

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Asignatura: Desarrollo Lógico Y Algoritmo
Taller Práctico 1
Profesor: Napoleón Ibarra
Estudiantes: Carlos Jiménez (4-845-1014) y Arvin Guerra (4-836-808)

I. Parte. Investigación (15 puntos)

Tema 1: Procedimiento de búsqueda y ordenamiento de un arreglo

1.1 Búsqueda secuencial

La búsqueda secuencial es el método más simple para localizar un elemento dentro de un arreglo. Consiste en recorrer el arreglo desde el primer elemento hasta el último, comparando cada valor con el dato buscado. Si el elemento se encuentra, se devuelve su posición. Si no está, el algoritmo termina sin encontrar coincidencias. Se usa cuando el arreglo no está ordenado o cuando el tamaño del arreglo es pequeño, ya que su costo de procesamiento crece linealmente con el número de elementos.

Características principales:

- No requiere que los datos estén ordenados.
- Su complejidad temporal es $O(n)$.
- Es fácil de implementar y entender.
- Ineficiente para grandes volúmenes de datos.

Ejemplo práctico:

Supón que tienes un arreglo con los valores [10, 25, 33, 47, 55]. Si buscas el número 33, el algoritmo compara 10, luego 25, y al llegar a 33 se detiene y devuelve la posición 2.

Pseudocódigo:

```
INICIO
    i ← 0
    encontrado ← FALSO
    MIENTRAS i < n Y encontrado = FALSO HACER
        SI arreglo[i] = clave ENTONCES
            encontrado ← VERDADERO
        FIN SI
        i ← i + 1
    FIN MIENTRAS
    SI encontrado ENTONCES
        ESCRIBIR "Elemento encontrado en la posición", i - 1
    SINO
        ESCRIBIR "Elemento no encontrado"
    FIN
```

Ejemplo en Python:

```
def busqueda_secuencial(arreglo, clave):
    for i in range(len(arreglo)):
        if arreglo[i] == clave:
            return i
    return -1

datos = [10, 25, 33, 47, 55]
pos = busqueda_secuencial(datos, 33)
print("Posición:", pos)
```

Salida: Posición: 2

Simulación paso a paso:

Arreglo = [10, 25, 33, 47, 55], clave = 33

- Paso 1: compara $10 \neq 33$
- Paso 2: compara $25 \neq 33$
- Paso 3: compara $33 = 33 \rightarrow$ encontrado

Ventajas:

- Sencillo de aplicar.
- Útil para estructuras pequeñas o no ordenadas.

Desventajas:

- Tarda más con muchos datos.
- No aprovecha arreglos ordenados.

1.2 Push Down (estructura de pila o “pila push-down”)

El método *Push Down* se relaciona con la estructura de datos tipo pila. Una pila almacena elementos de forma ordenada pero con acceso limitado. Solo se puede insertar o eliminar elementos por un extremo llamado “tope”. Este comportamiento se conoce como LIFO (Last In, First Out).

Operaciones principales:

- **Push:** inserta un elemento en el tope de la pila.
- **Pop:** elimina el elemento del tope.
- **Peek o Top:** permite ver el último elemento sin quitarlo.

Concepto:

Las pilas se utilizan para manejar operaciones anidadas, deshacer acciones o gestionar llamadas a funciones. Ejemplo clásico: el navegador web usa una pila para guardar las páginas visitadas y regresar con el botón “Atrás”.

Pseudocódigo:

```
CREAR pila vacía  
PUSH(pila, elemento):  
    añadir elemento al final de la pila  
  
POP(pila):  
    si la pila está vacía  
        mostrar error  
    si no, eliminar y devolver el último elemento
```

Ejemplo en Python:

```
class Pila:  
    def __init__(self):  
        self.items = []  
    def push(self, dato):  
        self.items.append(dato)  
    def pop(self):  
        if len(self.items) == 0:  
            return "Pila vacía"  
        return self.items.pop()  
    def mostrar(self):  
        return self.items  
  
p = Pila()  
p.push(5)  
p.push(10)  
p.push(15)  
print("Contenido actual:", p.mostrar())  
print("Elemento retirado:", p.pop())  
print("Pila final:", p.mostrar())
```

Salida:

Contenido actual: [5, 10, 15]

Elemento retirado: 15

Pila final: [5, 10]

Aplicaciones del Push Down:

- Control de recursividad en programas.
- Evaluación de expresiones matemáticas.
- Control de operaciones deshacer/rehacer.
- Seguimiento de rutas en navegadores o sistemas de archivos.

Tema 2: Base de Datos MySQL

Definición:

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) basado en lenguaje SQL (Structured Query Language). Permite almacenar, organizar y recuperar datos mediante tablas relacionadas. Es software libre y ampliamente utilizado en entornos web, empresariales y académicos.

Requisitos para ser instalado:

- Sistema operativo compatible: Windows, Linux o macOS.
- Espacio en disco de al menos 500 MB.
- Conexión a internet para descargar el paquete.

- Permisos de administrador.
- Usuario “root” inicial con contraseña segura.

Instalación básica:

- En Linux (Ubuntu):


```
sudo apt update
sudo apt install mysql-server
sudo systemctl start mysql
```
- En Windows:

Descargar *MySQL Installer* desde la web oficial y seguir el asistente.

Conexión Python–MySQL:

Para conectar Python con MySQL se necesita instalar una biblioteca que actúe como intermediario. Las más comunes son `mysql-connector-python` o `PyMySQL`.

Ejemplo de conexión:

```
import mysql.connector

conexion = mysql.connector.connect(
    host="localhost",
    user="root",
    password="tu_contraseña",
    database="alquiler")

cursor = conexion.cursor()
cursor.execute("SELECT DATABASE();")
print("Conectado a:", cursor.fetchone())
conexion.close()
```

Requisitos para la conexión:

- Tener el servidor MySQL corriendo.
- Crear una base de datos.
- Configurar usuario y permisos adecuados.
- Instalar el conector con `pip install mysql-connector-python`.

Replicación en una base de datos:

La replicación es el proceso de copiar y mantener datos idénticos en varios servidores. Un servidor maestro envía las actualizaciones a uno o más servidores esclavos. Esto mejora la disponibilidad, el rendimiento y la seguridad de la información.

Tipos de replicación:

- **Asíncrona:** el maestro no espera confirmación del esclavo.
- **Semisíncrona:** el maestro espera que al menos un esclavo confirme la recepción.
- **Síncrona:** el maestro y los esclavos actualizan los datos al mismo tiempo.

Beneficios de la replicación:

- Copias de respaldo actualizadas.
- Mayor velocidad en consultas distribuidas.
- Reducción del riesgo de pérdida de información.
- Posibilidad de balancear carga entre servidores.

Ejemplo simple de configuración conceptual:

- Servidor Maestro (IP 192.168.1.10): guarda los datos originales.
 - Servidor Esclavo (IP 192.168.1.11): recibe las copias del maestro.
- El esclavo se sincroniza mediante binlogs (archivos de registro) que contienen todas las operaciones realizadas en el maestro.

II. Laboratorio (15 puntos)

Procedimiento mínimo para realizar localmente

1. Instalar MySQL Linux: `sudo apt install mysql-server` o usar repositorio oficial Windows: usar instalador MySQL Installer Mac: usar Homebrew o paquete oficial
2. Configurar Ejecutar `mysql_secure_installation` para asegurar la instalación Crear usuario con permisos y base de datos de pruebas
3. Probar funcionamiento Conectar desde línea de comandos: `mysql -u usuario -p` Crear tabla de ejemplo y hacer inserciones y consultas
4. Comandos SQL básicos para crear base de datos `CREATE DATABASE alquiler; USE alquiler;` `CREATE TABLE equipos (id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100), tipo VARCHAR(50), disponible BOOLEAN, precio_hora DECIMAL(10,2));`
5. Verificación desde Python Usar el ejemplo de conexión anterior y ejecutar un `SELECT` para confirmar.

III. Desarrollo prototipo (30 puntos)

Caso: sistema de alquiler/asignación de equipos. Requisitos: insertar, actualizar, eliminar, listar, calcular costo final, guardar registros y generar factura a pedido. Control de estado activo/no activo.

MySQL Workbench

MySQL Model (BD.mwb) ×

File Edit View Arrange Model Database Tools Scripting Help

Description Editor

Model Overview

clientes: MySQL Table

Physical Schemas

BD MySQL Schema

Tables (3 items)

Add Table alquileres clientes equipos

Views (0 items)

Add View

Routines (0 items)

Add Routine

Routine Groups (0 items)

Add Group

Schema Privileges

SQL Scripts

Model Notes

Description

User Types List

Type Definition Flags

```
mysql> SHOW TABLES;
+-----+
| Tables_in_bd |
+-----+
| alquileres   |
| clientes     |
| equipos       |
+-----+
3 rows in set (0.03 sec)

mysql> DESCRIBE equipos;
+-----+-----+-----+-----+-----+-----+
| Field | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idequipos | int            | NO   | PRI | NULL    | auto_increment |
| tipo      | enum('laptop','PC','Impresora') | NO   |     | NULL    |                |
| estado    | enum('activo','inactivo')        | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> DESCRIBE clientes;
+-----+-----+-----+-----+-----+-----+
| Field | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_cliente | int            | NO   | PRI | NULL    | auto_increment |
| nombre     | varchar(45)    | NO   |     | NULL    |                |
| contacto   | varchar(45)    | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

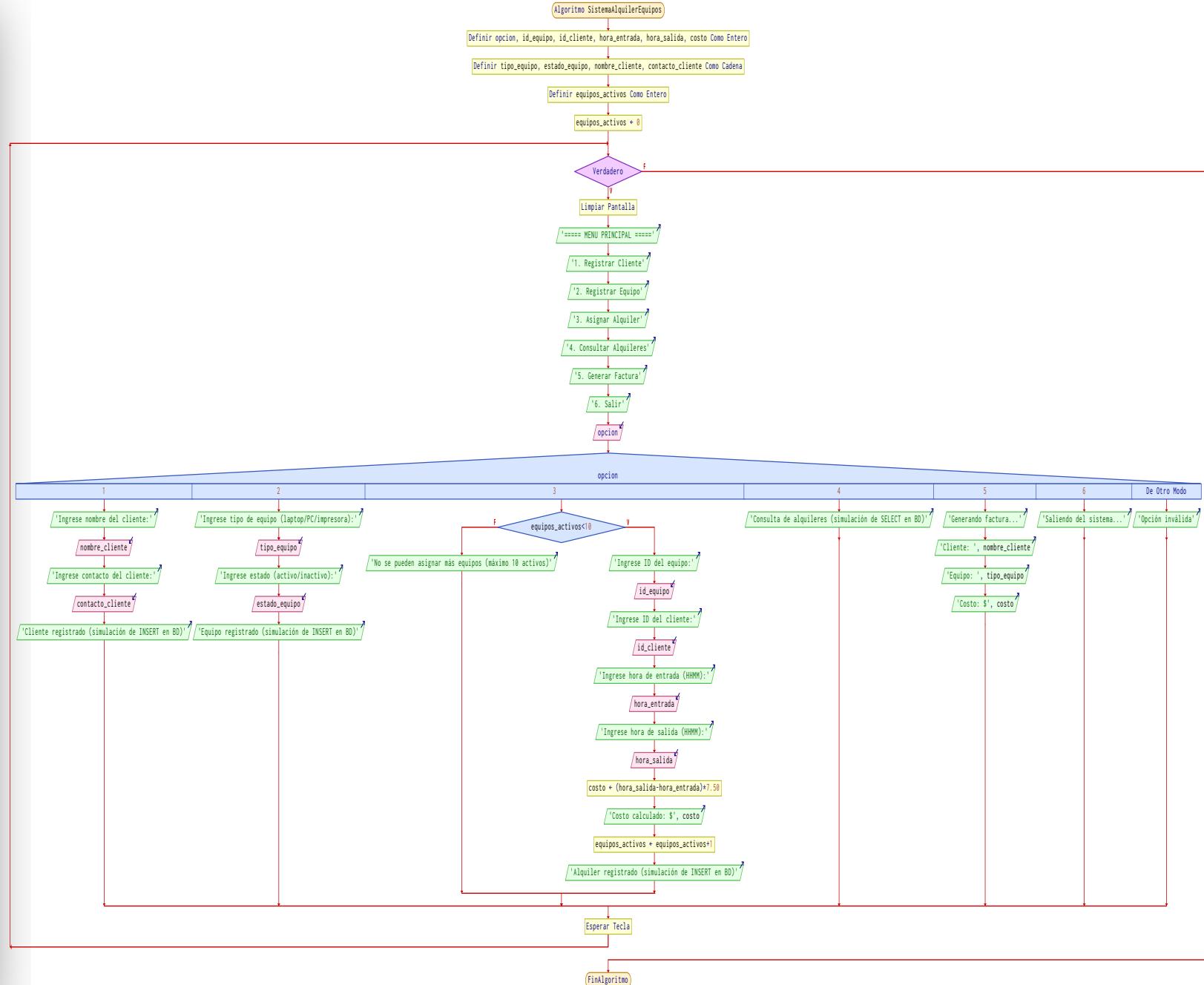
mysql> DESCRIBE alquileres;
+-----+-----+-----+-----+-----+-----+
| Field | Type           | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_alquiler | int            | NO   | PRI | NULL    | auto_increment |
| id_equipo   | int            | NO   | MUL | NULL    |                |
| id_cliente  | int            | NO   | MUL | NULL    |                |
| hora_entrada | datetime       | NO   |     | NULL    |                |
| hora_salida | datetime       | NO   |     | NULL    |                |
| costo       | decimal(10,2)  | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
```

Pseudocódigo: Base de Datos

```

1 Proceso SistemaAlquilerEquipos
2 Definir opcion, id_equipo, id_cliente, hora_entrada, hora_salida, costo Como Entero
3 Definir tipo_equipo, estado_equipo, nombre_cliente, contacto_cliente Como Cadena
4 Definir equipos_activos Como Entero
5 equipos_activos <- 0
6
7 Mientras Verdadero Hacer
8   Limpiar Pantalla
9   Escribir "===== MENU PRINCIPAL ====="
10  Escribir "1. Registrar Cliente"
11  Escribir "2. Registrar Equipo"
12  Escribir "3. Asignar Alquiler"
13  Escribir "4. Consultar Alquileres"
14  Escribir "5. Generar Factura"
15  Escribir "6. Salir"
16  Leer opcion
17
18 Segun opcion Hacer
19   1:
20     Escribir "Ingrese nombre del cliente:"
21     Leer nombre_cliente
22     Escribir "Ingrese contacto del cliente:"
23     Leer contacto_cliente
24     Escribir "Cliente registrado (simulación de INSERT en BD)"
25
26   2:
27     Escribir "Ingrese tipo de equipo (laptop/PC/impresora):"
28     Leer tipo_equipo
29     Escribir "Ingrese estado (activo/inactivo):"
30     Leer estado_equipo
31     Escribir "Equipo registrado (simulación de INSERT en BD)"
32
33   Si equipos_activos < 10 Entonces
34     Escribir "Ingrese ID del equipo:"
35     Leer id_equipo
36     Escribir "Ingrese ID del cliente:"
37     Leer id_cliente
38     Escribir "Ingrese hora de entrada (HHMM):"
39     Leer hora_entrada
40     Escribir "Ingrese hora de salida (HHMM):"
41     Leer hora_salida
42     costo <- (hora_salida - hora_entrada) * 7.50
43     Escribir "Costo calculado: $", costo
44     equipos_activos <- equipos_activos + 1
45     Escribir "Alquiler registrado (simulación de INSERT en BD)"
46   SiNo
47     Escribir "No se pueden asignar más equipos (máximo 10 activos)"
48   FinSi
49
50   4:
51     Escribir "Consulta de alquileres (simulación de SELECT en BD)"
52
53   5:
54     Escribir "Generando factura..."
55     Escribir "Cliente: ", nombre_cliente
56     Escribir "Equipo: ", tipo_equipo
57     Escribir "Costo: $", costo
58
59   6:
60     Escribir "Saliendo del sistema..."
61 De Otro Modo:
62   Escribir "Opción inválida"
63 FinSegun
64 Esperar Tecla
65 FinMientras
66 FinProceso

```



Programa: Python

Programa adjunto como “alquiler_app”

Sistema de Alquiler de Equipos - XYZ (MySQL)

Cliente Nombre:	arvin	Contacto:	26			
Tipo de Equipo:	PC					
Hora Entrada (YYYY-MM-DD HH:MM:SS):	2025-10-28 13:45:00	Hora Salida (YYYY-MM-DD HH:MM:SS):	2025-10-28 15:45:00			
Costo por Hora (\$):	2.50					
Asignar Equipo		Nuevo	Salir			
Tiempo Transcurrido (horas): 2.00 horas Costo Estimado: \$5.00						
Ver Registros	Generar Factura	Actualizar Registro	Eliminar Registro			
ID	Equipo	Estado	Cliente	Entrada	Salida	Costo

IV. PARTE. Diagrama de RED LAN. Valor 10 puntos.

Procedimiento: Utilizando la herramienta Packet Tracer confeccione la propuesta de la Red LAN de acuerdo al caso de estudio de la l1 parte. Verifique su funcionamiento.

