

# Instructivo del Proyecto

## 1. Requisitos Previos

- Un entorno tipo Linux (puede ser una instalación nativa de Linux o a través de WSL en Windows).
- Tener instalado el paquete *build-essential*, que incluye el compilador *gcc* y la herramienta *make*. Para instalarlo, ejecuta: *sudo apt update && sudo apt install build-essential*.

## 2. Organización de Archivos

Asegúrate de tener todos los siguientes archivos en una única carpeta de proyecto:

- *generador.c*
- *servidor.c*
- *cliente.c*
- *makefile*
- *validate.awk*

## 3. Compilación

El proyecto incluye un *Makefile* que compila todos los programas necesarios.

1. Abre una terminal en la carpeta del proyecto.
2. Ejecuta el comando:

```
make
```

3. Esto creará tres archivos ejecutables: **generador**, **servidor** y **cliente**.

Nota: Si necesitas recompilar todo desde cero, puedes usar *make clean* para borrar los ejecutables anteriores y luego *make* de nuevo.

# Generador de Datos de Prueba con Procesos y Memoria Compartida

## Generar el Archivo de Datos

Primero hay que generar el archivo *output.csv* con los datos.

- **Comando:**

```
./generador <cantidad_procesos> <total_registros>
```

- **Parametros:**

- *<cantidad\_procesos>*: Número de procesos que generarán datos en paralelo.
- *<total\_registros>*: Cantidad total de registros a crear en el archivo.

- **Ejemplo:**

```
./generador 10 50000
```

- **Verificación:** Para asegurarte de que el archivo se generó correctamente, puedes usar el script de validación *awk*:

```
awk -f validate.awk output.csv
```

- **Prueba Automatizada:** El script *test.sh* automatiza todo el proceso de compilación, ejecución y validación del generador. Para ejecutarlo, puedes usar el intérprete *bash* directamente:

```
bash test.sh
```

# Cliente-Servidor de Micro Base de Datos con Transacciones

## Iniciar el Servidor

Una vez que *output.csv* existe, puedes iniciar el servidor.

- **Comando:**

```
./servidor <puerto> <clientes_concurrentes> <clientes_en_espera>
```

- **Parámetros:**

- *<puerto>*: El número de puerto en el que escuchará el servidor (ej: **8080**).
- *<clientes\_concurrentes>*: Número máximo de clientes que pueden ser atendidos al mismo tiempo.
- *<clientes\_en\_espera>*: Número máximo de clientes que pueden quedar en la cola de espera si el servidor está lleno.

- **Ejemplo:**

```
./servidor 8080 2 1
```

## Conectar Clientes

Abre nuevas terminales (una por cada cliente) para conectarte al servidor.

- **Comando:**

```
./cliente <IP_servidor> <puerto>
```

- **Parámetros:**

- *<IP\_servidor>*: La dirección IP de la máquina donde se está ejecutando el servidor. Si es la misma máquina, usa 127.0.0.1.
- *<puerto>*: El mismo puerto que usaste para iniciar el servidor.

- **Ejemplo:**

```
./cliente 127.0.0.1 8080
```

# Lista de Comandos del Cliente

Una vez conectado, puedes usar los siguientes comandos.

Las operaciones de modificación (UPDATE, ADD, DELETE) deben realizarse dentro de una transacción.

- **GET <ID>**

- **Descripción:** Busca y muestra el registro con el ID especificado.
- **Ejemplo:**

```
GET 15
```

- **UPDATE <ID> <Nro\_Campo> <Nuevo\_Valor>**

- **Descripción:** Modifica un campo de un registro existente.
- **Parámetros:**
  - <Nro\_Campo>: 1 para *NOMBRE\_PRODUCTO*, 2 para *CANTIDAD*, 3 para *PRECIO*.
- **Ejemplo:**

```
UPDATE 15 2 100
```

- **ADD <NOMBRE>,<CANTIDAD>,<PRECIO>**

- **Descripción:** Agrega un nuevo registro. El ID se asigna automáticamente.
- **Ejemplo:**

```
ADD Tableta Gráfica,20,150.75
```

- **DELETE <ID>**

- **Descripción:** Elimina el registro con el ID especificado.
- **Ejemplo:**

```
DELETE 22
```

- **BEGIN TRANSACTION**

- **Descripción:** Inicia una transacción y solicita un bloqueo exclusivo sobre la base de datos.

- **COMMIT TRANSACTION**

- **Descripción:** Confirma los cambios realizados y libera el bloqueo.

- **HELP**

- **Descripción:** Muestra una lista de los comandos disponibles.

- **EXIT**

- **Descripción:** Se desconecta del servidor y cierra el cliente.