 8
- ESTÁNDARES DE CALIDAD 2 2 6
- MÉTRICAS DE SOFTWARE 2 2 6
- ANÁLISIS DE LENGUAJES DE PROGRAMACIÓN 2 2 6
- LENGUAJE DE PROGRAMACIÓN ESTRUCTURADO 2 1 5
- LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS 2 1 5
- LENGUAJE DE PROGRAMACIÓN VISUAL 2 1 5

INTERACCIÓN HOMBRE – MÁQUINA

- LÓGICA 2 1 5
- MULTIMEDIA 2 2 6
- TRATAMIENTO DE IMÁGENES 2 1 5
- VISIÓN ARTIFICIAL 2 0 4
- ALGORITMOS GENÉTICOS 1 2 4
- SISTEMAS EXPERTOS 2 1 5
- GRAFICACIÓN 2 1 5
- INTERACCIÓN HOMBRE-MAQUINA 2 1 5
- DESARROLLO MULTIMEDIA 1 2 4
- FUNDAMENTOS DE ROBÓTICA 2 1 5
- ROBÓTICA AVANZADA 1 2 4
- REDES NEURONALES 1 2 4

SIMBOLOGÍA

8 ÁREA S CURRICULARES

HT Horas Teóricas
HP Horas Prácticas
CR Créditos

12 LÍNEAS DE SERIACIÓN

NÚCLEO BÁSICO OBLIGATORIAS CURSAR Y ACREDITAR16 UA	49.5 HT 8 HP 107 CR
NÚCLEO SUSTANTIVO OBLIGATORIAS CURSAR Y ACREDITAR 39 UA	113 HT 56 HP 282 CR
NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 3 UA	5 HT 7 HP 17 CR

NÚCLEO BÁSICO OPTATIVAS: ACREDITAR 1 (CUALQUIERA) ó 2 UA (1 DE 2 CR +1 DE 5 CR ó 4CR) PARA CUBRIR DE 2 A 7 CRÉDITOS

NÚCLEO INTEGRAL OPTATIVAS: LÍNEA DE ACENTUACIÓN ADMINISTRACIÓN DE PROYECTOS INFORMÁTICOS

NÚCLEO INTEGRAL OPTATIVAS: LÍNEA DE ACENTUACIÓN REDES Y COMUNICACIONES

NÚCLEO INTEGRAL OPTATIVAS: LÍNEA DE ACENTUACIÓN INTERACCIÓN HOMBRE-MAQUINA E INTELIGENCIA COMPUTACIONAL

NÚCLEO INTEGRAL OPTATIVAS: LÍNEA DE ACENTUACIÓN DESARROLLO DE SOFTWARE DE APLICACIÓN

ACREDITAR LAS UA DE LA LÍNEA DE ACENTUACIÓN ELEGIDA PARA CUBRIR DE 22 A 37 CRÉDITOS.

TOTAL DEL NÚCLEO BÁSICO 17 A 18 UA PARA CUBRIR 109 A 114 CRÉDITOS

TOTAL DEL NÚCLEO SUSTANTIVO 39 UA PARA CUBRIR 282 CRÉDITOS

TOTAL DEL NÚCLEO INTEGRAL 8 A 10 PARA CUBRIR DE 39 A 54 CRÉDITOS

TOTAL DEL PLAN DE ESTUDIOS

UA OBLIGATORIAS	58
UA OPTATIVAS	6 A 9
UA A ACREDITAR	64 A 67
CRÉDITOS	430-450

Dirección de Estudios Profesionales Coordinación de Desarrollo Curricular

Índice de Contenidos

PROGRAMA DE ESTUDIO POR COMPETENCIAS

ESTRUCTURAS DE DATOS

I. IDENTIFICACIÓN DEL CURSO

<table>
<tr><td colspan="7">ESPACIO EDUCATIVO: Facultad de Ingeniería</td></tr>
<tr><td colspan="3">LICENCIATURA: Ingeniería en Computación

AÑO DE APROBACIÓN POR EL CONSEJO UNIVERSITARIO:</td><td colspan="4">ÁREA DE DOCENCIA: Programación e Ingeniería de Software</td></tr>
<tr><td colspan="2">APROBACIÓN POR LOS HH. CONSEJOS ACADÉMICO Y DE GOBIERNO</td><td>FECHA:</td><td colspan="2">PROGRAMA ELABORADO POR:
Mtro. Eduardo Trujillo Flores
Mtra. Mireya Salgado Gallegos</td><td colspan="2">PROGRAMA REVISADO POR:
Integrantes de la Academia de Programación e Ingeniería de Software

Centro Universitario Valle de Chalco</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="2">FECHA DE ELABORACIÓN :
Mayo de 2007</td><td colspan="2">FECHA DE REVISIÓN :
Mayo 2011</td></tr>
<tr><td>CLAVE</td><td>HORAS DE TEORÍA</td><td>HORAS DE PRÁCTICA</td><td>TOTAL DE HORAS</td><td>CRÉDITOS</td><td>TIPO DE CURSO</td><td>NÚCLEO DE FORMACIÓN</td></tr>
<tr><td>L41054</td><td>3</td><td>2</td><td>5</td><td>8</td><td>Curso</td><td>Sustantivo</td></tr>
<tr><td colspan="4">UNIDAD DE APRENDIZAJE ANTECEDENTE
Programación estructurada</td><td colspan="3">UNIDAD DE APRENDIZAJE CONSECUENTE
Organización de Archivos</td></tr>
<tr><td colspan="7">PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE:
Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)</td></tr>
</table>

Descripción de la unidad de aprendizaje

Identificación del Curso

Licenciatura en Ingeniería en Computación

Horas de Teoría: 3 hrs.

Horas de Práctica: 2 hrs.

Créditos: 8

Unidad de Aprendizaje Antecedente:
Programación Estructurada

Unidad de Aprendizaje Consecuente:
Organización de Archivos

Presentación

El estudio de las estructuras de datos, sin duda es uno de los más importantes dentro de las carreras relacionadas con la Computación, ya que en el conocimiento eficiente de las estructuras de datos suele ser imprescindible en la formación de los alumnos debido a la trascendencia que un aprendizaje teórico-práctico de las misma supondrá en su carrera.

Lineamientos

Del Profesor

- Cumplir en tiempo y contenido el programa de la unidad de aprendizaje
- Establecer tolerancia para el inicio de clases
- Proponer y respetar formas de evaluación
- Generar en sus alumnos una visión integradora de la unidad de aprendizaje
- Respetar el número de horas teóricas y practicas de la unidad de aprendizaje
- Cada sesión deberá concluir con una serie de ejercicios de repaso que permitirán reafirmar los conocimientos del curso
- Preferentemente las fechas de exámenes y entrega del proyecto final deberán establecerse desde el principio

Del Alumno

- Contar con el 80% de asistencia para presentar examen ordinario
- Contar con el 60% de asistencia para presentar examen extraordinario
- Contar con el 30% de asistencia para presentar examen a titulo de suficiencia
- Utilizar un lenguaje estructurado para la elaboración de programas
- Tener sentido de responsabilidad en los trabajos clase y extraclase
- Entregar en tiempo y forma los trabajos clase y extraclase
- Tener sentido de interrogación y participación dentro del salón de clases
- El alumno deberá entregar todas y cada una de las series de ejercicios.

Propósito

Que el alumno identifique las herramientas teóricas fundamentales para la representación y manipulación de información en la computadora haciendo énfasis en el tipo de datos dinámicos.

Competencias genéricas

Desarrollar programas analizando y diseñando soluciones a problemas reales del entorno a través del uso de herramientas lógicas

Ámbito de desempeño

Empresas públicas y privadas del sector industrial, educativo, comercial y de servicios.

Estructura

Unidad 1.	• Reconocer y manejar las variables dinámicas
Unidad 2.	• Aplicar las principales estructuras de datos lineales
Unidad 3.	• Aplicar la estructura de datos de árbol
Unidad 4.	• Aplicar la estructura de datos de grafo

Estructura por unidad

- **Unidad 1.** Reconocer y manejar las variables dinámicas
  - *Estructuras de datos*: Definición, Tipos.
  - *Abstracción*: Definición. TAD.
  - *Variables Dinámicas*: Apuntadores, Operaciones básicas.

Estructura por unidad

- **Unidad 2.** Aplicar las principales estructuras de datos lineales
  - *Pilas*: Representación. Operaciones (Inserción, Eliminación, Pila Llena, Pila Vacía). Tratamiento de expresiones aritméticas: notación infija, prefija, posfija. aplicaciones
  - *Colas*: Representación. Operaciones (Inserción, Eliminación, Cola Llena, Cola Vacía). Cola circular. Aplicaciones.
  - *Listas*. Representación. Operaciones (Inserción, Eliminación, Recorrido, Búsqueda). Listas doblemente ligadas. Aplicaciones

Estructura por unidad

- **Unidad 3.** Aplicar la estructura de datos de árbol
  - *Recursividad Directa*: Definición. Funcionamiento
  - *Árboles*: Usos. Características.
  - *Árboles binarios de Expresiones*: Características. Evaluación de expresiones aritméticas.
  - *Árboles Binarios de Búsqueda*: Características. Operaciones (Inserción, eliminación, búsqueda)

Estructura por unidad

- **Unidad 4.** Aplicar la estructura de datos de grafo
  - *Grafos*: Características. Tipos. Representación y construcción. Operaciones Librería estándar, interfaz con el sistema operativo
  - *Grafos Dirigidos*: Algoritmos para la obtención del camino más corto.
  - *Grafos No dirigidos*: Algoritmos para la obtención de costo mínimo

Procedimientos de Evaluación

La calificación ordinaria se obtiene como la suma del Trabajo semestral y el Proyecto final.
Para tener derecho a presentar el Proyecto final se debe obtener un promedio mínimo de 6.0/10.0 en el trabajo semestral.
Trabajo semestral:

- Programas producto y actividades clase y extra clase **30%**
- Exámenes parciales escritos **20%**
- Proyecto Final **50%**

Se podrá exentar el Proyecto final siempre y cuando se tenga un mínimo de 80% de asistencias, se aprueben los exámenes parciales y el promedio del Trabajo semestral sea de al menos 8.0/10.0
Extraordinaria y a Título de Suficiencia:

- Examen escrito **60%**
- Proyecto **40%**

Unidad 2.

Estructuras de Datos Lineales - Pilas

Contenido

- Breve repaso de arreglos y listas
- Pilas
- Aplicaciones
- Implementación
  - Pilas con arreglos estáticos
  - Pilas con arreglos dinámicos
  - Pilas con listas enlazadas
- Ejercicios

Arreglos

- Los arreglos tienen el operador de acceso aleatorio

  [ ]

- Lo que significa que podemos recuperar o modificar cualquier elemento de el en un solo acceso
- O bien, sin recorrer todos los elementos anteriores al elemento de interés
  - Cada elemento es accesado directamente en un tiempo constante

Arreglos

- Un ejemplo común de acceso aleatorio es un libro
  - Cada página de un libro es independiente de la otra

- Así, el acceso aleatorio es importante en diversos algoritmos, por ejemplo en
  - La búsqueda binaria

Listas ligadas

- En otro ejemplo
  - Las listas ligadas son otro tipo de estructura de datos
  - Cada elemento puede ser accesado en un orden determinado
  - Es decir, quedan bien para representar:
    - un royo de papel
    - una cinta
  - Todos los elementos anteriores tienen que ser recorridos para llegar al elemento de interes

Listas ligadas

- Este juego de diapositivas contiene un caso de estructuras de datos secuenciales.
- Llamadas de acceso limitado
- Para nuestro caso, veremos la estructura de datos tipo Pila

Pilas

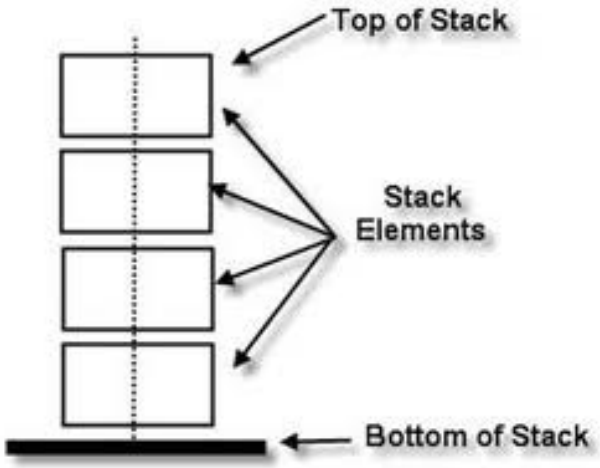
- Es un contenedor de objetos
- Insertados y removidos de acuerdo
  - Último en llegar – Primero en salir
  - LIFO (Least input – First Output)
- En la cabeza (parte más alta) de la pila solo se permiten dos operaciones
  - Push
  - Pop

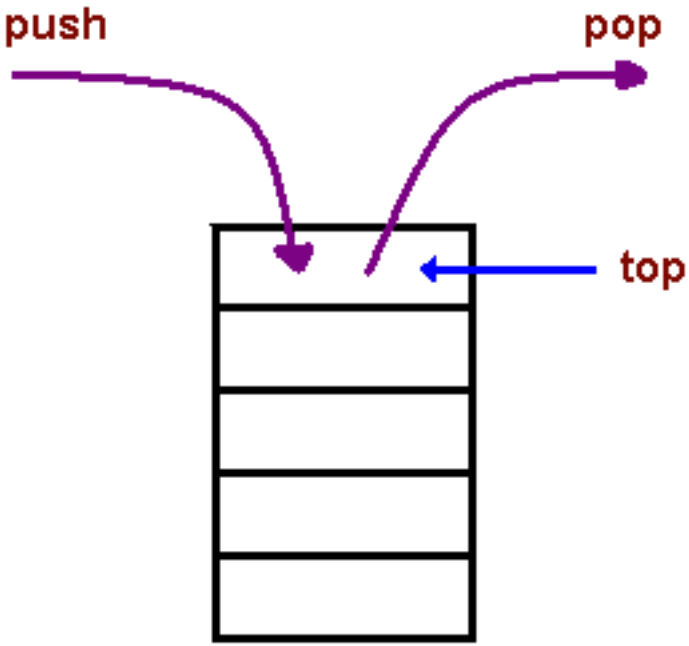
Pilas

- Push
  - Inserta elemento en el tope de la pila
- Pop
  - Remueve un elemento del tope de la pila
- Ejemplo:
  - Libros apilados, o uno encima del otro
  - Puedes agregar un libro encima, y retirar uno de encima

Pilas

Pila (*Stack*)





Pilas

- Una pila es una estructura de datos recursiva
  - La pila esta vacía o
  - Consiste de un tope y el resto de la pila
- Aplicaciones
  - Reservar una palabra
  - Operación deshacer
  - Backtraking
  - Procesamiento de lenguaje, ...

Aplicaciones

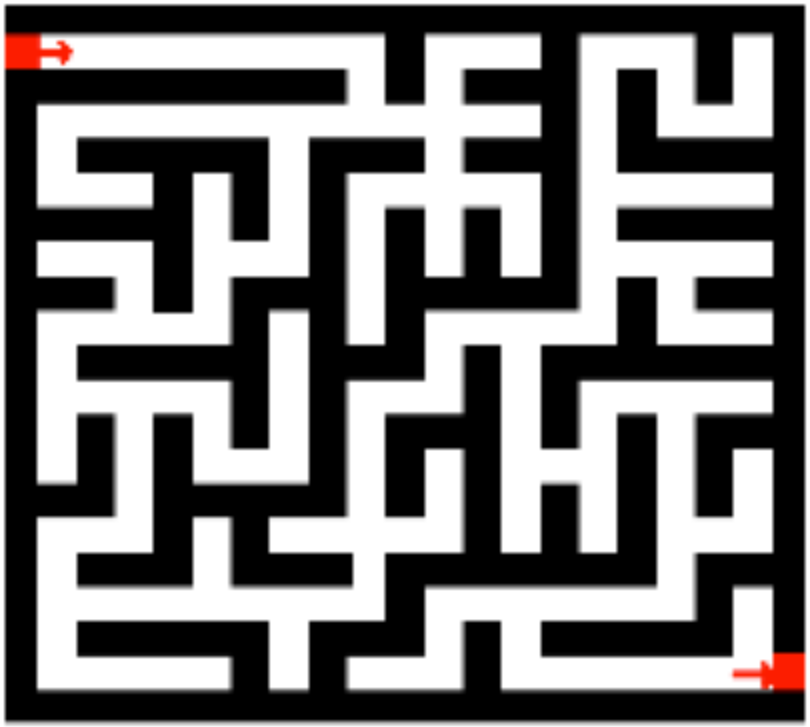
- Reservar una palabra:
  - Insertas una palabra en la pila – letra por letra
  - Luego las sacas según sea necesario
- Deshacer (undo):
  - Se mantiene un historial de todos los cambios en el texto
  - Cada cambio se apila
  - Por ejemplo en MS Word
  - Si quieres regresar al estado anterior, ¿te imaginaras que pasa es esa estructura?

Backtraking

- Aplicación en los laberintos
  - ¿Como encuentras un camino del inicio al final?

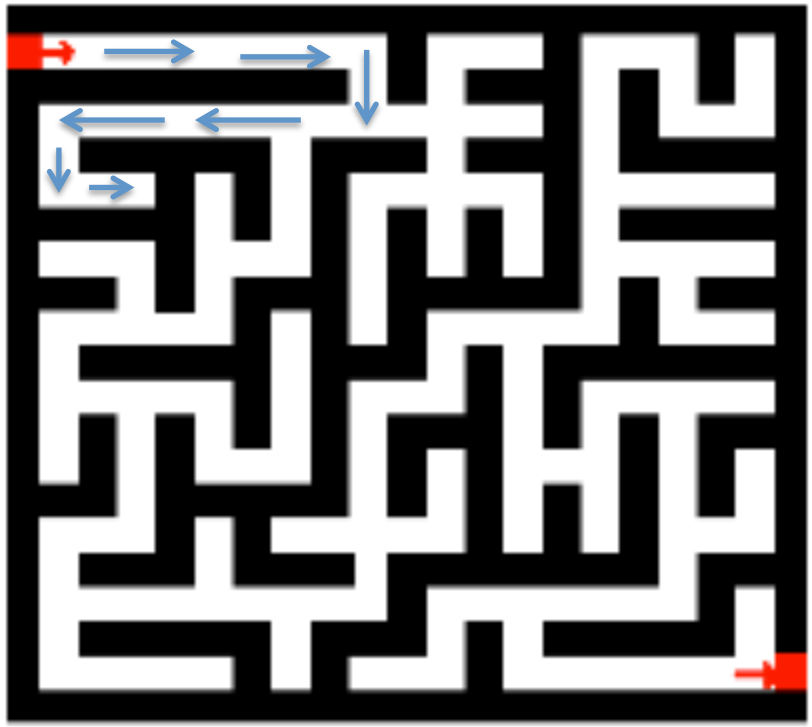
- Una vez que llegas a un camino muerto
  - Te tienes que regresar (Backtrak)
  - Tienes que regresar varios pasos hasta que encuentres otro camino potencialmente útil

Backtraking



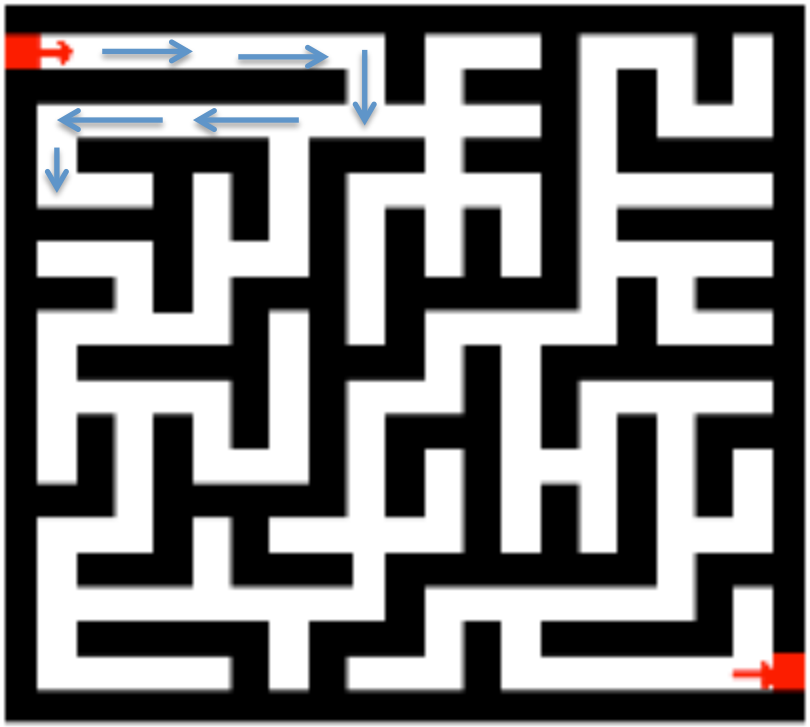
Backtraking

Se llega a un camino muerto



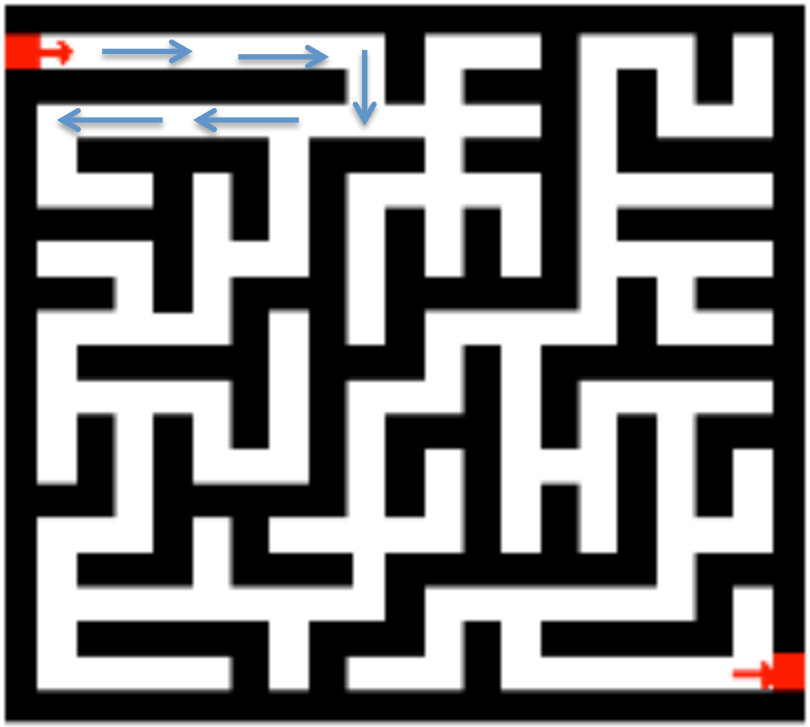
Backtraking

Empezamos a regresar en esos pasos dados



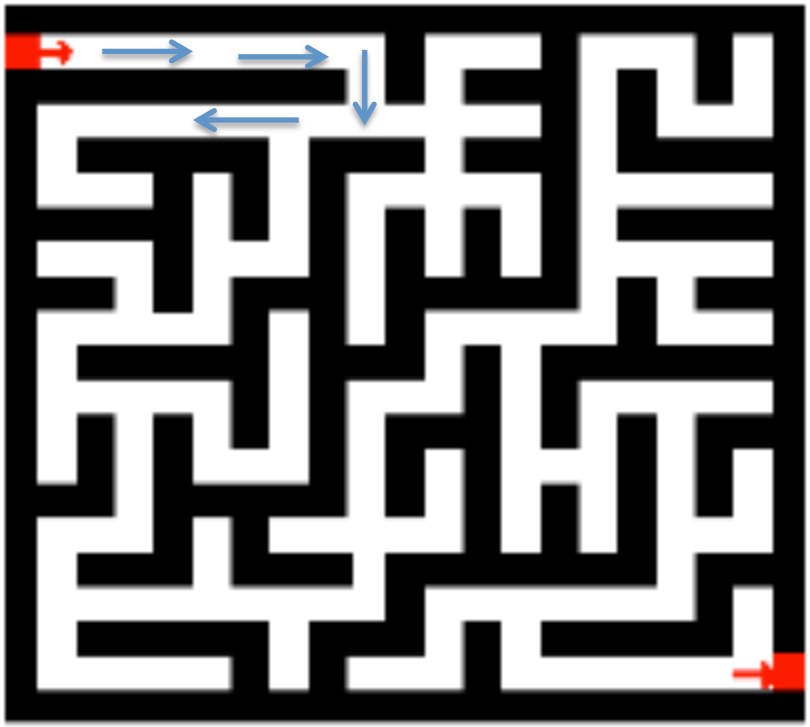
Backtraking

Empezamos a regresar en esos pasos dados



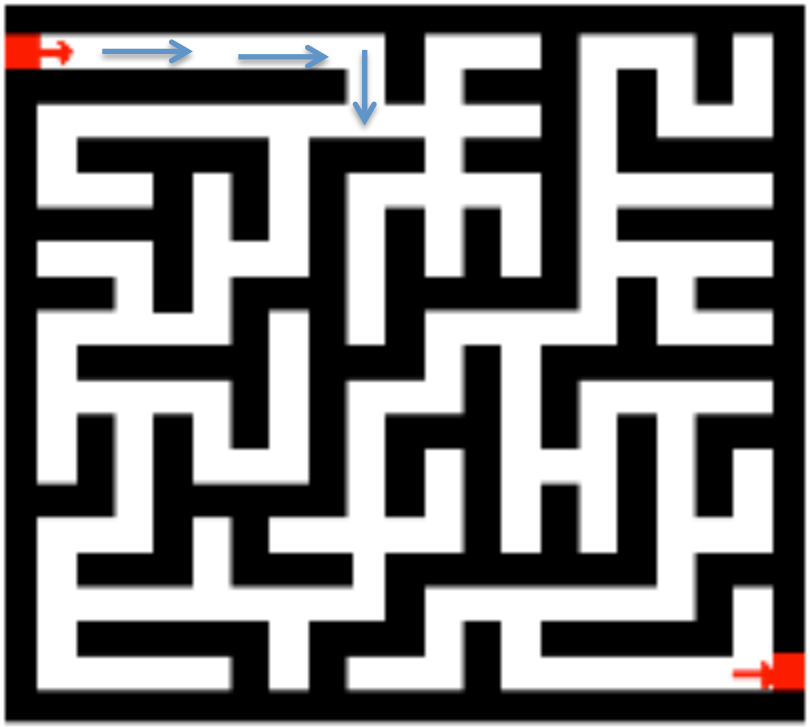
Backtraking

Empezamos a regresar en esos pasos dados



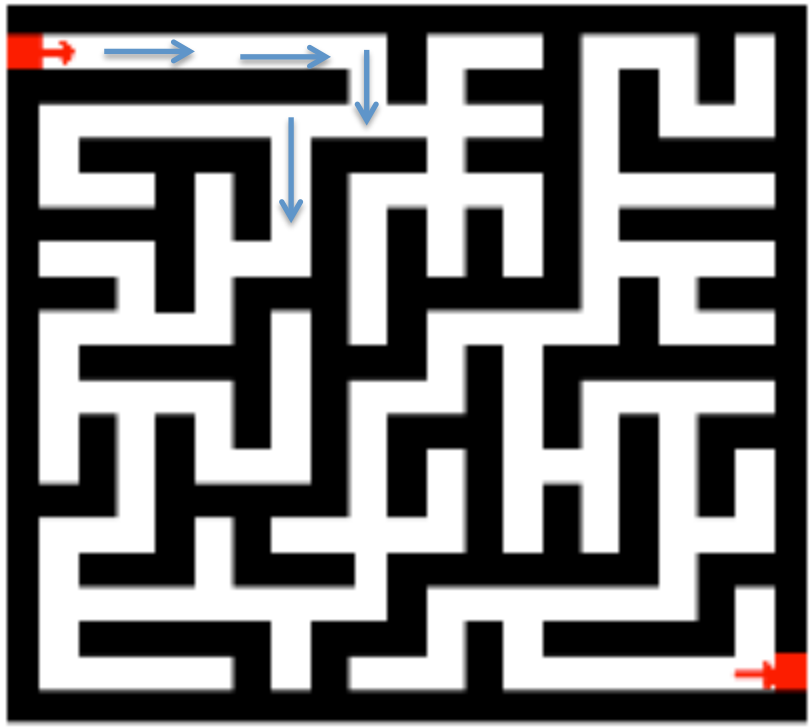
Backtraking

Empezamos a regresar en esos pasos dados



Backtraking

Al sacar los necesarios de la pila, se encuentra otra ruta, entonces empezamos a recorrerla, introduciendo de nueva cuenta esos pasos en la pila



Avanzar significa Push

Retroceder significa Pop

Procesamiento de Lenguaje

- Las pilas son utilizadas para:
  - Parámetros
  - Variables internas
  - Revisor de sintaxis para macheo de paréntesis
  - En las funciones recursivas

Implementación de Pilas

Implementación

- Existen diversas formas de implementar una estructura de tipo Pila:
  - Con arreglos estáticos
  - Arreglos Dinámicos
  - Listas ligadas

- Empecemos con los arreglos

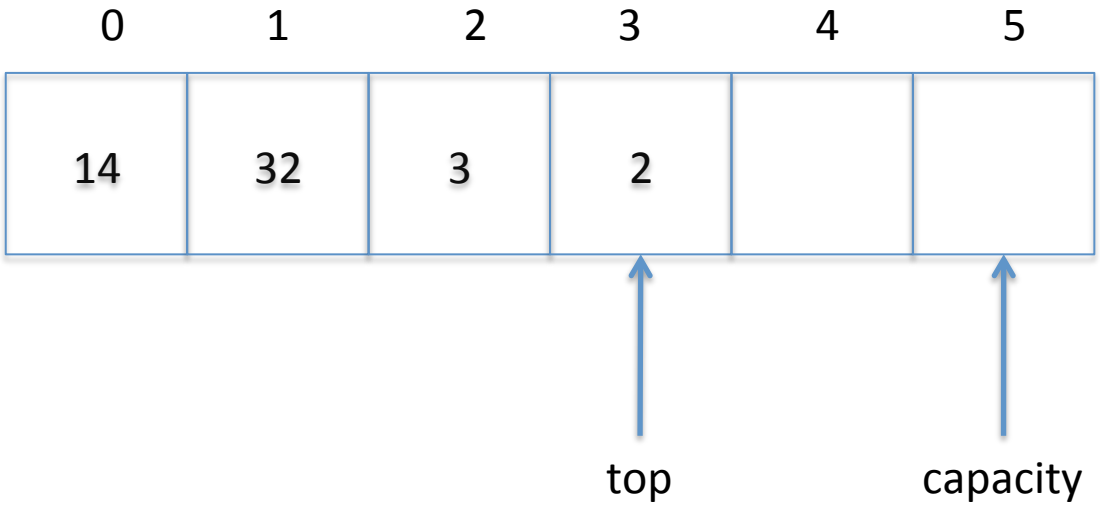
Pilas usando arreglos estáticos

- Elementos:
  - Arreglo con un tamaño >1
  - Una variable “top” que hace referencia al tope
  - una variable “capacity” que hace referencia a la capacidad máxima soportada por la pila
    - Recuerda que este es un arreglo estático
  - Top inicia en -1
    - Termina en capacity – 1
  - Se dice que la pila esta vacía si top = -1
    - y llena cunado top = capacity -1

Pilas usando arreglos estáticos

- Elementos:
  - El tamaño permanece constante
  - Si top = capacity
    - Existiría un desbordamiento, dando un error de segmentación o causando que el programa se detenga abruptamente.
    - Recuerda que si estas usando apuntadores para un tamaño fijo de pila, es posible que este no marque un error
      - En caso de existir mas información de usario adelante del arreglo
    - Pero podrías estar acarreando un error sin date cuenta

Pilas usando arreglos estáticos

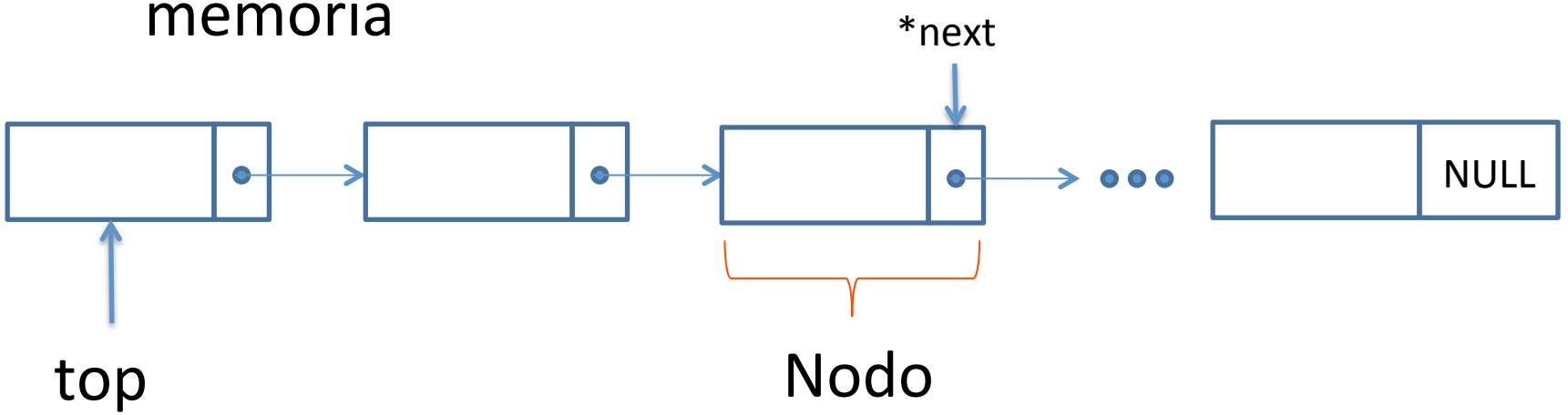


Pilas usando arreglos dinámicos

- Elementos:
  - Los mismos que los anteriores, sin embargo ahora el arreglo se asume que es apuntador
    - Por ejemplo: usando **malloc** o **calloc** para dar memoria
      - y usando **realloc** para hacerlo más grande o pequeño.
  - Cuando top = capacity
    - Se puede redimensionar el arregle con más espacio
      - Por ejemplo con el doble de espacio actual

Pilas usando listas ligadas

- Las listas ligadas dan la mejor implementación
  - Desde un punto de eficiencia
  - El primer elemento tiene su apuntador next = NULL
  - Cada que llega un nuevo elemento se solicita memoria



Primer ejercicio de Pilas - 1

- Con los conocimientos actuales que tienes de arreglos realiza el siguiente ejercicio:
  - Crear una pila de tamaño fijo de 5 elementos
  - Crea 3 funciones:
    - push – inserta elemento en la pila
    - pop – saca elemento de la pila
    - altura – regresa al altura de la pila
  - En este momento no verifiques el desbordamiento de la pila, y tus funciones pueden utilizar paso por referencia.

Primer ejercicio de Pilas - 1

- Por ejemplo, puedes usar un for para
  - Ingresar 3 elementos a la pila
  - Imprime antes y después de ello el tamaño de la pila
  - Al terminar
    - Saca dos elementos de la pila
    - Vuelve a imprimir el tamaño de la pila
- Habiendo terminado, crea una copia de tu programa y modifícalo para ingresar 100 elementos en la pila, con un tamaño de pila de 5
  - Observa el error generado y coméntalo con tus compañeros

Ejercicio - 2

- Teniendo el ejercicio anterior
  - Realiza una validación para evitar que la pila no llegue a si límite
  - Sigue utilizando arreglos estáticos

- Cuando termines el ejercicio, estaremos listos para usar arreglos dinámicos

Ejercicio - 3

- Repetir el mismo ejercicio pero ahora usando arreglos dinámicos

- Modifica la función push, para que cunado llegues al tope, duplicas el tamaño de la pila
- Comenta con tus compañeros que tan fácil o difícil es el ejercicio
  - ¿Crees que es mejor este tipo de técnicas para problemas de la vida real?

Resumen

- Este juego de diapositivas presentó el concepto de estructura de datos tipo Pila.
- Se describio como hacer una pila estática y dinámica, usando arreglos o listas enlazadas.
- Así mismo, realizaste varios ejercicios poniendo a prueba los conocimientos previos de la matería.
- Ahora sabes como programarlas y los problemas en que aplica esta estructura de datos.

Guion Explicativo

- **Estas diapositivas se deben de leer en el orden que aparecen.**
- **Debes de haber revisado a detalle el bloque de diapositivas de memoria dinámica, así como el de listas simplemente y doblemente enlazadas para iniciar con este juego de diapositivas**
- **Posteriormente, puedes revisar las estructuras de datos tipo colas.**

Bibliografía Básica

- Aguilar, Joyanes Luis, Programación en C++, Algoritmos, estructuras de datos y objetos, Ed. McGraw Hill,2002.
- Weiss. Allen Mark, Estructuras de datos y Algoritmos, Addison-Wesley Iberoamericana, S.A., 1995.
- A.,Euán Jorge I., B. Cordero Luis G., Estructura de datos, UNAM, 1984.
- Cairó, Osvaldo y Guardati, Silvia. Estructura de datos. Ed. McGraw Hill,2006.
- Criado, Ma. Asunción. Programación de lenguajes estructurados. AlfaOmega Ra-Ma, 2006.
- López, Leobardo. Programación estructurada. Un enfoque algorítmico. AlfaOmega,2004.

Bibliografía Complementaria

- Drozdeck,Adam. Estrcuturas de datos y algortimos en Java. Ed. Thomson, 2007.
- Joyanes, Luis M. Fernández, L. Sánchez, I. Zahonero. Estructuras de datos en C. Ed. McGraw Hill,2005.
- Koffman, Elliot y wolfang, Paul. Estrcutura de datos con C++. Objetos, abstracciones y diseño. Ed. McGraw Hill,2008.
- Nyhoff, Larry. TADs, Estructuras de datos y resolución de problemas en C++. Ed. Pearson-Prentice Hall, 2006.