

# “Implementación aplicación informática del caso SuperAndes”

Jaime Andrés Torres Bermejo, Juan Sebastián Alegría Zúñiga  
ISIS2304 – Iteración 3 – Proyecto de curso - Caso SuperAndes  
Universidad de los Andes, Bogotá, Colombia  
[j.torres16@uniandes.edu.co](mailto:j.torres16@uniandes.edu.co), [j.alegria@uniandes.edu.co](mailto:j.alegria@uniandes.edu.co)  
Fecha de presentación: noviembre 14 de 2022

## Tabla de contenido

1	Introducción .....	1
2	Resultados .....	1
2.1	Modelos .....	1
2.2	Resultados logrados.....	2
2.3	Estructura del proyecto.....	2
2.4	Resultados no logrados.....	3
2.5	Balance de pruebas .....	3
2.6	Conclusiones .....	3
3	Consultas SQL.....	3
3.1	RF15 .....	3
3.2	RF16 .....	4
3.3	RF17 .....	4
3.4	RF18 .....	4
3.5	RF19 .....	5
3.6	RF20 .....	5
3.7	RF21 .....	5
3.8	RF22 .....	5
3.9	RFC7 .....	5
3.10	RFC8 .....	6
3.11	RFC9 .....	6
4	Bibliografía.....	6

## 1 Introducción

Este documento presenta los resultados obtenidos para la tercera iteración del proyecto SuperAndes en el curso Sistemas Transaccionales de la Universidad de Los Andes.

## 2 Resultados

### 2.1 Modelos

A continuación, se presenta tanto el modelo conceptual como relacional, luego de haberlo normalizado y agregado atributos útiles para la implementación de los requerimientos del caso:

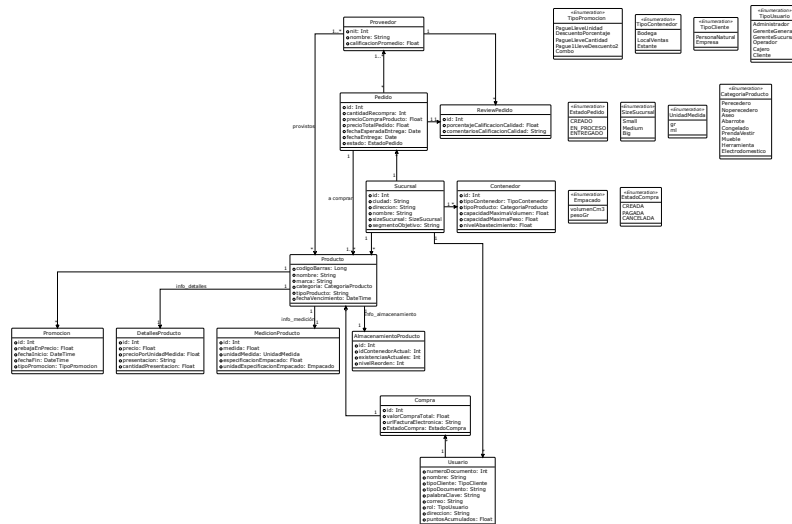


Figura 1. Modelo conceptual

Producto	codigoBarra	nombre	marca	especificaciones	tipoProducto	fechaVencimiento	idPedido	idRecurso	idRecursoFisico	idRecursoLogico	idContenedor	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN, CK in ('Proveedor', 'NoProveedor', 'Asno', 'Alimento', 'Complemento', 'ProductoVet', 'Medicamento', 'Vaccinacion', 'Tratamiento')	NN	NN	NN, FK Pedido.id	NN, FK Recurso.id	NN, FK RecursoFisico.id	NN, FK RecursoLogico.id	NN, FK Contenedor.id	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Promocion	id	etiquetaPrecio	tipoPromocion	fechaInicio	fechaFin	idProducto	cantidadDisponible	cantidadReservada	idContenedorActual	idContenedorAnterior	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	DD	NN	NN, CK in F	NN	NN	FK, FK Producto.codigoBarra	NN	NN	NN, FK Contenedor.id	NN, FK Contenedor.id	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Detalle Producto	precio	precioPorCantidad	precioPromocion	cantidadReservada	cantidadPromocion
PK, SA	NN	NN	NN	NN	NN

Medicacion Producto	medicacion	cantidadMedicacion	cantidadReservada	cantidadPromocion	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN	NN, CK in ('gr', 'ml')	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id

Compra	id	valorCompraTotal	idDetalleProducto	cantidadCompra	cantidadReservada	fechaCompra	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	DD	NN	NN, CK in ('CREADA', 'PAGADA', 'CANCELADA')	NN, FK Usuario.numeroDocumento	NN	NN	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Usuario	numeroDocumento	nombre	tipoUsuario	idDetalleProducto	cantidadReservada	cantidadPromocion	idDetalleProducto	idMedicion	idAbonoProducto
PK, UA	NN	NN	NN, CK in ('Proveedor', 'NoProveedor', 'Asno', 'Alimento', 'Complemento', 'ProductoVet', 'Medicamento', 'Vaccinacion', 'Tratamiento')	NN	NN	NN	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Contenedor	id	tipoContenedor	idDetalleProducto	cantidadReservada	cantidadPromocion	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN, CK in ('Bodega', 'Laboratorio', 'Farmacia', 'Hospital', 'Clinica', 'Centro de Diagnóstico', 'Centro de Tratamiento', 'Centro de Vacunación', 'Centro de Diagnóstico y Tratamiento')	NN	NN	NN	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Recurso	id	nombre	descripcion	stock	stockReservado	stockPromocion	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN	NN, CK in ('Small', 'Medium', 'Big')	NN	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Pedido	id	cantidadReservada	precioCompraProducto	precioTotalPedido	fechaPedido	fechaEntrega	estado	idRecurso	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN	NN	NN	NN, CK in ('CREADO', 'EN PROCESO', 'ENTREGADO')	NN, FK Recurso.id	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Review Pedido	id	cantidadReservada	precioCompraProducto	precioTotalPedido	fechaPedido	fechaEntrega	estado	idRecurso	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN	NN	NN	NN, CK in ('CREADO', 'EN PROCESO', 'ENTREGADO')	NN, FK Recurso.id	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Producto Promocion	id	cantidadReservada	precioCompraProducto	precioTotalPedido	fechaPedido	fechaEntrega	estado	idRecurso	idDetalleProducto	idMedicion	idAbonoProducto
PK, SA	NN	NN	NN	NN	NN	NN	NN, CK in ('CREADO', 'EN PROCESO', 'ENTREGADO')	NN, FK Recurso.id	NN, FK DetalleProducto.id	NN, FK MedicionProducto.id	NN, FK AbonoProducto.id

Figura 2. Modelo relacional

## 2.2 Resultados logrados

- Se logró dar formato a tanto las consultas requeridas para los RF como las consultas requeridas para los RFC, en ambos casos partimos del modelo relacional diseñado en las dos iteraciones anteriores, y a partir de esto poder responder a las preguntas que se nos fueron formuladas. Sin embargo, es de entenderse que, aunque las tablas guardan toda la información, limitaciones como el manejo de operaciones GROUP BY dentro de la lógica interna de Oracle SQL o la organización y elegancia de las consultas, nos llevaron a desarrollar vistas que empaquetaran esta información dentro de tablas auxiliares. Esto nos permitió responder elegantemente a las preguntas formuladas a partir de la reorganización de la información en presentaciones más trabajables. Es posible que esto haga a la base de datos tomar más espacio en memoria, sin embargo, este es un trade-off al cuál estamos dispuesto con el fin de resolver elegantemente el trabajo.

## 2.3 Estructura del proyecto

- En `docs/javadocs/` podrá encontrar la documentación autogenerada del proyecto JDO.

- En `docs/modelos/` podrá encontrar el modelo relacional y conceptual utilizado para este proyecto.
- En `docs/sql/create-tables/` podrá encontrar todas las sentencias de creación de tablas.
- En `docs/sql/data` podrá encontrar sentencias de limpieza de base de datos y scripts de creación del esquema completo.
- En `docs/sql/data-population` podrá encontrar sentencias de inserción de registros de prueba para poblar la base de datos y soportar los requerimientos funcionales.
- En `docs/sql/rfc` podrá encontrar sentencias de consulta para soportar los requerimientos funcionales de consulta.
- En `src/` podrá encontrar el código fuente del proyecto.

## 2.4 Resultados no logrados

A pesar de que se realizaron todas las consultas SQL necesarias para ejecutar en su totalidad los requerimientos de esta iteración, no se logró crear todos los métodos necesarios para visualizar su funcionamiento en la interfaz gráfica. Se crearon muchos de los métodos requeridos para satisfacer los requerimientos en la interfaz, interfazSuperAndes, negocio/SuperAndes, la mayoría en PersistenciaSuperAndes, y únicamente quedó faltando los SQL<Entidad> que se encargaban de ejecutar las sentencias SQL ya creadas en docs.

## 2.5 Balance de pruebas

A pesar de que no se realizaron pruebas unitarias o manejo de QA como se habría esperado del enunciado, se pudo generar una base de datos la cual nos permitiera correr los query especificados con datos relativamente realistas, y a la hora de probar los datos generados por esta mock data y compararlos con lo que nos debería dar una consulta de este tipo, se estableció que las consultas SQL realizadas funcionaban de forma adecuada para solucionar los problemas establecidos. Y esta carga de datos nos implicó el manejo de tuplas con los datos y condiciones establecidas, a partir de los cuales se puede establecer que las pruebas de integridad, de ser construidas formalmente, funcionarían prácticamente a la perfección.

## 2.6 Conclusiones

Nuestro modelo inicial soportaba, en su mayoría, todas las reglas de negocio y supuestos adicionales de implementación. Sin embargo, a diferencia de la iteración 2, en este caso nos tocó implementar vistas y estructuras auxiliares las cuales nos permitieran filtrar y construir alrededor de las limitaciones de Oracle SQL y del proyecto 'Parranderos'. En general, se lograron construir satisfactoriamente los requerimientos funcionales de consulta sin necesidad de añadir tablas enteras adicionales.

# 3 Consultas SQL

## 3.1 RF15

```
INSERT INTO COMPRA (
    ID,
    VALOR_COMPRA_TOTAL,
    URL_FACTURA_ELECTRONICA,
    ESTADO_COMPRA,
    COMPRADOR,
    FECHA_COMPRA
) VALUES (
```

```

        (SELECT (MAX(ID) + 1) FROM COMPRA),
        0,
        'https://mashable.com/mauris.jsp',
        'CREADA',
        '3896442104',
        '28-Nov-2021'
    );

SELECT COMPRA.ID AS ID_COMPRA,
       ID_PRODUCTO,
       CANT_UNIDADES_COMPRADAS
FROM PRODUCTOS_COMPRA
     FULL OUTER JOIN COMPRA
       ON ID_COMPRA = COMPRA.ID
     FULL OUTER JOIN USUARIO
       ON COMPRADOR = USUARIO.NUMERO_DOCUMENTO
WHERE COMPRADOR = 3896442104 AND
       COMPRA.ID = 1;

```

### 3.2 RF16

```

INSERT INTO PRODUCTOS_COMPRA (
    ID_COMPRA,
    ID_PRODUCTO,
    CANT_UNIDADES_COMPRADAS
) VALUES (
    1,
    6371316388088419,
    1
);

```

### 3.3 RF17

```

DELETE FROM PRODUCTOS_COMPRA
WHERE
    ID_PRODUCTO = 6371316388088419 AND
    ID_COMPRA = 1;

```

### 3.4 RF18

```

UPDATE COMPRA
SET
    ESTADO_COMPRA = 'PAGADA'
WHERE
    ID = 1;

```

### 3.5 RF19

```
UPDATE COMPRA
SET
    ESTADO_COMPRA = 'CANCELADA'
WHERE
    ID = 1;
```

### 3.6 RF20

```
SELECT ID_PRODUCTO, CANT_UNIDADES_COMPRADAS
FROM COMPRA
    FULL OUTER JOIN PRODUCTOS_COMPRA
    ON COMPRA.ID = ID_COMPRA
WHERE ESTADO_COMPRA = 'CANCELADA';
```

### 3.7 RF21

```
SELECT ID_PRODUCTO, SUM(CANT_UNIDADES_COMPRADAS), ID_PROVEEDOR
FROM COMPRA
    FULL OUTER JOIN PRODUCTOS_COMPRA
    ON COMPRA.ID = ID_COMPRA
    FULL OUTER JOIN PRODUCTO
    ON PRODUCTO.CODIGO_BARRAS = ID_PRODUCTO
    INNER JOIN PEDIDOS_PROVEEDOR
    ON PRODUCTO.ID_PEDIDO = PEDIDOS_PROVEEDOR.ID_PEDIDO
WHERE ESTADO_COMPRA = 'PAGADA'
GROUP BY ID_PROVEEDOR, ID_PRODUCTO;
```

### 3.8 RF22

```
UPDATE PEDIDO
SET
    ESTADO = 'ENTREGADO'
WHERE
    ID = 1;
```

### 3.9 RFC7

```
SELECT * FROM
    (((SELECT CODIGO_BARRAS, NOMBRE, SUM(VALOR_COMPRA_TOTAL) AS
    DINERO_ADQUIRIDO, NULL AS CANTIDADES_ADQUIRIDAS, 'MAYORES INGRESOS'
    AS CATEGORIA,
    ID_SUCURSAL FROM DATOS_COMPRAS
    GROUP BY CODIGO_BARRAS, NOMBRE, ID_SUCURSAL)
    UNION ALL
    (SELECT CODIGO_BARRAS, NOMBRE, NULL AS DINERO_ADQUIRIDO,
    SUM(CANT_UNIDADES_COMPRADAS) AS CANTIDADES_ADQUIRIDAS, 'MENOR
    DEMANDA' AS CATEGORIA, ID_SUCURSAL
```

```

FROM DATOS_COMPRAS
GROUP BY CODIGO_BARRAS, NOMBRE, ID_SUCURSAL))
UNION
(SELECT * FROM MAYOR_DEMANDA)
)
WHERE ID_SUCURSAL=?
ORDER BY ID_SUCURSAL;

```

### 3.10 RFC8

```

SELECT * FROM (
    SELECT NUMERO_DOCUMENTO, NOMBRE, COUNT(NUMERO_DOCUMENTO) AS
    COMPRAS_REALIZADAS,
    ID_SUCURSAL, to_char(FECHA_COMPRA, 'YYYY-MM') FROM
    (
    SELECT * FROM
    (
    (SELECT COMPRADOR, FECHA_COMPRA FROM COMPRA)
    INNER JOIN
    (SELECT * FROM USUARIO)
    ON COMPRADOR = NUMERO_DOCUMENTO
    )
    )
    WHERE ID_SUCURSAL = ?
    GROUP BY NUMERO_DOCUMENTO, NOMBRE, ID_SUCURSAL, FECHA_COMPRA
    ) WHERE COMPRAS_REALIZADAS >= 2;

```

### 3.11 RFC9

```

SELECT CODIGO_BARRAS, NOMBRE, CATEGORIA, ID_PEDIDO, SUCURSAL,
fecha_entrega AS ULTIMA_ENTREGA
FROM ULTIMOS_PEDIDOS
INNER JOIN
(SELECT CODIGO_BARRAS AS CBARRAS, FECHA_ENTREGA AS
PENULTIMA_ENTREGA FROM PENULTIMOS_PEDIDOS)
ON CBARRAS = ULTIMOS_PEDIDOS.CODIGO_BARRAS
WHERE PENULTIMA_ENTREGA <= FECHA_ENTREGA - INTERVAL '60' DAY
AND SUCURSAL = ?

```

## 4 Bibliografía

1. *Universidad de los Andes*. [En línea] [Citado el: 1 de Septiembre de 2022.] [https://bloqueneon.uniandes.edu.co//content/enforced/140479-UN\\_202220\\_ISIS2304\\_I/Proyecto/ST-Pry-DocMarco-It1.pdf?isCourseFile=true&\\_&d2lSessionVal=ZXW7mHJ8YHBaf9ZZ0mPgH8UUe&ou=140479](https://bloqueneon.uniandes.edu.co//content/enforced/140479-UN_202220_ISIS2304_I/Proyecto/ST-Pry-DocMarco-It1.pdf?isCourseFile=true&_&d2lSessionVal=ZXW7mHJ8YHBaf9ZZ0mPgH8UUe&ou=140479).

