localhost:8080/api/v1.0/clientes/grupos

{

    "grupo": "Clientes varios"

}

localhost:8080/api/v1.0/clientes

{

    "cliente": "Samy",

    "numeroIdentificacion": "1717748444001",

    "direcciones": [

        {

            "ciudad": "Quito",

            "direccion": "UIO",

            "telefonos": "22222",

            "contacto": "Luis mismo",

            "emails": [

                 "luis1@gmail.com",

                 "luis2@gmail.com"

            ],

            "predeterminada": true

        },

        {

            "ciudad": "GYE",

            "direccion":"Republica",

            "telefonos": "333333333",

            "contacto": "Pepe",

            "emails": [

                 "pepe1@gmail.com"

            ],

            "predeterminada": false

        }

    ]

}

localhost:8080/api/v1.0/items

{

    "item": "Monitor",

    "codigoItem": "000001"

}

localhost:8080/api/v1.0/ventas

{

    "fechaEmision": "08/06/2023",

    "secuencia": "000000001",

    "idCliente": " 64d74da7-60d4-4561-927a-6321e6bd6767",

    "items": "2",

    "detalleItems": [

        {

            "idItem": " cf1fb454-1b4f-4762-9b21-a3a723db25a7",

            "codigoItem": "001",

            "item": "Venta de libros",

            "precioUnitario": 100,

            "cantidad": 3,

            "descuento": 10

        },

        {

           "idItem": " cf1fb454-1b4f-4762-9b21-a3a723db25a7",

            "codigoItem": "002",

            "item": "Venta de libros Java",

            "precioUnitario": 200,

            "cantidad": 1,

            "descuento": 20

        }

    ]

}

El proyecto consta de las siguientes entidades:

VtClientesEntity: Clientes

VtClientesDireccionesEntity: Direcciones de clientes

VtClientesGruposEntity: Grupos de clientes

GeItemsEntity: Items

GeItemsImpuestosEntity: Impuestos del Item

VtVentaEntity: Ventas cabecera

VtVentaDetalleEntity: Ventas detalle

VtVentaValoresEntity: Ventas valores totales

Lo que se requiere:

Actualmente existe create para agregar registros uno por uno, se requiere recibir en listas, guardar la información y en caso de error (si el largo de numeroIdentificacion del cliente es menor a 13) y devolver una lista de errores en caso de existir.

1. Recibir listas de grupos de clientes
2. Recibir lista de clientes (guardar las direcciones)
3. Recibir lista de ítems (guardar los impuestos)
4. Recibir lista de facturas (guardar los valores totales)

Para el caso de grupos de ítems, buscar por el nombre, si ya existe se debe devolver como error en la lista de errores.

Para el caso de clientes, si el numeroIdentificacion ya existe se debe actualizar con los nuevos datos, eliminar las direcciones actuales y guardar las nuevas direcciones.

Guardar las direcciones de las listas de clientes recibidas (VtClientesDireccionesEntity). Cada dirección puede tener una lista de email.

Para el caso de ítems, si el codigoItem ya existe se debe actualizar con los nuevos datos, eliminar los impuestos del item y guardar los nuevos impuestos.

Guardar los impuestos de las listas de ítems recibidas (GeItemsImpuestosEntity)

Para el caso de ventas, si la secuencia ya existe, se debe devolver como error.

Guardar los valores totales de la factura en la tabla (VtVentaValoresEntity)

Implementar findAll para todas las entidades que devuelva la lista paginada con jpa Specification con diferentes filtros de búsqueda, ordenados según criterio ASC, DESC.

Para el caso de clientes, devolver únicamente la dirección predeterminada para cada cliente (no devolver todas las direcciones).

Para el caso de ventas devolver la pagina solicitada que incluya los valores totales de totas las paginas, para que el front pueda mostrar la pagina solicitada pero también los valores totales de toda la búsqueda.

Implementar findById para todas las entidades.