

## Documentación de Carga Masiva de Datos (Dummy Data)

**Proyecto:** Base de datos tam (Sistema de E-commerce)

**Fecha:** 13 de marzo de 2026

**Herramienta:** MariaDB / MySQL Stored Procedures

### 1. Objetivo

Poblar las tablas principales del sistema (productos y usuarios) con un volumen considerable de datos (500 registros por tabla) para realizar pruebas de rendimiento, paginación y diseño de interfaz de usuario sin necesidad de registro manual.

### 2. Metodología: Procedimientos Almacenados

Se optó por el uso de **Procedimientos Almacenados (Stored Procedures)** sobre scripts de INSERT masivos por las siguientes ventajas:

- **Eficiencia:** El motor de la base de datos procesa la lógica internamente.
- **Dinamicidad:** Uso de funciones como RAND() para evitar datos idénticos.
- **Escalabilidad:** Permite cambiar el número de iteraciones (ej. pasar de 500 a 1000) modificando una sola variable.

### 3. Resumen de Tablas Afectadas

Tabla	Registros	Lógica de Datos
productos	500	Se generaron nombres incrementales, precios aleatorios coherentes con su costo (margen del 30%) y se vincularon a las 7 categorías existentes.
usuarios	500	Se generaron identificaciones, correos y nombres de usuario únicos para evitar conflictos con las restricciones UNIQUE. Se asignaron roles (Admin/Cliente) y contraseñas encriptadas.

## 4. Detalles de Implementación Técnica

### A. Generación de Productos

Se implementó un bucle que calcula:

- **Subcategorías:** Asignación aleatoria entre los IDs 1 y 7 para mantener la integridad referencial con la tabla subcategorias.
- **Precios:** Valores flotantes entre **100.000** y **5.100.000**.
- **Integridad:** Se utilizaron las rutas de imágenes estándar `assets/img/` para facilitar la visualización en el frontend.

```
DELIMITER //
CREATE PROCEDURE CargarProductosMasivos()
BEGIN
  DECLARE i INT DEFAULT 1;
  DECLARE v_precio DECIMAL(10,2);
  DECLARE v_costo DECIMAL(10,2);
  WHILE i <= 500 DO
    SET v_precio = 100000 + (RAND() * (5100000 - 100000));
    SET v_costo = v_precio * 0.70;
    INSERT INTO productos (nombre, precio, marca, características_basicas, características_tecnicas, imagen, stock, costo, subcategoria_id, created_at, updated_at)
    VALUES (CONCAT('Producto Genérico de Prueba ', i), v_precio, 'Marca Dummy',
    CONCAT('Características básicas de prueba para el producto número ', i, '.'), 'Procesador genérico, RAM variable, almacenamiento de prueba.',
    CONCAT('assets/img/dummy_img_', i, '.jpg'),
    FLOOR(RAND() * 100),
    v_costo,
    FLOOR(1 + (RAND() * 7)), -- Asignación aleatoria de subcategoría 1-7
    NOW(),
    NOW()
    );
    SET i = i + 1;
  END WHILE;
END //
DELIMITER ;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0053 segundos.)

```
CREATE PROCEDURE CargarProductosMasivos() BEGIN DECLARE i INT DEFAULT 1; DECLARE v_precio DECIMAL(10,2); DECLARE v_costo DECIMAL(10,2); WHILE i <= 500 DO SET v_precio = 100000 + (RAND() * (5100000 - 100000)); SET v_costo = v_precio * 0.70; INSERT INTO productos (nombre, precio, marca, características_basicas, características_tecnicas, imagen, stock, costo, subcategoria_id, created_at, updated_at) VALUES (CONCAT('Producto Genérico de Prueba ', i), v_precio, 'Marca Dummy', CONCAT('Características básicas de prueba para el producto número ', i, '.'), 'Procesador genérico, RAM variable, almacenamiento de prueba.', CONCAT('assets/img/dummy_img_', i, '.jpg'), FLOOR(RAND() * 100), v_costo, FLOOR(1 + (RAND() * 7)), -- Asignación aleatoria de subcategoría 1-7 NOW(), NOW()); SET i = i + 1; END WHILE; END;
```

[Editar en línea](#) | [\[Editar\]](#) | [\[Crear código PHP\]](#)

### B. Generación de Usuarios

Para cumplir con las reglas de negocio y seguridad:

- **Seguridad:** Se utilizó el hash `$2y$10$...` que corresponde a la cadena "password" cifrada, permitiendo pruebas de login funcionales.
- **Unicidad:** Se concatenó el índice del bucle (i) a los campos de texto únicos para prevenir errores de duplicidad en email y user\_name.
- **Edad:** Se calculó la fecha\_nacimiento restando un intervalo aleatorio de entre 18 y 58 años a la fecha actual (`CURDATE()`).

```

DELIMITER //
CREATE PROCEDURE CargarUsuariosMasivos()
BEGIN
DECLARE i INT DEFAULT 1;
WHILE i <= 500 DO
INSERT INTO usuarios (nombre, tipo_documento, numero_documento, fecha_nacimiento, email, password, email_verified_at, user_name, imagen, rol_id, estado,
created_at,updated_at)
VALUES (CONCAT('Usuario de Prueba ', i),'CC',
CONCAT(FLOOR(1000000000 + (RAND() * 9000000000)), i),
DATE_SUB(CURDATE(), INTERVAL FLOOR(18 + (RAND() * 41)) YEAR),
CONCAT('usuario_', i, '_', FLOOR(RAND() * 9999), '@dominio.com'),
'$2y$10$92IXunpkj00r0Q5byMI.Ye4oKoEa3Ro911C/.og/at2.uhewG/igi', -- Hash de "password"
1,CONCAT('user_dummy_', i, '_', FLOOR(RAND() * 9999)),
CONCAT('assets/img/avatar_', i, '.png'), 2, 1,
NOW(),
NOW()
);
SET i = i + 1;
END WHILE;
END //
DELIMITER ;

```

## 5. Cómo Replicar

Para volver a generar estos datos en otro entorno:

1. Asegurarse de que las tablas roles y subcategorias tengan sus datos base.
2. Ejecutar el bloque de código del procedimiento.
3. Llamar al procedimiento con la instrucción CALL.
4. (Recomendado) Eliminar el procedimiento con DROP para no dejar objetos temporales en producción.

## Consideraciones y Advertencias

Durante el uso del sistema, puede presentarse la siguiente advertencia:

Advertencia: un formulario en esta página tiene más de 1000 campos. Al enviarlo se podrían ignorar algunos campos debido a la configuración «max\_input\_vars» de PHP.

Advertencia: un formulario en esta página tiene más de 1000 campos. Al enviarlo se podrían ignorar algunos campos debido a la configuración «max\_input\_vars» de PHP.

## **Presentado por equipo de testing**

- Alejandro Guerrero
- Samuel Possu
- Yorlan Castro