



**INSTITUTO TECNOLÓGICO DE SANTO DOMINGO**

**Asignatura**

ING-ING231-04 EXPERIMENTACION EN INGENIERIA

**Estudiante**

Angel Concepcion Capellán 1124530

Pedro Angel Encarnación 112181

**Docente**

KATHERINE ROWENA CANDELIER FIGUERO

**Tarea**

Practica 2

**Fecha de entrega**

28/11/2025

Santo Domingo, D.N, República Dominicana

## 1. Entender y delimitar el problema en estudio

En un proyecto pequeño de desarrollo de software, un equipo de estudiantes de Ingeniería de Software está implementando una **API REST** que será usada por una aplicación web.

En las pruebas iniciales, se observa que **el tiempo de respuesta de la API aumenta mucho** cuando hay varios usuarios conectados al mismo tiempo, y no está claro si este desempeño depende más de la **configuración del servidor** o del **tipo de base de datos** que se use.

### Problema:

No sabemos cómo influyen el **número de usuarios concurrentes** y el **tipo de base de datos** en el **tiempo de respuesta promedio** de la API.

### Objetivo del experimento:

Determinar cómo afectan:

- El número de usuarios concurrentes (bajo / alto), y
  - El tipo de base de datos (SQLite / PostgreSQL)
- al tiempo de respuesta promedio (en milisegundos) de una API REST desplegada en un servidor de pruebas.

## 2. Elegir la variable de respuesta y verificar que se mida correctamente

- **Variable de respuesta (Y):**  
Tiempo de respuesta promedio de la API en milisegundos (ms) para un conjunto de peticiones.
- **Cómo se mide:**
  1. Se selecciona un endpoint de la API (por ejemplo, /api/productos).

2. Se usa una herramienta de prueba de carga (por ejemplo, JMeter o k6) para enviar varias solicitudes simultáneas.
3. Para cada combinación de factores (tratamiento), se ejecuta la prueba durante un tiempo fijo (por ejemplo, 2 minutos).
4. La herramienta reporta el **tiempo de respuesta promedio (ms)** de todas las peticiones en esa ejecución.

De esta manera, la variable de respuesta es **cuantitativa**, se mide con el mismo procedimiento en todos los tratamientos y permite comparar objetivamente los resultados.

### **3. Determinar los factores a estudiar y/o investigar**

#### **Factores controlables (los que vamos a manipular)**

1. **Factor A – Número de usuarios concurrentes**
  - A1: Carga baja (10 usuarios concurrentes)
  - A2: Carga alta (50 usuarios concurrentes)
2. **Factor B – Tipo de base de datos**
  - B1: SQLite
  - B2: PostgreSQL

#### **Factores no controlables (ruido, solo se reconocen)**

- Diferencias de rendimiento del hardware del servidor en distintos momentos.
- Variaciones de la red local o del Wi-Fi.
- Procesos del sistema operativo ejecutándose en segundo plano.
- Uso ocasional del equipo por otros programas durante la prueba.

Estos factores **no se controlan directamente**, pero se intenta que afecten lo menos posible ejecutando las pruebas en condiciones similares y aleatorizando el orden de los tratamientos.

#### 4. Seleccionar los niveles de cada factor y el diseño de experimento

Cada factor tendrá **2 niveles**, por lo que se utilizará un **diseño factorial completo 2×2**, similar al ejemplo del video y al documento de tu compañero.

Práctica 2. Introducción al dis...

Los tratamientos serán:

1. **T1 – A1B1:** 10 usuarios concurrentes, base de datos SQLite.
2. **T2 – A1B2:** 10 usuarios concurrentes, base de datos PostgreSQL.
3. **T3 – A2B1:** 50 usuarios concurrentes, base de datos SQLite.
4. **T4 – A2B2:** 50 usuarios concurrentes, base de datos PostgreSQL.

Si participan dos estudiantes en el laboratorio, cada uno puede ejecutar los cuatro tratamientos, de forma que haya **2 réplicas por tratamiento** (una por cada estudiante).

#### 5. Planear y organizar el diseño de experimentos

##### 1. Participantes:

- Estudiante 1 de Ingeniería de Software.
- Estudiante 2 de Ingeniería de Software.

##### 2. Materiales:

- Servidor o laptop con la API REST desplegada.
- Las dos bases de datos configuradas (SQLite y PostgreSQL) con la misma información.
- Herramienta de pruebas de carga (JMeter, k6 u otra).
- Hoja de registro de resultados (tabla de tratamientos y tiempos de respuesta).

### 3. Procedimiento general:

- Se define un **orden aleatorio** para los cuatro tratamientos (por ejemplo: T3 → T1 → T4 → T2) para evitar sesgos.
- Para cada tratamiento:
  1. Se configura la API para usar el tipo de base de datos correspondiente (SQLite o PostgreSQL).
  2. Se ajusta la herramienta de pruebas al número de usuarios concurrentes definido (10 o 50).
  3. Se ejecuta la prueba durante un tiempo fijo (ej. 2 minutos).
  4. Se registra el **tiempo de respuesta promedio (ms)** que da la herramienta.
- Cada estudiante repite el mismo conjunto de tratamientos, obteniendo así dos valores de tiempo promedio por tratamiento.

### 4. Registro de datos:

- Se crea una tabla con columnas:  
Estudiante, Tratamiento (T1, T2, T3, T4), Número de usuarios, Tipo de BD, Tiempo de respuesta promedio (ms).
- Luego esos datos se usarán para analizar qué combinación de factores produce **menor tiempo de respuesta**.

Tabla 1. Factores

Tipo de factor	Nombre del factor	Símbolo	Nivel 1	Nivel 2
Controlable	Número de usuarios concurrentes	A	A1: 10 usuarios	A2: 50 usuarios
Controlable	Tipo de base de datos	B	B1: SQLite	B2: PostgreSQL
No controlable	Rendimiento del hardware del servidor	—	Carga baja de otros procesos	Carga alta de otros procesos
No controlable	Estabilidad de la red	—	Latencia baja	Latencia a algo más alta

<b>No controlable</b>	Procesos del sistema operativo	—	Pocos procesos en segundo plano	Varios procesos en segundo plano
-----------------------	--------------------------------	---	---------------------------------	----------------------------------

Tabla 2. Tratamientos del experimento

<b>Tratamiento</b>	<b>Número de usuarios (A)</b>	<b>Tip o de BD (B)</b>	<b>Descripción del tratamiento</b>	<b>Réplicas *</b>
<b>T1</b>	A1: 10 usuarios	B1: SQLite	Prueba de carga con 10 usuarios usando una base de datos SQLite	2
<b>T2</b>	A1: 10 usuarios	B2: PostgreSQL	Prueba de carga con 10 usuarios usando una base de datos PostgreSQL	2
<b>T3</b>	A2: 50 usuarios	B1: SQLite	Prueba de carga con 50 usuarios usando una base de datos SQLite	2
<b>T4</b>	A2: 50 usuarios	B2: PostgreSQL	Prueba de carga con 50 usuarios usando una base de datos PostgreSQL	2

Tabla “Niveles y tratamientos”

Nivel de usuarios (A)	Nivel de BD (B)	Tratamiento	Y (tiempo de respuesta promedio en ms)
1	1	1	?
2	1	2	?
1	2	3	?
2	2	4	?

PRUEBAS DE CARGA SOBRE UNA API REST

