



# UAEM

Universidad Autónoma  
del Estado de México



## Simulación de Inteligencia Artificial Unidad 2. Representación del Conocimiento – Algoritmos de búsqueda y ubicación de Datos

**Centro Universitario UAEM Valle de México**



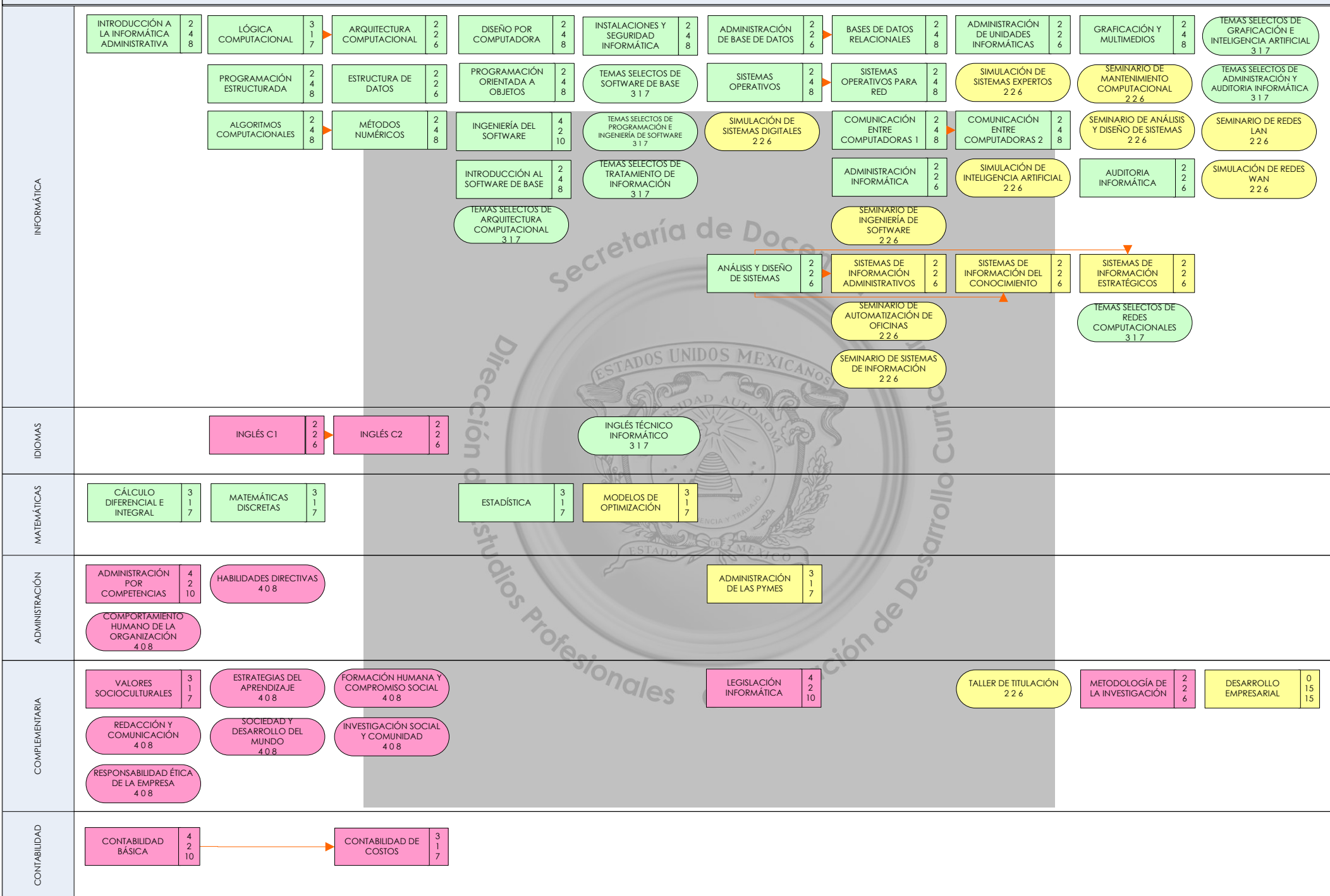
Fecha de elaboración: Julio 2016  
Semestre: Agosto – Diciembre 2016

**Licenciatura en Informática Administrativa**

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

MAPA CURRICULAR DE LA LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA 2003

CRÉDITOS TOTALES: 400



FINANZAS

ANÁLISIS Y PLANEACIÓN FINANCIERA	3 1 7
----------------------------------	-------------

PROYECTOS DE INVERSIÓN	3 1 7
------------------------	-------------

ECONOMÍA

MICROECONOMÍA	3 1 7	MACROECONOMÍA	4 2 10
---------------	-------------	---------------	--------------

SIMBOLOGÍA

8 ÁREAS CURRICULARES

HT HORAS TEÓRICAS  
HP HORAS PRÁCTICAS  
CR CRÉDITOS

11 LÍNEAS DE SERIACIÓN

NÚCLEO BÁSICO OBLIGATORIAS CURSAR Y ACREDITAR 11 UA	34 HT 18 HP 86 CR
---	-------------------------

NÚCLEO BÁSICO OPTATIVAS ACREDITAR 3 UA PARA CUBRIR 24 CRÉDITOS
--

TOTAL DEL NÚCLEO BÁSICO 14 UA 110 CRÉDITOS
--

NÚCLEO SUSTANTIVO OBLIGATORIAS CURSAR Y ACREDITAR 26 UA	58 HT 76 HP 192 CR
---	--------------------------

NÚCLEO SUSTANTIVO OPTATIVAS ACREDITAR 2 UA PARA CUBRIR 14 CRÉDITOS
--

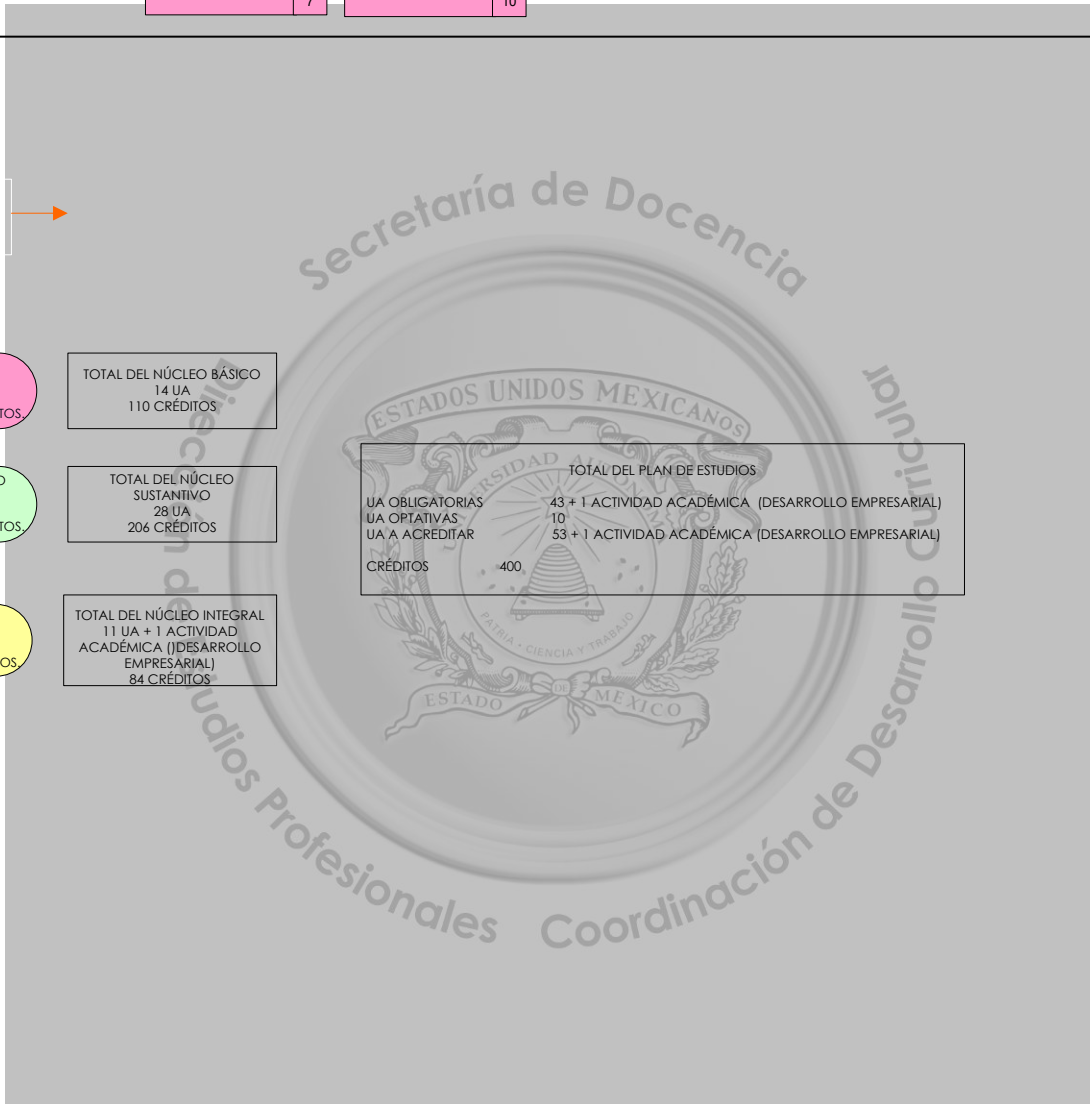
TOTAL DEL NÚCLEO SUSTANTIVO 28 UA 206 CRÉDITOS
--

NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 6 UA + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)	15 HT 24 HP 54 CR
---	-------------------------

NÚCLEO INTEGRAL OPTATIVAS ACREDITAR 5 UA PARA CUBRIR 30 CRÉDITOS
--

TOTAL DEL NÚCLEO INTEGRAL 11 UA + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL) 84 CRÉDITOS
--

TOTAL DEL PLAN DE ESTUDIOS	
UA OBLIGATORIAS	43 + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)
UA OPTATIVAS	10
UA A ACREDITAR	53 + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)
CRÉDITOS	400



# Índice de Contenidos

- Visión global de la Unidad de Aprendizaje (UDA)
- Descripción de la unidad de aprendizaje
- Unidad 2. Representación del Conocimiento
  - 2.1 Algoritmos de búsqueda de datos
  - 2.2 Algoritmos de ubicación de datos
- Resumen
- Guión Explicativo
- Bibliografía



**Programa de Estudio por Competencias  
 SIMULACIÓN DE INTELIGENCIA ARTIFICIAL**

**1. IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE**

<b>ORGANISMO ACADÉMICO: Facultad de Contaduría y Administración.</b>								
<b>Programa Educativo:</b> Licenciatura en Informática Administrativa.					<b>Área de docencia:</b> Informática			
<b>Aprobación por los H.H. Consejos Académico y de Gobierno</b>			<b>Fecha: 21 y 27 Marzo 2007</b>		<b>Programa elaborado por:</b> M en Sc. Román Ramírez Nájera MA Ed. Ma. de la Luz Sánchez Paz		<b>Fecha de elaboración</b>  15 febrero 2007	
Clave	Horas de teoría	Horas de práctica	Total de horas	Créditos	Tipo de Unidad de Aprendizaje	Carácter de la Unidad de Aprendizaje	Núcleo de formación	Modalidad
L30060	2	2	4	6	T	optativa	integral	Presencial
<b>Prerrequisitos</b>					<b>Unidad de Aprendizaje</b>		<b>Unidad de Aprendizaje Consecuente:</b>	
<b>Programas educativos en los que se imparte:</b> Licenciatura en Informática Administrativa								

# Descripción de la unidad de aprendizaje

# Identificación del Curso

Licenciatura en Informática Administrativa

**Horas de Teoría: 2 hrs.**

**Horas de Práctica: 2 hrs.**

**Créditos: 6**

**Unidad de Aprendizaje Antecedente:**

Ninguna

**Unidad de Aprendizaje Consecuente:**

Ninguna

# Presentación

Debido a los requerimientos en el manejo de la información, se hace imprescindible el desarrollo de estrategias y algoritmos que permitan el manejo eficiente de la misma en todas las áreas del saber humano. La inteligencia artificial desarrolla métodos para el manejo y administración de ésta.



# Lineamientos

## Del Profesor

- **Cubrir el programa en su totalidad**
- **Puntualidad**
- **Asistencia del 100% de las sesiones**
- **Aplicación del marco teórico y práctico de lo visto en una sesión de clase proporcionalmente a un 33% de peso en el tiempo de la sesión, de tal forma que se pueda asegurar el aprendizaje y puesta en práctica de cada alumno.**
- **Coordinar asistencia a una exposición nacional de equipo de cómputo**
- **Elaborar y publicar material didáctico en el portal académico de la universidad**

## Del Alumno

- **Asistencia mínima de un 80% del curso.**
- **Puntualidad con un máximo de tolerancia de 15 minutos**
- **Elaborar y cumplir los trabajos teóricos y prácticos que se soliciten**
- **Propuesta de talleres externos sobre el armado de computadoras**
- **Asistir una exposición nacional de equipo de cómputo**
- **Consultas en el portal de Internet de la Universidad**
- **Comunicarse por lo menos en dos foros de consulta en el portal académico de la Universidad**
- **Contar con una cuenta de correo electrónico para intercambio de material didáctico de la materia**

# Propósito

**El propósito será la simulación de los sistemas de inteligencia artificial para proponer soluciones prácticas a problemas reales.**

# Competencias genéricas

**Identificar los componentes y funcionamiento de un sistema de simulación de inteligencia artificial.**

# Ámbito de desempeño

**El proceso de enseñanza-aprendizaje se realizará mediante sesiones de explicación en clase, ejercicios grupales y realización de trabajos extraclase que complementen lo enseñado en el aula. Foros y consulta de materiales en internet.**

# Estructura

## Unidad 1.

- Nociones Básicas de Lógica

## Unidad 2.

- Representación del Conocimiento

## Unidad 3.

- Introducción al lenguaje Prolog

## Unidad 4.

- Introducción al lenguaje Clips

## Unidad 5.

- Aplicaciones de Inteligencia Artificial

# Estructura por unidad

- **Unidad 1.** Nociones Básicas de Lógica
  - *Tablas de verdad*
  - *Semántica del lenguaje*
  - *Formulación de Oraciones Lógicas*
- **Unidad 2.** Representación del Conocimiento
  - *Algoritmos de búsqueda de datos*
  - *Algoritmos de ubicación de datos*
  - *Máquina de inferencia*

# Estructura por unidad

- **Unidad 3.** Introducción al lenguaje Prolog
  - *Conocimiento del sistema operativo de la PC*
  - *Operación básica de la PC*
  - *Instalación del software*
  - *Comandos básicos del lenguaje*
- **Unidad 4.** Introducción al lenguaje Clips
  - *Instalación del software*
  - *Conceptos básicos del lenguaje*

# Estructura por unidad

- **Unidad 5.** Aplicaciones de la Inteligencia Artificial
  - *Desarrollo de Modelos de Simulación de Inteligencia Artificial*



# Procedimientos de Evaluación

2 Exámenes escritos	<b>40%</b>
Proyecto	<b>40%</b>
Trabajo, investigaciones y tareas	<b>20%</b>

# Representación del Conocimiento

# Contenido

- Unidad 2 Representación del Conocimiento
  - 2.1 Algoritmos de búsqueda de datos
  - 2.2 Algoritmos de Ubicación de datos

# 2.1 Algoritmos de búsqueda de datos

# Algoritmo de Búsqueda de Datos

- Localizar un elemento concreto dentro de una estructura de datos.
- Dará solución a un problema existencial o de un elemento determinado en un conjunto finito de elementos
  - Si el elemento en cuestión pertenece o no a dicho conjunto, además de su localización dentro de éste.

# Algoritmo de Búsqueda de Datos

- A continuación, se describen aquellos que se utilizan para buscar un elemento en un arreglo o lista:
- **Búsqueda secuencial:**
  - Se utiliza cuando el contenido de la estructura de datos no se encuentra o no puede ser ordenado.

# Búsqueda Secuencial

- Trabaja de manera sencilla
  - Consiste en buscar el elemento comparándolo secuencialmente, con cada elemento del arreglo o conjunto de datos hasta que se encuentre su ubicación, o hasta que se llegue al final del arreglo.
- La existencia se puede asegurar desde el momento en que el elemento es localizado
  - No se puede asegurar la no existencia hasta que no se hayan analizado todos los elementos del arreglo.

# Búsqueda Secuencial

## *Ventajas*

- Es el mejor método de búsqueda para registros desordenados
- Revisa nodo por nodo sin brincar ninguno.

## *Desventajas*

- Es muy lento
- Si los datos no están en orden es el único método que puede emplear



# Búsqueda Secuencial

- Si los valores de la llave no son únicos, para encontrar todos los registros con una llave particular, se requiere buscar en toda la lista.
- Rendimiento pobre en los archivos secuenciales
  - Aplicaciones interactivas que incluyen peticiones o actualizaciones de registros individuales

# Búsqueda Secuencial

## *Principales Aplicaciones*

- De proceso de lotes
  - Óptimos para dichas aplicaciones si se procesan todos los registros.
- Esta técnica es fuerte, ya que se aplica a un patrón de búsqueda pequeño, sencillo y manejable.

# Búsqueda Secuencial - - Ejemplo

Índice	0	1	2	3	4
Valor	<b>24</b>	<b>2</b>	<b>10</b>	<b>30</b>	<b>20</b>

*Objetivo: el valor 33*

- Paso 1: Comparar 33 con índice 1
- Paso 2: Comparar 33 con índice 2
- Paso 3: Comparar 33 con índice 3
- Paso 4: Comparar 33 con índice 4

**Resultado: Valor no encontrado**

## Búsqueda binaria (dicotómica)

- Se utiliza cuando el vector en el que queremos determinar la existencia o no de un elemento está ordenado, o puede estarlo
- Este algoritmo reduce el tiempo de búsqueda considerablemente.

# Búsqueda binaria - Algoritmo

- Se compara el elemento a buscar con un elemento cualquiera del arreglo o conjunto de datos
- Si el valor de éste es mayor que el del elemento buscado se repite el procedimiento en la parte del arreglo que va desde el inicio de éste hasta el elemento tomado
- En caso contrario:

# Búsqueda binaria - Algoritmo

- Se toma la parte del arreglo que va desde el elemento tomado hasta el final.
- Así, se van obteniendo intervalos cada vez más pequeños, hasta obtener un intervalo indivisible, con el elemento buscado como elemento central.
- Si el elemento no se encuentra dentro de este último
  - Se deduce que el elemento buscado no se encuentra en el arreglo.

## Búsqueda binaria - *Características*

- Se compara la llave buscada con la llave localizada al centro del arreglo.
- Si la llave analizada corresponde a la buscada
  - Fin de búsqueda, si no:

## Búsqueda binaria - *Características*

- Si la llave buscada es menor que la analizada repetir proceso en mitad superior
  - Si no en la mitad inferior.
- El proceso de partir por la mitad el arreglo se repite hasta encontrar el registro o hasta que el tamaño de la lista restante sea cero, lo cual implica que el valor de la llave buscada no está en la lista.



## Búsqueda binaria - *Ventajas*

- Método eficiente siempre que el vector esté ordenado.
  - Exige una ordenación previa del archivo.
- Proporciona un medio para reducir el tiempo requerido para buscar en una lista.
- Es más rápido por su recursividad, su mayor ventaja es con los archivos extensos.

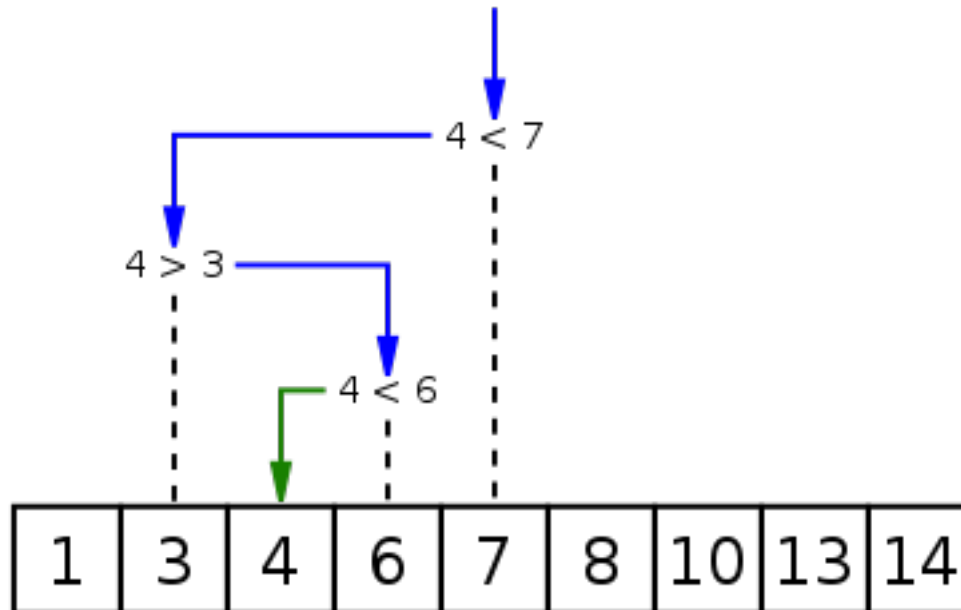
## Búsqueda binaria - *Ventajas*

- El código del procedimiento de esta búsqueda es corto en comparación con las demás técnicas de búsqueda.
- Con una sola comparación eliminamos la mitad de la tabla
  - Método más eficiente de buscar en una lista ordenada sin emplear tablas o índices adicionales.

## Búsqueda binaria - *Desventajas*

- El archivo debe estar ordenado
  - Suele plantear problemas en las inserciones y eliminaciones de elementos.
- No revisa todos los elementos del archivo
  - Requiere que todos los elementos estén ordenados.
- Mantener grandes archivos ordenados es muy costoso.

# Búsqueda binaria - - *Ejemplo*



## 2.2 Algoritmos de ubicación de datos

# Algoritmos de ubicación de datos

Al momento de generar algoritmos para este efecto, es necesario diseñar estrategias de búsqueda y control, a lo cual se debe saber que:

- La palabra búsqueda se refiere a la búsqueda de la solución en el espacio problema.

# Algoritmos de ubicación de datos

- Una búsqueda procede con diferentes tipos de estrategias de control.
  - Una estrategia se define escogiendo el orden en el que los nodos se expanden en dicha búsqueda.
- **Árbol:** Un árbol es un tipo de estructura jerarquizada alternativo a las listas enlazadas

# Árboles de búsqueda

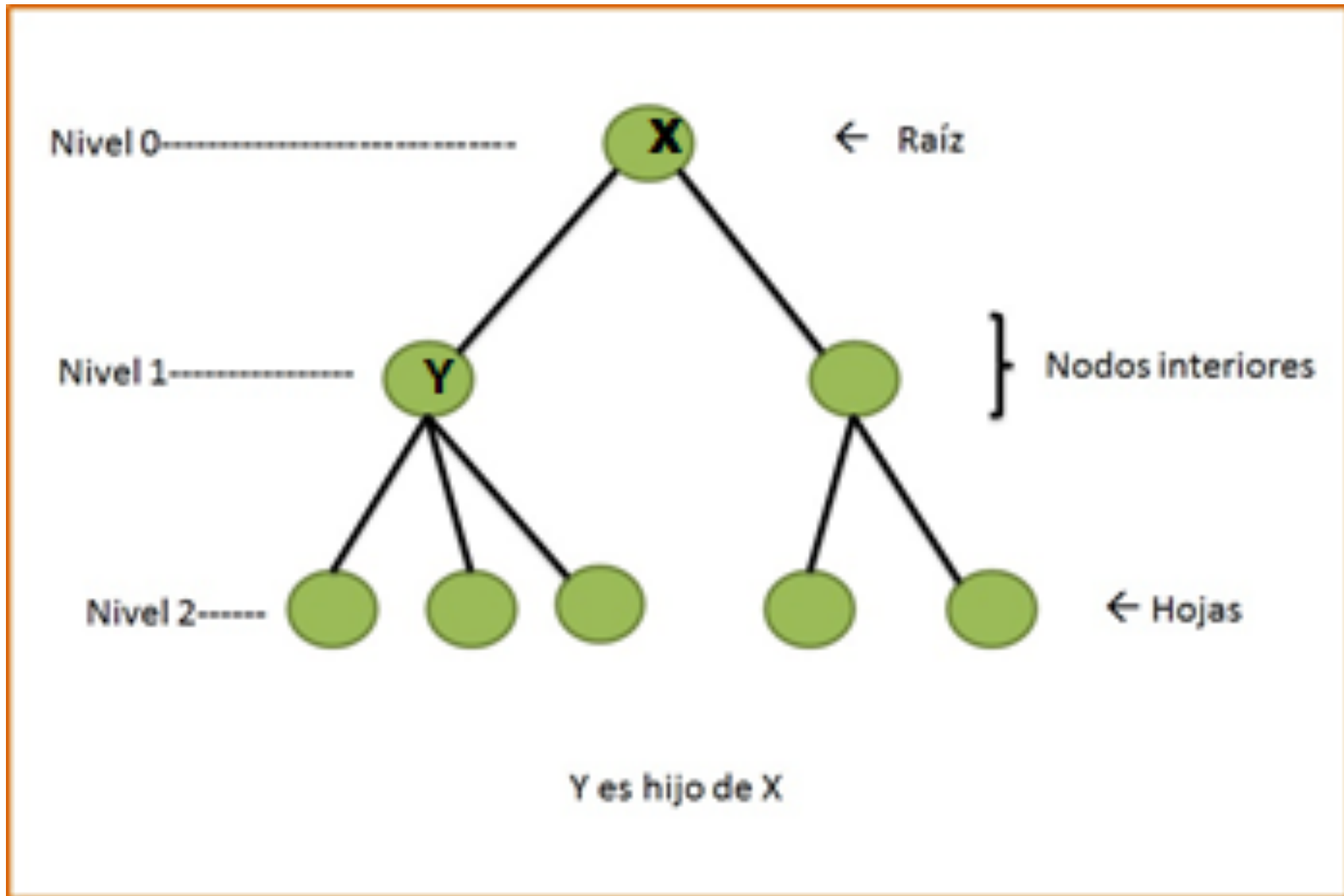
- Son utilizadas en la programación de hojas de calculo y en inteligencia artificial.
- Es una forma particular de grafo dirigido.
- La estructura es tipo árbol y se usa para la representación de datos.



# Árboles de búsqueda - Notación

- *Nodo*
  - Cada elemento de la estructura.
  - Cada nodo se ramifica con otros siguiendo algún criterio.
- *Raíz*
  - Es el elemento superior del árbol.
- *Hojas o terminales*
  - Son los nodos que no tienen subárboles.
- *Nivel o profundidad*
  - Es el largo del camino de la raíz al nodo.
- *Altura*
  - Son las capas que crecen a partir de la raíz.
- *Grado*
  - Es el número de descendientes directos de un nodo.

# Árboles de búsqueda - Notación



# Clasificación de los árboles binarios

- *Árbol binario distinto*
  - Cuando las sus estructuras de dos árboles binarios son diferentes.
- *Árbol binario similar*
  - Cuando sus estructuras son idénticas
  - La información que contiene sus nodos son diferentes.
- *Árbol binario equivalente*
  - Son arboles similares
  - Los nodos contienen la misma información.
- *Árbol binario completo*
  - Todos sus nodos excepto los del ultimo nivel, tiene dos hijos; el sub árbol izquierdo y sub árbol derecho.

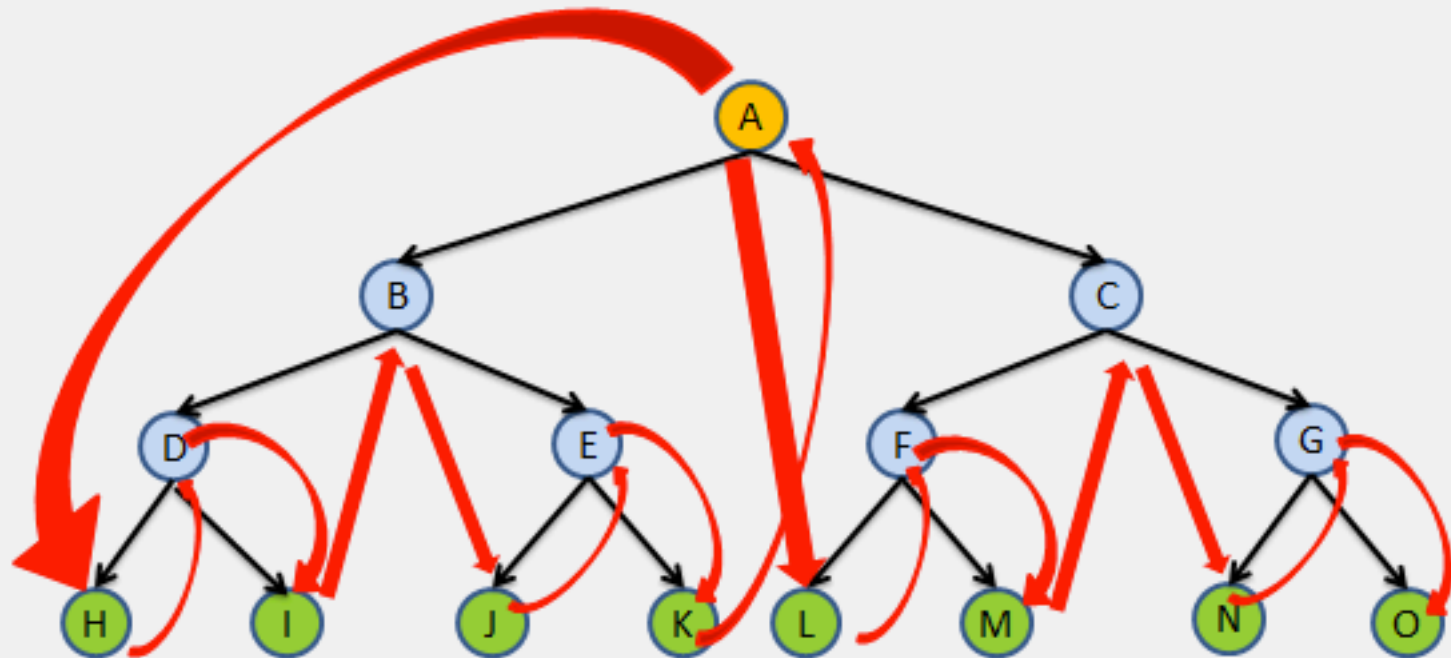
## *Recorridos para un árbol binario*

- Al estar en el nodo actual
  - Visitando el nodo
- Recorrido se puede dividir en dos
  - Visitando el:
    - hijo de la izquierda
    - hijo de la derecha
  - Para cada caso se aplica la recursividad

## *Recorridos para un árbol binario*

- En orden (in-orden)
  - Consiste en recorrer el árbol en el orden que está.
  - **i.e. (izquierdo, raíz, derecho)**
    - **Recorrer todo el sub-árbol izquierdo**
    - **Visitar la raíz**
    - **Recorrer todo el sub-árbol derecho**

### Recorrido en In-Orden

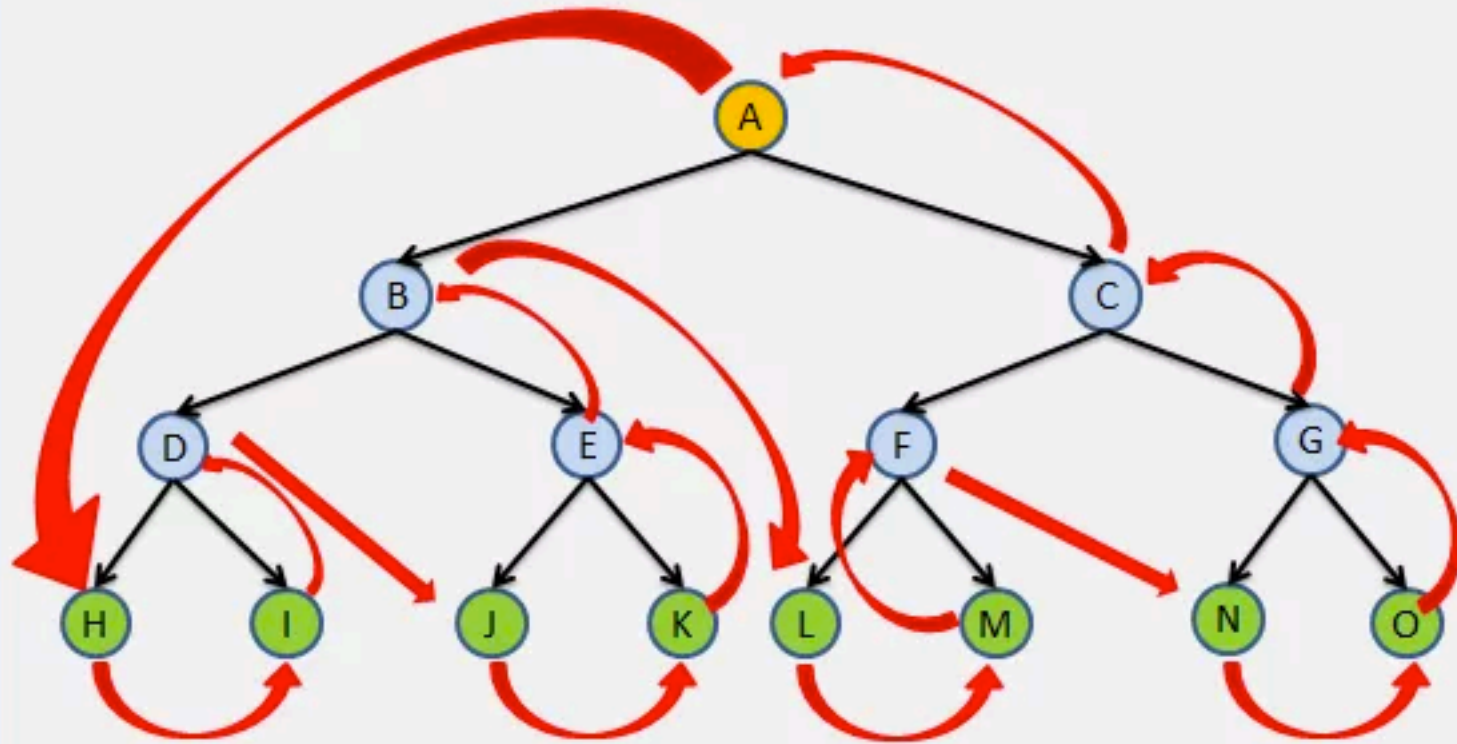


**Recorrido: H, D, I, B, J, E, K, A, L, F, M, C, N, G, O**

## *Recorridos para un árbol binario*

- Post-orden
  - Consiste en recorrer las ramas en orden inverso.
  - (Izquierdo, derecho, raíz)
- Pre-orden
  - Consiste en recorrer ramas desde la raíz hacia abajo.
  - (raíz, izquierdo, derecho)

### Recorrido en Post-Orden



**Recorrido: H, I, D, J, K, E, B, L, M, F, N, O, G, C, A**



## *Recorrido Pre-orden*

- Viendo los ejemplos de
  - In-orden y Post-orden

¿Puede hacer un ejemplo similar con un recorrido Pre-orden?

# *Pilas y Colas*

- Las pilas y colas son estructuras de datos.
- Mantiene el siguiente orden:
  - Pilas
    - Último en entrar, primero en salir
  - Colas
    - Primero en entrar, primero en salir
- Ambas a menudo se implementan como listas enlazadas
  - Pero esa no es la única manera posible.

# *Pilas*

- Una lista ordenada trabaja como
  - Último en entrar primero en salir (LIFO)
- Aquí los artículos (bloques - nodos) están en secuencia
  - Se apilan una encima de la otra, como en una pila de libros

# *Pilas*

- Las inserciones y eliminaciones se realizan en un solo extremo
  - Llamado superior.
- Si la pila es  **$S = a[1], a[2], \dots, a[n]$** 
  - **Entonces**
    - **$a[1]$**  es el elemento inferior.
    - Sería el ultimo en salir.

## *Pilas*

- Cualquier elemento intermedio  $a[i]$  está en la parte superior de elemento  $a[i-1]$ 
  - donde  $1 < i \leq n$
- Todas las operaciones tienen lugar en la parte superior.

# Pilas

- La operación **POP**
  - Elimina elemento desde la parte superior
- La operación **PUSH**
  - Inserta un elemento en la parte superior.

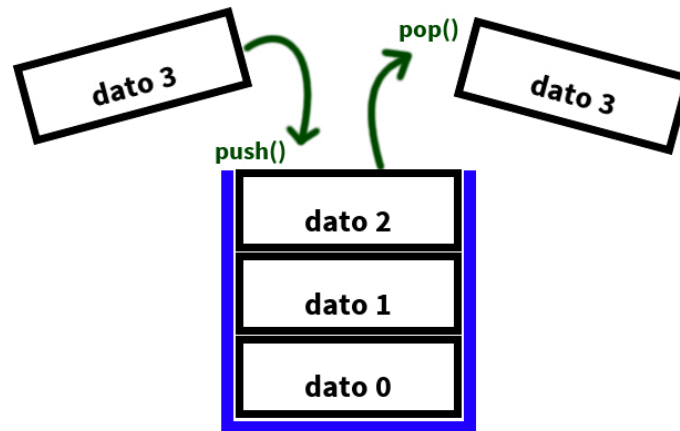


Figura tomada de: <https://miltonlab.gitbooks.io/estructuras-de-datos-2/content/chapter5.html>

## *Colas*

- Una lista ordenada trabaja como First-In First-out (FIFO).
  - Primero en entrar-primero en salir
  - Los elementos en una fila se agregan al final y se retiran de inicio
  - Marca restricciones sobre cómo los elementos se pueden agregar y quitar de la lista en secuencia.

## *Colas*

- Una cola tiene dos extremos.
- Todas las inserciones se llevan a cabo en un extremo, llamado **Posterior o Atrás**
- Todas las eliminaciones se llevan a cabo en el otro extremo, llamado **Frente o cabeza**.



## Colas

- Si la cola tiene un elemento  $[n]$  como la trasera  $a[i-1]$  es detrás de una  $[i]$ , donde  $1 < i \leq n$ .
- Todas las operaciones de inserción tienen lugar en un extremo **Final – Atrás**
- La eliminación se efectúa en la parte delantera – **Frente-Cabeza**

## *Resumen*

- Dentro de la representación del conocimiento, aquí se mostraron algoritmos de búsqueda y ubicación de datos, como la búsqueda secuencial y binaria, así como recorridos en arboles y estructuras de tipo Colas y Pilas

# Guión Explicativo

- Este juego de diapositivas debe leerse en el orden que aparece.
- Anterior a este juego, se debe revisar la Unidad 1. Nociones básicas de Lógica.
- Posteriormente se deberá revisar las diapositivas de la sección 2.3. Máquina de inferencia.

## Bibliografía básica

- Allen Collin, Hand Michael, Logic Primer, ed. MIT Press, 2001.
- Giarratano Joseph C., Riley Gary, Expert Systems: Principles and Programming, ed. Thompson, 2005.
- Rosen Kenneth H., Discrete Mathematics and its Applications, ed. McGraw Hill, 2007.
- Johnsonbaugh Richard, Matemáticas Discretas, ed. Prentice Hall, 2005.