

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN
Y CIENCIAS DE LA COMPUTACIÓN
BASE DE DATOS 1
ING. JUAN ALVARO DIAZ ARDAVIN



PROYECTO FASE 2

Nombre	Carné
Cruz Francisco Estrada Gregorio	7960-23-5339

04/10/2025

Tabla de contenido

Sistema de Control - Empresa de Confección Internacional	3
1. CENSO DE ENTIDADES	3
1.1 País	3
1.2 Empresa	3
1.3 Localidad	3
1.4 Terceros	4
1.5 Usuario_portal	4
1.6 Empleado	4
1.7 Planilla	5
1.8 Artículo	5
1.9 Inventario	6
1.10 Activo_Fijo	6
1.11 Pedido	6
1.12 Detalle_pedido	7
1.13 Aprobación	7
1.14 Orden_produccion	7
1.15 Despacho	8
1.16 Ruta	8
1.17 Movimiento	9
1.18 Tipo	9
2. REVISIÓN DE ENTIDADES	9
2.1 Generalización / Particularización	9
2.2 Relaciones Identificadas	11
3. RAZONAMIENTO DEL DISEÑO	13
3.1 Jerarquía Organizacional (País → Empresa → Localidad)	13
3.2 Tabla Terceros: Generalización de Clientes y Proveedores	13
3.3 Movimiento: Tabla Universal de Transacciones	14
3.4 Detalle_pedido como Puente	15
3.5 Orden_produccion: Planificación vs Ejecución	15
3.6 Ruta: Optimización de Logística	16
3.7 Tipo: Tabla Catálogo Universal	17
3.8 Auditoría y Trazabilidad	18
4. ELABORACIÓN DEL DIAGRAMA E/R	19
5. SCRIPTS DE CREACIÓN DE TODOS LOS OBJETOS DE LA BASE DE DATOS DBMS DE ORACLE	20
5.1 Creación De Tablas	20
5.2 Vistas	25
5.3 Insercion De Datos	26

Sistema de Control - Empresa de Confección Internacional

1. CENSO DE ENTIDADES

Se identificaron **18 entidades principales** necesarias para el sistema integral de control. Cada entidad representa un objeto de negocio con ciclo de vida propio y atributos únicos.

1.1 País

Descripción: Representa cada país donde la empresa tiene operaciones. Es el nivel más alto de la jerarquía organizacional.

Atributos:

- codigo_pais (PK)
- nombre
- numero_area
- moneda

Justificación: Necesaria para consolidación a nivel internacional, manejo de múltiples monedas y cumplimiento de regulaciones por país.

1.2 Empresa

Descripción: Representa las filiales o subsidiarias de la organización en cada país.

Atributos:

- codigo_empresa (PK)
- codigo_pais (FK)
- nombre
- identificador_fiscal
- tipo_empresa (matriz, filial)

Justificación: Permite tener múltiples empresas legales por país, cada una con su propio identificador fiscal para cumplimiento tributario local.

1.3 Localidad

Descripción: Representa ubicaciones físicas (fábricas, bodegas, oficinas) donde opera la empresa.

Atributos:

- codigo_localidad (PK)
- codigo_empresa (FK)
- nombre

- tipo_localidad (fábrica, bodega, oficina)
- direccion

Justificación: Permite control granular de inventarios, activos y operaciones por ubicación física. Esencial para tracking de transferencias.

1.4 Terceros

Descripción: Entidad generalizada que representa tanto Clientes como Proveedores, evitando duplicación.

Atributos:

- codigo_tercero (PK)
- nombre
- codigo_pais (FK)
- codigo_cliente_aprobador (FK recursiva)
- telefono
- direccion
- tipo_tercero (Cliente, Proveedor, Ambos)
- cuenta_bancaria

Justificación: Generalización - Un tercero puede ser cliente Y proveedor. La relación recursiva permite jerarquías de aprobación. La cuenta bancaria facilita pagos/cobros.

1.5 Usuario_portal

Descripción: Credenciales de acceso al portal web para que clientes realicen pedidos.

Atributos:

- codigo_usuario_portal (PK)
- codigo_empleado (FK)
- codigo_tercero (FK)
- estado
- tipo_usuario
- usuario
- clave

Justificación: Separa la autenticación del negocio. Un usuario puede ser empleado interno o cliente externo.

1.6 Empleado

Descripción: Personal que labora en las diferentes localidades de la empresa.

Atributos:

- codigo_empleado (PK)
- codigo_empresa (FK)
- codigo_pais (FK)
- codigo_localidad (FK)
- nombre
- puesto
- salario_base
- fecha_ingreso
- es_aprobador (si/no)

Justificación: Central para planilla, aprobaciones y auditoría. El campo es_aprobador identifica quién puede aprobar pedidos/órdenes.

1.7 Planilla

Descripción: Registro de pagos de salarios a empleados por periodo.

Atributos:

- codigo_planilla (PK)
- codigo_empresa (FK)
- codigo_empleado (FK)
- salario_bruto
- descuentos
- salario_netto
- fecha_pago
- periodo_semanal_quincenal_mensual

Justificación: Requerimiento explícito - auditar gastos en planilla. Permite cálculo de costos laborales por empresa/localidad.

1.8 Artículo

Descripción: Catálogo maestro de todos los items: materia prima, productos semi-elaborados y productos terminados.

Atributos:

- codigo_articulo (PK)
- nombre
- tipo_articulo (materia prima, producto terminado, Servicio)
- unidad_medida (paquete, libra, unidad)
- precio_referencia

Justificación: Tabla maestra. La clasificación por tipo_articulo es esencial para producción. Incluye también "servicios" para movimientos de efectivo en la tabla de Movimientos.

1.9 Inventario

Descripción: Stock actual de artículos en cada localidad.

Atributos:

- codigo_inventario (PK)
- codigo_localidad (FK)
- codigo_articulo (FK)
- cantidad_actual
- cantidad_reservada

Justificación: Control de existencias por ubicación. cantidad_reservada permite gestión de compromisos (pedidos pendientes).

1.10 Activo_Fijo

Descripción: Bienes de capital (maquinaria, vehículos, edificios) propiedad de la empresa.

Atributos:

- codigo_activo_fijo (PK)
- codigo_empresa (FK)
- codigo_localidad (FK)
- descripcion
- valor_compra
- fecha_compra
- depreciacion

Justificación: Requerimiento - auditar valor de activos fijos. Permite cálculo de valor contable y depreciación por localidad.

1.11 Pedido

Descripción: Solicitud de productos realizada por un cliente, ya sea vía portal o medio tradicional.

Atributos:

- codigo_pedido (PK)
- codigo_cliente (FK → Terceros)
- fecha_pedido
- fecha_requerida

- estado
- origen (portal web, normal)

Justificación: Central al negocio. Origen diferencia pedidos automáticos vs manuales. Estado permite workflow de aprobación.

1.12 Detalle_pedido

Descripción: Líneas individuales de cada pedido, especificando artículos y cantidades.

Atributos:

- codigo_detalle_pedido (PK)
- codigo_pedido (FK)
- codigo_articulo (FK)
- cantidad_solicitada
- cantidad_despachada

Justificación: Normalización - un pedido tiene múltiples líneas. cantidad_despachada permite despachos parciales y tracking.

1.13 Aprobación

Descripción: Registro de decisiones de aprobación/rechazo de pedidos por personal autorizado.

Atributos:

- codigo_aprobacion (PK)
- codigo_pedido (FK)
- codigo_empleado (FK)
- fecha
- resultado (aprobado, rechazado)
- comentarios

Justificación: Requerimiento - alta gerencia aprueba/rechaza pedidos. Proporciona auditoría de decisiones y justificaciones.

1.14 Orden_produccion

Descripción: Planificación de manufactura para producir artículos necesarios para cumplir pedidos.

Atributos:

- codigo_orden_produccion (PK)
- codigo_producto_terminado (FK → Artículo)

- codigo_materia_prima (FK → Articulo)
- codigo_localidad (FK)
- codigo_pedido (FK)
- fecha_inicio
- fecha_estimada
- fecha_real
- estado (planificada, en proceso, completa, cancelada)

Justificación: Requerimiento - calcular tiempos de producción. Relaciona qué se produce, dónde, para qué pedido y en qué timeframe.

1.15 Despacho

Descripción: Envío físico de mercancía entre localidades o hacia clientes.

Atributos:

- codigo_despacho (PK)
- codigo_pedido (FK)
- codigo_localidad_origen (FK)
- codigo_localidad_destino (FK)
- fecha_envio
- fecha_estimada
- fecha_real
- estado

Justificación: Requerimiento - coordinar entrega entre fábricas y cliente final. Tracking de logística con fechas estimadas vs reales.

1.16 Ruta

Descripción: Definición de caminos de transporte entre localidades con tiempos y costos.

Atributos:

- codigo_ruta (PK)
- codigo_localidad_origen (FK)
- codigo_localidad_destino (FK)
- tiempo_horas
- costo_transporte
- tipo_transporte (aereo, maritimo, terrestre)

Justificación: Requerimiento - tiempo estimado de transporte. Permite cálculo automático de fecha_estimada en despachos.

1.17 Movimiento

Descripción: Registro transaccional de todas las operaciones: entrada/salida de inventario, transferencias, y movimientos de efectivo.

Atributos:

- codigo_movimiento (PK)
- tipo_movimiento (FK → Tipo)
- codigo_articulo (FK)
- codigo_localidad_origen (FK)
- codigo_localidad_destino (FK)
- codigo_pedido (FK)
- codigo_planilla (FK)
- fecha
- monto
- cantidad

Justificación: Tabla universal de transacciones. Consolida inventario y efectivo usando artículos tipo "servicio". Permite auditoría completa de flujo de efectivo y materiales.

1.18 Tipo

Descripción: Tabla catálogo universal para clasificaciones reutilizables en el sistema.

Atributos:

- codigo_tipo (PK)
- descripcion1
- descripcion2
- campo

Justificación: Patrón de diseño - evita múltiples tablas pequeñas de catálogo. Soporta tipos de movimiento, tipos de empleado, tipos de localidad, etc.

2. REVISIÓN DE ENTIDADES

2.1 Generalización / Particularización

Generalización 1: Terceros (Cliente + Proveedor)

Tipo: Generalización Total

Análisis: Se unificaron las entidades Cliente y Proveedor en una sola entidad "Terceros" porque:

- Comparten la mayoría de atributos (nombre, dirección, teléfono, país)

- En la práctica empresarial, un mismo tercero puede actuar en ambos roles
- Ejemplo: Un proveedor de tela que también compra excedentes de producción

Implementación:

```
1 tipo_tercero: ['Cliente', 'Proveedor', 'Ambos']
2
```

Ventajas:

- Reduce redundancia de datos
- Facilita mantenimiento
- Evita inconsistencias
- Permite relaciones comerciales bidireccionales

Generalización 2: Artículo (Materia Prima + Producto Terminado + Servicio)

Tipo: Generalización Total con Extensibilidad

Análisis: Todos los items del catálogo comparten:

- Estructura base (código, nombre, unidad, precio)
- Relaciones comunes (inventario, movimientos)
- Procesos de valoración

Implementación:

```
1 tipo_articulo: ['materia_prima', 'producto_terminado',
'servicio']
2
```

Particularidades por tipo:

- **Materia prima:** Se consume en producción (BOM), no se vende directamente
- **Producto terminado:** Se vende a clientes, es resultado de producción
- **Servicio:** Representa conceptos para movimientos de efectivo (cantidad=1, se valora por monto)

Generalización 3: Localidad (Fábrica + Bodega + Oficina)

Tipo: Generalización Parcial

Análisis: Todas las ubicaciones físicas comparten atributos base pero tienen capacidades diferentes:

- **Fábrica:** Puede producir (Orden_produccion)
- **Bodega:** Solo almacena (Inventario)
- **Oficina:** Personal administrativo (Empleado)

Implementación:

```
1 tipo_localidad: ['fabrica', 'bodega', 'oficina']
2
```

2.2 Relaciones Identificadas

Relaciones 1:N (Uno a Muchos)

Entidad Padre	Entidad Hija	Cardinalidad	Descripción
País	Empresa	1:N	Un país alberga múltiples empresas (filiales)
Empresa	Localidad	1:N	Una empresa tiene múltiples ubicaciones físicas
Empresa	Empleado	1:N	Una empresa tiene múltiples empleados
Terceros	Pedido	1:N	Un cliente realiza múltiples pedidos
Pedido	Detalle_pedido	1:N	Un pedido contiene múltiples líneas de productos
Pedido	Aprobacion	1:N	Un pedido puede tener múltiples aprobaciones/rechazos
Pedido	Despacho	1:N	Un pedido puede requerir múltiples despachos
Pedido	Orden_produccion	1:N	Un pedido genera órdenes de producción
Artículo	Inventario	1:N	Un artículo existe en múltiples localidades
Artículo	Movimiento	1:N	Un artículo tiene múltiples movimientos
Localidad	Inventario	1:N	Una localidad almacena múltiples artículos
Localidad	Ruta (origen)	1:N	De una localidad salen múltiples rutas
Localidad	Ruta (destino)	1:N	A una localidad llegan múltiples rutas
Empleado	Aprobacion	1:N	Un empleado realiza múltiples aprobaciones
Empleado	Planilla	1:N	Un empleado tiene múltiples registros de planilla
Tipo	Movimiento	1:N	Un tipo clasifica múltiples movimientos

Relaciones Especiales

1. Relación Recursiva en Terceros

```
1 Terceros.codigo_cliente_aprobador → Terceros.codigo_tercero
2 Cardinalidad: 1:N opcional
3
```

Propósito: Permite jerarquías de aprobación donde un cliente corporativo puede aprobar a sus subsidiarias.

Ejemplo:

```
1 Cliente Corporativo (código: CLI-001)
2 |─ Subsidiaria A (CLI-002, aprobador: CLI-001)
3 |─ Subsidiaria B (CLI-003, aprobador: CLI-001)
4 |─ Subsidiaria C (CLI-004, aprobador: CLI-001)
5
```

2. Relación N:M entre Pedido y Despacho (resuelta con Detalle_pedido)

```
1 Pedido (1) ↔ (N) Detalle_pedido (N) ↔ (1) Despacho
2
```

Propósito: Un despacho puede contener líneas de varios pedidos, y un pedido puede requerir varios despachos parciales.

Ejemplo:

```
1 Pedido #100: 1000 camisas
2 |─ Despacho #1: 400 camisas (40% del pedido)
3 |─ Despacho #2: 350 camisas (35% del pedido)
4 |─ Despacho #3: 250 camisas (25% del pedido)
5
```

3. Doble FK en Movimiento para Transferencias

```
1 Movimiento:
2   - codigo_localidad_origen (FK → Localidad)
3   - codigo_localidad_destino (FK → Localidad)
4
```

Propósito: Permite rastrear movimientos de inventario entre ubicaciones.

Tipos de movimiento:

- **Entrada:** origen=NULL, destino=Localidad (compra a proveedor)
- **Salida:** origen=Localidad, destino=NULL (venta a cliente)
- **Transferencia:** origen=Localidad A, destino=Localidad B
- **Efectivo:** origen y destino representan cuentas/cajas

3. RAZONAMIENTO DEL DISEÑO

3.1 Jerarquía Organizacional (País → Empresa → Localidad)

Decisión: Establecer estructura de 3 niveles

Razonamiento:

1. **Nivel País:** Necesario para consolidación financiera a nivel internacional. Cada país tiene su moneda, regulaciones tributarias y sistema contable propio.
2. **Nivel Empresa:** Una misma organización puede tener múltiples entidades legales en un país (por razones fiscales, operativas o estratégicas).
3. **Nivel Localidad:** Granularidad operativa. Control de inventarios, activos y personal por ubicación física.

Beneficios:

- Consolidación flexible (por localidad → empresa → país)
- Cumplimiento regulatorio por jurisdicción
- Optimización de operaciones logísticas
- Reporting multinivel para diferentes stakeholders

Ejemplo práctico:

1	País: Guatemala (GT)
2	├ Empresa: Confecciones GT S.A. (RUC: 12345678-9)
3	└ Localidad: Fábrica Zona 12
4	└ Localidad: Bodega Puerto Quetzal
5	└ Localidad: Oficina Administrativa
6	└ Empresa: Textiles GT S.A. (RUC: 98765432-1)
7	└ Localidad: Planta Textil Mixco
8	

3.2 Tabla Terceros: Generalización de Clientes y Proveedores

Decisión: Unificar en una sola entidad con discriminador

Razonamiento:

1. Se estima que el 80% de los atributos son iguales en ambas entidades.

2. **Realidad del negocio:** Un partner puede ser ambos roles simultáneamente
3. **Mantenimiento:** Una sola fuente de verdad para datos de contacto
4. **Escalabilidad:** Facilita agregar nuevos tipos (ej: "Socio estratégico")

Implementación clave:

```
1 tipo_tercero IN ('Cliente', 'Proveedor', 'Ambos')
2 cuenta_bancaria -- Para pagos y cobros
3 codigo_cliente_aprobador -- Jerarquías comerciales
4
```

3.3 Movimiento: Tabla Universal de Transacciones

Decisión: Consolidar movimientos de inventario y efectivo en una tabla

Razonamiento:

A favor de consolidación:

1. Ambos comparten estructura (origen, destino, fecha, tipo, cantidad/monto)
2. Simplifica auditoría: una sola consulta para flujo de materiales y dinero
3. Integridad referencial: artículos tipo "servicio" permiten registrar efectivo
4. Facilita reportes integrados (ej: costo de materia prima + costo de transporte)

Contra separación:

- Duplicaría lógica de auditoría
- Complicaría trazabilidad de operaciones complejas (ej: venta = salida inventario + entrada efectivo)

Implementación:

```
1 -- Movimiento de inventario
2 INSERT INTO Movimiento VALUES (
3     tipo_movimiento: 'ENTRADA_COMPRA',
4     codigo_articulo: 'MAT-TELA-001',
5     cantidad: 500,
6     monto: 3000
7 );
8
9 -- Movimiento de efectivo (usando artículo servicio)
10 INSERT INTO Movimiento VALUES (
11     tipo_movimiento: 'PAGO_PLANILLA',
12     codigo_articulo: 'SERV-PLANILLA',
13     cantidad: 1,
14     monto: 15000.00 -- El valor real
```

```
15 );  
16
```

3.4 Detalle_pedido como Puente

Decisión: Normalizar a 3FN separando cabecera y líneas de pedido

Razonamiento:

1. **Eliminación de redundancia:** Datos de cliente, fechas no se repiten por cada artículo
2. **Flexibilidad:** Un pedido puede tener N productos sin modificar estructura
3. **Tracking granular:** Cada línea puede tener estado independiente (cantidad_despachada)
4. **Relación con Despacho:** Permite despachos parciales por línea

Ejemplo:

```
1 Pedido #PED-2025-001  
2 | Cliente: Textiles Corp  
3 | Fecha: 2025-09-29  
4 | Estado: Aprobado  
5 | Detalle_pedido:  
6 |   | Línea 1: Camisa Blanca, solicitado: 1000, despachado:  
7 |   | 600  
8 |   | Línea 2: Camisa Azul, solicitado: 500, despachado: 500  
9 |   | Línea 3: Pantalón, solicitado: 300, despachado: 0
```

3.5 Orden_produccion: Planificación vs Ejecución

Decisión: Separar la intención (orden) de la ejecución (movimientos)

Razonamiento:

1. **Planificación:** La orden define QUÉ, CUÁNTO, DÓNDE y CUÁNDO producir
2. **Ejecución:** Los movimientos de inventario registran el consumo real de MP y generación de PT
3. **Medición:** Comparar estimado vs real para KPIs (eficiencia, tiempos)
4. **Trazabilidad:** ¿Qué pedido originó esta producción?

Flujo de trabajo:

```
1 1. Cliente hace Pedido de 1000 camisas  
2 2. Sistema verifica Inventario: solo hay 200 camisas  
3 3. Sistema crea Orden_produccion: producir 800 camisas
```

```
4 4. Orden_produccion consulta qué MP se necesita (tabla
imaginaria BOM)
5 5. Sistema crea Movimientos:
6   - Salida de MP (tela, botones, hilo)
7   - Entrada de PT (800 camisas)
8 6. Sistema actualiza Inventario
9 7. Sistema crea Despacho para entregar al cliente
10
```

Campos críticos:

- fecha_estimada vs fecha_real: Medir eficiencia
- estado: Flujo de trabajo (planificada → en proceso → completa)
- codigo_pedido: Trazabilidad hacia atrás

3.6 Ruta: Optimización de Logística

Decisión: Tabla maestra de rutas precalculadas

Razonamiento:

1. **Performance:** No calcular tiempos en tiempo real
2. **Realismo:** Los tiempos de transporte no son simétricos ni proporcionales a distancia
3. **Costos:** Agregar el costo de transporte al cálculo del precio.
4. **Optimización:** Algoritmo puede elegir ruta óptima (más rápida vs más barata)

Ejemplo:

```
1 -- Ruta directa
2 Ruta(origen: Fábrica GT, destino: Cliente USA, tiempo: 72h,
costo: $500, tipo: Aéreo)
3
4 -- Ruta indirecta (más barata pero lenta)
5 Ruta(origen: Fábrica GT, destino: Bodega MX, tiempo: 24h,
costo: $100, tipo: Terrestre)
6 Ruta(origen: Bodega MX, destino: Cliente USA, tiempo: 48h,
costo: $150, tipo: Terrestre)
7 Total: 72h, $250 -- Mismo tiempo, mitad de costo
8
```

Cálculo automático en Despacho:

```
1 UPDATE Despacho
2 SET fecha_estimada = fecha_envio + (
3     SELECT tiempo_horas FROM Ruta
4     WHERE origen = Despacho.localidad_origen
```



```
5      AND destino = Despacho.localidad_destino
6  );
7
```

3.7 Tipo: Tabla Catálogo Universal

Decisión: Usar una sola tabla para múltiples clasificaciones

Razonamiento:

1. **Extensibilidad:** Agregar nuevos tipos sin ALTER TABLE
2. **Reducción de tablas:** Evitar proliferación de tablas pequeñas
3. **Consistencia:** Mismo patrón para todos los catálogos
4. **Flexibilidad:** Permite descripciones multinivel

Alternativa rechazada: Una tabla por cada catálogo

```
1 Tipo_movimiento (X)
2 Tipo_empleado (X)
3 Tipo_localidad (X)
4 Tipo_tercero (X)
5 ...
6
```

Implementación:

```
1 -- Tipos de movimiento
2 INSERT INTO Tipo VALUES ('MOV-001', 'Entrada por compra',
3 'Compra MP', 'tipo_movimiento');
4 INSERT INTO Tipo VALUES ('MOV-002', 'Salida por venta', 'Venta
5 PT', 'tipo_movimiento');
6 INSERT INTO Tipo VALUES ('MOV-003', 'Transferencia', 'Entre
7 localidades', 'tipo_movimiento');
8
9 -- Tipos de empleado
10 INSERT INTO Tipo VALUES ('EMP-001', 'Operario', 'Producción',
11 'tipo_empleado');
12 INSERT INTO Tipo VALUES ('EMP-002', 'Supervisor', 'Jefatura',
13 'tipo_empleado');
14
15 -- El campo "campo" discrimina el contexto
16
```

Ventaja adicional: Facilita internacionalización (descripcion1: español, descripcion2: inglés)

3.8 Auditoría y Trazabilidad

Decisión: Diseño orientado a auditoría desde el inicio

Razonamiento:

- Empresa quiere entrar a bolsa de valores → Requiere controles internos robustos
- Operaciones multinacionales → Necesita consolidación transparente
- Alto volumen de transacciones → Debe poder rastrear cada operación

Mecanismos implementados:

1. Registro de transacciones:

```
1 Movimiento: TODA operación queda registrada con fecha, tipo,
  origen, destino
2
```

2. Aprobaciones explícitas:

```
1 Aprobacion: Quién, cuándo, por qué se aprobó/rechazó cada
  pedido
2
```

3. Fechas estimadas vs reales:

```
1 Despacho: fecha_envio, fecha_estimada, fecha_real
2 Orden_produccion: fecha_inicio, fecha_estimada, fecha_real
3 -- Permite medir performance y detectar desviaciones
4
```

4. Estados para Flujo de trabajo:

```
1 Pedido.estado: ['pendiente', 'aprobado', 'en_produccion',
  'despachado', 'completado', 'cancelado']
2 -- Reconstruye historia del pedido
3
```

5. Relaciones de trazabilidad:

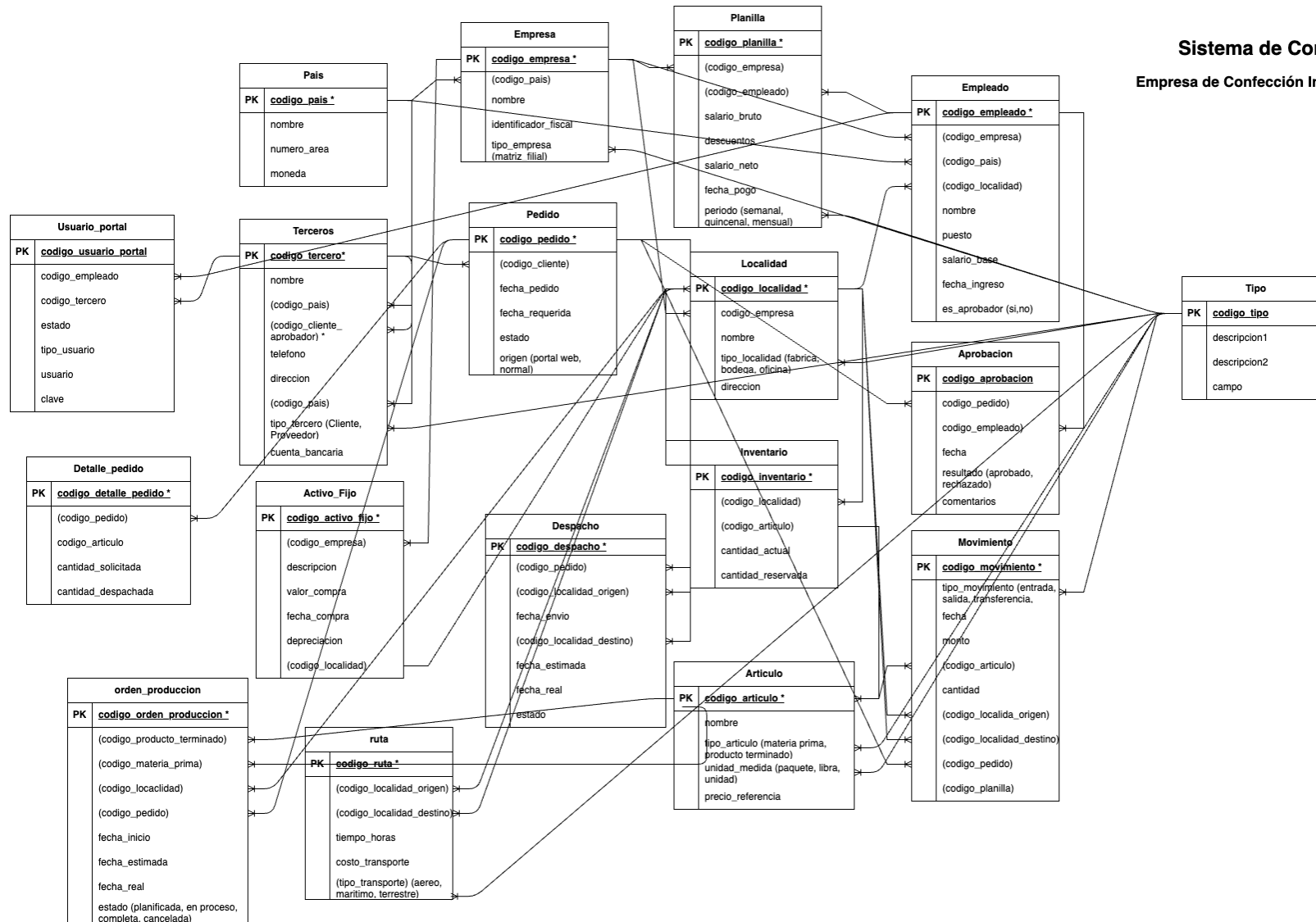
```
1 Movimiento.codigo_pedido: ¿Qué pedido originó este movimiento?
2 Orden_produccion.codigo_pedido: ¿Por qué estamos produciendo
  esto?
3 Despacho.codigo_pedido: ¿A quién va este envío?
4
```

Beneficio: Auditor puede responder preguntas como:

- "¿Dónde está el inventario de tela comprada el 15 de marzo?"
- "¿Quién aprobó el pedido PED-2025-001?"
- "¿Cuánto efectivo se movió de Guatemala a México en septiembre?"

4. ELABORACIÓN DEL DIAGRAMA E/R

Sistema de Control Empresa de Confección Internacional



5. SCRIPTS DE CREACIÓN DE TODOS LOS OBJETOS DE LA BASE DE DATOS DBMS DE ORACLE

5.1 Creación De Tablas

-- 1. PAIS

```
CREATE TABLE pais (  
  codigo_pais NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  nombre      VARCHAR2(100) NOT NULL,  
  moneda      VARCHAR2(20)  
);
```

-- 2. TIPO (Catálogo universal)

```
CREATE TABLE tipo (  
  codigo_tipo  NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  descripcion1 VARCHAR2(100) NOT NULL, -- Nombre del tipo  
  descripcion2 VARCHAR2(100),         -- Campo opcional  
  campo        VARCHAR2(50) NOT NULL  -- Categoría del tipo  
);
```

-- 3. EMPRESA

```
CREATE TABLE empresa (  
  codigo_empresa NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_pais    NUMBER NOT NULL REFERENCES pais(codigo_pais),  
  nombre         VARCHAR2(150) NOT NULL,  
  identificador_fiscal VARCHAR2(50),  
  tipo_empresa   NUMBER NOT NULL REFERENCES tipo(codigo_tipo)  
);
```

-- 4. LOCALIDAD

```
CREATE TABLE localidad (  
  codigo_localidad NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_empresa   NUMBER NOT NULL REFERENCES empresa(codigo_empresa),  
  nombre           VARCHAR2(150),  
  tipo_localidad   NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  direccion        VARCHAR2(200)  
);
```

-- 5. TERCEROS (Clientes / Proveedores)

```
CREATE TABLE terceros (  
  codigo_tercero NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_pais  NUMBER NOT NULL REFERENCES pais(codigo_pais),  
  nombre      VARCHAR2(150),  
  tipo_tercero NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  telefono    VARCHAR2(30),  
  direccion   VARCHAR2(200),  
  cuenta_bancaria VARCHAR2(50)  
);
```

-- 6. EMPLEADO

```
CREATE TABLE empleado (  
  codigo_empleado NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_empresa  NUMBER NOT NULL REFERENCES empresa(codigo_empresa),  
  codigo_localidad NUMBER REFERENCES localidad(codigo_localidad),  
  nombre          VARCHAR2(100),  
  puesto          VARCHAR2(50),  
  salario_base   NUMBER(12,2),  
  fecha_ingreso  DATE,  
  es_aprobador   CHAR(1) DEFAULT 'N'  
);
```

-- 7. USUARIO_PORTAL

```
CREATE TABLE usuario_portal (  
  codigo_usuario_portal NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY  
  KEY,  
  codigo_empleado  NUMBER REFERENCES empleado(codigo_empleado),  
  codigo_tercero   NUMBER REFERENCES terceros(codigo_tercero),  
  usuario          VARCHAR2(50) UNIQUE NOT NULL,  
  clave            VARCHAR2(100) NOT NULL,  
  tipo_usuario     NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  estado           NUMBER NOT NULL REFERENCES tipo(codigo_tipo)  
);
```

-- 8. ARTICULO

```
CREATE TABLE articulo (  
  codigo_articulo NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  nombre          VARCHAR2(100) NOT NULL,  
  tipo_articulo   NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  unidad_medida   NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  precio_referencia NUMBER(12,2)  
);
```

-- 9. INVENTARIO

```
CREATE TABLE inventario (  
  codigo_inventario NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_localidad  NUMBER NOT NULL REFERENCES localidad(codigo_localidad),  
  codigo_articulo   NUMBER NOT NULL REFERENCES articulo(codigo_articulo),  
  cantidad_actual   NUMBER DEFAULT 0,  
  cantidad_reservada NUMBER DEFAULT 0  
);
```

-- 10. PEDIDO

```
CREATE TABLE pedido (  
  codigo_pedido NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_cliente NUMBER NOT NULL REFERENCES terceros(codigo_tercero),  
  fecha_pedido  DATE DEFAULT SYSDATE,  
  fecha_requerida DATE,  
  estado        VARCHAR2(20),  
  origen        VARCHAR2(20)  
);
```

-- 11. DETALLE_PEDIDO

```
CREATE TABLE detalle_pedido (  
  codigo_detalle_pedido NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY  
KEY,  
  codigo_pedido  NUMBER NOT NULL REFERENCES pedido(codigo_pedido),  
  codigo_articulo NUMBER NOT NULL REFERENCES articulo(codigo_articulo),  
  cantidad_solicitada NUMBER NOT NULL,  
  cantidad_despachada  NUMBER DEFAULT 0  
);
```

-- 12. APROBACION

```
CREATE TABLE aprobacion (  
  codigo_aprobacion NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_pedido     NUMBER NOT NULL REFERENCES pedido(codigo_pedido),  
  codigo_empleado   NUMBER NOT NULL REFERENCES empleado(codigo_empleado),  
  fecha             DATE DEFAULT SYSDATE,  
  resultado         NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  comentarios       VARCHAR2(200)  
);
```

-- 13. DESPACHO

```
CREATE TABLE despacho (  
  codigo_despacho NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_pedido  NUMBER NOT NULL REFERENCES pedido(codigo_pedido),  
  codigo_localidad_origen NUMBER REFERENCES localidad(codigo_localidad),  
  codigo_localidad_destino NUMBER REFERENCES localidad(codigo_localidad),  
  fecha_envio    DATE,  
  fecha_estimada DATE,  
  fecha_real     DATE,  
  estado_despacho NUMBER NOT NULL REFERENCES tipo(codigo_tipo)  
);
```

-- 14. MOVIMIENTO

```
CREATE TABLE movimiento (  
  codigo_movimiento NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY  
  KEY,  
  tipo_movimiento  NUMBER NOT NULL REFERENCES tipo(codigo_tipo),  
  codigo_articulo  NUMBER REFERENCES articulo(codigo_articulo),  
  codigo_localidad_origen NUMBER REFERENCES localidad(codigo_localidad),  
  codigo_localidad_destino NUMBER REFERENCES localidad(codigo_localidad),  
  codigo_pedido    NUMBER REFERENCES pedido(codigo_pedido),  
  fecha            DATE DEFAULT SYSDATE,  
  monto            NUMBER(12,2),  
  cantidad         NUMBER  
);
```

-- 15. PLANILLA

```
CREATE TABLE planilla (  
  codigo_planilla NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_empleado NUMBER NOT NULL REFERENCES empleado(codigo_empleado),  
  salario_bruto   NUMBER(12,2),  
  descuentos      NUMBER(12,2),  
  salario_netos   NUMBER(12,2),  
  fecha_pago      DATE  
);
```

-- 16. ACTIVO_FIJO

```
CREATE TABLE activo_fijo (  
  codigo_activo_fijo NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_empresa    NUMBER REFERENCES empresa(codigo_empresa),  
  codigo_localidad  NUMBER REFERENCES localidad(codigo_localidad),  
  descripcion       VARCHAR2(200),  
  valor_compra      NUMBER(12,2),  
  fecha_compra      DATE,  
  depreciacion      NUMBER(12,2)  
);
```

-- 17. RUTA

```
CREATE TABLE ruta (  
  codigo_ruta NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  codigo_localidad_origen NUMBER REFERENCES localidad(codigo_localidad),  
  codigo_localidad_destino NUMBER REFERENCES localidad(codigo_localidad),  
  tiempo_horas  NUMBER,  
  costo_transporte NUMBER(12,2),  
  tipo_transporte NUMBER NOT NULL REFERENCES tipo(codigo_tipo)  
);
```


5.2 Vistas

-- 1) Inventario valorizado

```
CREATE OR REPLACE VIEW v_inventario_valorizado AS
SELECT
    p.nombre AS pais,
    e.nombre AS empresa,
    l.nombre AS localidad,
    a.nombre AS articulo,
    i.cantidad_actual,
    a.precio_referencia,
    (i.cantidad_actual * a.precio_referencia) AS valor_total
FROM inventario i
JOIN articulo a ON i.codigo_articulo = a.codigo_articulo
JOIN localidad l ON i.codigo_localidad = l.codigo_localidad
JOIN empresa e ON l.codigo_empresa = e.codigo_empresa
JOIN pais p ON e.codigo_pais = p.codigo_pais;
```

-- 2) Flujo de efectivo

```
CREATE OR REPLACE VIEW v_flujo_efectivo AS
SELECT
    p.nombre AS pais,
    e.nombre AS empresa,
    m.fecha,
    t.descripcion1 AS tipo_movimiento,
    m.monto
FROM movimiento m
JOIN tipo t ON m.tipo_movimiento = t.codigo_tipo
JOIN localidad l ON NVL(m.codigo_localidad_origen, m.codigo_localidad_destino) =
l.codigo_localidad
JOIN empresa e ON l.codigo_empresa = e.codigo_empresa
JOIN pais p ON e.codigo_pais = p.codigo_pais;
```

-- 3) Pedidos pendientes de aprobación

```
CREATE OR REPLACE VIEW v_pedidos_pendientes AS
SELECT
    ped.codigo_pedido,
    ter.nombre AS cliente,
    ped.fecha_pedido,
    ped.fecha_requerida,
    ped.estado
FROM pedido ped
JOIN terceros ter ON ped.codigo_cliente = ter.codigo_tercero
WHERE ped.estado = 'PENDIENTE';
```

5.3 Insercion De Datos

-- 1. PAISES

```
INSERT INTO pais (nombre, moneda) VALUES ('Guatemala', 'GTQ');
INSERT INTO pais (nombre, moneda) VALUES ('México', 'MXN');
INSERT INTO pais (nombre, moneda) VALUES ('USA', 'USD');
```

-- 2. TIPOS (catálogo universal)

-- Tipos de empresa

```
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_EMPRESA','Matriz');
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_EMPRESA','Filial');
```

-- Tipos de localidad

```
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_LOCALIDAD','Fábrica');
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_LOCALIDAD','Bodega');
```

-- Tipos de transporte

```
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_TRANSPORTE','Terrestre');
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_TRANSPORTE','Marítimo');
```

-- Estados de despacho

```
INSERT INTO tipo (campo, descripcion1) VALUES ('ESTADO_DESPACHO','Pendiente');
INSERT INTO tipo (campo, descripcion1) VALUES ('ESTADO_DESPACHO','En tránsito');
INSERT INTO tipo (campo, descripcion1) VALUES ('ESTADO_DESPACHO','Entregado');
```

-- Resultado de aprobación

```
INSERT INTO tipo (campo, descripcion1) VALUES
('RESULTADO_APROBACION','Aprobado');
INSERT INTO tipo (campo, descripcion1) VALUES
('RESULTADO_APROBACION','Rechazado');
```

-- Tipos de artículo

```
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_ARTICULO','Materia Prima');
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_ARTICULO','Producto
Terminado');
```

-- Unidades de medida

```
INSERT INTO tipo (campo, descripcion1) VALUES ('UNIDAD_MEDIDA','Metro');
INSERT INTO tipo (campo, descripcion1) VALUES ('UNIDAD_MEDIDA','Kilogramo');
```

-- Estados de usuario

```
INSERT INTO tipo (campo, descripcion1) VALUES ('ESTADO_USUARIO','Activo');
INSERT INTO tipo (campo, descripcion1) VALUES ('ESTADO_USUARIO','Inactivo');
```

-- Tipos de usuario

```
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_USUARIO','Empleado');
INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_USUARIO','Cliente');
```

-- Tipos de movimiento

INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_MOVIMIENTO','Compra MP');

INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_MOVIMIENTO','Venta Producto');

INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_MOVIMIENTO','Pago Nómina');

INSERT INTO tipo (campo, descripcion1) VALUES ('TIPO_MOVIMIENTO','Efectivo Ingreso');

-- 3. EMPRESAS

INSERT INTO empresa (codigo_pais, nombre, identificador_fiscal, tipo_empresa) VALUES (1, 'Confecciones Global S.A.', 'CGT-12345', 1);

INSERT INTO empresa (codigo_pais, nombre, identificador_fiscal, tipo_empresa) VALUES (2, 'Confecciones México', 'CMX-98765', 2);

-- 4. LOCALIDADES

INSERT INTO localidad (codigo_empresa, nombre, tipo_localidad, direccion) VALUES (1, 'Fábrica Central', 3, 'Zona 4, Ciudad de Guatemala');

INSERT INTO localidad (codigo_empresa, nombre, tipo_localidad, direccion) VALUES (1, 'Bodega Central', 4, 'Zona 12, Ciudad de Guatemala');

INSERT INTO localidad (codigo_empresa, nombre, tipo_localidad, direccion) VALUES (2, 'Fábrica CDMX', 3, 'Parque Industrial, CDMX');

-- 5. TERCEROS

INSERT INTO terceros (codigo_pais, nombre, tipo_tercero, telefono, direccion, cuenta_bancaria) VALUES (1, 'Textiles Maya', 1, '502-5555-1111', 'Mixco, Guatemala', 'GT12345');

INSERT INTO terceros (codigo_pais, nombre, tipo_tercero, telefono, direccion, cuenta_bancaria) VALUES (1, 'Tiendas Estilo', 1, '502-5555-2222', 'Zona 1, Guatemala', 'GT67890');

INSERT INTO terceros (codigo_pais, nombre, tipo_tercero, telefono, direccion, cuenta_bancaria) VALUES (2, 'Algodones Premium', 1, '55-5555-3333', 'CDMX, México', 'MX54321');

-- 6. EMPLEADOS

```
INSERT INTO empleado (codigo_empresa, codigo_localidad, nombre, puesto, salario_base,
fecha_ingreso, es_aprobador)
VALUES (1, 1, 'Carlos Pérez', 'Gerente Producción', 8000, DATE '2022-01-10', 'Y');
```

```
INSERT INTO empleado (codigo_empresa, codigo_localidad, nombre, puesto, salario_base,
fecha_ingreso)
VALUES (1, 2, 'Ana López', 'Operaria', 3500, DATE '2023-06-15');
```

-- 7. USUARIOS DEL PORTAL

```
INSERT INTO usuario_portal (codigo_empleado, usuario, clave, tipo_usuario, estado)
VALUES (1, 'cperez', '1234', 23, 21); -- tipo_usuario=Empleado, estado=Activo
```

```
INSERT INTO usuario_portal (codigo_tercero, usuario, clave, tipo_usuario, estado)
VALUES (2, 'tienda_estilo', 'abcd', 24, 21); -- cliente activo
```

-- 8. ARTÍCULOS

```
INSERT INTO articulo (nombre, tipo_articulo, unidad_medida, precio_referencia)
VALUES ('Tela algodón', 13, 15, 50.00);
```

```
INSERT INTO articulo (nombre, tipo_articulo, unidad_medida, precio_referencia)
VALUES ('Camisa Hombre', 14, 15, 120.00);
```

-- 9. INVENTARIO

```
INSERT INTO inventario (codigo_localidad, codigo_articulo, cantidad_actual)
VALUES (1, 1, 500); -- Fábrica Central tiene 500 metros de Tela
```

```
INSERT INTO inventario (codigo_localidad, codigo_articulo, cantidad_actual)
VALUES (2, 2, 300); -- Bodega Central tiene 300 camisas
```

-- 10. PEDIDO Y DETALLE

```
INSERT INTO pedido (codigo_cliente, fecha_pedido, fecha_requerida, estado, origen)
VALUES (2, SYSDATE, SYSDATE + 10, 'PENDIENTE', 'WEB');
```

```
INSERT INTO detalle_pedido (codigo_pedido, codigo_articulo, cantidad_solicitada)
VALUES (1, 2, 50); -- Cliente pide 50 camisas
```

-- 11. APROBACIÓN

```
INSERT INTO aprobacion (codigo_pedido, codigo_empleado, resultado, comentarios)
VALUES (1, 1, 19, 'Aprobado por disponibilidad');
```

-- 12. DESPACHO

```
INSERT INTO despacho (codigo_pedido, codigo_localidad_origen,  
codigo_localidad_destino, fecha_envio, estado_despacho)  
VALUES (1, 2, 2, SYSDATE, 17); -- pendiente de envío
```

-- 13. MOVIMIENTOS

```
INSERT INTO movimiento (tipo_movimiento, codigo_articulo, codigo_localidad_origen,  
codigo_localidad_destino, codigo_pedido, monto, cantidad)  
VALUES (25, 2, 2, NULL, 1, 6000.00, 50); -- venta producto
```

-- 14. PLANILLA

```
INSERT INTO planilla (codigo_empleado, salario_bruto, descuentos, salario_netto,  
fecha_pago)  
VALUES (1, 8000, 800, 7200, SYSDATE);
```

-- 15. ACTIVOS FIJOS

```
INSERT INTO activo_fijo (codigo_empresa, codigo_localidad, descripcion, valor_compra,  
fecha_compra, depreciacion)  
VALUES (1, 1, 'Máquina de coser industrial', 25000, DATE '2022-05-01', 5000);
```

-- 16. RUTAS

```
INSERT INTO ruta (codigo_localidad_origen, codigo_localidad_destino, tiempo_horas,  
costo_transporte, tipo_transporte)  
VALUES (1, 2, 5, 1500, 5); -- terrestre entre fábrica y bodega
```

Sistema de Control - Empresa de Confección Internacional

Implementación de Base de Datos para control integral de operaciones
multinacionales

Estudiante: Cruz Francisco Estrada Gregorio

Carné: 7960-23-5339



Estructura Organizacional



País

Nivel superior para consolidación internacional. Manejo de múltiples monedas y regulaciones locales.



Empresa

Entidades legales por país con identificadores fiscales únicos para cumplimiento tributario.



Localidad

Ubicaciones físicas (fábricas, bodegas, oficinas) para control granular de operaciones.



Entidades Principales Identificadas

18

Entidades Totales

Objetos de negocio con ciclo de vida propio

3

Generalizaciones

Terceros, Artículos y Localidades
unificadas

25

Relaciones

Conexiones entre entidades para integridad

Entidades Core

- País, Empresa, Localidad
- Terceros (Clientes/Proveedores)
- Empleado, Usuario Portal
- Artículo, Inventario

Procesos Operativos

- Pedido, Detalle Pedido
- Aprobación, Orden Producción
- Despacho, Ruta
- Movimiento, Planilla

Generalizaciones Implementadas

Terceros

Unifica **Clientes y Proveedores** en una sola entidad. Un tercero puede actuar en ambos roles simultáneamente.

- Elimina redundancia de datos
- Facilita relaciones bidireccionales
- Incluye jerarquías de aprobación

Artículos

Consolida **Materia Prima, Productos Terminados y Servicios** con discriminador por tipo.

- Estructura base común
- Servicios para movimientos de efectivo
- Procesos de valoración unificados

Localidades

Agrupar **Fábricas, Bodegas y Oficinas** con capacidades diferenciadas por tipo.

- Atributos base compartidos
- Funcionalidades específicas por tipo
- Control granular de operaciones



Tabla Universal de Movimientos

Decisión Clave: Consolidar inventario y efectivo

Ventajas del Diseño

- Auditoría integral en una consulta
- Trazabilidad completa de operaciones
- Artículos tipo "servicio" para efectivo
- Simplifica reportes integrados

Tipos de Movimiento

- **Entrada:** Compras a proveedores
- **Salida:** Ventas a clientes
- **Transferencia:** Entre localidades
- **Efectivo:** Pagos y cobros

Flujo de Procesos Operativos



Pedido del Cliente

Cliente realiza pedido vía portal web o medio tradicional. Se registra con fecha requerida y estado inicial.



Aprobación

Personal autorizado aprueba o rechaza pedidos. Se registra quién, cuándo y por qué se tomó la decisión.



Orden de Producción

Sistema verifica inventario y genera órdenes de producción para artículos faltantes con fechas estimadas.



Despacho

Coordinación de entrega entre localidades y cliente final con tracking de fechas estimadas vs reales.

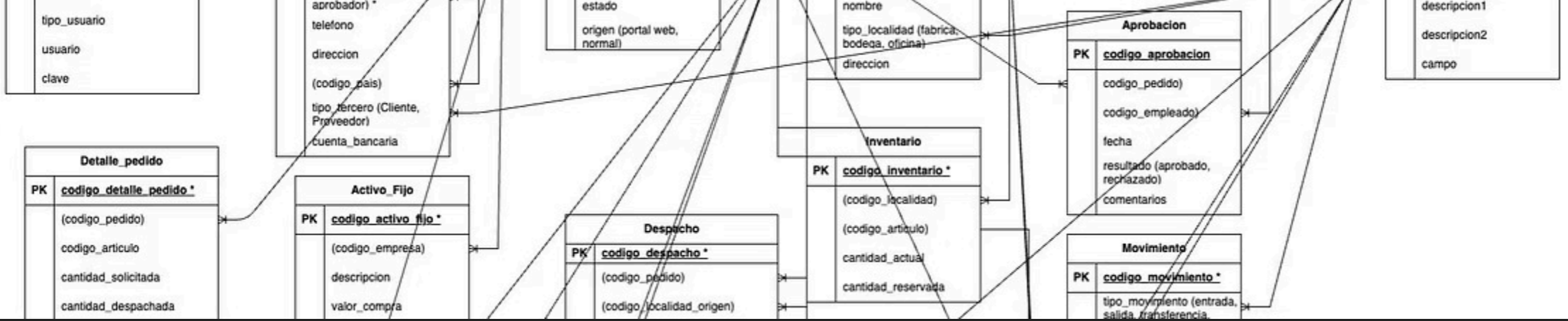


Diagrama Entidad-Relación

Diseño normalizado a 3FN con 18 entidades principales y relaciones que garantizan integridad referencial y trazabilidad completa.

Características de Auditoría y Trazabilidad



Registro Transaccional

Toda operación queda registrada en la tabla Movimiento con fecha, tipo, origen y destino para auditoría completa.



Aprobaciones Explícitas

Sistema registra quién, cuándo y por qué se aprobó cada pedido, proporcionando trail de decisiones.



Fechas Estimadas vs Reales

Comparación entre tiempos planificados y ejecutados para medir performance y detectar desviaciones.



Estados de Flujo

Control de estados permite reconstruir historia completa de pedidos y órdenes de producción.

Implementación en Oracle

01

Creación de Tablas

17 tablas principales con claves primarias auto-generadas y restricciones de integridad referencial.

02

Tabla Tipo Universal

Catálogo centralizado para clasificaciones reutilizables evitando proliferación de tablas pequeñas.

03

Vistas de Negocio

Vistas especializadas para inventario valorizado, flujo de efectivo y pedidos pendientes.

04

Datos de Prueba

Inserción de datos maestros y transaccionales para validar funcionalidad del sistema.



Beneficios del Diseño



Escalabilidad

Diseño preparado para crecimiento multinacional con estructura jerárquica flexible.



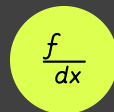
Auditoría

Trazabilidad completa de operaciones preparada para controles internos robustos.



Eficiencia

Consolidación de entidades reduce redundancia y facilita mantenimiento de datos.



Integración

Tabla universal de movimientos permite reportes integrados de inventario y efectivo.

Sistema listo para operación

Base de datos implementada con éxito, preparada para soportar operaciones de confección internacional con control integral y auditoría completa..



¡Muchas gracias!

¿Tienen alguna pregunta?

