

## **Informe de Análisis y Diseño**

**Realizado por:** Kevin Rosero

### **Evaluación Práctica Progreso 1 – Caso BioNet**

#### **1. Identificación del Problema**

La empresa BioNet administra una red de laboratorios clínicos distribuidos en distintas ciudades, actualmente, los sistemas de laboratorio no están unificados, lo que genera una serie de problemas técnicos y operativos:

- Cada laboratorio genera archivos .csv de forma local, sin un estándar centralizado.
- Los archivos se copian manualmente en un servidor FTP compartido, lo que implica errores humanos frecuentes.
- El sistema central accede a una base de datos compartida, lo que genera problemas de concurrencia y sincronización.
- Se presentan inconsistencias frecuentes en los datos debido a:
  - Registros duplicados o sobrescritos.
  - Errores de sincronización al subir archivos incompletos.
  - Condiciones de carrera al intentar insertar múltiples archivos simultáneamente.

#### **2. Justificación de los Patrones Utilizados**

##### **Patrón: Transferencia de Archivos**

- Los laboratorios tienen conectividad limitada y no se puede implementar una API REST centralizada.
- Permite la transferencia segura y controlada de archivos .csv.
- Se pueden establecer reglas de validación para aceptar o rechazar archivos antes de su procesamiento.

### Patrón: Base de Datos Compartida

- Consolidar todos los resultados de exámenes en una única fuente de verdad.
- Facilitar reportes, consultas y auditorías desde un solo punto.
- Implementar triggers de auditoría, validación de duplicados, e integridad referencial.

### 3. Diseño de Alto Nivel de la Solución

- Estructura de Carpetas

#### Examen\_progreso\_1/

```
|— input-labs/      # Archivos .csv recibidos
|— processed/      # Archivos válidos procesados
|— error/          # Archivos inválidos
|— scripts/        # Scripts Python
|  |— process_files.py # Módulo de validación y clasificación
|  |— insert_to_db.py  # Módulo de ingesta a la base de datos
|  |— test_connection.py # script de prueba para la conexión
|— sql/            # Scripts SQL (opcional)
|— logs/           # (opcional) Logs del sistema
|— .env            # Configuración sensible del entorno
```

- **Flujo de Integración**

1. Los laboratorios generan archivos .csv con resultados de exámenes.
2. Los archivos se colocan en la carpeta input-labs/.
3. El módulo process\_files.py:
  - Valida que el archivo tenga la estructura esperada.
  - Mueve archivos válidos a processed/ e inválidos a error/.
4. El módulo insert\_to\_db.py:
  - Lee cada archivo en processed/.
  - Evita duplicados mediante verificación por paciente, tipo de examen y fecha.
  - Ejecuta un trigger que registra los cambios en la tabla log\_cambios\_resultados.

- **Esquema de Base de Datos Sugerido**

**Base de Datos:** BioNetDB

**Tabla:** resultados\_examenes

Campo	Tipo	Descripción
id	INT (PK)	Identificador único
laboratorio_id	INT	Código del laboratorio
paciente_id	INT	ID del paciente
tipo_examen	NVARCHAR(100)	Tipo de examen realizado
resultado	NVARCHAR(100)	Resultado del examen
fecha_examen	DATE	Fecha del examen

**Restricción:** índice único sobre (paciente\_id, tipo\_examen, fecha\_examen)

**Tabla:** log\_cambios\_resultados

Campo	Tipo	Descripción
id	INT (PK)	Identificador del log
operacion	NVARCHAR(10)	Tipo de operación: 'INSERT' o 'UPDATE'
paciente_id	INT	Paciente afectado
tipo_examen	NVARCHAR(100)	Tipo de examen afectado
fecha	DATETIME	Fecha y hora del cambio

**Trigger asociado:** trg\_log\_resultados para auditar operaciones sobre resultados\_examenes.

#### 4. Conclusión

La solución propuesta resuelve los principales problemas del sistema actual de BioNet mediante un diseño modular, seguro y funcional, cumple con todos los criterios de evaluación definidos:

- Control de archivos y errores
- Consolidación en una base de datos segura
- Validación y trazabilidad completa

Además, se han proporcionado archivos de prueba (buenos.csv, duplicados.csv, malos.csv) para demostrar el funcionamiento.