

### Descripción del Mecanismo de Carga Masiva de Datos

El objetivo de esta fase del trabajo integrador fue generar 500.000 registros ficticios por tabla principal de manera consistente, respetando el modelo relacional predefinido. Para lograrlo, se emplearon exclusivamente sentencias SQL y un procedimiento almacenado en MySQL, sin recurrir a herramientas externas o scripts de programación.

Se implementó el procedimiento almacenado `generamos_datos`, que utiliza un bucle `WHILE` para insertar filas de forma controlada. Esta elección se debió a la necesidad de mantener la integridad referencial entre las tablas `Empleado` y `Legajo`, ya que cada empleado debe estar asociado a un legajo válido mediante la clave foránea correspondiente.

Para garantizar la relación uno a uno (dado que el legajo es `UNIQUE` en `Empleado`), se aprovechó la secuencia incremental idéntica de los identificadores (`id` en `Legajo` e `i` en el bucle). Así, al insertar un registro en `Legajo` con `id = i`, se pudo insertar en `Empleado` con `legajo = i`, asegurando que todas las claves foráneas apunten a registros existentes. Este enfoque previene problemas de claves foráneas huérfanas y respeta la cardinalidad 1:1 entre ambas entidades.

Los valores de los campos se generaron utilizando funciones nativas de MySQL:

- **Número de legajo:** Formato 'LEGAJO-00000001' usando `LPAD(i, 8, '0')`.
- **Categorías (A a F) y áreas (QA, DEV, etc.):** Asignación aleatoria mediante `ELT()` y `RAND()`, simulando distribuciones realistas.
- **Estados (ACTIVO/INACTIVO):** Sesgo del 90% hacia "ACTIVO", coherente con la mayoría de los empleados activos en un sistema.
- **Fechas de alta e ingreso:** Generadas en rangos plausibles (entre 2022 y 2026 aproximadamente) usando `DATE_ADD` con intervalos aleatorios.
- **Correos electrónicos y nombres:** Ficticios pero estructurados (nombreX, apellidoX, emailX@empresa.com), facilitando la trazabilidad en pruebas.

Además, se incorporó una baja lógica mediante el campo `eliminado BOOLEAN DEFAULT FALSE`. Inicialmente, todos los registros se cargan como no eliminados, dejando abierta la posibilidad de simular borrados en futuras consultas sin afectar la integridad física de los datos.

Durante el desarrollo, se utilizó una herramienta de IA (como parte del acompañamiento pedagógico) para revisar la estrategia de integridad referencial y optimizar el uso de funciones como `RAND()` y `ELT()`. Esto contribuyó a evitar errores comunes, como la generación de valores fuera de rango o la ruptura de la unicidad.

El volumen final alcanza las 500.000 filas en cada tabla, lo que permitirá realizar análisis de rendimiento confiables en las siguientes etapas, especialmente al evaluar el impacto de los índices en consultas complejas.