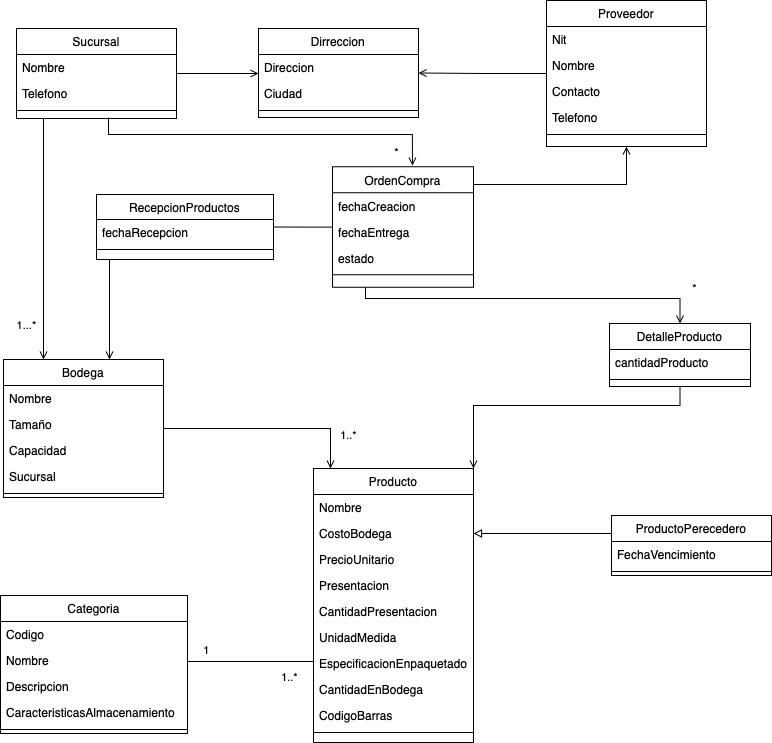
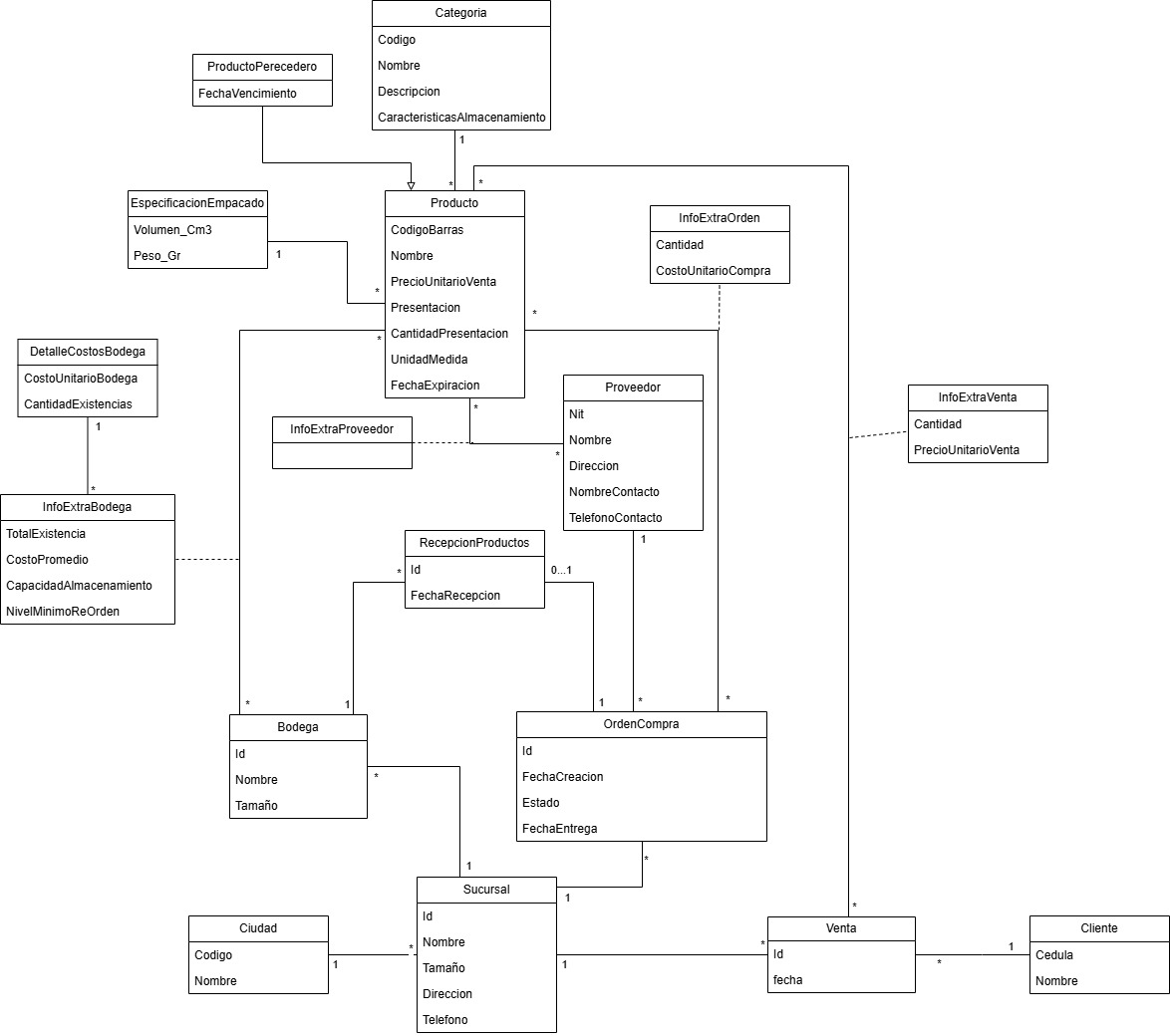
**Documento de informe Taller #1 Sistemas Transaccionales**

Paula Mora Cárdenas - 202211158

Julián Roberto Ramirez Alemán - 202310826

**Primera versión**



**Segunda versión**

**1. Modelos UML**

**Primera Versión**

**Entidades:**

Sucursal

Bodega

RecepcionProductos

Producto

Categoria

OrdenCompra

DetalleProducto

Proveedor

Cliente

Venta

**Relaciones:**

Sucursal a Bodega: Uno a muchos.

Bodega a Producto: Uno a muchos.

OrdenCompra a RecepcionProductos: Uno a muchos.

Producto a DetalleProducto: Uno a muchos.

Categoria a Producto: Uno a muchos.

Proveedor a OrdenCompra: Uno a muchos.

**Segunda Versión**

**Entidades:**

Sucursal

Bodega

RecepcionProductos

Producto

EspecificacionEmpacado

Categoria

OrdenCompra

Proveedor

Venta

Cliente

**Relaciones:**

Sucursal a Bodega: Uno a muchos.

Bodega a Producto: Uno a muchos.

OrdenCompra a RecepcionProductos: Uno a muchos.

Producto a EspecificacionEmpacado: Uno a uno.

Producto a Categoria: Uno a muchos.

Proveedor a OrdenCompra: Uno a muchos.

Venta a Cliente: Uno a muchos.

**2. Descripción de las Modificaciones**

* Adición de la Entidad EspecificacionEmpacado: En la segunda versión, se introdujo esta nueva entidad para detallar las características de empaque de los productos (volumen y peso).
* Atributos Adicionales: Se han agregado atributos como CostoUnitarioCompra en InfoExtraOrden y CostoPromedio, NivelMinimoReOrden en InfoExtraBodega.
* Eliminación de DetalleProducto: Esta entidad fue eliminada en la segunda versión, simplificando la relación entre productos y órdenes.

**3. Comprobación de la Forma Normal**

**Primera Versión**

1NF: Todas las entidades cumplen con la primera forma normal (atributos atómicos).

2NF: Todas las entidades cumplen con la segunda forma normal, ya que todos los atributos dependen completamente de la clave primaria.

3NF: Las entidades también cumplen con la tercera forma normal, sin dependencias transitivas.

**Segunda Versión**

1NF: Todas las entidades cumplen con la primera forma normal (atributos atómicos).

2NF: Todas las entidades cumplen con la segunda forma normal, asegurando que todos los atributos dependen completamente de la clave primaria.

3NF: Las entidades cumplen con la tercera forma normal, sin dependencias transitivas.

**4. Documentación de las clases**

**Clase Bodega**

**Descripción:** La clase Bodega representa una bodega en el sistema, que está asociada a una sucursal específica. Esta clase es parte del modelo de datos y se utiliza para mapear la entidad Bodega en la base de datos.

**Atributos:**

* private Long id: Identificador único de la bodega.
* private String nombre: Nombre de la bodega.
* private Double tamanio: Tamaño de la bodega en metros cuadrados.
* private Sucursal sucursal: Relación con la sucursal a la que pertenece la bodega.

**Métodos:**

* public Long getId(): Retorna el identificador de la bodega.
* public void setId(Long id): Establece el identificador de la bodega.
* public String getNombre(): Retorna el nombre de la bodega.
* public void setNombre(String nombre): Establece el nombre de la bodega.
* public Double getTamanio(): Retorna el tamaño de la bodega.
* public void setTamanio(Double tamanio): Establece el tamaño de la bodega.
* public Sucursal getSucursal(): Retorna la sucursal asociada a la bodega.
* public void setSucursal(Sucursal sucursal): Establece la sucursal asociada a la bodega.

**Clase Categoria**

**Descripción:** La clase Categoria representa una categoría de productos en el sistema. Esta clase se utiliza para organizar y clasificar los productos según sus características y requisitos de almacenamiento.

**Atributos:**

* private Long codigo: Identificador único de la categoría.
* private String nombre: Nombre de la categoría.
* private String descripcion: Descripción de la categoría.
* private String caracteristicasAlmacenamiento: Características específicas de almacenamiento para los productos de esta categoría.

**Métodos:**

* public Long getCodigo(): Retorna el código de la categoría.
* public void setCodigo(Long codigo): Establece el código de la categoría.
* public String getNombre(): Retorna el nombre de la categoría.
* public void setNombre(String nombre): Establece el nombre de la categoría.
* public String getDescripcion(): Retorna la descripción de la categoría.
* public void setDescripcion(String descripcion): Establece la descripción de la categoría.
* public String getCaracteristicasAlmacenamiento(): Retorna las características de almacenamiento de la categoría.
* public void setCaracteristicasAlmacenamiento(String caracteristicasAlmacenamiento): Establece las características de almacenamiento de la categoría.

**Clase Ciudad**

**Descripción:** La clase Ciudad representa una ciudad en el sistema. Se utiliza para almacenar información sobre las ciudades donde se encuentran las sucursales o bodegas.

**Atributos:**

* private Long codigo: Identificador único de la ciudad.
* private String nombre: Nombre de la ciudad.

**Métodos:**

* public Long getCodigo(): Retorna el código de la ciudad.
* public void setCodigo(Long codigo): Establece el código de la ciudad.
* public String getNombre(): Retorna el nombre de la ciudad.
* public void setNombre(String nombre): Establece el nombre de la ciudad.

**Clase Cliente**

**Descripción:** La clase Cliente representa a un cliente en el sistema. Se utiliza para almacenar información sobre los clientes, incluyendo su identificación y nombre.

**Atributos:**

* private Long cedula: Identificador único del cliente (número de cédula).
* private String nombre: Nombre del cliente.

**Métodos:**

* public Long getCedula(): Retorna la cédula del cliente.
* public void setCedula(Long cedula): Establece la cédula del cliente.
* public String getNombre(): Retorna el nombre del cliente.
* public void setNombre(String nombre): Establece el nombre del cliente.

**Clase DetalleCostoBodega**

**Descripción:** La clase DetalleCostoBodega representa los detalles del costo de los productos almacenados en una bodega. Se utiliza para gestionar la información sobre el costo unitario y la cantidad de existencias en la bodega.

**Atributos:**

* private Long id: Identificador único del detalle de costo de la bodega.
* private Double costoUnitarioBodega: Costo unitario de los productos en la bodega.
* private Integer cantidadExistencias: Cantidad de existencias disponibles en la bodega.

**Métodos:**

* public Long getId(): Retorna el identificador del detalle de costo de la bodega.
* public void setId(Long id): Establece el identificador del detalle de costo de la bodega.
* public Double getCostoUnitarioBodega(): Retorna el costo unitario de la bodega.
* public void setCostoUnitarioBodega(Double costoUnitarioBodega): Establece el costo unitario de la bodega.
* public Integer getCantidadExistencias(): Retorna la cantidad de existencias en la bodega.
* public void setCantidadExistencias(Integer cantidadExistencias): Establece la cantidad de existencias en la bodega.

**Clase EspecificacionEmpacado**

**Descripción:** La clase EspecificacionEmpacado representa las especificaciones de empaque de un producto, incluyendo su volumen y peso. Se utiliza para gestionar la información relacionada con el empaque de los productos.

**Atributos:**

* private Long id: Identificador único de la especificación de empaque.
* private Double volumen\_CM3: Volumen del producto en centímetros cúbicos.
* private Double peso\_Gr: Peso del producto en gramos.

**Métodos:**

* public Long getId(): Retorna el identificador de la especificación de empaque.
* public Double getVolumen\_CM3(): Retorna el volumen del producto en centímetros cúbicos.
* public void setVolumen\_CM3(Double volumen\_CM3): Establece el volumen del producto en centímetros cúbicos.
* public Double getPeso\_Gr(): Retorna el peso del producto en gramos.
* public void setPeso\_Gr(Double peso\_Gr): Establece el peso del producto en gramos.

**Clase InfoExtraBodega**

**Descripción:** Representa información adicional sobre los productos almacenados en una bodega, utilizando una clave primaria compuesta.

**Atributos:**

* @EmbeddedId private InfoExtraBodegaPK pk: Clave primaria compuesta.

**Métodos:**

* public InfoExtraBodegaPK getPk(): Retorna la clave primaria.
* public void setPk(InfoExtraBodegaPK pk): Establece la clave primaria.

**Clase InfoExtraBodegaPK**

**Descripción:** Representa la clave primaria compuesta para la entidad InfoExtraBodega, incluyendo referencias a otros objetos y atributos adicionales.

**Atributos:**

* @ManyToOne private Producto codBarras: Referencia al producto.
* @OneToOne private Bodega idBodega: Referencia a la bodega.
* @OneToOne private DetalleCostoBodega idCostobodega: Referencia al detalle de costo de la bodega.
* private Integer totalExistencia: Total de existencias.
* private Double costoPromedio: Costo promedio de los productos.
* private Double capacidadAlmacenamiento: Capacidad de almacenamiento.
* private Double nivelMinimoReorden: Nivel mínimo de reorden.

**Métodos:**

* public Integer getTotalExistencia(): Retorna el total de existencias.
* public void setTotalExistencia(Integer totalExistencia): Establece el total de existencias.
* public Double getCostoPromedio(): Retorna el costo promedio.
* public void setCostoPromedio(Double costoPromedio): Establece el costo promedio.
* public Double getCapacidadAlmacenamiento(): Retorna la capacidad de almacenamiento.
* public void setCapacidadAlmacenamiento(Double capacidadAlmacenamiento): Establece la capacidad de almacenamiento.
* public Double getNivelMinimoReorden(): Retorna el nivel mínimo de reorden.
* public void setNivelMinimoReorden(Double nivelMinimoReorden): Establece el nivel mínimo de reorden.
* public Producto getCodBarras(): Retorna el producto.
* public void setCodBarras(Producto codBarras): Establece el producto.
* public Bodega getIdBodega(): Retorna la bodega.
* public void setIdBodega(Bodega idBodega): Establece la bodega.

Clase InfoExtraOrden

**Descripción:** Representa información adicional sobre las órdenes de compra, utilizando una clave primaria compuesta para identificar de manera única cada entrada.

**Atributos**

* @EmbeddedId private InfoExtraOrdenPK pk: Clave primaria compuesta que incluye referencias a otros objetos y atributos relevantes.

**Métodos:**

* public InfoExtraOrdenPK getPk(): Retorna la clave primaria compuesta.
* public void setPk(InfoExtraOrdenPK pk): Establece la clave primaria compuesta.

**Clase InfoExtraOrdenPK**

**Descripción:** Representa la clave primaria compuesta para la entidad InfoExtraOrden, incluyendo referencias a otros objetos y atributos adicionales relacionados con las órdenes de compra.

**Atributos:**

* @ManyToOne private Producto codigoBarras: Referencia al producto.
* @OneToOne private OrdenCompra ordenCompra: Referencia a la orden de compra.
* private Integer cantidad: Cantidad de productos en la orden.
* private Double costoUnitarioCompra: Costo unitario de compra del producto.

**Métodos:**

* public Integer getCantidad(): Retorna la cantidad de productos.
* public void setCantidad(Integer cantidad): Establece la cantidad de productos.
* public Double getCostoUnitarioCompra(): Retorna el costo unitario de compra.
* public void setCostoUnitarioCompra(Double costoUnitarioCompra): Establece el costo unitario de compra.
* public Producto getCodigoBarras(): Retorna el producto.
* public void setCodigoBarras(Producto codigoBarras): Establece el producto.
* public OrdenCompra getOrdenCompra(): Retorna la orden de compra.
* public void setOrdenCompra(OrdenCompra ordenCompra): Establece la orden de compra.

**Clase InfoExtraProveedor**

**Descripción:** Representa información adi:cional sobre los proveedores de productos, utilizando una clave primaria compuesta para identificar de manera única cada entrada.

**Atributos:**

* @EmbeddedId private InfoExtraProveedorPK pk: Clave primaria compuesta que incluye referencias a otros objetos relacionados con el proveedor y el producto.

**Métodos:**

* public InfoExtraProveedorPK getPk(): Retorna la clave primaria compuesta.
* public void setPk(InfoExtraProveedorPK pk): Establece la clave primaria compuesta.

**Clase InfoExtraProveedorPK**

**Descripción:** Representa la clave primaria compuesta para la entidad InfoExtraProveedor, incluyendo referencias a otros objetos relacionados con el proveedor y el producto.

**Atributos:**

* @ManyToMany private Proveedor nit: Referencia al proveedor.
* @ManyToMany private Producto codBar: Referencia al producto.

**Métodos:**

* public Proveedor getNit(): Retorna el proveedor.
* public void setNit(Proveedor nit): Establece el proveedor.
* public Producto getCodBar(): Retorna el código de barras del producto.
* public void setCodBar(Producto codBar): Establece el código de barras del producto.

**Clase InfoExtraVenta**

**Descripción:** Representa información adicional sobre las ventas, utilizando una clave primaria compuesta para identificar de manera única cada entrada relacionada con una venta específica.

**Atributos:**

* @EmbeddedId private InfoExtraVentaPK pk: Clave primaria compuesta que incluye referencias a otros objetos y atributos relevantes para la venta.

**Métodos:**

* public InfoExtraVentaPK getPk(): Retorna la clave primaria compuesta.
* public void setPk(InfoExtraVentaPK pk): Establece la clave primaria compuesta.

**Clase InfoExtraVentaPK**

**Descripción:** Representa la clave primaria compuesta para la entidad InfoExtraVenta, incluyendo referencias a otros objetos y atributos adicionales relacionados con la venta.

**Atributos:**

* @ManyToOne private Producto codigoBarras: Referencia al producto asociado a la venta.
* @OneToOne private Venta id\_venta: Referencia a la venta específica.
* private Integer cantidad: Cantidad de productos vendidos.
* private Double precioUnitarioVenta: Precio unitario de venta del producto.

**Métodos:**

* public Integer getCantidad(): Retorna la cantidad de productos vendidos.
* public void setCantidad(Integer cantidad): Establece la cantidad de productos vendidos.
* public Double getPrecioUnitarioVenta(): Retorna el precio unitario de venta.
* public void setPrecioUnitarioVenta(Double precioUnitarioVenta): Establece el precio unitario de venta.
* public Producto getCodigoBarras(): Retorna el producto asociado a la venta.
* public void setCodigoBarras(Producto codigoBarras): Establece el producto asociado a la venta.
* public Venta getId\_venta(): Retorna la venta específica.
* public void setId\_venta(Venta id\_venta): Establece la venta específica.

**Clase OrdenCompra**

**Descripción:** Representa una orden de compra en el sistema, incluyendo detalles sobre la fecha de creación, estado, fecha de entrega y referencias a un proveedor y una sucursal.

**Atributos:**

* @Id @GeneratedValue(strategy=GenerationType.AUTO) private Long id: Identificador único de la orden de compra.
* private Date fechaCreacion: Fecha en que se creó la orden de compra.
* private String estado: Estado actual de la orden de compra.
* private Date fechaEntrega: Fecha programada para la entrega de la orden.
* @ManyToOne @JoinColumn(name = "Nit\_proveedor", referencedColumnName = "Nit\_proveedor") private Proveedor Nit\_proveedor: Referencia al proveedor asociado a la orden.
* @OneToOne @JoinColumn(name = "Id\_Sucursal", referencedColumnName = "Id\_Sucursal") private Sucursal Id\_Sucursal: Referencia a la sucursal donde se realiza la orden.

**Métodos:**

* public Long getId(): Retorna el identificador de la orden de compra.
* public void setId(Long id): Establece el identificador de la orden de compra.
* public Date getFechaCreacion(): Retorna la fecha de creación de la orden de compra.
* public void setFechaCreacion(Date fechaCreacion): Establece la fecha de creación de la orden de compra.
* public String getEstado(): Retorna el estado de la orden de compra.
* public void setEstado(String estado): Establece el estado de la orden de compra.
* public Date getFechaEntrega(): Retorna la fecha de entrega de la orden.
* public void setFechaEntrega(Date fechaEntrega): Establece la fecha de entrega de la orden.
* public Proveedor getNit\_proveedor(): Retorna el proveedor asociado a la orden.
* public void setNit\_proveedor(Proveedor nit\_proveedor): Establece el proveedor asociado a la orden.
* public Sucursal getId\_Sucursal(): Retorna la sucursal asociada a la orden.
* public void setId\_Sucursal(Sucursal id\_Sucursal): Establece la sucursal asociada a la orden.

**Clase Producto**

**Descripción:** Representa un producto en el sistema, incluyendo detalles como nombre, precio, presentación y referencias a categorías y especificaciones de empaque.

**Atributos:**

* @Id @GeneratedValue(strategy = GenerationType.AUTO) private Long codigoBarras: Identificador único del producto.
* private String nombre: Nombre del producto.
* private Double precioUnitarioVenta: Precio unitario de venta del producto.
* private String presentacion: Presentación del producto (por ejemplo, caja, botella).
* private Integer cantidadPresentacion: Cantidad de unidades en la presentación.
* private String unidadMedida: Unidad de medida del producto (por ejemplo, litros, kilogramos).
* private Date fechaExpiracion: Fecha de expiración del producto.
* @ManyToOne @JoinColumn(name = "codigo\_categoria", referencedColumnName = "Codigo\_Categoria") private Categoria codigo\_categoria: Referencia a la categoría del producto.
* @ManyToOne @JoinColumn(name = "id\_especificacionEmpacado", referencedColumnName = "Id\_especificacionEmpacado") private EspecificacionEmpacado id\_especificacionEmpacado: Referencia a la especificación de empaque del producto.

**Métodos:**

* public Long getCodigoBarras(): Retorna el código de barras del producto.
* public void setCodigoBarras(Long codigoBarras): Establece el código de barras del producto.
* public String getNombre(): Retorna el nombre del producto.
* public void setNombre(String nombre): Establece el nombre del producto.
* public Double getPrecioUnitarioVenta(): Retorna el precio unitario de venta.
* public void setPrecioUnitarioVenta(Double precioUnitarioVenta): Establece el precio unitario de venta.
* public String getPresentacion(): Retorna la presentación del producto.
* public void setPresentacion(String presentacion): Establece la presentación del producto.
* public Integer getCantidadPresentacion(): Retorna la cantidad en la presentación.
* public void setCantidadPresentacion(Integer cantidadPresentacion): Establece la cantidad en la presentación.
* public String getUnidadMedida(): Retorna la unidad de medida del producto.
* public void setUnidadMedida(String unidadMedida): Establece la unidad de medida del producto.
* public Date getFechaExpiracion(): Retorna la fecha de expiración del producto.
* public void setFechaExpiracion(Date fechaExpiracion): Establece la fecha de expiración del producto.
* public Categoria getCodigo\_categoria(): Retorna la categoría del producto.
* public void setCodigo\_categoria(Categoria categoria): Establece la categoría del producto.
* public EspecificacionEmpacado getId\_EspecificacionEmpacado(): Retorna la especificación de empaque del producto.
* public void setId\_EspecificacionEmpacado(EspecificacionEmpacado especificacionEmpacado): Establece la especificación de empaque del producto.

**Clase ProductoPerecedero**

**Descripción:** Representa un producto perecedero, que incluye una fecha de vencimiento adicional además de las características de un producto regular.

**Atributos:**

* private Date fechaVencimiento: Fecha en que el producto perecedero deja de ser apto para el consumo.

**Métodos:**

* public Date getFechaVencimiento(): Retorna la fecha de vencimiento del producto perecedero.
* public void setFechaVencimiento(Date fechaVencimiento): Establece la fecha de vencimiento del producto perecedero.

**Clase Proveedor**

**Descripción:** Representa un proveedor en el sistema, incluyendo información básica como nombre, dirección y datos de contacto.

**Atributos:**

* @Id private Long NIT: Número de Identificación Tributaria del proveedor.
* private String nombre: Nombre del proveedor.
* private String direccion: Dirección del proveedor.
* private String nombreContacto: Nombre de la persona de contacto en el proveedor.
* private Long telefonoContacto: Teléfono de contacto del proveedor.

**Métodos:**

* public Long getNIT(): Retorna el NIT del proveedor.
* public void setNIT(Long nIT): Establece el NIT del proveedor.
* public String getNombre(): Retorna el nombre del proveedor.
* public void setNombre(String nombre): Establece el nombre del proveedor.
* public String getDireccion(): Retorna la dirección del proveedor.
* public void setDireccion(String direccion): Establece la dirección del proveedor.
* public String getNombreContacto(): Retorna el nombre de contacto del proveedor.
* public void setNombreContacto(String nombreContacto): Establece el nombre de contacto del proveedor.
* public Long getTelefonoContacto(): Retorna el teléfono de contacto del proveedor.
* public void setTelefonoContacto(Long telefonoContacto): Establece el teléfono de contacto del proveedor.

**Clase RecepcionProductos**

**Descripción:** Representa la recepción de productos en el sistema, incluyendo detalles sobre la fecha de recepción y referencias a la bodega y la orden de compra asociadas.

**Atributos:**

* @Id @GeneratedValue(strategy=GenerationType.AUTO) private Long id: Identificador único de la recepción de productos.
* private Date fechaRecepcion: Fecha en que se recibe el producto.
* @OneToOne @JoinColumn(name = "id\_Bodega", referencedColumnName = "Id\_Bodega") private Bodega id\_Bodega: Referencia a la bodega donde se reciben los productos.
* @OneToOne @JoinColumn(name = "id\_OrdenCompra", referencedColumnName = "Id\_OrdenCompra") private OrdenCompra id\_OrdenCompra: Referencia a la orden de compra asociada a la recepción.

**Métodos:**

* public Long getId(): Retorna el identificador de la recepción de productos.
* public void setId(Long id): Establece el identificador de la recepción de productos.
* public Date getFechaRecepcion(): Retorna la fecha de recepción de productos.
* public void setFechaRecepcion(Date fechaRecepcion): Establece la fecha de recepción de productos.
* public Bodega getId\_Bodega(): Retorna la bodega asociada a la recepción.
* public void setId\_Bodega(Bodega id\_Bodega): Establece la bodega asociada a la recepción.
* public OrdenCompra getId\_OrdenCompra(): Retorna la orden de compra asociada a la recepción.
* public void setId\_OrdenCompra(OrdenCompra id\_OrdenCompra): Establece la orden de compra asociada a la recepción.

**Clase Sucursal**

**Descripción:** Representa una sucursal en el sistema, incluyendo información sobre su ubicación, tamaño y contacto, así como una referencia a la ciudad en la que se encuentra.

**Atributos:**

* @Id @GeneratedValue(strategy=GenerationType.AUTO) private Long id: Identificador único de la sucursal.
* private String nombre: Nombre de la sucursal.
* private Double tamanio: Tamaño de la sucursal.
* private String direccion: Dirección de la sucursal.
* private Integer telefono: Número de teléfono de la sucursal.
* @ManyToOne @JoinColumn(name = "codigo\_Ciudad", referencedColumnName = "codigo\_Ciudad") private Ciudad codigo\_Ciudad: Referencia a la ciudad asociada a la sucursal.

**Métodos:**

* public Long getId(): Retorna el identificador de la sucursal.
* public void setId(Long id): Establece el identificador de la sucursal.
* public String getNombre(): Retorna el nombre de la sucursal.
* public void setNombre(String nombre): Establece el nombre de la sucursal.
* public Double getTamanio(): Retorna el tamaño de la sucursal.
* public void setTamanio(Double tamanio): Establece el tamaño de la sucursal.
* public String getDireccion(): Retorna la dirección de la sucursal.
* public void setDireccion(String direccion): Establece la dirección de la sucursal.
* public Integer getTelefono(): Retorna el número de teléfono de la sucursal.
* public void setTelefono(Integer telefono): Establece el número de teléfono de la sucursal.
* public Ciudad getCodigo\_Ciudad(): Retorna la ciudad asociada a la sucursal.
* public void setCodigo\_Ciudad(Ciudad codigo\_Ciudad): Establece la ciudad asociada a la sucursal.

**Clase Venta**

**Descripción:** Representa una venta en el sistema, incluyendo la fecha de la venta, la sucursal donde se realizó y el cliente asociado.

**Atributos:**

* @Id @GeneratedValue(strategy=GenerationType.AUTO) private Long id: Identificador único de la venta.
* private Date fecha: Fecha en que se realizó la venta.
* @ManyToOne @JoinColumn(name = "id\_Sucursal", referencedColumnName = "Id\_Sucursal") private Sucursal id\_Sucursal: Referencia a la sucursal donde se realizó la venta.
* @OneToOne @JoinColumn(name = "id\_Cliente", referencedColumnName = "Id\_Cliente") private Cliente id\_Cliente: Referencia al cliente asociado a la venta.

**Métodos:**

* public Long getId(): Retorna el identificador de la venta.
* public void setId(Long id): Establece el identificador de la venta.
* public Date getFecha(): Retorna la fecha de la venta.
* public void setFecha(Date fecha): Establece la fecha de la venta.
* public Sucursal getId\_Sucursal(): Retorna la sucursal asociada a la venta.
* public void setId\_Sucursal(Sucursal id\_Sucursal): Establece la sucursal asociada a la venta.
* public Cliente getId\_Cliente(): Retorna el cliente asociado a la venta.
* public void setId\_Cliente(Cliente id\_Cliente): Establece el cliente asociado a la venta.