

**Universidad Tecnológica de Panamá**  
**Facultad de Sistemas Computacionales**  
**Asignatura: Desarrollo Lógico y Algoritmo**  
**Taller Práctico1**

Profesor: Napoleón Ibarra

Valor: 100 puntos

Estudiante: Rafael Rivera 4-834-473

Olmedo Alvarado 4-839-1939

**Fecha Inicio: 27/10/2025 --> 4:10 PM**

**Fecha Entrega: 28/10/2025 -->3:20 PM**

**Procedimiento:**

1. De manera individual o en grupo de trabajo de 2 personas, realizar la asignación. Utilizando la herramienta Internet, investigue, desarrolle los conceptos solicitados.
2. Entregar el trabajo en formato digital (Parte I, II, III, IV) en PDF en la plataforma.

**Criterios de Evaluación:**

Criterios	Puntos (Mínimo 1, Máximo 5)	Porcentaje
Sustentación	1 - 5	15 %
Puntualidad	1 - 5	15 %
Desarrollo	1 - 5	70 %

**I PARTE. Investigación. *Valor 15 puntos***

**Temas:**

- 1) *Procedimiento de búsqueda y ordenamiento de un arreglo.*
  - 1.1. *Búsqueda secuencial.*
  - 1.2. *Push Down.*
- 2) *Base de Datos MYSQL: Definición, ¿Qué se requiere para ser instalado?, ¿Qué se necesita para hacer una conexión PYTHON-MYSQL? Explique, ¿Qué es una replicación en una Base de Datos? Sustente su respuesta.*

**Procedimiento:**

1. Utilice la técnica (a su criterio) para desarrollar el tema propuesto.
2. En su desarrollo (Ponencia) debe estar los siguientes puntos:
  - 2.1. *Un ejemplo (código sencillo funcional) de Arreglos. Su pseudocódigo y simulación.*
  - 2.2. *Concepto.*
  - 2.3. *Su sintaxis.*

**II PARTE. Laboratorio. *Valor 15 puntos***

**Procedimiento:**

1. Escenario local: instale, configure, haga pruebas de funcionamiento a MySQL en su equipo de producción.
2. Una vez instalado confeccione una base de datos (Usted elige su nombre). Tome en cuenta la III parte de la actividad.

**III PARTE. Desarrollo prototipo. *Valor 30 Puntos.***

Caso de Estudio: La empresa XYZ tiene sus servicios tecnológicos a disposición de sus clientes: alquiler de equipos (PC), impresiones, fotocopiado, otros. Requiere que Usted elabore/programe un prototipo de pseudocódigo y programa (PYTHON) que permita asignar/alquilar equipos,

calcule el costo final, guarde, almacene los registros, genere la factura en caso de que el cliente la solicite.

Actualmente la organización cuenta con 11 equipos entre laptop, PC escritorio, impresora multifuncional. El programa debe ser capaz de insertar, actualizar, eliminar registros.

Tome en cuenta los equipos (10), puede darse el caso que los mismos todos se estén utilizando a la misma vez, a modo de sugerencia controle quienes están activos/no activos.



Figura 1. Ejemplo propuesta de cálculo de uso

### III 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 II parte. Verifique su funcionamiento.

## BUENA SUERTE

### Desarrollo

Un arreglo (o array) es una estructura de datos fundamental que almacena una colección de elementos del mismo tipo en posiciones de memoria contiguas. A cada elemento se le puede acceder directamente a través de su índice (su posición numérica, usualmente comenzando en 0).

Sintaxis (Ejemplo en Python): La sintaxis para crear un arreglo (llamado "lista" en Python, que es más flexible, pero cumple el mismo propósito).

#### 1.1. Búsqueda Secuencial

Concepto: La búsqueda secuencial (o búsqueda lineal) es el método de búsqueda más simple. Consiste en recorrer un arreglo elemento por elemento, desde el primero hasta el último, comparando cada elemento con el valor que se desea encontrar (el "valor objetivo").

Si se encuentra el elemento, la búsqueda termina exitosamente (a menudo devolviendo el índice donde se encontró).

Si se llega al final del arreglo sin encontrar el elemento, la búsqueda termina sin éxito.

Es un método simple, pero puede ser ineficiente (lento) en arreglos muy grandes. Funciona tanto para arreglos ordenados como desordenados.

## 1.2. Push Down

Concepto: El término "Push Down" no es un algoritmo estándar de búsqueda ni de ordenamiento en el contexto de arreglos.

Lo más probable es que se refiera a la operación "push" (empujar), que es la acción de agregar un nuevo elemento al final de un arreglo. En Python, esta operación se realiza comúnmente con el método `append()`.

Esta operación es fundamental en la estructura de datos "Pila" (Stack), donde se "apila" (push) un elemento en la cima. Cuando se implementa una pila usando un arreglo, "push" simplemente significa agregar el elemento al final.

## 2. Base de Datos MySQL

Definición: MySQL es uno de los sistemas de gestión de bases de datos relacionales (RDBMS, por sus siglas en inglés) de código abierto más populares del mundo. Se basa en SQL (Lenguaje de Consulta Estructurado). Funciona bajo un modelo cliente-servidor, donde un "servidor" (el software MySQL) almacena y gestiona los datos, y los "clientes" (aplicaciones, como un programa de Python) se conectan a él para solicitar o modificar dichos datos.

¿Qué se requiere para ser instalado? Para instalar el servidor MySQL (por ejemplo, MySQL Community Server) en un equipo, se requiere:

- Software de Instalación: Descargar el instalador oficial de MySQL (MySQL Installer para Windows, o los paquetes .deb/.rpm para Linux, .dmg para macOS) desde su sitio web.
- Sistema Operativo Compatible: MySQL es multiplataforma y funciona en los sistemas operativos más comunes como Windows, macOS y diversas distribuciones de Linux.
- Hardware: Cumplir con los requisitos mínimos de memoria RAM, espacio en disco duro y CPU (generalmente modestos para una instalación básica).
- Dependencias: En algunos sistemas (como Windows), puede requerir componentes adicionales como el paquete Microsoft Visual C++ Redistributable.
- Herramientas (Opcional pero recomendado): Una herramienta gráfica como MySQL Workbench para administrar la base de datos visualmente.

¿Qué se necesita para hacer una conexión PYTHON-MYSQL? Para que un programa de Python pueda "hablar" con una base de datos MySQL, se necesitan dos cosas principales:

Una Biblioteca "Conector": Es un software intermediario (un driver) que traduce las instrucciones de Python al protocolo que entiende MySQL. El conector oficial y más recomendado es mysql-connector-python.

Se instala en el entorno de Python usando el gestor de paquetes pip:

Bash

```
pip install mysql-connector-python
```

- El Código de Conexión: Dentro del script de Python, se deben seguir estos pasos:
  - Importar la biblioteca: `import mysql.connector`.
  - Establecer la conexión: Usar la función `connect()` con los parámetros de acceso: `host` (dirección del servidor, ej: "localhost"), `user` (usuario de la BD), `passwd` (la contraseña) y `database` (el nombre de la base de datos a la que te quieres conectar).
  - Crear un "cursor": Un objeto cursor es el que permite ejecutar los comandos SQL.
  - Ejecutar consultas: Usar `cursor.execute("SELECT * FROM ...")` o `cursor.execute("INSERT INTO ...")`.
  - Confirmar cambios: Si se hicieron cambios (INSERT, UPDATE, DELETE), se debe llamar a `db.commit()` (donde db es la conexión).
  - Cerrar la conexión: Es crucial cerrar el cursor y la conexión al terminar: `cursor.close()` y `db.close()`.

¿Qué es una replicación en una Base de Datos? (Sustento) La replicación es el proceso de crear y mantener múltiples copias de una base de datos en diferentes servidores, de forma sincronizada.

Funciona con un modelo Origen-Réplica (o Maestro-Esclavo). El servidor "Origen" (Master) es la fuente principal de datos; todas las escrituras (INSERT, UPDATE, DELETE) ocurren allí. Los servidores "Réplica" (Slaves) copian automáticamente todos los cambios realizados en el Origen.

Sustento (Utilidad de la replicación): La replicación se utiliza por tres razones principales:

Alta Disponibilidad y Tolerancia a Fallos: Si el servidor Origen falla (se cae, se daña), una de las Réplicas puede ser "promovida" rápidamente para convertirse en el nuevo Origen. Esto minimiza el tiempo de inactividad del sistema.

Escalabilidad y Rendimiento (Balanceo de Carga): En aplicaciones con muchas lecturas (muchos usuarios consultando datos, pero pocos escribiendo), la carga se puede distribuir. El servidor Origen maneja todas las escrituras, mientras que las consultas de lectura (SELECTs) se reparten entre todas las Réplicas. Esto mejora drásticamente el rendimiento.

Respaldo (Backup) en Caliente: Se pueden realizar copias de seguridad pesadas sobre un servidor Réplica sin afectar el rendimiento ni bloquear el servidor Origen, que sigue operando y recibiendo datos de los usuarios.

```
Arreglo original: ['PC-01', 'Laptop-02', 'Impresora-01']
Arreglo después de 'Push Down': ['PC-01', 'Laptop-02', 'Impresora-01', 'PC-03']
-----
Simulación: El equipo 'Laptop-02' SÍ fue encontrado.
Se encuentra en la posición (índice): 1
Simulación 2: El equipo 'PC-10' NO fue encontrado.
```