Universidad de los Andes

Proyecto: Entrega 2

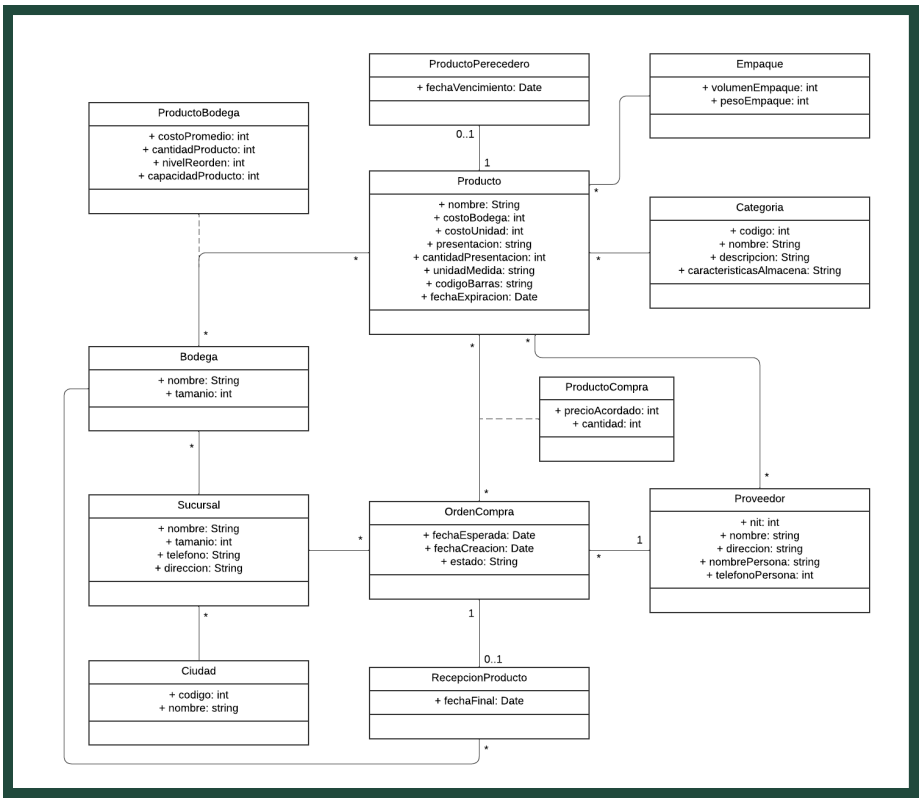
Sistemas Transaccionales 2024-20

Estudiante 1:	Johan Camilo Suarez Sinisterra
Estudiante 2:	Kalia Angelica Gonzalez Jimenez
Estudiante 3:	Juan Miguel Delgado

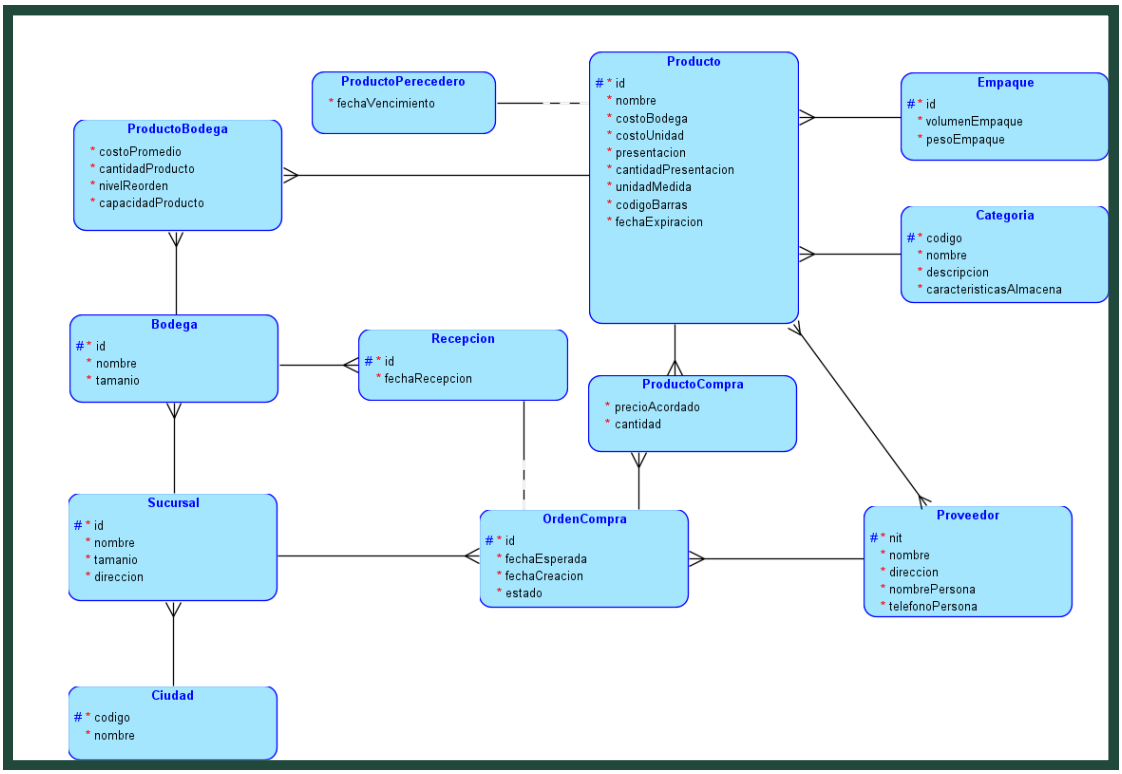
Contenido

Modelo de Datos

Modelo UML: Modificado



Modelo E/R: Modificado



Modelo Relacional: Modificado

Link del Excel con el modelo relacional:

ModeloRelacional.xlsx

Documentación nuevo RF

Para este nuevo requerimiento, se tenia que realizar una transacción en donde se registrara la entrada de distintos productos luego de que la orden de compra ya fuera recibida en la sucursal que la hizo. Para esto, se hizo una nueva clase llamada RecepcionService.java, en donde se maneja la transacción completamente para que en caso de un error, esta pueda hacer rollback sin ningún inconveniente.

Así mismo, se realizaron nuevas funciones en las clases RecepcionRepository.java y RecepcionController.java para manejar cada las consultas y endpoints de este nuevo requerimiento. A continuación se muestran los cambios realizados para el cumplimiento de esta nueva funcionalidad:

```
@Modifying
@Transactional
@Query(value = "INSERT INTO recepcion (id, fecharecepcion, bodega_id, ordencompra_id, sucursal_nombre, proveedor_nombre, bodega_nombre) " +//
               "VALUES(recepcion_sequence.NEXTVAL, "+//
               "(SELECT OrdenCompra.fechaesperada FROM OrdenCompra WHERE OrdenCompra.id = :ordencompra_id),:bodega_id,:ordencompra_id, "+//
               "(SELECT Sucursal.nombre FROM Sucursal INNER JOIN OrdenCompra ON OrdenCompra.sucursal_id = Sucursal.id WHERE OrdenCompra.id = :ordencompra_id), " +//
               "(SELECT Proveedor.nombre FROM Proveedor INNER JOIN OrdenCompra.proveedor_nit = Proveedor.nit WHERE OrdenCompra.id = :ordencompra_id), "+//
               "(SELECT Bodega.nombre FROM Bodega WHERE Bodega.id = :bodega_id)) ",nativeQuery = true)
void insertarRecepcion(@Param("ordencompra_id") int ordencompra_id, @Param("bodega_id") int bodega_id);
```

Estas sentencias se encargan de poder registrar cada uno de los documentos de recepción de productos con toda la información solicitada. En esta, se puede observar que se reciben dos parámetros, el id de la orden de compra y el id de la bodega donde se van a guardar los productos, los cuales son los únicos datos que ingresa el usuario.

Luego, se actualiza el costo promedio en bodega, la cantidad en bodega y la orden de compra pasa a estado “Entregada”. Esto se hace con las siguientes sentencias:

```
@Modifying
@Transactional
@Query(value = "UPDATE ProductoBodega PB SET "+//
               "PB.cantidadproducto = PB.cantidadproducto + (SELECT ProductoCompra.cantidad FROM OrdenCompra INNER JOIN ProductoCompra ON OrdenCompra.id = ProductoCompra.ordencompra_id WHERE OrdenCompra.id = :ordencompra_id) "+//
               "PB.costopromedio = ROUND(((PB.cantidadproducto * PB.costopromedio) + ((SELECT SELECT ProductoCompra.precioacordado FROM OrdenCompra INNER JOIN ProductoCompra ON OrdenCompra.id = ProductoCompra.ordencompra_id WHERE Or
               "WHERE PB.bodega_id = :bodega_id AND PB.producto_codigobarras IN (SELECT ProductoCompra.producto_codigobarras FROM OrdenCompra INNER JOIN ProductoCompra,ordencompra_id = OrdenCompra.id WHERE OrdenCompra.id = :ordencom
void actualizarProductoBodega(@Param("bodega_id") int bodega_id, @Param("ordencompra_id") int ordencompra_id);
```

```
@Modifying
@Transactional
@Query(value = "UPDATE OrdenCompra SET estado=:estado WHERE id=:id", nativeQuery = true)
void actualizarOrdenCompra(@Param("id") int id, @Param("estado") String estado);
```

Ya con las sentencias realizadas, lo único que se debe hacer, es conectar el service y el controller adecuadamente para que el requerimiento pueda funcionar correctamente. Esto lo hacemos mediante los siguientes métodos:

```java
@PostMapping("/recepcion/new/save")
public ResponseEntity<String> guardarRecepcion(@RequestBody Recepcion recepcion) {
    try {
        int ordenCompraId = recepcion.getOrdenCompra().getId();
        int bodegaId = recepcion.getBodega().getId();
        System.out.println("OrdenCompra ID: " + ordenCompraId);
        System.out.println("Bodega ID: " + bodegaId);

        recepcionService.recepcionRF10(ordenCompraId, bodegaId);
        recepcionService.actualizarProductoBodega(ordenCompraId, bodegaId);
        return new ResponseEntity<>(body:"Recepción creada exitosamente", HttpStatus.CREATED);
    } catch (Exception e) {
        return new ResponseEntity<>(body:"Error al crear la Recepción", HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

```java
@Transactional(isolation = Isolation.SERIALIZABLE)
public void recepcionRF10(int ordencompra_id, int bodega_id){

    try{

        recepcionRepository.insertarRecepcion(ordencompra_id, bodega_id);

    }catch (Exception e){
        System.out.println(x:"Error al registrar la recepcion de productos.");
    }

}

@Transactional(isolation = Isolation.SERIALIZABLE)
public void actualizarProductoBodega(int ordencompra_id, int bodega_id){

    try{
        recepcionRepository.actualizarProductoBodega(bodega_id, ordencompra_id);
        ordenCompraRepository.actualizarOrdenCompra(ordencompra_id, estado:"Entregada");
    }catch (Exception e){
        System.out.println(x:"No se puede actualizar la recepcion de productos");
    }

}
```

Documentación nuevos RFC

- **RFC6 Consulta Documentos de ingreso de Productos a Bodega - SERIALIZABLE**
  - RecepcionRepository: En el repósitorio de la identidad , se tiene elmetodo obtenerRecepcion(idB,idS) cuyos parametros son el id de la bodega y el id de la sucursal. Este retorna una coleccion de recepciones que cumple con lo ingresado.
  - RecepecionService: En el servicio, se tiene el metodo consultarRfc6(idB,idS) donde llava al metodo obtenerRecepcion explicado anteriormente y realiza una pausa de 30000 para mantener el bloqueo y luego proseguir a retornar la coleccion de recepciones despues de dicho tiempo.
  - RecepcionController: En el controlador , se tiene el metodo consultaRfc6(idS,idB) donde llama el metodo del servicio explicado anteriormente y si no aparece error alguno, retorna el nombre de la recepcion que esta en la bodega y sucursal ingresados.
- **RFC7 Consulta Documentos de ingreso de Productos a Bodega - READ COMMITED**

- RecepcionRepository: En el repósitorio de la identidad , se tiene elmetodo obtenerRecepcion(idB,idS) cuyos parametros son el id de la bodega y el id de la sucursal. Este retorna una coleccion de recepciones que cumple con lo ingresado.
- RecepecionService: En el servicio, se tiene el metodo consultarRfc7(idB,idS) donde llava al metodo obtenerRecepcion explicado anteriormente y realiza una pausa de 30000 para mantener el bloqueo y luego proseguir a retornar la coleccion de recepciones despues de dicho tiempo.
- RecepcionController: En el controlador , se tiene el metodo consultaRfc7(idS,idB) donde llama el metodo del servicio explicado anteriormente y si no aparece error alguno, retorna el nombre de la recepcion que esta en la bodega y sucursal ingresados.

Escenarios de prueba

Usuario Oracle

Usuario: ISIS2304A2824020

En este usuario se encuentra la base de datos de nuestra aplicación.