

PRUEBA TECNICA DE: Joel Amós Siquinajay Ojot

FECHA DE ENTREGA: 12/02/2025

DNC COFFEE

PROBLEMAS IDENTIFICADOS

IDENTIFICACION DE PROBLEMAS		
PARTE 1, PROBLEMAS DE RENDIMIENTO		
CONTROLADOR:	KardexController	Funcion: Index
PROBLEMA:	N+1 Queries: Se ejecuta un consulta para obtener todos los productos	1. Un query para sumar entradas. 2. Un query para sumar salidas 3. Un query para calcular promedio
	Cálculos redundantes	El método calcularCostoPromedio() se llama por cada producto, ejecutando otra consulta por producto
	Falta de índices	No hay índices optimizados en la tabla kardex para las consultas frecuentes.
	Impacto con 1,000 registros de productos:	- Queries ejecutados: $1 (\text{productos}) + 1,000 \times 2 (\text{entradas/salidas}) + 1,000 (\text{costo}) = 3,001 \text{ queries}$. - Tiempo estimado: Si cada query toma 40ms a 120 segundos (2 minutos aprox.) - Memoria: Cada producto crea arrays temporales en memoria.
	Estimación de queries ejecutados:	1 query: SELECT * FROM productos 1,000 queries: SUM(cantidad) FROM kardex WHERE producto_id = ? AND tipo = 'entrada' 1,000 queries: SUM(cantidad) FROM kardex WHERE producto_id = ? AND tipo = 'salida' 1,000 queries: SELECT * FROM kardex WHERE producto_id = ? AND tipo = 'entrada' TOTAL: 3,001 queries
	Propuesta:	- Utilizar paginación desde el inicio - Calculo sobre las 14 filas propuestas en el proceso. - Cuando el usuario pida la siguiente página se hace lo mismo.

SOLUCION EN PANTALLA:		<ul style="list-style-type: none"> - Agregar filtros a la pantalla de usuario para que pueda filtrar por fechas. - Agregar filtros por categoría o grupo de productos, por producto y cualquier otro que ayude a reducir la búsqueda
PROBLEMA DE NEGOCIO		
PROBLEMA	El costo promedio está mal calculado:	<p>El código actual calcula un promedio simple sobre todas las entradas</p> <p>No considera las salidas: Solo usa entradas, ignorando que algunos productos ya se vendieron/transformaron</p> <p>Orden incorrecto: Usa ORDER BY fecha DESC pero no aplica el método FIFO/LIFO correctamente</p> <p>Sin considerar stock actual: Un promedio ponderado debe considerar el stock disponible, no todas las compras históricas</p>
SOLUCION:	<p>Método de costeo recomendado: Promedio ponderado</p> <p>Formula promedio costo ponderado:</p>	<ul style="list-style-type: none"> - Aceptado por normas contables - Más práctico de implementar que FIFO/LIFO - Refleja mejor el costo real - Menos expuesto a manipulación <p>Costo Promedio = (Valor Inventario Actual) / (Cantidad Inventario Actual)</p> <p>Valor Inventario Actual = $\Sigma(\text{cantidad disponible} * \text{costo unitario})$</p> <p>Cantidad Inventario Actual = $\Sigma(\text{cantidad disponible})$</p>
PARTES 2, BUG DE PRODUCCION CRITICO		
CONTROLADOR	TransformacionController	Función: ProcesarTransformacion
PROBLEMA:	Misma cantidad para salida y para entrada	<ul style="list-style-type: none"> - Se está usando la variable \$cantidad_entrada tanto para la entrada como para la salida, ignorando por completo el factor de conversión o el valor real que debería tener el producto procesado - No hay registro de merma
MIGRACION DE DATOS		

PROUESTA:	Archivo:	dnc-erp-prueba-tecnica\docs\PROCESO_DE_MIGRACION.sql
PARTE 3, OPTIMIZACION DE CONSULTA PARA DASHBOARD		
PROUESTA:	Archivo:	dnc-erp-prueba-tecnica\docs\QUERY_DASHBOARD_PRINCIPAL.sql
TABLA MATERIALIZADA		
PROUESTA:	Archivo:	dnc-erp-prueba-tecnica\docs\TABLA_MATERIALIZADA.sql
INDICES		
PROUESTA:	Archivo:	dnc-erp-prueba-tecnica\docs\INDICES_PROPUESTOS_PARA_RENDIMIENTO.sql
PARTE 4, ARQUITECTURA DECISION TECNICA REAL		
PROUESTA:	Sistema de colas con Redis/RabbitMQ y cron Job de procesamiento por lotes	- Las colas garantizan entrega al menos una vez. - Se puede procesar en múltiples workers
	Arquitectura:	<p>Usuario → [Frontend] → [API: /facturas/generar-lote] → [Redis Queue] →</p> <p>→ [Worker PHP 1] → [SAT API] → [Worker PHP 2] → [SAT API] → [Worker PHP 3] → [SAT API] → [WebSocket Server] → [Frontend: Progreso]</p> <p>→ [Base de Datos: Estado facturas]</p>
BONUS DEBUGGING DE PROBLEMA BIZARRO		
PROUESTA:	Archivo:	dnc-erp-prueba-tecnica\docs\ReporteExcelController.php
CARGA DE DATOS DE PRUEBA		
INSERTS:	Archivo:	dnc-erp-prueba-tecnica\docs\SCHEMA_CON_DATA.sql Se ejecutaron los scripts indicados para carga de información para escenarios de pruebas.
CONCLUSIONES	- Esta prueba está basada en problemas reales que regularmente enfrentamos los desarrolladores, a través del recorrido se encuentra código legacy, bugs de producción, optimización de queries y decisiones arquitectónicas bajo presión.	

- Algo sumamente impórtate es entender el negocio para poder aplicar procesos de mejora continua.
- Existen muchas formas de mejorar el rendimiento donde seguramente requerirá pruebas exhaustivas y banlanceo de carga entre otros componentes.
- Es de suma importancia establecer canales de comunicación usuario cliente con área de informática para poder dar soluciones definitivas y no realizar tareas repetitivas.
- El recorrido de la vida es un aprendizaje constante, me ha servido esta prueba para avanzar en conocimiento. Ha sido un reto sumamente interesante como también ampliar conocimiento en este campo.