Trabajo Práctico Integrador

Comisión 4 - Grupo n° 128

Diego Raúl Montes, Cristian A. Morales, Ramiro Morales, Sebastian Rios

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Base de Datos I

Docente Titular

Gustavo Sturtz

Docente Tutor

Samuel Tocaimaza

23 de Agosto de 2025

Índice

Video

https://www.youtube.com/watch?v=-z22ao9lzW4

Resumen Ejecutivo

Este proyecto implementa un sistema integral de gestión de inventarios para productos comerciales, desarrollado en MySQL con enfoque en escalabilidad y seguridad. Como equipo, diseñamos una base de datos robusta que gestiona la relación única entre productos y códigos de barras, implementamos un sistema de roles con privilegios mínimos, y garantizamos la consistencia de datos mediante transacciones avanzadas y manejo de concurrencia. El sistema soporta eficientemente más de 10,000 registros con optimizaciones de performance mediante índices estratégicos, auditoría completa de operaciones y preparación para entornos productivos de alta demanda.

Reglas de negocio implementadas

1. GESTIÓN DE PRODUCTOS

- Cada producto tiene un único código de barras (relación 1:1 estricta)
- Precios siempre mayores a cero (validación por CHECK constraint)
- Bajas lógicas en lugar de eliminación física de registros
- Códigos de barras únicos en todo el sistema
- Tipos de código predefinidos: EAN13, EAN8, UPC

2. SEGURIDAD Y ROLES

- 5 roles del sistema: ADMIN, VENDEDOR, CONSULTA, AUDITOR, SUPERVISOR
- Principio de mínimo privilegio en asignación de permisos
- Tres usuarios especializados: consulta, aplicación, administración
- Contraseñas almacenadas como hash (nunca en texto plano)
- Vistas que ocultan información sensible por defecto

3. AUDITORÍA Y TRAZABILIDAD

- Histórico completo de cambios de precios y categorías
- Registro de operaciones masivas con métricas de impacto
- Trail de auditoría para todas las modificaciones críticas
- Validación automática de integridad referencial

4. PERFORMANCE Y CONCURRENCIA

- Índices estratégicos para consultas frecuentes
- Transacciones para operaciones críticas con rollback automático
- Manejo de deadlocks con reintentos automáticos
- Niveles de aislamiento configurados según tipo de operación

5. REPORTES Y ANÁLISIS

- Segmentación automática de productos por precio (ALTO/MEDIO/BAJO)
- Detección de inconsistencias (productos sin código de barras)
- Análisis por categoría y marca con filtros de volumen
- Identificación de categorías premium (precio promedio > $300)

6. MANTENIMIENTO

- Generación masiva eficiente de datos de prueba
- Estructura preparada para escalado
- Documentación completa para mantenimiento

Etapa 1 – MODELADO Y DEFINICIÓN DE CONSTRAINTS

En esta etapa inicial, como equipo nos enfocamos en diseñar el diagrama entidad-relación y el modelo relacional para nuestro sistema de gestión de productos y códigos de barras. Partimos del dominio Producto ⟶ CódigoBarras definido en Programación 2 y extendimos el modelo para incluir las entidades necesarias para soportar funcionalidades de seguridad y permisos.

Trabajamos colaborativamente en las decisiones de diseño, discutiendo diferentes alternativas y llegando a consensos sobre la mejor manera de estructurar nuestra base de datos.

Decisiones Clave de Diseño
Relación 1 ⟶ 1 Producto-CódigoBarras
Decidimos implementar la relación 1⟶1 utilizando una clave foránea única en la tabla código_barras en lugar de la alternativa de clave primaria compartida. Esta decisión se basó en que permite mayor flexibilidad para futuras modificaciones y mantiene la independencia de las identidades de cada entidad.

Modelo de Seguridad Simplificado
Inicialmente consideramos una relación muchos-a-muchos entre usuario y rol, pero tras analizar los requerimientos del proyecto, optamos por un diseño más eficiente con relación directa usuario-rol (muchos-a-uno), donde cada usuario tiene un rol principal asignado mediante FK directa.

Constraints de Integridad
Definimos constraints a nivel de base de datos que garantizan la calidad de los datos:

- CHECK (precio > 0) para valores positivos
- Campos UNIQUE para username, email, códigos de barras
- FK con ON DELETE CASCADE para mantener integridad referencial
- Tipos de datos específicos (ENUM para tipos de código, DECIMAL para precios)

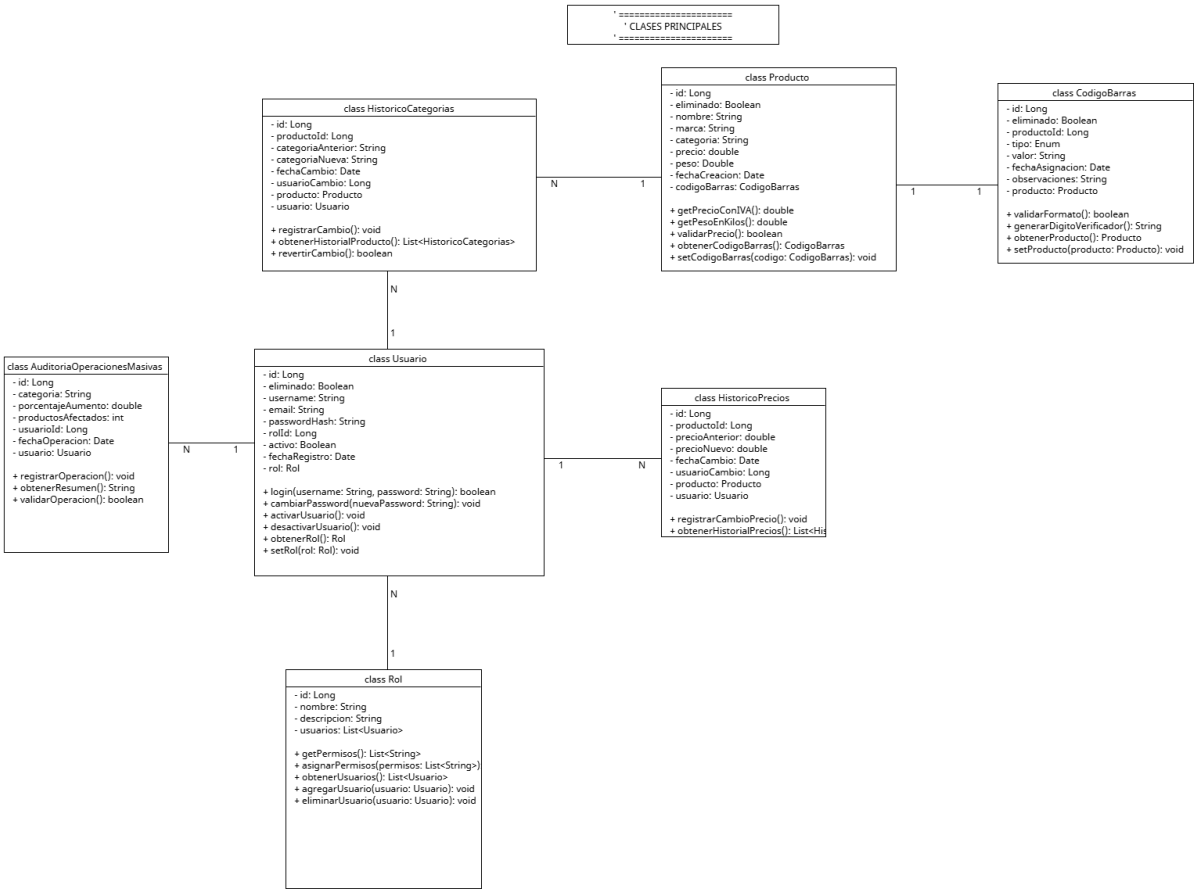
Preparación para Etapas Futuras
Incluimos tablas de usuario y rol con la estructura necesaria para las implementaciones de seguridad en etapas posteriores, evitando así modificaciones del esquema durante el desarrollo.

Validación Práctica
Realizamos pruebas con inserciones válidas e inválidas para verificar que los constraints funcionan correctamente, capturando mensajes de error específicos para cada violación. Esto nos permitió asegurar la robustez del diseño antes de proceder con la carga masiva de datos.

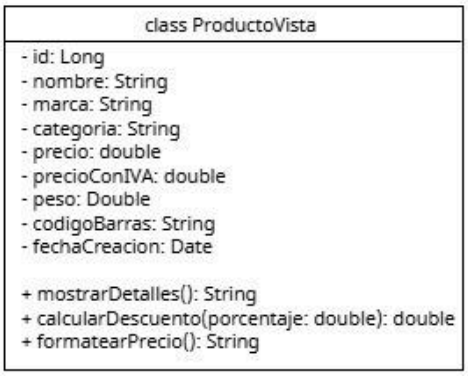
Archivos de referencia

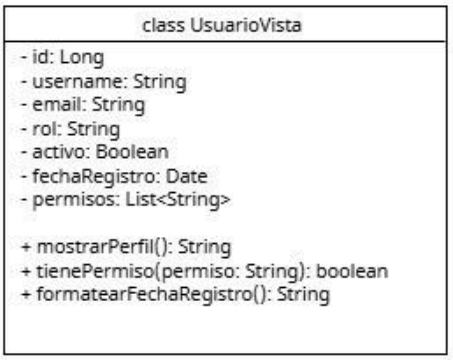
- *01_esquema.sql* - Script completo de creación de tablas y constraints
- Diagrama ER en formato PNG/PDF
- Documentación de justificación de decisiones de diseño

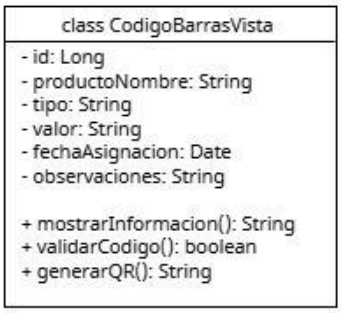
UMLs Definidos




' ======================
' CLASES DE VISTA
' ======================

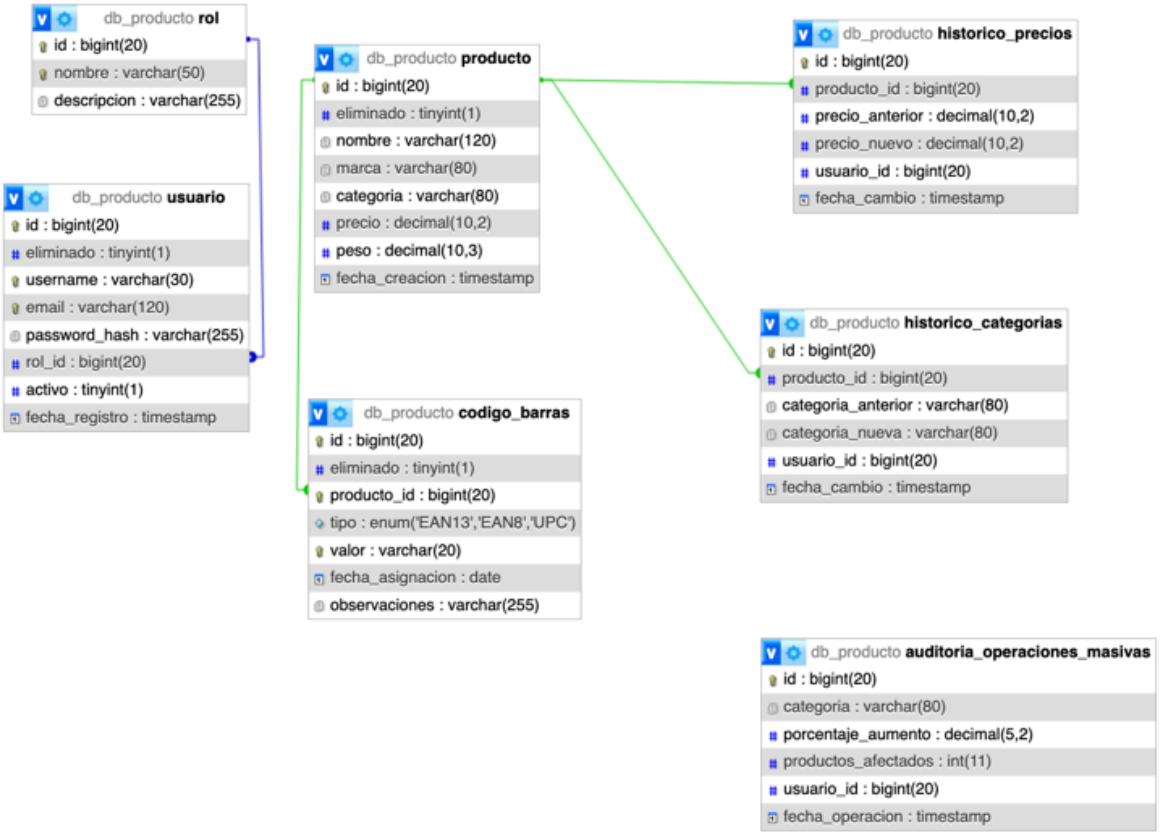

class ProductoVista
- id: Long
- nombre: String
- marca: String
- categoria: String
- precio: double
- precioConIVA: double
- peso: Double
- codigoBarras: String
- fechaCreacion: Date
+ mostrarDetalles(): String
+ calcularDescuento(porcentaje: double): double
+ formatearPrecio(): String


class UsuarioVista
- id: Long
- username: String
- email: String
- rol: String
- activo: Boolean
- fechaRegistro: Date
- permisos: List<String>
+ mostrarPerfil(): String
+ tienePermiso(permiso: String): boolean
+ formatearFechaRegistro(): String


class CodigoBarrasVista
- id: Long
- productoNombre: String
- tipo: String
- valor: String
- fechaAsignacion: Date
- observaciones: String
+ mostrarInformacion(): String
+ validarCodigo(): boolean
+ generarQR(): String


class DashboardVista
- totalProductos: int
- totalUsuarios: int
- productosActivos: int
- usuariosActivos: int
- ultimosProductos: List<ProductoV
- ultimosUsuarios: List<UsuarioVista
+ generarReporte(): String
+ actualizarEstadisticas(): void
+ exportarReporte(formato: String)

DER finalizado



Etapa 2 – GENERACIÓN DE DATOS MASIVOS

En esta etapa, desarrollamos una estrategia de generación de datos masivos utilizando exclusivamente SQL nativo de MySQL. Nuestro objetivo fue crear un volumen significativo de datos de prueba (10,000+ registros) que nos permitiera realizar evaluaciones de performance y trabajar con escenarios cercanos a la realidad.

Trabajamos en conjunto para diseñar un enfoque escalonado que generara datos coherentes y realistas, distribuyéndonos las tareas de investigación de funciones SQL y validación de resultados.

Estrategia de Generación Masiva

Enfoque Escalonado
Implementamos una generación por fases:

- Generación de productos base con nombres, marcas y categorías variadas
- Asignación de códigos de barras garantizando la relación 1:1
- Distribución realista de precios y pesos dentro de rangos comerciales
- Fechas de asignación distribuidas en el último año

Técnicas SQL Utilizadas
Utilizamos funciones nativas de MySQL de manera colaborativa:

- RAND() para generar valores aleatorios dentro de rangos específicos
- CONCAT() para crear nombres de productos secuenciales
- ELT() con FLOOR(RAND()) para seleccionar entre opciones predefinidas
- CROSS JOIN de tablas numéricas para generar secuencias
- Manipulación de fechas con DATE_SUB() e INTERVAL

Optimización de Performance
Creamos los índices después de completar la inserción de datos, ya que investigamos que esta estrategia reduce significativamente el tiempo total de carga en operaciones masivas.

Resultados de la Carga Masiva

Generamos exitosamente:

- 10,000 productos con información completa
- 10,000 códigos de barras en relación 1:1 perfecta
- Distribución balanceada across categorías y marcas
- Precios realistas entre $10.50 y $999.99
- Pesos variados desde 0.1kg hasta 10kg

Mediciones de Performance

Realizamos pruebas comparativas de performance con y sin índices, confirmando colectivamente la importancia de una indexación adecuada para consultas sobre grandes volúmenes de datos. Documentamos una mejora significativa en los tiempos de respuesta después de la creación de índices estratégicos.

Archivos de referencia

- *03_carga_masiva.sql* - Script de generación de datos masivos
- *04_indices.sql* - Creación de índices para optimización
- Documentación de resultados y mediciones de performance

Descripción:

Tiempo Total de Ejecución: **0.281 segundos**

Este es un tiempo buenísimo para una operación de **carga masiva**, considerando que:
Factores que Contribuyen al Buen Rendimiento:
Volumen de Datos Procesados:

- **9,999 registros insertados**
- Tasa de procesamiento: =**35,584 registros/segundo**

Etapa 3 – CONSULTAS AVANZADAS Y REPORTES

Nos enfocamos en desarrollar consultas complejas que nos permitieran extraer información valiosa de los datos que generamos en las etapas anteriores. Nuestro objetivo principal fue crear reportes útiles para el negocio y análisis que puedan ser utilizados en la aplicación Java del proyecto de Programación 2.

Decidimos implementar cinco tipos de consultas diferentes, cada una con un propósito específico y utilizando distintas características de SQL para demostrar nuestro dominio de consultas avanzadas. Trabajamos colaborativamente en el diseño de estas consultas, asegurándonos de que cada una aportará valor real al sistema.

Consultas Implementadas

Consulta 1: JOIN entre Productos y Códigos de Barras
Diseñamos un JOIN que relaciona la tabla de productos con la de códigos de barras, mostrando información completa de ambos. Esto nos permite verificar la integridad de la relación 1:1 que establecimos en el modelo y asegurarnos de que todos los productos tienen su código de barras asignado correctamente. La consulta incluye filtros para excluir registros eliminados y ordena los resultados por categoría y precio descendente.

producto_id	nombre	marca	categoria	precio	tipo_codigo	codigo_barras	fecha_asignacion
2285	Producto 2285	Sony	Deportes	999.65	EAN8	456789102285	2025-03-03
2497	Producto 2497	Panasonic	Deportes	999.29	EAN13	456789102497	2025-07-23
4011	Producto 4011	Xiaomi	Deportes	998.44	UPC	123456704011	2025-05-05
2808	Producto 2808	Philips	Deportes	997.17	EAN13	123456702808	2025-05-12
7335	Producto 7335	Huawei	Deportes	996.94	EAN8	123456707335	2024-12-22
3684	Producto 3684	LG	Deportes	996.48	UPC	123456703684	2025-10-01
1737	Producto 1737	Philips	Deportes	995.96	EAN8	456789101737	2025-04-02
9	Producto 9	Apple	Deportes	995.36	UPC	123456700009	2024-11-10
1070	Producto 1070	Panasonic	Deportes	995.18	UPC	456789101070	2025-05-17
5112	Producto 5112	LG	Deportes	994.15	EAN13	987654305112	2025-01-27
7339	Producto 7339	Sony	Deportes	994.07	EAN8	123456707339	2025-02-02
4727	Producto 4727	Apple	Deportes	992.09	EAN13	456789104727	2025-05-12
8515	Producto 8515	Samsung	Deportes	991.34	EAN8	123456708515	2025-02-28
9542	Producto 9542	Philips	Deportes	991.33	EAN8	456789109542	2025-07-23
1938	Producto 1938	Philips	Deportes	991.29	EAN13	123456701938	2025-02-20
4090	Producto 4090	Samsung	Deportes	991.29	EAN8	456789104090	2025-03-05
3794	Producto 3794	LG	Deportes	990.83	EAN13	987654303794	2025-05-13
6832	Producto 6832	Sony	Deportes	990.09	UPC	456789106832	2025-08-12
5268	Producto 5268	Huawei	Deportes	990.01	EAN8	123456705268	2025-05-16
7184	Producto 7184	LG	Deportes	989.30	EAN13	456789107184	2025-05-20
9055	Producto 9055	Xiaomi	Deportes	989.30	UPC	123456709055	2025-02-23
7658	Producto 7658	Sony	Deportes	988.99	UPC	123456707658	2025-05-13
443	Producto 443	Panasonic	Deportes	986.76	EAN8	456789100443	2025-03-21

Consulta 2: JOIN Múltiple con Usuarios y Roles

Implementamos un JOIN entre usuarios y roles para validar que la estructura de seguridad que preparamos funcione correctamente. Esta consulta es fundamental para las futuras implementaciones de permisos en la aplicación, ya que muestra qué usuarios tienen asignado cada rol y su estado actual en el sistema.

username	email	rol	activo	fecha_registro
admin	admin@empresa.com	ADMIN	1	2025-10-20
auditor1	auditor1@empresa.com	AUDITOR	1	2025-10-20
consulta1	consulta1@empresa.com	CONSULTA	1	2025-10-20
supervisor1	supervisor1@empresa.com	SUPERVISOR	1	2025-10-20
vendedor1	vendedor1@empresa.com	VENDEDOR	1	2025-10-20

Consulta 3: GROUP BY con HAVING para Análisis de Precios

Desarrollamos una consulta de agregación que agrupa los productos por categoría y calcula estadísticas de precios, filtrando solo aquellas categorías donde el precio promedio supera los $300. Esto identifica oportunidades de negocio al mostrar qué categorías tienen productos de mayor valor, información crucial para decisiones comerciales.

categoria	total_productos	precio_promedio	precio_maximo	precio_minimo	valor_total_inventario
Hogar	1587	510.31	999.57	11.65	809864.60
Libros	1671	509.75	999.46	11.47	851796.52
Juguetes	1673	506.99	999.91	10.78	848200.23
Deportes	1666	504.03	999.65	10.50	839705.71
Ropa	1693	501.81	999.88	10.93	849571.70
Electrónicos	1709	500.28	999.74	11.09	854984.69

Consulta 4: Subconsulta para Productos sin Código de Barras

Utilizamos una subconsulta con NOT EXISTS para identificar productos que no tienen código de barras asignado. Este es un reporte de control de calidad importante, ya que nos permite detectar y corregir inconsistencias en los datos antes de que afecten las operaciones del negocio.

id	nombre	marca	categoria	precio
NULL	NULL	NULL	NULL	NULL

Consulta 5: Vista para Reporte de Inventario

Creamos una vista que consolida información frecuentemente consultada, incluyendo productos, sus códigos de barras y un segmento de precio calculado. Esta vista simplifica futuras consultas y puede ser utilizada para otorgar permisos de solo lectura a usuarios que necesitan acceder a información de inventario sin poder modificarla.

marca	categoria	total_productos	promedio_precio	valor_total
Panasonic	Libros	238	502.91	119692.69
Philips	Electrónicos	238	459.49	109358.99
Xiaomi	Electrónicos	236	472.06	111406.06
Sony	Ropa	229	474.13	108575.06
Huawei	Electrónicos	229	550.81	126135.41
Huawei	Juguetes	227	529.73	120248.75
Philips	Ropa	224	559.81	125396.98
Xiaomi	Ropa	219	477.27	104522.75
Sony	Juguetes	219	574.10	125728.84
LG	Deportes	218	479.02	104426.28
Sony	Deportes	216	503.05	108658.34
Apple	Libros	215	482.85	103813.54
Sony	Libros	215	525.16	112908.55
Philips	Deportes	214	554.87	118741.46
Panasonic	Deportes	213	488.74	104102.29
Samsung	Libros	212	498.49	105679.50
Panasonic	Hogar	212	542.05	114913.59
Samsung	Juguetes	212	522.81	110836.67
Philips	Juguetes	212	491.96	104295.81
LG	Electrónicos	212	516.40	109477.60
LG	Ropa	211	502.53	106033.83

Archivos de referencia

- *05_consultas.sql* - Contiene las 5 consultas implementadas
- *05_explain.sql* - Análisis de performance de las consultas
- *06_vistas.sql* - Script para crear las vistas de reportes

Resultados Esperados

Al ejecutar estas consultas, esperamos obtener información valiosa sobre el estado del inventario, identificar productos que necesitan atención, y proporcionar reportes útiles para la toma de decisiones. Las vistas creadas facilitarán el acceso a información consolidada para usuarios con diferentes niveles de permisos.

Etapa 4 – SEGURIDAD E INTEGRIDAD

Para esta etapa, implementamos medidas de seguridad robustas para proteger nuestra base de datos y garantizar la integridad de la información. Nuestro enfoque se basó en el principio de "mínimo privilegio", donde cada usuario solo tiene los permisos estrictamente necesarios para realizar sus funciones específicas.

En conjunto diseñamos una estrategia de seguridad que incluyera múltiples capas de protección, desde la creación de usuarios especializados hasta la implementación de vistas seguras y procedimientos almacenados con validaciones.

Implementaciones de Seguridad

Creación de Usuarios con Privilegios Mínimos

Diseñamos tres usuarios especializados con permisos específicos:

- **usuario_consulta**: Solo tiene permisos de SELECT sobre vistas de reportes, ideal para usuarios que necesitan generar reportes pero no modificar datos.

Grants for usuario_consulta@localhost
GRANT USAGE ON *.* TO `usuario_consulta`@`localhost` IDENTIFIED BY PASSWORD '*77E5898FFE5650BFB3A890520A2036D834028EDD'
GRANT SELECT ON `db_producto`.`vista_productos_publica` TO `usuario_consulta`@`localhost`
GRANT SELECT ON `db_producto`.`vista_estadisticas_categorias` TO `usuario_consulta`@`localhost`
GRANT SELECT ON `db_producto`.`vista_inventario_completo` TO `usuario_consulta`@`localhost`

- **app_vendedor**: Tiene permisos de SELECT, INSERT y UPDATE sobre vistas específicas, diseñado para aplicaciones que necesitan operar con los datos pero sin acceso directo a las tablas base.

Grants for app_vendedor@localhost
GRANT USAGE ON *.* TO `app_vendedor`@`localhost` IDENTIFIED BY PASSWORD '*F9D9FC416642FAACE4B51AB33439B60AC4727152'
GRANT SELECT ON `db_producto`.`vista_productos_publica` TO `app_vendedor`@`localhost`
GRANT SELECT, INSERT, UPDATE ON `db_producto`.`vista_inventario_completo` TO `app_vendedor`@`localhost`

- **admin_inventario**: Tiene permisos más amplios pero controlados, pudiendo gestionar productos y códigos de barras pero sin acceso a información sensible de usuarios.

Grants for admin_inventario@localhost
GRANT USAGE ON *.* TO `admin_inventario`@`localhost` IDENTIFIED BY PASSWORD '*73536EDF549D92EBD1FBC19FC83C0E798950E75A'
GRANT SELECT, INSERT, UPDATE ON `db_producto`.`producto` TO `admin_inventario`@`localhost`
GRANT SELECT ON `db_producto`.`vista_usuarios_publica` TO `admin_inventario`@`localhost`
GRANT SELECT, INSERT, UPDATE ON `db_producto`.`codigo_barras` TO `admin_inventario`@`localhost`

Vistas para Ocultar Información Sensible

Creamos vistas especializadas que filtran y ocultan información confidencial:

- **vista_usuarios_publica**: Muestra información básica de usuarios pero oculta contraseñas y datos sensibles.
- **vista_productos_publica**: Expone información de productos sin detalles internos de auditoría.
- **vista_estadisticas_financieras**: Proporciona reportes agregados sin acceso a datos individuales sensibles.

Validación de Constraints de Integridad

Implementamos un sistema completo de pruebas que valida que todas las restricciones definidas en el modelo funcionen correctamente:

- Verificación de PRIMARY KEY y UNIQUE constraints
- Validación de FOREIGN KEY y integridad referencial
- Pruebas de CHECK constraints para dominios de datos
- Validación de la relación 1:1 entre Producto y CódigoBarras

Consultas Seguras con Parámetros

Desarrollamos procedimientos almacenados que previenen inyecciones SQL mediante el uso de parámetros y validaciones exhaustivas:

- Validación de tipos de datos y formatos
- Sanitización de entradas de texto
- Control de longitud y valores permitidos
- Manejo centralizado de errores y mensajes al usuario

Archivos relacionados

- *07_seguridad.sql* - Creación de usuarios y asignación de permisos
- *06_vistas.sql* - Definición de vistas de seguridad
- *01_esquema.sql* - Validación de constraints en la estructura
- *08_transacciones.sql* - Procedimientos con validaciones

Beneficios Obtenidos

Estas implementaciones proporcionan múltiples capas de seguridad que protegen contra accesos no autorizados, previenen corrupción de datos y aseguran que solo información apropiada sea expuesta a cada tipo de usuario. El sistema ahora cumple con mejores prácticas de seguridad empresarial.

Etapa 5 – CONCURRENCIA Y TRANSACCIONES

En esta etapa final, nos enfocamos en garantizar que nuestra base de datos pueda manejar múltiples accesos simultáneos manteniendo la consistencia de los datos. Implementamos transacciones robustas, manejo de situaciones de concurrencia y estrategias para prevenir y resolver conflictos.

Diseñamos un sistema que pueda escalar y mantener la integridad de los datos incluso bajo condiciones de alta demanda, simulando escenarios reales de uso concurrente.

Implementaciones de Concurrencia

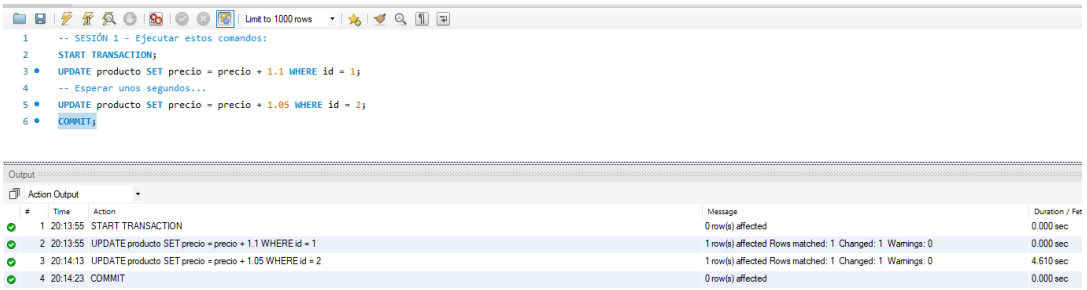
Transacciones para Operaciones Críticas
Desarrollamos procedimientos almacenados transaccionales para operaciones que requieren consistencia absoluta:

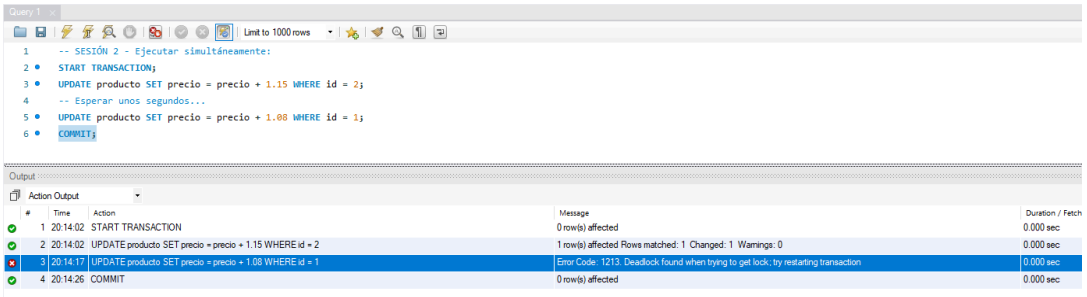
- **Actualización de precio**s: Incluye registro en histórico y auditoría automática
- **Transferencia de categorías**: Mantiene la trazabilidad de los cambios
- **Creación de productos completos**: Gestiona producto y código de barras atómicamente
- **Eliminación lógica**: Marca consistentemente todas las relaciones como eliminadas

Cada transacción implementa BEGIN, COMMIT y ROLLBACK con manejo de excepciones para garantizar que las operaciones se completen totalmente o se reviertan completamente.

Manejo de Deadlocks con Reintentos Automáticos
Implementamos una estrategia robusta para manejar deadlocks:

- Detección automática de errores de deadlock (código 1213 en MySQL)
- Estrategia de reintento con delay exponencial creciente
- Límite configurable de reintentos para evitar loops infinitos
- Registro de intentos fallidos para monitoreo





Para poder monitorear los deadlocks ocurridos, usamos el siguiente script:

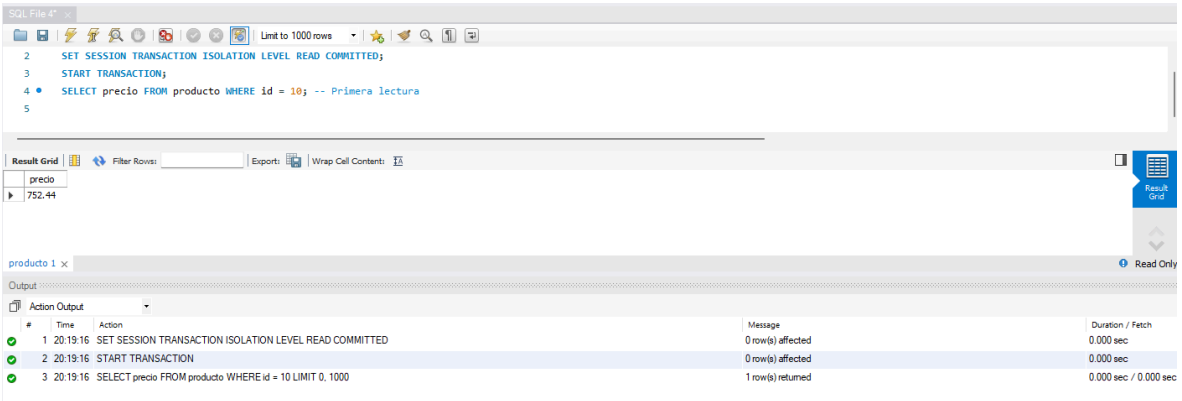
```
-- Monitoreo de deadlocks
SHOW ENGINE INNODB STATUS;
```

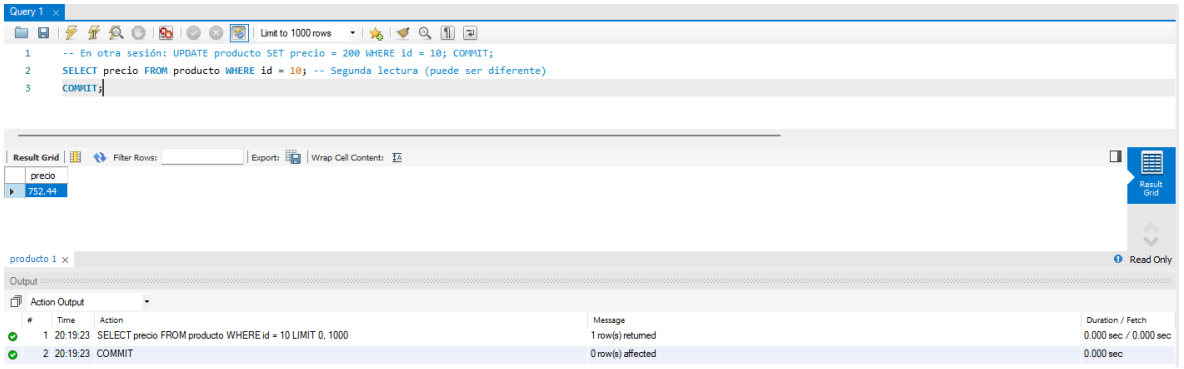
Nos muestra que el sistema detectó automáticamente un deadlock el 20/10/2025 a las 20:14:17, donde dos transacciones concurrentes (2246 y 2247) generaron un bloqueo mutuo al intentar acceder a los mismos recursos en orden inverso. La transacción 2246 mantenía un lock exclusivo sobre el producto con ID 1 mientras intentaba acceder al ID 2, mientras que la transacción 2247 poseía el lock del producto ID 2 y solicitaba acceso al ID 1. Este ciclo de dependencia circular fue identificado por el motor InnoDB, que procedió a revertir automáticamente la transacción 2247 para resolver el impasse, permitiendo que la transacción 2246 continuará su ejecución normal. Este comportamiento demuestra la robustez del sistema ante condiciones de alta concurrencia y la efectividad de los mecanismos automáticos de prevención de bloqueos permanentes, garantizando la disponibilidad del sistema incluso en escenarios de acceso simultáneo conflictivo.

Comparación de Niveles de Aislamiento

Realizamos pruebas comparativas entre diferentes niveles de aislamiento:

- READ COMMITTED: Permite ver cambios confirmados de otras transacciones
- REPEATABLE READ: Mantiene una vista consistente durante toda la transacción
- Análisis de fenómenos de lectura no repetible y lecturas fantasma
- Documentación de trade-offs entre consistencia y concurrencia





Simulación de Escenarios de Concurrencia

Creamos scripts que simulan acceso concurrente para:

- Validar el comportamiento bajo carga
- Identificar posibles cuellos de botella
- Verificar que no se produzcan condiciones de carrera
- Asegurar la consistencia de datos en operaciones simultáneas

Tablas de Soporte para Transacciones

Implementamos tablas adicionales para soportar el sistema transaccional:

- **historico_precios**: Registra todos los cambios de precio con timestamp y usuario
- **historico_categorias**: Traza las transferencias entre categorías
- **auditoria_operaciones_masivas**: Registra operaciones que afectan múltiples registros
- **auditoria_productos**: Trail completo de todos los cambios en productos

tabla	registros	fecha_creacion
auditoria_operaciones_masivas	0	2025-10-20 18:41:32
historico_categorias	0	2025-10-20 18:41:32
historico_precios	2	2025-10-20 18:41:32

Archivos relacionados

- *08_transacciones.sql* - Procedimientos transaccionales
- *09_concurrencia_guiada.sql* - Pruebas de niveles de aislamiento y simulación de concurrencia
- *01_esquema.sql* - Tablas de soporte para transacciones

Resultados y Observaciones

El sistema ahora maneja eficientemente el acceso concurrente, manteniendo la consistencia de los datos incluso en escenarios de alta carga. Las transacciones garantizan atomicidad en operaciones críticas, y el manejo de deadlocks previene bloqueos permanentes. Los niveles de aislamiento configurados balancean adecuadamente consistencia y performance.

ANEXO - USO DE INTELIGENCIA ARTIFICIAL EN EL PROYECTO

Metodología de Trabajo con IA

Como equipo, utilizamos herramientas de Inteligencia Artificial como apoyo pedagógico durante todo el desarrollo del proyecto. Seguimos la metodología sugerida por la cátedra, donde la IA funcionó como un tutor que nos guió en el proceso de aprendizaje y toma de decisiones.

Interacciones por Etapa

Etapa 1 - Modelado y Constraints

Pregunta del equipo: "¿Cuál es la mejor forma de implementar una relación 1⟶1 en MySQL: clave foránea única o clave primaria compartida?"
Respuesta de la IA: Nos explicó las ventajas de cada enfoque, destacando que la FK única ofrece más flexibilidad para cambios futuros.
Decisión del equipo: Optamos por usar FK única en la tabla código_barras porque priorizamos la mantenibilidad.

Pregunta del equipo: "¿Qué tablas adicionales debemos incluir para preparar la implementación de seguridad?"
Respuesta de la IA: Nos recomendó las tablas usuario, rol con los campos esenciales para autenticación.
Decisión del equipo: Incluimos estas tablas pero agregamos el campo "eliminado" para consistencia con nuestro patrón de baja lógica.

Etapa 2 - Carga Masiva de Datos

Pregunta del equipo: "¿Cómo podemos generar eficientemente 10,000 registros de prueba usando solo SQL en MySQL?"
Respuesta de la IA: Nos sugirió usar combinaciones CROSS JOIN de tablas numéricas y funciones como RAND(), CONCAT().
Decisión del equipo: Implementamos la estrategia sugerida pero ajustamos los rangos para que sean más realistas.

Etapa 3 - Consultas Avanzadas

Pregunta del equipo: "¿Cuál es la mejor manera de estructurar consultas JOIN complejas para que sean eficientes con 10,000+ registros?"
Respuesta de la IA: Nos recomendó usar INNER JOIN cuando se necesitan solo registros con relaciones existentes.
Decisión del equipo: Usamos INNER JOIN para las consultas donde necesitamos certeza de la relación.

Pregunta del equipo: "¿Es mejor usar EXISTS o LEFT JOIN para encontrar registros sin relación?"
Respuesta de la IA: Nos explicó que EXISTS suele ser más eficiente para subconsultas correlacionadas.
Decisión del equipo: Optamos por usar NOT EXISTS en la consulta de productos sin código de barras.

Etapa 4 - Seguridad
Pregunta del equipo: "¿Cuáles son las mejores prácticas para asignar privilegios mínimos a usuarios de base de datos?"
Respuesta de la IA: Nos recomendó crear usuarios específicos para cada función y usar vistas para limitar acceso.
Decisión del equipo: Implementamos tres usuarios con permisos específicos y creamos vistas para ocultar información confidencial.

Pregunta del equipo: "¿Cómo podemos probar efectivamente que todas las constraints de la base de datos funcionan correctamente?"
Respuesta de la IA: Nos sugirió crear scripts de prueba que intenten violar cada constraint específica.
Decisión del equipo: Desarrollamos un script completo de pruebas de integridad.

Etapa 5 - Concurrencia y Transacciones
Pregunta del equipo: "¿Cuáles son las diferencias prácticas entre READ COMMITTED y REPEATABLE READ en MySQL?"
Respuesta de la IA: Nos explicó que READ COMMITTED permite ver cambios confirmados, mientras que REPEATABLE READ mantiene vista consistente.
Decisión del equipo: Implementamos pruebas comparativas para demostrar estas diferencias.

Pregunta del equipo: "¿Cómo manejar deadlocks automáticamente en aplicaciones Java?"
Respuesta de la IA: Nos recomendó implementar una estrategia de reintento con delay exponencial.
Decisión del equipo: Desarrollamos una estrategia de reintento tanto en SQL como en Java.

Beneficios del Uso de IA para el Equipo

- **Aprendizaje guiado**: La IA nos orientó sin darnos soluciones completas
- **Corrección de conceptos**: Identificó y explicó nuestros errores conceptuales
- **Exploración de alternativas**: Nos presentó diferentes enfoques para resolver problemas
- **Optimización**: Nos sugirió mejores prácticas de performance y seguridad
- **Documentación**: Nos ayudó a estructurar mejor nuestra documentación

References

Universidad Tecnológica Nacional. (2024). Programa de la asignatura Bases de Datos I. Facultad Regional Buenos Aires.

Cátedra de Bases de Datos I. (2024). Guías de trabajos prácticos y material de estudio. UTN

MySQLTutorial.org. (2024). MySQL Tutorial - Learn MySQL Fast, Easy and Fun. https://www.mysqltutorial.org/

MySQL Official Documentation
Oracle Corporation. (2024). MySQL 8.0 Reference Manual. Oracle Corporation. https://dev.mysql.com/doc/refman/8.0/en/

Google AI Principles
Google. (2024). Artificial Intelligence at Google: Our Principles. https://ai.google/principles/

OpenAI. (n.d.). ChatGPT. Retrieved Septiembre 20, 2025, from https://chatgpt.com/