

Documento de Diseño de Software

Software: Integración E-commerce Iquique - Magento

Autor: Cristhian Moya

Versión: 1.0

Fecha: julio 2021

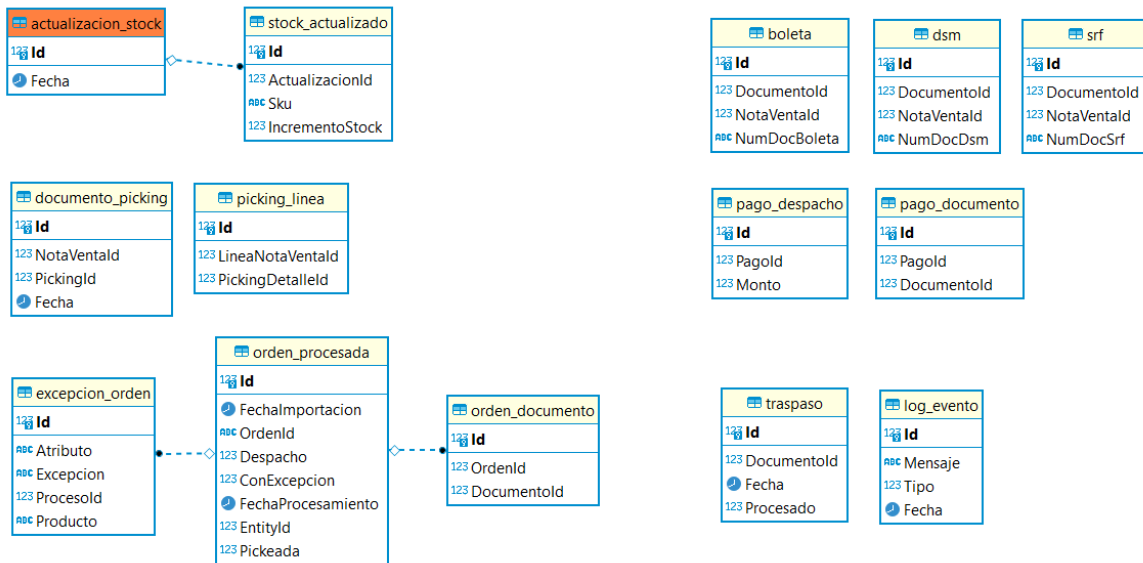
Contenido

1.	Modelo Entidad Relación.....	4
2.	Diccionario de Datos	5
2.1	Tablas agregadas	5
2.2	Nuevo trigger creado	9
3.	Diagrama de clases.....	10
3.1	Controladores	10
3.1.1	EventLogController	10
3.1.2	HomeController	10
3.1.3	NotaVentaController	11
3.1.4	ReportController.....	11
3.1.5	ServicioController	12
3.2	Modelos	13
3.2.1	Modelos utilizados para la importación desde la API de Magento	13
3.2.2	Modelos utilizados para empaquetado y transporte de datos	17
3.2.3	Modelos utilizados como Entidades de la Base de Datos BDIQUIQUE.....	20
3.3	Clase de servicio autónomo	23
3.4	Clases de utilidades	24
3.4.1	API.....	24
3.4.2	Util	24
3.4.3	Logger	25
3.4.4	LogType	25
3.4.5	Extensions	25
3.5	Clases de validación.....	25
3.5.1	MinFechaValidator	25
3.6	Clase de contexto de datos	26
3.6.1	AppDbContext	26
3.7	Clases de parámetros	27
3.7.1	Shared	27
3.7.2	TAMAÑO_CELDA	27
3.8	Clases de Filtros.....	27
3.8.1	BeforeActionFilter	27
4.	Diagrama de Contexto.....	28

4.1 Diagrama de interconexión de arquitectura.....	28
--	----

1. Modelo Entidad Relación

Se presenta el MER para las tablas que se han añadido a la base de datos BDIQUIQUE. Algunas relaciones no se muestran debido a que son llaves foráneas a de tablas ya existentes en la base de datos como DOCUMENTO, DOCUMENTOD, o PAGOE. Se dan mas detalles en el Diccionario de Datos.



2. Diccionario de Datos

El diccionario de datos describe las tablas y un trigger que se agregaron a la base de datos BDIQUIQUE. El encabezado de cada grilla muestra el nombre de la tabla y una breve descripción de ella. Luego se muestran los campos, sus tipos de datos y algunos comentarios explicativos sobre ellos.

2.1 Tablas agregadas

actualizacion_stock: Intentos de actualización automática de stock que se realizan hacia la API de Magento		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Fecha	datetime2	Timestamp de la actualización

boleto: Referencia a documentos de tipo boleto generados desde la aplicación de integración.		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
DocumentId	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la boleto
NotaVentaId	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la nota de venta asociada a la boleto
NumDocBoleto	nvarchar(MAX)	Número de documento del registro en la tabla DOCUMENTO, correspondiente a la boleto

documento_picking: Referencia a documentos de tipo picking generados desde la aplicación de integración.		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
NotaVentaId	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la nota de venta asociada al picking
PickingId	int	Identificador del registro en la tabla DOCUMENTO correspondiente al picking
Fecha	datetime2	Fecha en que se realiza el picking

dsm: Referencia a documentos de tipo DSM generados desde la aplicación de integración.		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
DocumentId	int	Identificador del registro en la tabla DOCUMENTO correspondiente al DSM
NotaVentaId	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la nota de venta asociada al DSM
NumDocDsm	nvarchar(MAX)	Número de documento del registro en la tabla DOCUMENTO, correspondiente al DSM

excepcion_orden: Excepciones producidas al intentar importar datos desde la API de Magento		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Atributo	nvarchar(MAX)	Campo en la entidad que se intentó importar desde la API, para el cual se produjo la excepción
Excepcion	nvarchar(MAX)	Detalle de la excepción producida
Procesold	int	Orden para la cual se produjo la excepción. Es una referencia a la tabla orden_procesada
Producto	nvarchar(MAX)	Campo opcional. Cuando la excepción está asociada a un producto, este campo contiene el SKU del producto para el cual se produjo la excepción.

log_evento: Eventos producidos en la comunicación con la API de Magento		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Mensaje	nvarchar(MAX)	Mensaje del evento producido
Tipo	int	Tipo del evento producido (1: ERROR, 0: INFORMACIÓN)
Fecha	datetime2	Timestamp del evento

orden_documento: Relaciona la orden importada desde la API de Magento y su correspondiente Nota de Venta		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
OrdenId	int	Identificador del registro en la tabla orden_procesada correspondiente a la orden que se está importando.
Documentold	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la Nota de Venta asociada a la orden que se está importando.

orden_procesada: Orden importada desde la API de Magento		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
FechaImportacion	datetime2	Timestamp de la importación
OrdenId	nvarchar(MAX)	Identificador alfanumérico de la orden de compra según registro en Magento
Despacho	float	Monto en pesos del valor asociado al despacho de la orden
ConExcepcion	bit	Indica si la orden fue cargada con o sin excepciones. (1: con excepción, 0: sin excepción)
FechaProcesamiento	datetime2	Campo opcional. Fecha en la que se procesa la orden ya sea como SRF o como Boleta. Si aún no ha sido procesada, este campo contendrá el valor NULL
EntityId	int	Identificador numérico de la orden de compra según registro en Magento
Pickeada	bit	Indica si ya se realizó el proceso de picking para esta orden o no. (1: ya fue pickeada, 0: no ha sido pickeada)

pago_despacho: Referencia a registro de pago generado, asociado al monto de despacho para una orden		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Pagold	int	Identificador del registro en la tabla PAGOE asociado al pago
Monto	float	Monto pagado relacionado con el despacho de la orden

pago_documento: Relaciona la orden importada desde la API de Magento y su correspondiente Nota de Venta		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Pagold	int	Identificador del registro en la tabla PAGOE asociado al pago
DocumentId	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la Nota de Venta a la que se le está asociando el pago

picking_linea: Relaciona una línea de picking con la línea de nota de venta que se está pickeando		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
LineaNotaVentald	int	Identificador del registro en la tabla DOCUMENTOD asociado a la línea de detalle de la nota de venta que se está pickeando
PickingDetalleld	int	Identificador del registro en la tabla DOCUMENTOD correspondiente a la línea de picking que se está creando

srf: Referencia a documentos de tipo SRF generados desde la aplicación de integración.		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Documentold	int	Identificador del registro en la tabla DOCUMENTO correspondiente al SRF
NotaVentald	int	Identificador del registro en la tabla DOCUMENTO correspondiente a la nota de venta asociada al SRF
NumDocSrf	nvarchar(MAX)	Número de documento del registro en la tabla DOCUMENTO, correspondiente al SRF

stock_actualizado: Actualizaciones a stock de productos que se han realizado. Cabe mencionar que la actualización de stock solo se realiza cuando se generan traspasos hacia la bodega B01 (ingreso de productos a esta bodega). Es decir, solo habrá aumento de stock hacia la API de Magento y nunca disminución.		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
ActualizacionId	int	Identificador del registro en la tabla actualizacion_stock correspondiente al proceso en el cual fue actualizado este stock
Sku	nvarchar(MAX)	SKU del producto para el que se está actualizando el stock
IncrementoStock	Int	Cantidad en la que se está aumentando el stock de este producto

traspaso: En esta tabla se inserta un registro cada vez que se realiza un traspaso hacia la bodega B01		
Campo	Tipo de datos	Comentario
Id	int (identity)	Identificador
Documentold	int	Identificador del registro en la tabla DOCUMENTO correspondiente al traspaso que se hizo
Fecha	datetime2	Timestamp del traspaso
Procesado	bit	Indica si el traspaso ya se procesó como actualización de stock en Magento o aún no. (1: ya fue procesado, 0: aún no ha sido procesado)

2.2 Nuevo trigger creado

El trigger que se creó tiene como objetivo insertar un registro en la tabla **traspaso**, cada vez que se genere un traspaso hacia la bodega B10 mediante el sistema legacy.

```
CREATE TRIGGER NuevoTraspasoRealizado on [DOCUMENTO]
FOR INSERT
AS DECLARE
    @TD VARCHAR(4),
    @BODENT VARCHAR(4),
    @BODSAL VARCHAR(4),
    @IDDOCTO INT

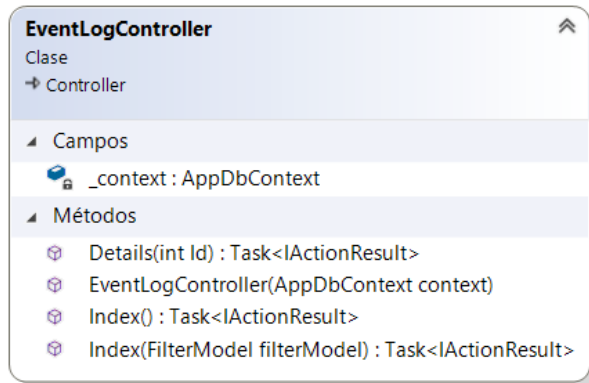
SELECT @TD = ins.TD FROM INSERTED ins;
SELECT @BODENT = ins.BODENT FROM INSERTED ins;
SELECT @BODSAL = ins.BODSAL FROM INSERTED ins;
SELECT @IDDOCTO = ins.IDDOCTO FROM INSERTED ins;

IF (@TD = 'TRP' AND @BODENT = 'B01' AND @BODSAL = 'CEN')
BEGIN
    INSERT INTO traspaso(
        [DocumentoId],
        [Fecha],
        [Procesado])
    VALUES (
        @IDDOCTO,
        GETDATE(),
        0);
END
```

3. Diagrama de clases

3.1 Controladores

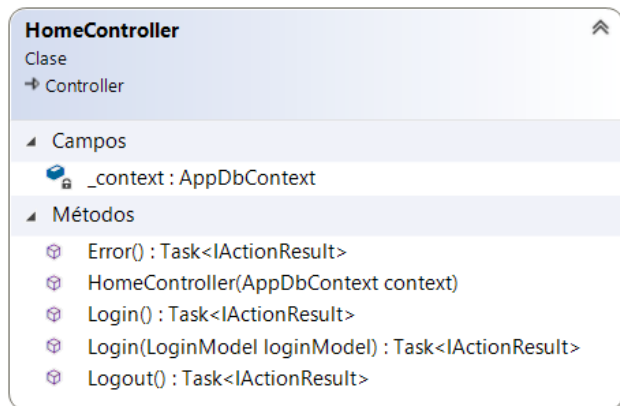
3.1.1 EventLogController



El método **Index** da acceso a la vista principal del controlador EventLog. En esta vista se muestra una grilla con los eventos que se han producido en las importaciones desde Magento. El modelo FilterModel se utiliza para capturar los parámetros de filtro que se aplican a la grilla.

El método **Details** despliega la vista de detalles de un evento en particular. El parámetro Id indica el identificador del registro log_evento que se desplegará.

3.1.2 HomeController

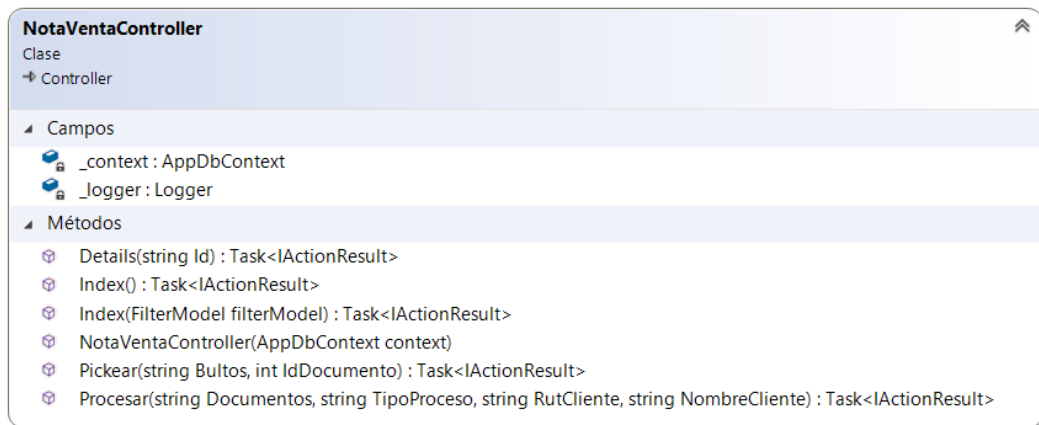


El método **Login** se utiliza para iniciar sesión en la aplicación. El modelo LoginModel captura los datos de inicio de sesión que ingresa el usuario.

El método **Logout** se utiliza para cerrar sesión en la aplicación.

El método **Error** se utiliza para desplegar una vista de error cuando ocurre una excepción en la aplicación.

3.1.3 NotaVentaController



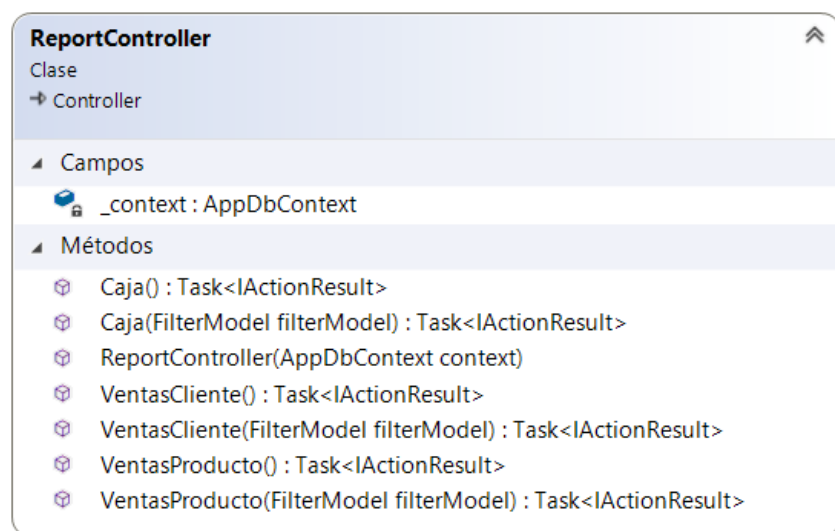
El método **Index** se utiliza para mostrar la vista principal que presenta la grilla de las notas de venta importadas. El modelo **FilterModel** captura los parámetros de filtro que se aplican a la grilla.

El método **Procesar** es utilizado para procesar una nota de venta ya sea como boleta o como SRF. El parámetro **Documentos** indica los ID de las notas de venta que ha seleccionado el usuario para procesar. El **TipoProceso** puede ser BOLETA o SRF. El **RutCliente** y **NombreCliente** se capturan desde la lista desplegable que aparece al procesar un SRF.

El método **Details** muestra los detalles de una nota de venta. El parámetro **Id** es el identificador de la nota de venta que se desplegará.

El método **Pickear** se utiliza para realizar un proceso de preparación de despacho para una nota de venta. El parámetro **Bultos** contiene un listado de objetos del tipo *BultoLinea* que contiene la información de bultos para cada línea de la nota de venta que se está pickeando. El parámetro **IdDocumento** tiene el identificador de la nota de venta que se está pickeando.

3.1.4 ReportController



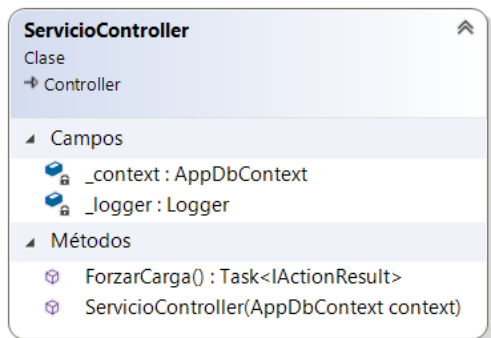
El controlador **ReportController** reúne los informes que puede desplegar el sistema.

El método **Caja** muestra el informe de caja. El modelo **FilterModel** captura los parámetros de filtro que se aplican a la grilla.

El método **VentasCliente** muestra el informe de ventas por cliente. El modelo **FilterModel** captura los parámetros de filtro que se aplican a la grilla.

El método **VentasProducto** muestra el informe de ventas por producto. El modelo **FilterModel** captura los parámetros de filtro que se aplican a la grilla.

3.1.5 ServicioController



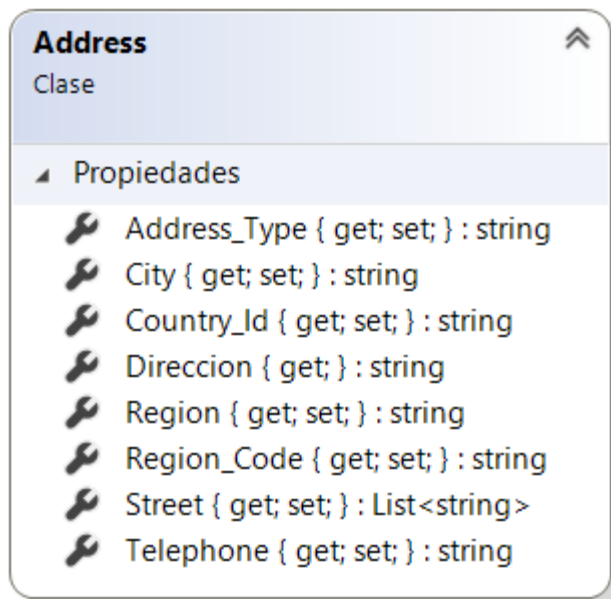
El controlador **ServicioController** se utiliza para llevar a cabo un método de servicio de importación o refresco de datos desde Magento, pero activado manualmente por el usuario.

El método **ForzarCarga** es llamado por el usuario, cuando ha hecho modificaciones o correcciones en los datos de Magento o la base de datos BDIQUIQUE y no desea esperar 1 hora para que el sistema actualice los cambios.

3.2 Modelos

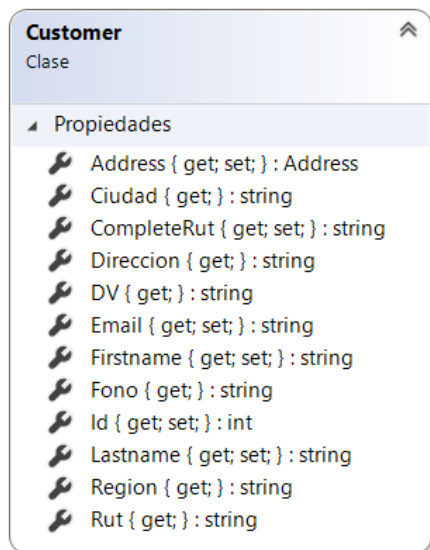
3.2.1 Modelos utilizados para la importación desde la API de Magento

3.2.1.1 Address



El modelo **Address** representa la dirección de un cliente. El atributo **Direccion** se genera automáticamente concatenando las cadenas de texto que vengan en el arreglo **Street**.

3.2.1.2 Customer






El modelo **Customer** representa a un cliente. Los campos **Direccion**, **Ciudad**, **Region** y **Fono** se obtienen automáticamente desde el objeto **Address** asociado a él. Los campos **Rut** y **DV** se obtienen automáticamente separando por “-” el campo **CompleteRut**.

3.2.1.3 *ExtensionAttribute*

ExtensionAttribute
Clase

▲ Propiedades




 Payment_Additional_Info { get; set; } : List<PaymentAdditionalInfo>
 Rut { get; set; } : string
 Shipping_Assignments { get; set; } : List<ShippingAssignment>

Se utiliza para capturar información adicional de los clientes, pagos y despachos de órdenes.

3.2.1.4 *MultiReponse*

MultiResponse
Clase

▲ Propiedades

 items { get; set; } : JArray
 search_criteria { get; set; } : object
 total_count { get; set; } : object

Se utiliza para recibir la respuesta de una llamada a la API que contiene múltiples objetos.
Ejemplo: Las órdenes completadas en un rango de fechas.

3.2.1.5 Order

Order	
Clase	
▲ Propiedades	
🔧	Created_At { get; set; } : DateTime
🔧	Customer { get; set; } : Customer
🔧	Customer_Email { get; set; } : string
🔧	Customer_Firstname { get; set; } : string
🔧	Customer_Id { get; set; } : int
🔧	Customer_Is_Guest { get; set; } : int
🔧	Customer_Lastname { get; set; } : string
🔧	Entity_Id { get; set; } : int
🔧	Extension_Attributes { get; set; } : ExtensionAttribute
🔧	Increment_Id { get; set; } : string
🔧	Items { get; set; } : List<ProductOrder>
🔧	Payment { get; set; } : Payment
🔧	Shipping_Amount { get; set; } : double?
🔧	State { get; set; } : string
🔧	Status { get; set; } : string
🔧	Subtotal_Incl_Tax { get; set; } : double?
🔧	Total_Due { get; set; } : double?

Contiene la información relativa a la orden de compra que se está importando. Cada orden de compra se traduce en una nota de venta en el sistema de Iquique.

3.2.1.6 Payment

Payment	
Clase	
▲ Propiedades	
🔧	FormaPago { get; } : string
🔧	Method { get; set; } : string

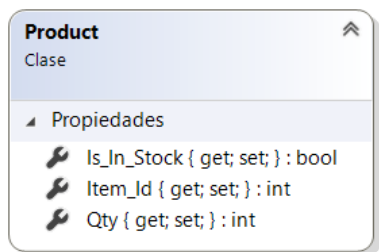
El atributo **FormaPago** se obtiene analizando si el atributo **Method** contiene las palabras “mercadopago” o “webpay”.

3.2.1.7 PaymentAdditionalInfo

PaymentAdditionalInfo	
Clase	
▲ Propiedades	
🔧	Key { get; set; } : string
🔧	Value { get; set; } : string

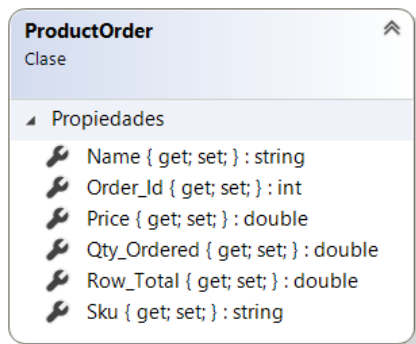
Es utilizada en la clase ExtensionAttribute para obtener más información respecto al pago

3.2.1.8 Product



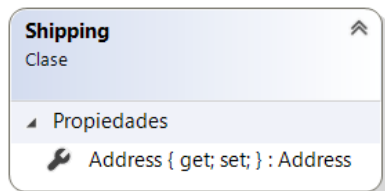
Representa un producto.

3.2.1.9 ProductOrder



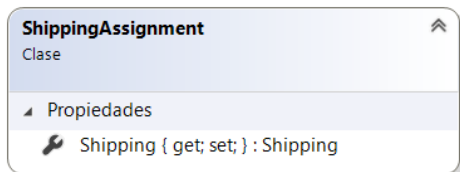
Entrega información de la asociación entre un producto y una orden.

3.2.1.10 Shipping



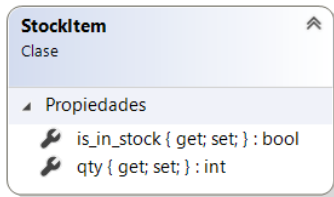
Se utiliza para entregar información respecto al despacho de una orden.

3.2.1.11 ShippingAssignment



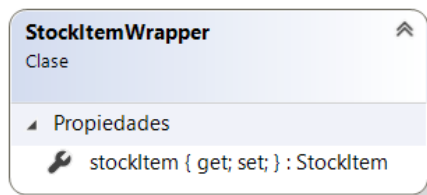
Se utiliza para crear la signación de un despacho a una orden

3.2.1.12 StockItem



Se utiliza para modificar el stock de un producto

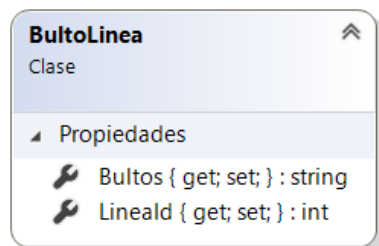
3.2.1.13 StockItemWrapper



Se utiliza para empaquetar y enviar una petición de modificación de stock.

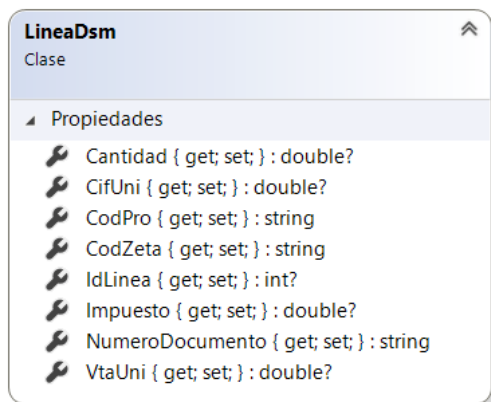
3.2.2 Modelos utilizados para empaquetado y transporte de datos

3.2.2.1 BultoLinea



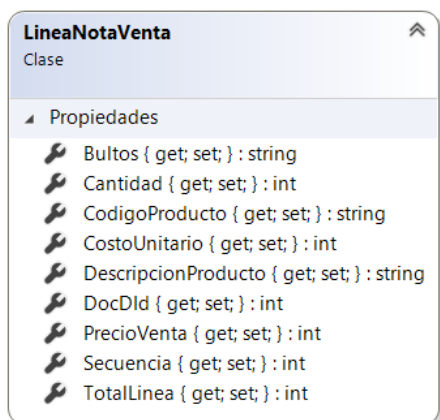
Contiene la información de bultos para una línea de nota de venta

3.2.2.2 LineaDsm



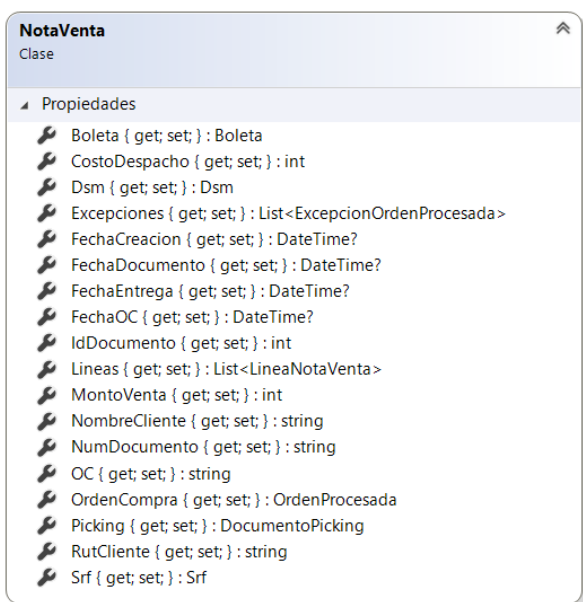
Contiene información de un registro de la tabla DOCUMENTOD relativa a la línea de detalle de un documento DSM. Es útil para realizar cálculos temporales del impuesto acumulado para un documento DSM (dependiendo de su código zeta).

3.2.2.3 LineaNotaVenta



Es usada para insertar y recuperar información desde y hacia la tabla DOCUMENTOD, cuando se trabaja con una línea de nota de venta.

3.2.2.4 NotaVenta



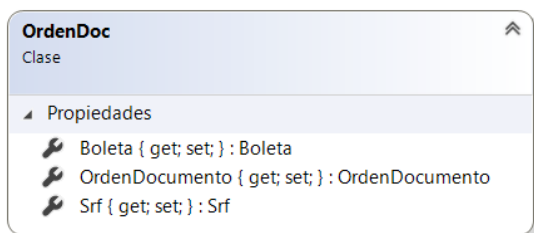
Se utiliza para insertar y recuperar información desde y hacia la tabla DOCUMENTO, cuando se trabaja con una nota de venta.

Los campos **Boleta**, **Dsm** y **Srf** son opcionales y corresponde al documento (objeto del tipo Boleta, Dsm o Srf según corresponda) referenciado si la nota de venta se procesa como boleta o SRF.

El campo **Excepciones** contiene una lista con todas las excepciones producidas al importar esta nota de venta.

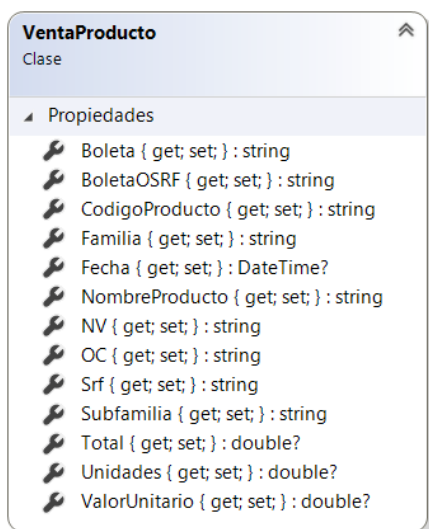
El campo **Picking** es opcional y corresponde al documento referenciado (objeto del tipo DocumentoPicking) si se ha hecho Picking a la nota de venta.

3.2.2.5 OrdenDoc



Se utiliza para listar los documentos asociados a una nota de venta (una boleta o un SRF según sea el caso) en la grilla de notas de venta.

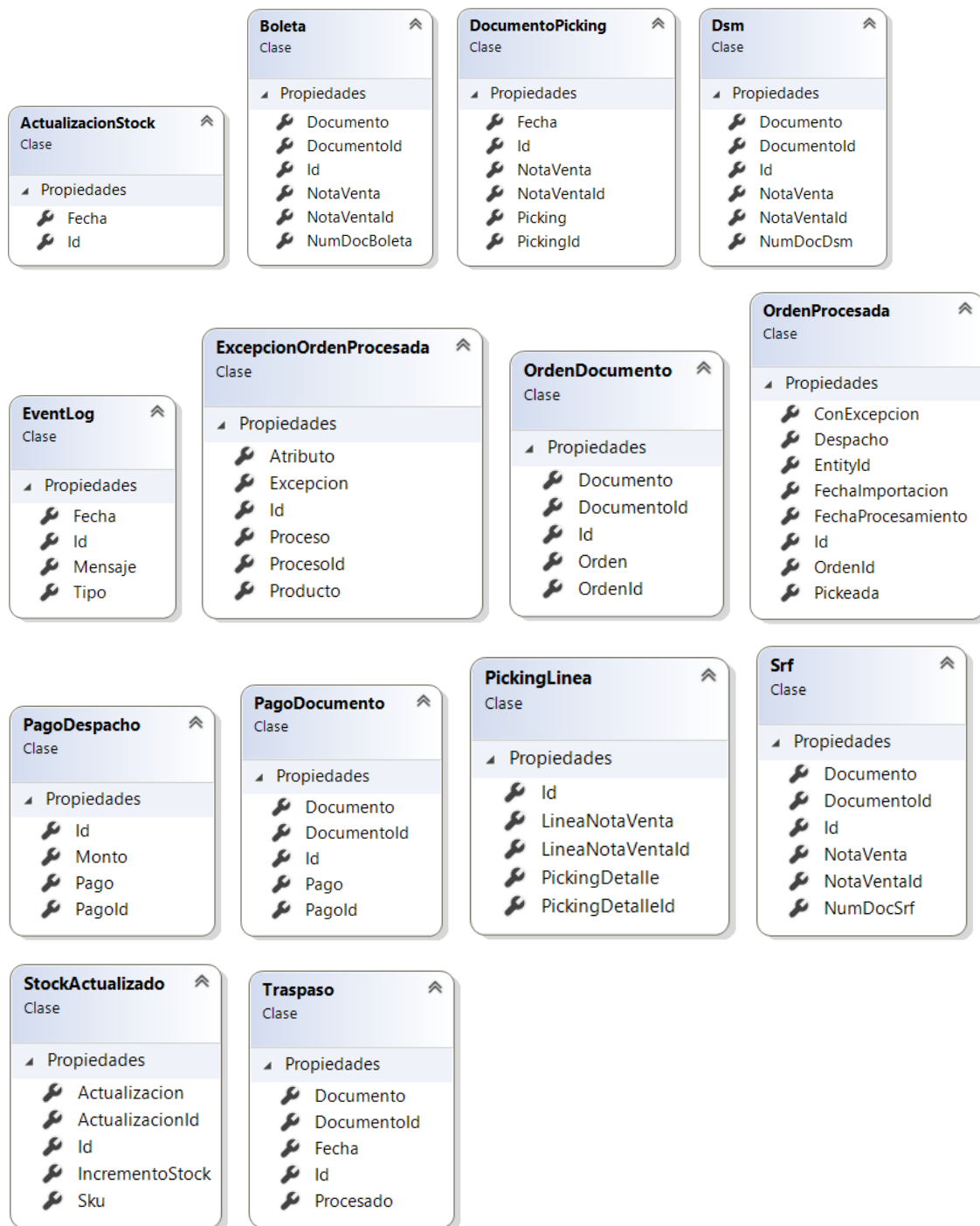
3.2.2.6 VentaProducto



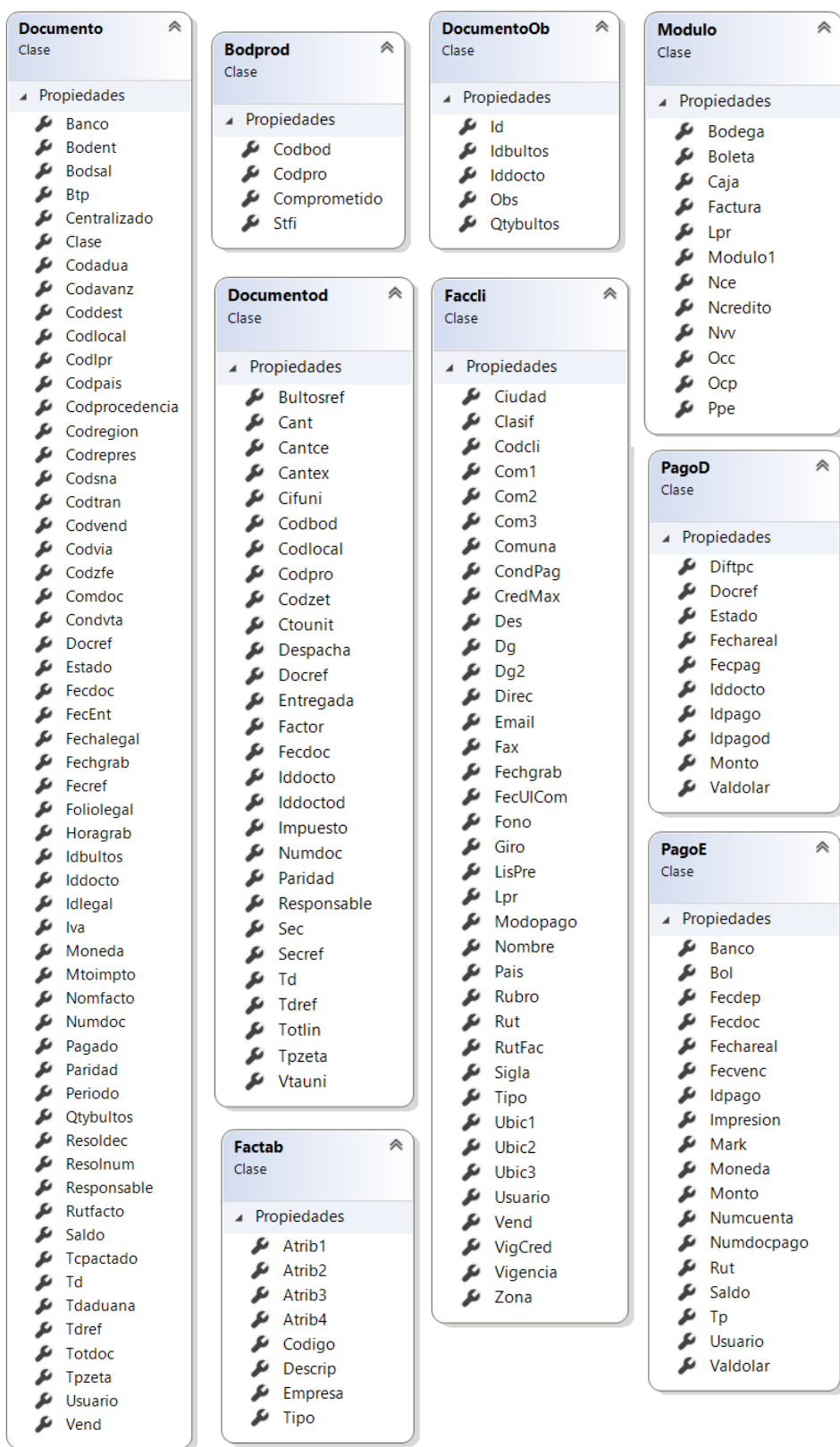
Se utiliza para mostrar información en el reporte Ventas por Producto

3.2.3 Modelos utilizados como Entidades de la Base de Datos BDIQUIQUE

3.2.3.1 Entidades nuevas agregadas a BDIQUIQUE



3.2.3.2 Entidades preexistentes en BDIQUIQUE



Parametro
Clase
Propiedades
Bodarriendo
Bodcorte
Bodent
Bodprod
Desktoppersonal
Dgempresa
Direccion
Direfacto
Emisor
Fechabl
FechaTope
Id
Manifiesto
Montomaxper
Nombrequempra
Numbl
Repre
Rutempra
Ubva
Uvbv
Ubvc

Stoart
Clase
Propiedades
Bmp
Cansto
Codaran
Codint
Codpro
Com1
Compac
Compo
Composicion
Comprometido
Cuma
Cup
CupMlocal
Cuue
Descri
Estado
Familia
Feccuma
Fechgrab
Feculen
Feculsa
Impuesto
Kopr
Mermac
Moncom
Monfac
Nomzofri
Proced
PtesMes
Puv
Stfi1
Stfi11
Stfi15
Stfi2
Stomax
Stomin
Subfam
Subtipo
Tipo
Tipoprod
Ubic
Unimed
Unimed2
Uniset
Usuario
Ventaac

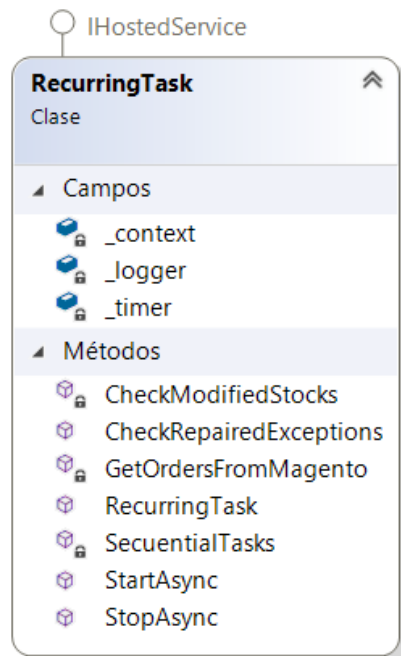
Tabparidad
Clase
Propiedades
Dia
Fecha
Moneda
Valor

Usuario
Clase
Propiedades
Cc
Inicio
Nombre
NomUsuario
Pwd

Tipodocumento
Clase
Propiedades
Clase
Codidoc
Cuenta
Extra
Factorinventario
Factormonto
Fechgrab
IdSecuencia
Impuesto
Modulo
Namedocumen...
Sistema
Tipodoc
Usuario

Zetart
Clase
Propiedades
Bodent
Cantini
Ciftot
Cifuni
Codnab
Codpro
Codume
Codzet
Descri
Docent
Empresa
Fecha
Moneda
Paridad
Salfinal
Salinic
Tipo
Umed

3.3 Clase de servicio autónomo



La clase **RecurringTask** implementa un servicio de tareas autónomas y recurrentes. Es la encargada de implementar la comunicación con la API de Magento.

El parámetro de sistema **Shared.FRECUENCIA_EN_MINUTOS** se puede utilizar para modificar la frecuencia con la que las tareas se ejecutan. Por defecto, está seteado en 60.

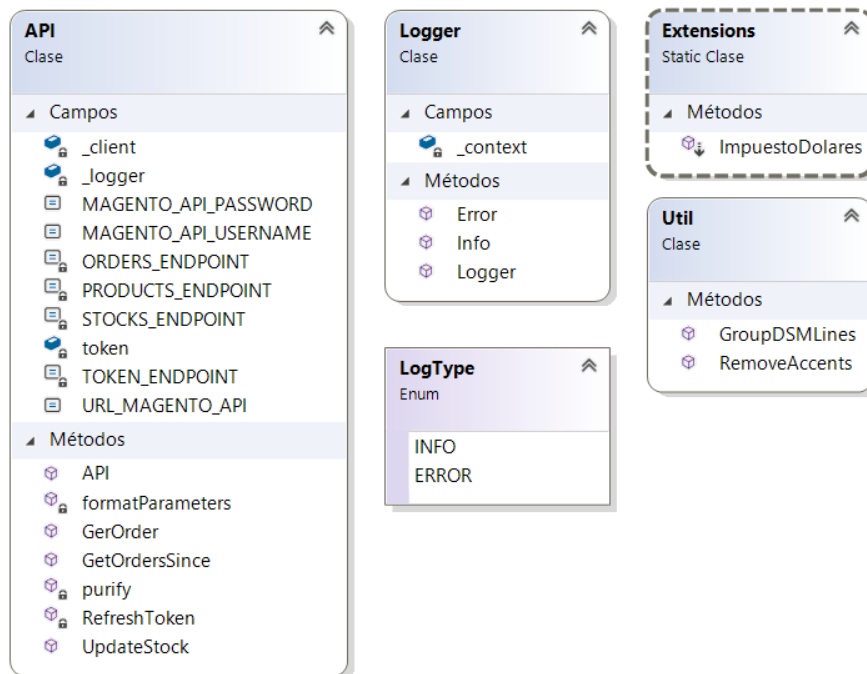
El método **CheckModifiedStocks** chequea si hay traspasos a la bodega B01 que se hayan realizado y aún no se hayan procesado. Si los hay, se ejecuta una actualización de stock para cada uno de los SKU que hayan sido afectados por el traspaso. En el caso de que la actualización sea exitosa, se crea un nuevo registro de **actualizacion_stock** en la BD.

El método **CheckRepairedExceptions** verifica si se han rectificado las excepciones producidas en las órdenes importadas. Si las correcciones efectuadas implican que la orden ya no tiene excepciones, entonces se marca la orden como “sin excepciones”, para que pueda ser procesada.

El método **GetOrdersFromMagento** obtiene un listado de órdenes confirmadas en Magento desde la fecha indicada en el parámetro **Shared.ORDERS_SINCE_X_DAYS_AGO**. Para cada una de ellas, se revisa que no haya sido procesada anteriormente y se procesa creando una nota de venta con los datos correspondientes a ella. En este proceso también se generan los pagos asociados a las notas de venta que se estén creando.

El método **SecuentialTasks** ejecuta las llamadas a los métodos `GetOrdersFromMagento`, `CheckRepairedExceptions` y `CheckModifiedStocks` de manera secuencial y síncrona.

3.4 Clases de utilidades



3.4.1 API

La clase **API** es la encargada de realizar la comunicación con la API de Magento.

El método **formatParameters** sirve para ajustar el formato de los parámetros de las peticiones a la API.

El método **GetOrder** trae desde la API la información de una orden por su ID.

El método **GetOrdersSince** trae todas las órdenes confirmadas desde la fecha indicada como argumento.

El método **purify** elimina los caracteres indeseados de las peticiones.

El método **RefreshToken** autentifica y autoriza al usuario autorizado para realizar las peticiones en la API.

El método **UpdateStock** actualiza el stock del producto identificado por el SKU entregado como primer argumento. Aumenta su stock en la cantidad entregada como segundo argumento.

3.4.2 Util

La clase **Util** contiene utilidades genéricas y transversales para el sistema.

El método **RemoveAccents** quita los acentos del string entregado como argumento.

El método **GroupDSMLines** recibe como argumento un arreglo de objetos LineaDSM y las agrupa y acumula de acuerdo con su código zeta. Es decir, las líneas que tengan el mismo código zeta se juntarán en una única línea. El impuesto y cantidad de las líneas acumuladas corresponde a la suma del total de los valores de un mismo grupo.

3.4.3 Logger

La clase **Logger** genera un historial de eventos ocurridos en las importaciones y lo registra en la base de datos.

El método **Info** registra un evento informativo.

El método **Error** registra un evento erróneo.

3.4.4 LogType

La enumeración **LogType** lista los tipos de log posibles.

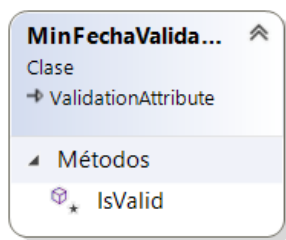
3.4.5 Extensions

La clase estática **Extensions** sirve para extender entidades del sistema, agregándoles métodos o propiedades como si fueran nativas.

El método **ImpuestoDolares** extiende la entidad **Zetart**. Sirve para calcular el impuesto en dólares de un registro específico de la tabla **ZETART**.

3.5 Clases de validación

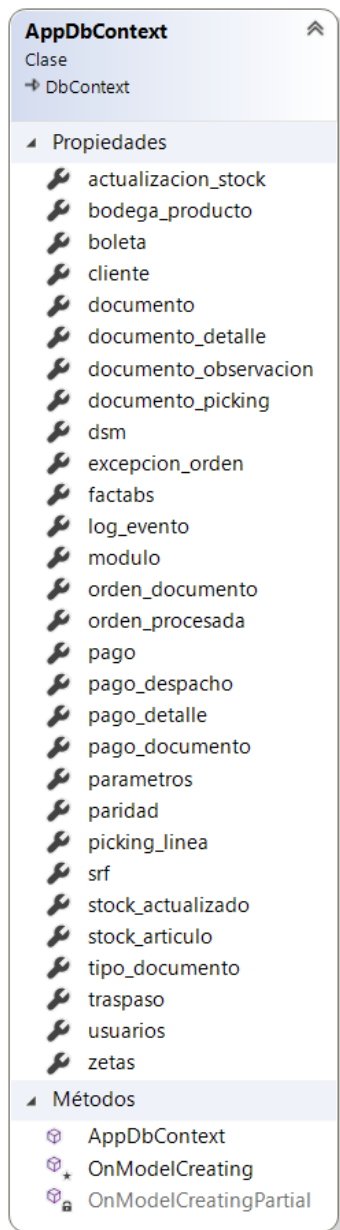
Las clases de validación dispones atributos de decoración a las entidades para validación compleja en formularios.



3.5.1 MinFechaValidator

La clase **MinFechaValidator** decora un atributo de tipo DateTime, para validar que sea mayor al 1 de enero de 1900.

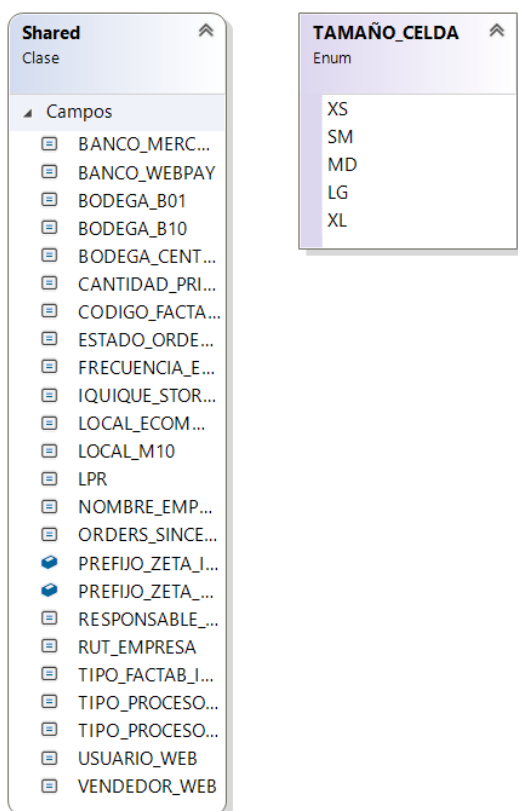
3.6 Clase de contexto de datos



3.6.1 AppDbContext

La clase **AppDbContext** corresponde al contexto de datos (que contiene el modelo de objetos mapeado a las tablas de la base de datos) de la aplicación.

3.7 Clases de parámetros



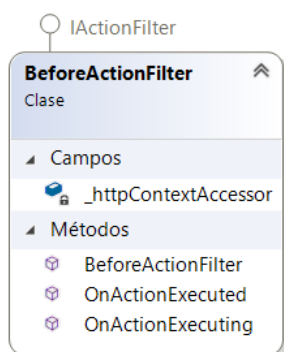
3.7.1 Shared

La clase **Shared** contiene constantes que juegan el rol de parámetros de configuración del sistema.

3.7.2 TAMAÑO_CELDA

La enumeración **TAMAÑO_CELDA** lista los posibles tamaños que tienen las celdas de las grillas.

3.8 Clases de Filtros



3.8.1 BeforeActionFilter

La clase **BeforeActionFilter** es un filtro que se ejecuta antes de cualquier llamada a controlador/acción. Se utiliza para validar que el usuario esté correctamente autenticado en el sistema para poder operar sobre él.

4. Diagrama de Contexto

4.1 Diagrama de interconexión de arquitectura

