
 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

## Iteración 3

Camilo Andres Salins Martinez, Juan Felipe Sánchez  
 Universidad de los Andes, Bogotá, Colombia  
 {ca.salinas, jf.sanchezg}@uniandes.edu.co  
 Fecha de presentación: Noviembre 4 de 2018

### 1 Construcción de la aplicación, ejecución de pruebas y análisis de resultados

Diseño de datos:

Se construyó un web scrapper en Node.js para scrapear mockaroo, y enviar así multiples peticiones de descarga de archivos a la web en simultaneo, se generaron cerca de 800000 datos para las tablas necesarias.

El programa consiste en abrir una instancia de puppeteer (Chrome para node.js) abrir la pestaña de mockaroo, loguearse en una cuenta con los schemas de tablas hechos, y ejecutar una serie de veces el scrappeo de cada schema para generar 1000 registros por ejecución de schema, logrando así con un solo ciclo del programa obtener 17000 datos, ademas se le agrego una funcion al programa de funcionar de manera asincrona para ejecutar mas instancias, y a la vez generar mas datos.

### 2 Análisis y modelo conceptual

En esta sección se encuentra el listado de requerimientos funcionales y no funcionales que debe ofrecer la aplicación. De igual modo, se encuentra el modelo conceptual y las reglas de negocio.

#### 2.1 Requerimientos funcionales

<b>Nombre</b>	RF1. Registrar proveedores.
<b>Resumen</b>	Registra los proveedores del supermercado en el sistema
<b>Entradas</b>	
NIT: Es el número de identificación tributario del proveedor.	
Nombre	
Productos: Es una lista de los productos ofrecidos, cada uno de ellos debe poseer un precio y una calificación.	
<b>Resultados</b>	
Se registra al proveedor en el sistema	

### RNF asociados

Persistencia.

<b>Nombre</b>	RF2. Registrar productos.
<b>Resumen</b>	Registra los productos ofrecidos en el supermercado.
<b>Entradas</b>	
Nombre	
Marca	
Precio unitario	
Presentación	
Precio por unidad de medida	
Cantidad por presentación	
Unidad de medida:	Es la unidad de masa o volumen con la que se mide el producto, e.g: g, k, ml
Volumen del empaque:	Es el volumen del empaque, e.g: 150 cm <sup>3</sup>
Peso total:	Es el peso total del producto, incluye el peso del empaque.
Código de barras:	Identificador único del producto está escrito en número hexadecimal.
Categoría:	Es la categoría a la que pertenece el producto, puede ser inicialmente: Perecedero, Abarrotes y Electrodomésticos.
Tipo:	Es la clasificación de un producto dentro de una categoría. Con base en las categorías mencionadas pueden ser: <ol style="list-style-type: none"> <li>1. Perecederos: carnes, verduras, lácteos, huevos, comida del mar.</li> <li>2. Abarrotes: Escobas, trapeadores, baldes, medicamentos, higiene personal.</li> <li>3. Electrodomésticos: Neveras, Lavadoras, Televisores, Horno microondas, Hornos, Estufas.</li> </ol>
<b>Resultados</b>	
Se registra al producto en el sistema.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF3. Registrar cliente.
---------------	-------------------------

<b>Resumen</b>	Registra los clientes que posee el supermercado.
<b>Entradas</b>	
Tipo de documento: En caso de personas naturales el documento de identificación, las empresas es el NIT	
Numero de documento	
Nombre	
Correo Electrónico	
Dirección: Si el cliente se trata de una empresa se solicita la dirección.	
<b>Resultados</b>	
Se registra al cliente en el sistema.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF4. Registrar sucursal.
<b>Resumen</b>	Registra una sucursal del supermercado.
<b>Entradas</b>	
Ciudad	
Dirección	
Nombre	
Tipo Bodega: Es el tipo de producto que se almacena en las bodegas.	
Tipo Estante: Es el tipo de producto que se almacena en el estante.	
<b>Resultados</b>	
Se registra una sucursal del supermercado en el sistema.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF5. Registrar bodega de una sucursal.
<b>Resumen</b>	Registra una bodega de una sucursal del supermercado.
<b>Entradas</b>	
Sucursal: Es la sucursal quien se le va a agregar la bodega.	

Tipo Producto: Es el tipo de producto que se almacena en las bodegas.

Capacidad de volumen: Es el volumen máximo que puede almacenar la bodega.

Capacidad por peso: Es el peso máximo que soporta la bodega.

### Resultados

Se registra la bodega de la sucursal en el sistema.

### RNF asociados

Persistencia.

<b>Nombre</b>	RF6. Registrar estante de una sucursal.
<b>Resumen</b>	Registra un estante de una sucursal del supermercado.
<b>Entradas</b>	
Sucursal: Es la sucursal quien se le va a agregar la bodega.	
Tipo Producto: Es el tipo de producto que se almacena en las bodegas.	
Capacidad de volumen: Es el volumen máximo que puede almacenar la bodega.	
Capacidad por peso: Es el peso máximo que soporta la bodega.	
<b>Resultados</b>	
Se registra el estante de la sucursal en el sistema.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF7. Registrar promoción
<b>Resumen</b>	Registra una promoción ofrecida por el supermercado.
<b>Entradas</b>	
Tipo: Es el tipo de promoción que se va a ofrecer.	
Producto: Es el producto sobre el cual se va a ofrecer la promoción.	
Unidades disponibles: Numero de unidades disponibles para vender.	
Fecha inicio: Es la fecha en la cual comienza a aplicar la promoción	
Fecha final: Fecha de cierre de la promoción.	
<b>Resultados</b>	

Se registra la promoción del producto de manera satisfactoria.

#### RNF asociados

Persistencia.

<b>Nombre</b>	RF8. Finalizar promoción
<b>Resumen</b>	Finaliza la promoción ofrecida por el supermercado.
<b>Entradas</b>	
<b>Resultados</b>	
Se elimina la promoción del sistema cuando se cumplan las condiciones dadas (llego a su fecha de clausura o se acabaron las unidades).	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF9. Registrar el pedido de un producto a un proveedor para una sucursal.
<b>Resumen</b>	Se registra el pedido de un producto a un proveedor para una cierta sucursal
<b>Entradas</b>	
Sucursal: Es la sucursal que realiza el pedido a un proveedor.	
Proveedor	
Producto	
Volumen: Es el numero o volumen de unidades necesarias que se van a solicitar del producto.	
Precio: Es el precio total de la orden del producto.	
Fecha de entrega: Es la fecha en la cual se espera que el producto sea entregado por el proveedor.	
<b>Resultados</b>	
Se registra la solicitud del producto al proveedor en el sistema.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF10. Registrar la llegada de un pedido a una sucursal.
---------------	---

<b>Resumen</b>	Se registra el pedido de un producto a un proveedor para una cierta sucursal
<b>Entradas</b>	
Pedido	
Fecha de llegada: Es la fecha en la que se recibe el producto.	
<b>Resultados</b>	
Se registra la llegada del pedido, se cambia el estado del pedido, se da una calificación y se actualizan existencias en bodega.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RF11. Registrar la venta de un producto a un cliente.
<b>Resumen</b>	Se registra la venta de n unidades de un producto. Se debe tener en cuenta si hay promociones vigentes del producto a vender.
<b>Entradas</b>	
Cliente	
Producto	
Unidades: Son el número de unidades compradas por un cliente.	
<b>Resultados</b>	
Se registra la venta, se actualizan inventarios del producto, se realiza la factura al cliente y se realizan pedidos de abastecimiento en caso de ser necesario.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RFC1. Mostrar el dinero recolectado por ventas en cada sucursal durante un periodo de tiempo y en el año corrido
<b>Resumen</b>	Se muestra el dinero recolectado por todas las ventas realizado en cada sucursal en un periodo de tiempo
<b>Entradas</b>	
Fecha inicial: Es la fecha de inicio del periodo de tiempo	
Fecha final: Es la fecha final del periodo de tiempo	
<b>Resultados</b>	

Se registra la venta, se actualizan inventarios del producto, se realiza la factura al cliente y se realizan pedidos de abastecimiento en caso de ser necesario.

#### RNF asociados

Persistencia.

<b>Nombre</b>	RFC2. Mostrar las 20 promociones más populares.
<b>Resumen</b>	Muestra las 20 promociones que más dinero vendieron en menor tiempo.
<b>Entradas</b>	
<b>Resultados</b>	
Se muestra las 20 promociones que mas vendieron en el menor lapso.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RFC3. Mostrar el índice de ocupación de cada una de las bodegas y estantes de una sucursal
<b>Resumen</b>	Dada una sucursal mencionar el índice de ocupación cada bodega y estante (estado actual / capacidad máxima) * 100
<b>Entradas</b>	
Sucursal: Es la sucursal sobre el cual se mostrará la capacidad de sus bodegas y estante.	
<b>Resultados</b>	
Lista con la capacidad individual de las bodegas y estantes de una sucursal.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RFC4. Mostrar los productos que cumplen cierta característica.
<b>Resumen</b>	De acuerdo con una característica de entrada, filtrar todos los productos de la cumplen.
<b>Entradas</b>	
Característica: Es el filtro que se desea hacer sobre los productos existentes en el supermercado	

### Resultados

Lista de productos que cumplen con la característica de entrada.

### RNF asociados

Persistencia.

<b>Nombre</b>	RFC5. Mostrar las compras hechas por SuperAndes a los proveedores
<b>Resumen</b>	Mostrar todas las ordenes de pedido hechas por el supermercado, es decir, por todas las sucursales a los proveedores.
<b>Entradas</b>	
<b>Resultados</b>	
Lista con todas las ordenes de pedido realizadas con su respectiva información y la información del proveedor a quien solicitaron.	
<b>RNF asociados</b>	
Persistencia.	
<b>Nombre</b>	RFC6. Mostrar las ventas de SuperAndes a un usuario dado, en un rango de fechas indicado.
<b>Resumen</b>	Mostrar todos los productos vendidos a un cliente en un rango de fecha dado.
<b>Entradas</b>	
Cliente	
Fecha inicial: Es la fecha inicial de consulta.	
Fecha final: Fecha final de la consulta.	
<b>Resultados</b>	
Lista con todos los productos vendidos al cliente en el rango de fecha indicado.	
<b>RNF asociados</b>	
Persistencia.	

<b>Nombre</b>	RFC7. ANALIZAR LA OPERACIÓN DE SUPERANDES
---------------	---



<b>Resumen</b>	Mostrar las fechas de mayor demanda según parámetros.
<b>Entradas</b>	
Rango de tiempo entre las cuales se van a dar las fechas de mayor demanda y un producto	
<b>Resultados</b>	
Lista de las fechas con mayor demanda dentro del rango de tiempo dado para el producto ingresado.	
<b>RNF asociados</b>	
Persistencia.	
<b>Nombre</b>	RFC8. ENCONTRAR LOS CLIENTES FRECUENTES.
<b>Resumen</b>	Mostrar todos los clientes frecuentes de una sucursal.
<b>Entradas</b>	
Fecha inicial: Es la fecha inicial de consulta.	
Fecha final: Fecha final de la consulta.	
Sucursal	
<b>Resultados</b>	
Lista con todos los clientes que han visitado la sucursal dada más de dos veces en el rango de tiempo.	
<b>RNF asociados</b>	
Persistencia.	

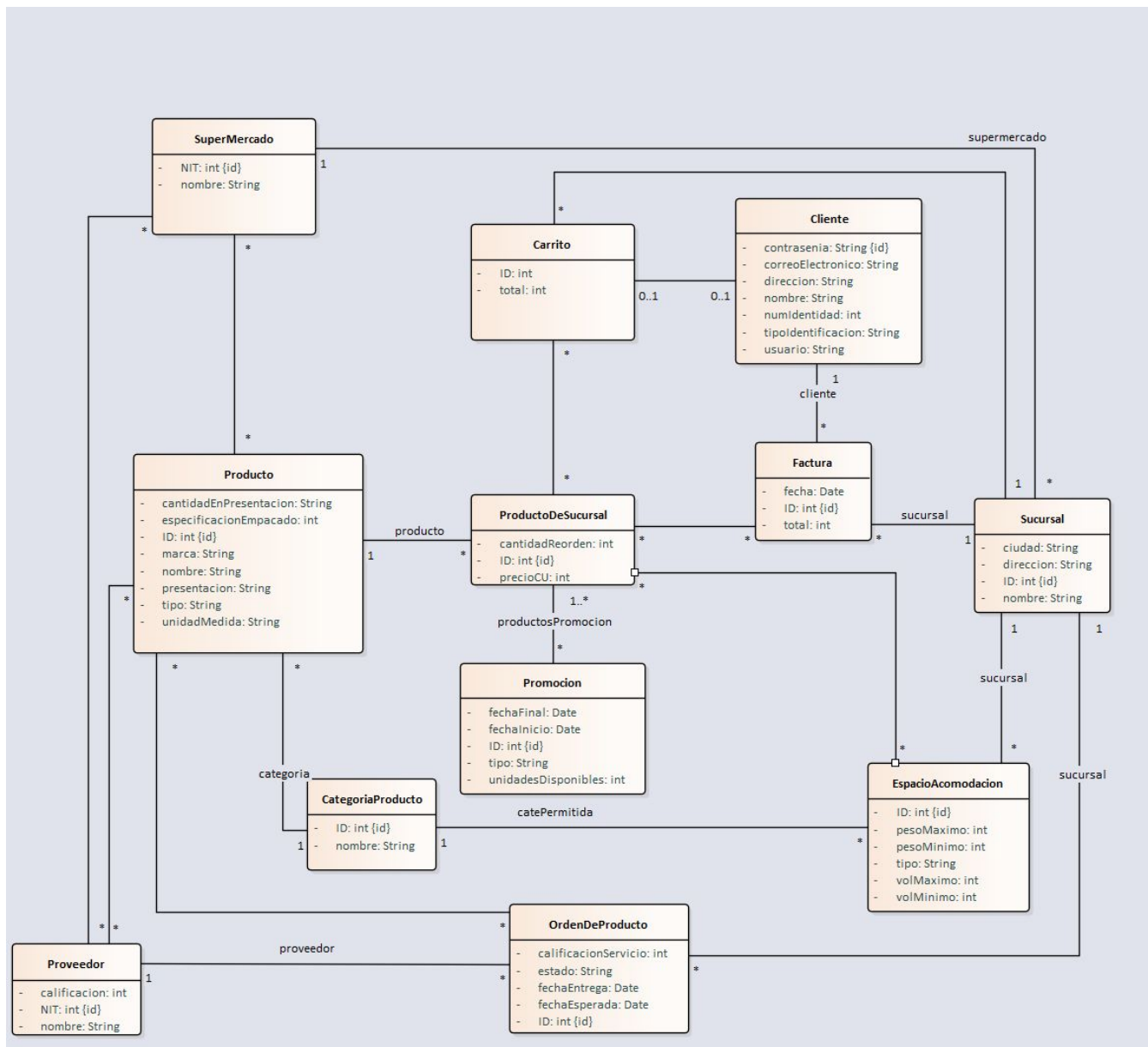
<b>Nombre</b>	RFC9. ENCONTRAR LOS PRODUCTOS QUE NO TIENEN MUCHA DEMANDA.
<b>Resumen</b>	Encontrar los productos que hayan tenido un periodo de reorden mayor a 2 meses.
<b>Entradas</b>	
Sucursal	
<b>Resultados</b>	
Lista con todos los productos con un periodo de reorden mayor a 2 meses en una sucursal dada.	



## RNF asociados

Persistencia.

## 2.2 Modelo conceptual y reglas de negocio





 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

Figura 1: Modelo conceptual del caso de estudio

### 3 Clases para persistir

Las clases cuya información debe persistir en la base de datos pues los datos que contienen son de vital importancia en el modelo son:

1. **Producto:** Contiene la información de los productos que ofrece la cadena de supermercados SuperAndes. Algunos datos que posee son, por ejemplo: nombre y marca. Esta clase debe persistir porque los productos se encuentran involucrados en todas las operaciones de negocio como la compra a proveedores de un producto específico y la venta de ellos al cliente.
2. **ProductoDeSucursal:** Contiene el identificador de un producto en específico que ofrece una sucursal de la cadena. De igual modo a esta clase se le asocia las promociones de un tipo de producto en una sucursal para controlar las existentes.
3. **CategoriaProducto:** Contiene la información de las categorías existentes y permitidas de todos los productos. Esta clase debe persistir debido a que con ella se corrobora la relación de todos los productos ofrecidos en el supermercado con una categoría existente.
4. **Sucursales:** Esta clase debe persistir debido a que contiene la información de las sucursales en todo el país de la cadena SuperAndes. En ella se encuentran datos asociados a su localización y el nombre que poseen. Con la información de esta clase también se asocian las bodegas y los estantes y los productos que en ella se ofrecen.
5. **EspacioAcomodacion:** El espacio de acomodación se define como los lugares donde se encuentran los productos en los supermercados. El espacio puede ser de tipo bodega o estante. Esta clase debe persistir pues contiene la información de los productos existentes en toda una sucursal al igual de permitir conocer si se debe realizar un pedido para reabastecer.
6. **Proveedores:** Contiene la información básica del proveedor de la cadena de supermercados, la lista de los productos que el ofrece y una calificación. Esta clase debe persistir pues en ella se consigna la información de los proveedores que es necesaria para la realización de ordenes de abastecimiento de productos.
7. **OrdenDeProducto:** Contiene la información del cual se va a hacer una compra al proveedor. En ella se encuentra la fecha en la que se realizó y la fecha aproximada de entrega de este. Al igual la cantidad del producto y el precio acordado con el proveedor.
8. **Cliente:** Contiene la información de todos los clientes que realizan compras en la cadena de supermercados SuperAndes. Esta clase debe persistir pues su información es necesaria para identificar las compras y realizar procesos de facturación.
9. **Promociones:** Contiene la información de todas las promociones que se realizan sobre un producto del supermercado en una sucursal. Esta clase debe persistir pues es necesaria para llevar un control de los productos vendidos en una promoción de un producto.
10. **Factura:** Esta clase lleva todo el control y registro de las compras realizadas por los clientes en la cadena de supermercados SuperAndes. Esta clase debe persistir pues permite llevar un

control sobre las ventas al igual que proporciona la información para realizar informes a las entidades de control como la DIAN.

11. Carrito: Esta clase lleva un registro de los productos que serán potencialmente vendidos a un cliente específico.

**Reglas de negocio:**

1. Los datos de ninguna clase deben ser valores nulos a excepción de la dirección de residencia del cliente y el nombre de la sucursal.
2. No puede haber clientes con el mismo tipo de documento y el mismo número de identificación
3. Para registrar una nueva sucursal en la ciudad se debe verificar que no exista una ya con la misma dirección.
4. El valor de una factura debe ser un número mayor a cero.
5. Los espacios de acomodación de los productos (estantes y bodegas) deben estar asociados a una sucursal existente.
6. El tipo de un espacio de acomodación debe estante o bodega únicamente.
7. El volumen máximo de un espacio de acomodación debe ser mayor a su volumen mínimo  $V_{max} > V_{min}$
8. La cantidad de existencias de un producto es un valor mayor a cero.
9. La fecha inicial y final de una promoción no puede ser la misma. Se espera que la fecha final sea posterior a la inicial.
10. El código de barras de un producto debe ser un valor en hexadecimal.
11. Los productos deben pertenecer a una categoría existente.
12. Se deben realizar ordenes de pedido para abastecimiento siempre que el nivel de reorden se alcance.
13. El estado de una orden de pedido es EN\_ESPERA o ENTREGADO

#### **4 Modelo de tablas relacional**

A continuación, se presenta el modelo de tablas relacional que fue creado en la base de datos y usado para suplir los requerimientos de negocio.



Universidad de  
los Andes  
Facultad de Ingeniería

Ingeniería de Sistemas y Computación  
**Pregrado**  
ISIS2304 – Sistemas Transaccionales



PRODUCTOS							
CODIGO	NOMBRE	PRESENTACION	MARCA	UNIDAD_MEDIDA	CANTIDAD_EN_PRESENTACION	ESPECIFICACION_EMPACADO	TIPO
PK	NN	NN	NN	NN	NN,CK(>0)	NN,CK(>0)	NN
SUPERMERCADOS			SUCURSALES				
NIT	NOMBRE		ID	CIUDAD	DIRECCION	NOMBRE	ID_SUPER
PK	NN		PK	NN	NN		FK(supermercado.ID), NN
PROVEEDORES			OFRECE				
NIT	NOMBRE	CALIFICACION	ID_SUPERMERCADO	ID_PRODUCTO	PRECIO_CU	PRECIO_UDIDAD_MEDIDA	
PK	NN	NN, CK(between 0 and 10)	PK, FK(supermercados.nit)	PK, FK(producto.codigo)	NN, CK(>0)	CK(>0)	



Figura 2: Modelo de las relaciones: Productos, Supermercados, Sucursales, Proveedores y Ofrece

PROVEE				PRODUCTO DE SUCURSAL			
ID_PROVEEDOR	ID_PRODUCTO	PRECIO_CU	PRECIO_UNIDAD_MEDIDA	ID	ID_PRODUCTO	PRECIO_CU	CANTIDAD_REORDEN
PK, FK(proveedores.nit)	PK, FK(productos.codigo)	NN, CK(>0)	CK(>0)	PK	NN, FK(productos.codigo)	NN	NN
SUPERMERCADO_PROVEEDOR				PROMOCION			
ID_SUPERMERCADO	ID_PROVEEDOR			ID	FECHA_INICIO	FECHA_FINAL	TIPO
PK, FK(supermercados.nit)	PK, FK(proveedores.nit)			PK	NN	NN	NN
CATEGORIA_PRODUCTO				PRODUCTOS EN PROMOCION			
ID	NOMBRE			ID_PRODUCTO_SUC	ID_PROMOCION		
PK	NN			PK, FK(producto_de_suc.id)	PK, FK(promocion.id)		

Figura 3: Modelo de las relaciones: Proveen, Supermercado\_Proveedor, Producto\_Sucursal, Promociones Categorías y Productos\_Promocion.

PEDIDOS_SUCURSAL					
ID_SUCURSAL	ID_PROVEEDOR	ESTADO	FECHA_ESPERADA	FECHA_ENTREGA	CALIFICACION
FK(sucursas.ID), NN	FK(proveedores.ID), NN	NN CK('EN_ESPERA', 'ENTREGADO')	NN	CH(>FECHA_ESPERADA)	NN, CK(CALIFICACION BETWEEN 0 AND 10)
ORDEN_PRODUCTO					
ID_ORDEN	ID_PRODUCTO	PRECIO_CU_ACORDADO	CANTIDAD		
FK(Pedidos_Sucursal.ID), NN	FK(productos.ID), NN	NN, CK(>0)	NN, CK(>0)		
CLIENTE					
ID	NOMBRE	CORREO	DIRECCION	NUM_IDENTIDAD	TIPO_IDENTIDAD
PK	NN	NN, CK(CORREO LIKE '%@%')		NN, CK(>0)	NN

Figura 4: Modelo de las relaciones: Pedidos\_Sucursal, Orden\_Producto y Cliente

FACTURA							
ID	FECHA	TOTAL	ID_CLIENTE	ID_SUCURSAL			
PK, SA	NN	NN, CK(>0)	FK(cliente.ID), NN	FK(sucursal.ID), NN			
VENTA_PRODUCTO				EXISTENCIAS			
ID_PRODUCTO_SUC	ID_FACTURA	CANTIDAD		ID_PRODUCTO_SUC	ID_ESPACIO_AC	CANTIDAD	
PK, FK(productos_de_suc.id)	PK, FK(factura.id)	NN, CH(>0)		PK, FK(productos_de_suc.id)	PK, FK(espacio_aco.id)	NN, CH(>0)	
ESPACIO_ACOMODACION							
ID	TIPO	VOLUMEN_MAXIMO	VOLUMEN_MINIMO	ID_CATEGORIA	PESO_MAXIMO	PESO_MINIMO	ID_SUCURSAL
PK	NN, CK (IN ('BODEGA', 'ESTANTE'))	NN, CK(>0)	NN, CK(>0)	FK(categoria_producto.ID)	NN, CK(>0)	NN, CK(>0)	FK(sucursal.ID), NN



 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

Figura 5: Modelo de las relaciones: Factura, Venta\_Producto, Existencias, Espacio\_Acomodacion.

LA TABLA DE CLIENTE SUFRIÓ MODIFICACIONES DEBIDO A LOS NUEVOS REQUERIMIENTOS FUNCIONALES.

CLIENTE							
ID	NOMBRE	CORREO	DIRECCION	NUM_IDENTIDAD	TIPO_IDENTIDAD	CONTRASEÑA	USUARIO
PK	NN	NN, CK(CORREO LIKE '%@%')		NN, CK(>0)	NN	NN	NN

Figura 6: Modelo de las relaciones:Usuario después de modificaciones.

Además se crearon 2 nuevas tablas

CARRITO_DE_COMPRAS			
ID	ID_SUCURSAL	TOTAL	ID_CLIENTE
PK	NN	NN, CK(>=0)	FK

CONTENIDO_CARRITO		
ID_CARRITO	ID_PRODUCTO_SUCURSAL	CANTIDAD
PK	PK	NN, CK(>0)

Figura 7: Modelo de las relaciones: Carrito de compras, y contenido de carrito.



## 5 Resultados

En esta sección se mencionan los resultados obtenidos.

### 5.1 Resultados logrados

Se logro cumplir los con lo requerimientos funcionales de los numerales del 1 al 6 y del 9 al 11. Por otro lado, se logro cumplir con los requerimientos de consulta numerales 1 y 2. Las pruebas realizadas en el proyecto con código Java para los requerimientos funcionales 9 al 11 con Junit solo fueron exhaustivas para los requerimientos 9 y 10. En el requerimiento 9 se verifica los 3 casos propuestos (PK no repetida, FK existente y CHK). Las pruebas para el requerimiento 10 solo se verificó actualización correcta debido a que es un requerimiento que no realiza ninguna labor de agregar información. Las pruebas para el requerimiento 11 solo verifica funcionalidad correcta para registrar la venta de un producto, cabe destacar que dentro de esta única prueba se contempla la posibilidad de fallos y se toman las medidas necesarias para prevenir desbordes en la transacción como también para



 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

conservar la integridad de la información existente. Por otro lado, para los requerimientos de consulta 1 y 2 realizan las peticiones de consulta de manera satisfactoria y solo se verifican por el resultado obtenido de la base de datos.

Por otro lado, para los numerales de diseño, se logró realizar todo el planteamiento correcto del problema y realizar los modelos requeridos (conceptual UML y relacional) al igual que la descripción de los requerimientos funcionales presentes en este documento.

## 5.2 Resultados no logrados

No se pudo cumplir con pruebas exhaustivas para todos los requerimientos solicitados y solo se lograron las mencionadas en la sección anterior. De igual modo, tampoco se logró el desarrollo de una interfaz para el usuario. No se pudo realizar los requerimientos 3 y 4 de consulta.

## 6 Balance de pruebas

En la siguiente sección se presenta evidencia de la satisfacción de las pruebas logradas. Se detallan específicamente las pruebas de conexión y la de los requerimientos 9 y 10. Cabe resaltar que para los requerimientos solicitados se realizaron pruebas de los numerales 9, 10, 11 y RFC2

### 6.1 Conexión a la base de datos

La prueba de conexión se ejecuta satisfactoriamente, para el caso de una conexión correcta, la base de datos contiene las tablas y credenciales correctas y se puede abrir una sesión. Como prueba de su culminación satisfactoria se anexa una imagen del log del datanucleus.



```
=====
Datastore Adapter : org.datanucleus.store.rdbms.adapter.OracleAdapter
Datastore : name="Oracle" version="Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
Real Application Testing options"
Datastore Driver : name="Oracle JDBC driver" version="12.2.0.1.0"
Primary Connection Factory : URL[jdbc:oracle:thin:@fn3.oracle.virtual.uniandes.edu.co:1521]
Secondary Connection Factory : URL[jdbc:oracle:thin:@fn3.oracle.virtual.uniandes.edu.co:1521]
Datastore Identifiers : factory="datanucleus2" case=UPPERCASE schema=ISIS2304A341826
Supported Identifier Cases : "MixedCase" UPPERCASE "MixedCase-Sensitive"
Supported Identifier Lengths (max) : Table=30 Column=30 Constraint=30 Index=30 Delimiter=
Support for Identifiers in DDL : catalog=false schema=true
Datastore : checkTableViewExistence, rdbmsConstraintCreateMode=DataNucleus, initialCatalog=
Support Statement Batching : yes (max-batch-size=50)
Queries : Results direction=forward, type=forward-only, concurrency=read-only
Java-Types : string-default-length=255
JDBC-Types : NCLOB, NCHAR, [id=2007], REF, CLOB, BLOB, ARRAY, STRUCT, DATE, TIME, TIMESTAMP,
=====
- ManagedConnection CERRADO : "org.datanucleus.store.rdbms.ConnectionFactoryImpl$ManagedConnection
StoreManager now created
- Started pool of ExecutionContext (maxPool=20, reaperThread=false)
- ===== NucleusContext =====
- DataNucleus Context : Versión: "5.1.10" con JRE "1.8.0_181" en "Windows 7"
- Persistence API : JDO
- Persistence-Unit : SuperAndes
- Plugin Registry : org.datanucleus.plugin.NonManagedPluginRegistry
- ClassLoading : default
- Persistence : pm-singlethreaded, nontransactional-read, nontransactional-write, nontransactional-read-write
- Transactions : type=RESOURCE_LOCAL, mode=datastore, isolation=read-committed
- ValueGeneration : txn-isolation=read-committed connection=New
```

Figura 8: Conexión satisfactoria a la SMD

Por otro lado para la ejecución de la prueba con tablas inexistentes en el esquema se tiene, el archivo ejecutado es TablasBD\_B.json

```
javax.jdo.JDODataStoreException: Error con la ejecución de Query de SQL "SELECT * FROM B_PROVEEDORES".
at org.datanucleus.api.jdo.NucleusJDOHelper.getJDOExceptionForNucleusException(NucleusJDOHelper.java:542)
at org.datanucleus.api.jdo.JDOQuery.executeInternal(JDOQuery.java:456)
at org.datanucleus.api.jdo.JDOQuery.executeList(JDOQuery.java:345)
at uniandes.isis2304.superAndes.persistencia.SQLProveedores.darProveedores(SQLProveedores.java:124)
at uniandes.isis2304.superAndes.persistencia.PersistenciaSuperAndes.darProveedores(PersistenciaSuperAndes.java:612)
at uniandes.isis2304.superAndes.negocio.SuperAndes.darProveedores(SuperAndes.java:141)
at uniandes.isis2304.superAndes.test.ConexionTest.tablaInexistenteTest(ConexionTest.java:182)
```

Figura 9: Intento fallido de conexión a la base de datos con tablas inexistentes

Para el tercer y cuarto caso corresponden a los intentos de abrir una conexión a la base de datos con credenciales inexistentes o con tablas que presentan errores. Se obtiene un resultado correcto el cual es que la conexión falle por las razones escritas. El resultado es el siguiente:

```
javax.jdo.JDOFatalUserException: A property named javax.jdo.PersistenceManagerFactoryClass must be specified, or a jar file with a META-INF,  
    at javax.jdo.JDOHelper.getPersistenceManagerFactory(JDOHelper.java:870)  
    at javax.jdo.JDOHelper.getPersistenceManagerFactory(JDOHelper.java:1109)  
    at javax.jdo.JDOHelper.getPersistenceManagerFactory(JDOHelper.java:924)  
    at uniandes.isis2304.superAndes.persistencia.PersistenceSuperAndes.<init>(PersistenciaSuperAndes.java:154)  
    at uniandes.isis2304.superAndes.persistencia.PersistenceSuperAndes.getInstance(PersistenciaSuperAndes.java:179)  
    at uniandes.isis2304.superAndes.negocio.SuperAndes.<init>(SuperAndes.java:53)  
    at uniandes.isis2304.superAndes.test.ConexionTest.baseDatosInaccesible(ConexionTest.java:112)  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
    at sun.reflect.NativeMethodAccessorImpl.invoke(Native Method)  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:46)
```

Figura 10: Conexión a la base de datos con una unidad de persistencia inexistente en la clase del proyecto SuperAndes

```
javax.jdo.JDOFatalDataStoreException: ORA-01017: invalid username/password; logon denied

    at org.datanucleus.api.jdo.NucleusJDOHelper.getJDOExceptionForNucleusException(NucleusJDOHelper.java:528)
    at org.datanucleus.api.jdo.JDOPersistenceManagerFactory.freezeConfiguration(JDOPersistenceManagerFactory.java:868)
    at org.datanucleus.api.jdo.JDOPersistenceManagerFactory.createPersistenceManagerFactory(JDOPersistenceManagerFactory.java:346)
    at org.datanucleus.api.jdo.JDOPersistenceManagerFactory.getPersistenceManagerFactory(JDOPersistenceManagerFactory.java:225)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at javax.jdo.JDOHelper$16.run(JDOHelper.java:1975)
    at java.security.AccessController.doPrivileged(Native Method)
```

Figura 11: Intento fallido de conexión pues no se tienen las credenciales correctas.

## 6.2 Requerimiento funcional #9

Para la verificación del requerimiento #9 se incluye en detalle la creación de solicitud, la siguiente imagen refleja la creación correcta de una orden de pedido.

Imprimiendo en consola el pedido realizado

Orden Producto [idOrden =9024, estado =EN ESPERA, fechaEsperada =2018/02/10 - 00:00, fechaEntrega =2018/04/10 - 00:00, calificacion =7]

Ahora deberia aparecer la excepcion





 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

Figura 10: Intento de registro de pedido correcto

Posteriormente cuando se intenta crear la misma orden de pedido se obtiene el siguiente error por violación de PK.

```
java.sql.SQLIntegrityConstraintViolationException: ORA-00001: restricción única (ISIS2304A341820.PK_PEDIDOS) violada
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:494)
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:446)
at oracle.jdbc.driver.T4C8Oall.processError(T4C8Oall.java:1054)
at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:59)
```

Figura 12: Intento de registro de un pedido existente

Por otro lado, si se le intenta asociar al pedido valores cuya llave foránea sea inexistente, como por ejemplo asociar el pedido para una sucursal inexistente se obtiene.

```
java.sql.SQLIntegrityConstraintViolationException: ORA-02291: restricción de integridad (ISIS2304A341820.FK_SUCURSAL_PEDIDOS) violada - clave
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:494)
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:446)
at oracle.jdbc.driver.T4C8Oall.processError(T4C8Oall.java:1054)
at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
```

Figura 13: Intento de registro con una sucursal inexistente

Finalmente, si se intenta agregar una tupla que incumple con los chequeos por ejemplo: Un pedido con calificación mayor que la máxima se obtiene:

```
java.sql.SQLIntegrityConstraintViolationException: ORA-02290: restricción de control (ISIS2304A341820.CHK_CALIFICACION_PEDIDOS) violada
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:494)
at oracle.jdbc.driver.T4CTTIoer11.processError(T4CTTIoer11.java:446)
at oracle.jdbc.driver.T4C8Oall.processError(T4C8Oall.java:1054)
at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:59)
at oracle.jdbc.driver.T4CPreparedStatement.executeForRows(T4CPreparedStatement.java:910)
```

Figura 14: Intento de registro de un pedido con calificación mayor a la permitida

### 6.3 Requerimiento funcional 10

Se actualiza las existencias en bodega del producto del pedido. La prueba se ejecuta satisfactoriamente



Nombre de la unidad persistencia: SuperAndes

Consulta: `SELECT * FROM EXISTENCIAS`

Estado del producto

ENTREGADO

Consulta: `SELECT * FROM EXISTENCIAS`

Cantidad de productos antes de la actualizacion: 155 Cantidad posterior: 185

ID	ID_SUCURSAL	ID_PROVEEDOR	ESTADO	FECHA_ESPERADA	FECHA_ENTREGA	CALIFICACION
1	1	1	123 ENTREGADO	02/04/18	10/11/19	5

ID_PRODUCTO_SUC	ID_ESPACIO_ACOMO	CANTIDAD
1	1	185

Figura 15: Actualizacion de las reservas del producto