

Sistema de Administración de Farmacias

Modelo de Implementación

Versión 1.0

Historia de revisiones

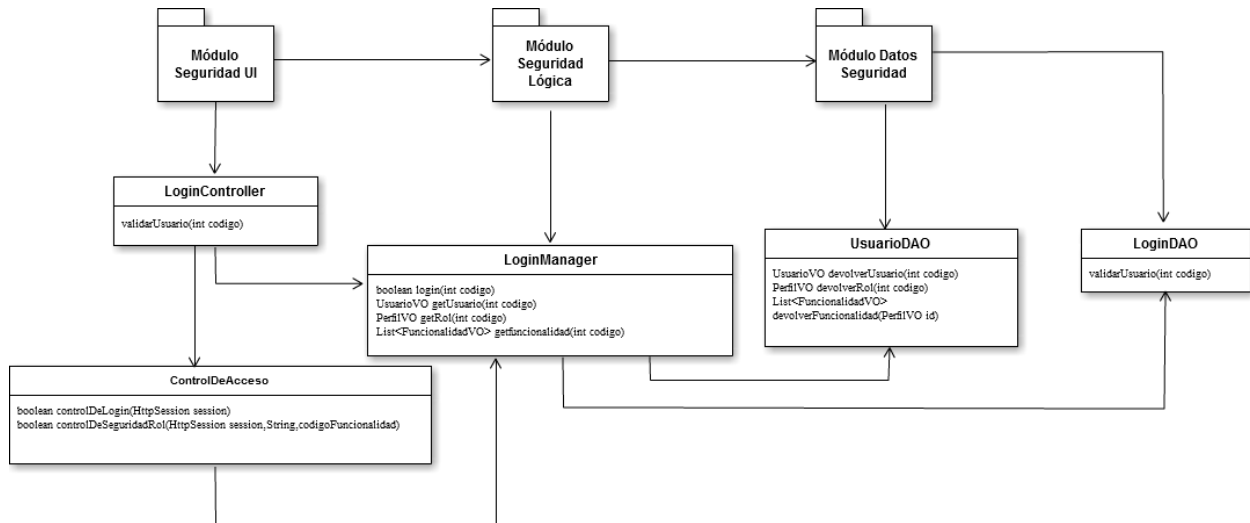
Fecha	Versión	Descripción	Autor
24/09/2014	1.0	Comienzo del documento.	Implementadores
27/09/2014	1.0	Realización de documento	Implementadores
28/09/2014	1.0	Revisión.	SQA

Contenido:

1. Trazabilidad desde el Modelo de Diseño	3
1.1 Caso de Uso Iniciar Sesión.....	3
1.2 Caso de Uso Buscar Producto	4
1.3 Caso de Uso Ver Información Producto	5
1.4 Caso de Uso Actualizar Precios y Stock	5
2. Subsistemas	8
2.1 [Subsistema 1].....	8
2.2 [Subsistema 2].....	8
3. Componentes	8
3.1. [Componente 1].....	8
3.2. [Componente 2].....	8
4. Interfases	9
4.1 [Interfase 1]	9
4.2 [Interfase 2]	9

1. Trazabilidad desde el Modelo de Diseño

1.1 Caso de Uso Iniciar Sesión



Controladores:

- LoginController
- ControlDeAcceso

Manejadores:

- LoginManager

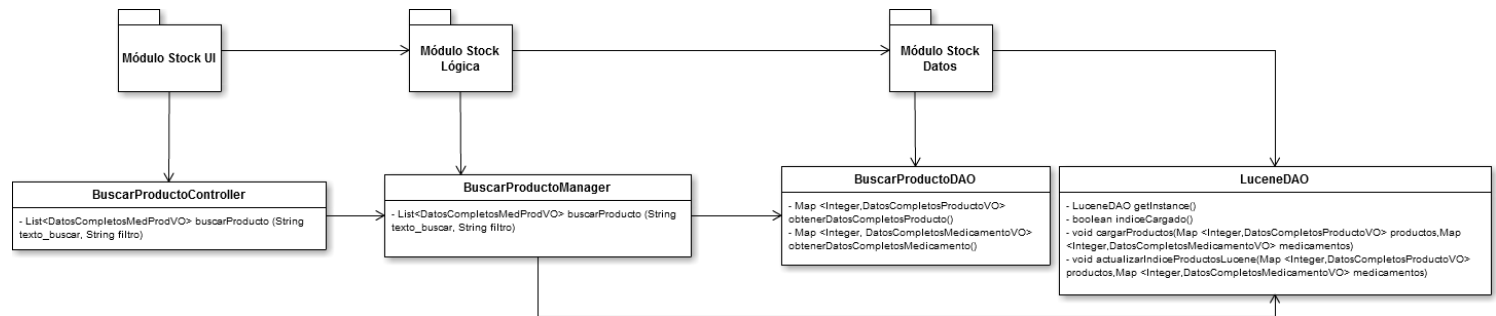
DAOs (Data Access Object):

- LoginDAO: Se encarga de validar si el usuario está registrado en el sistema.
- UsuarioDAO: Se encarga de las operaciones relacionadas a perfil del usuario y a las funcionalidades que puede acceder de acuerdo a dicho perfil

VOs (Objetos de Valor):

- UsuarioVO: contiene los datos del usuario.
Los datos son:
 - idUsuario: id del Usuario
 - nombre: nombre del Usuario
 - codigo: codigo asociado al Usuario
- PerfilVO: contiene los datos del perfil
 - idPerfil: id del perfil
 - descripcion: descripción del perfil
- FuncionalidadVO: contiene los datos de la funcionalidad.
 - idFuncionalidad: id de la funcionalidad
 - descripcion: descripción de la funcionalidad
 - codigo: descripción mas breve de la funcionalidad

1.2 Caso de Uso Buscar Producto



Controladores:

- BuscarProductoController

Manejadores:

- BuscarProductoManager

DAOs (Data Access Object):

- BuscarProductoDAO: Se encarga de las operaciones necesarias para la búsqueda del producto.
- Lucene DAO: Se encarga de las operaciones de indexación y búsqueda utilizando Lucene.

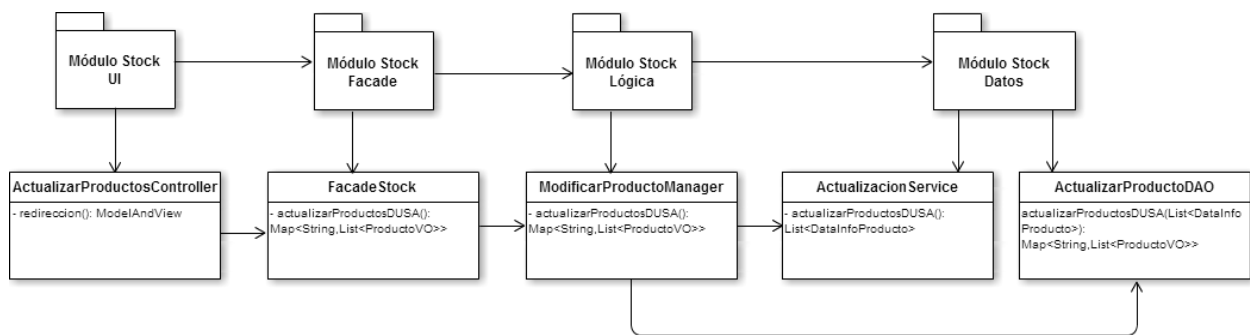
VOs (Objetos de Valor):

- DatosCompletoProducto: contiene todos los datos relevantes del producto que se desean obtener por medio de la búsqueda. Los datos son:
 - idProducto: id del Producto.
 - descripcion: descripción del Producto.
 - precioCompra: precio de compra del Producto.
 - priprecioVenta: precio de venta del Producto.
 - habilitado: representa si el producto está habilitado para la venta.
 - cantidadStock: cantidad de producto disponible en el stock.
 - vencimientoStock: fechas de vencimiento del producto en cuestión en el stock.
 - codigoBarras: códigos de barra del producto.
 - tipoIVA: tipo de IVA aplicado al producto.
 - proveedor: nombre del proveedor.
 - descuentos: lista de descuentos.
 - descripcionDescuento: lista de descripción de descuento.
- DatosCompletoMedicamento: contiene todos los datos relevantes de los productos que son además medicamentos. Los datos son:

- idProducto: id del producto.
- requiereReceta: indica si el medicamento requiere de presentación de receta.
- nombreLaboratorio: nombre del laboratorio.
- nombreDroga: drogas que componen el producto.
- accionTerapeutica: acciones terapéuticas del producto.
- presentacion: presentaciones del producto.
- DatosCompletosMedProdVO: merge necesaria de los últimos dos VO.

1.3 Caso de Uso Ver Información Producto

1.4 Caso de Uso Actualizar Precios y Stock



Controladores:

- **ActualizarProductosController:** La función redirección es llamada cuando se desea actualizar los precios y el stock y primero le pide a la Fachada que actualice y éste le devuelve las listas con los productos actualizados/agregados y luego se crea la vista actualizarProductosDUSA pasándole como parámetros dichas listas.

Fachada:

- **FacadeStock:** La función actualizarProductosDUSA simplemente le pide a la capa de lógica que actualice y mande las listas de lo cambiado y éste se las manda al Controlador.

Manejadores:

- **ModificarProductosManager:** La función actualizarProductosDUSA primero le pide al ActualizacionService que le mande la lista de DUSA con los productos a Actualizar y luego llama a la función del ActualizarProductoDAO

pasándole dicha lista como argumento. El DAO le devuelve las listas de lo cambiado y éste se las manda a la Fachada.

DAOs (Data Access Object):

- **ActualizacionService:** La función actualizarProductosDUSA accede al Web Service proporcionado por DUSA para obtener la lista de los productos a actualizar y luego se la manda al Manager.
- **ActualizarProductoDAO:** Finalmente el DAO es el encargado de recorrer la lista proporcionada por DUSA . Si ya existe el producto actualiza los datos y sino lo agrega a la base de datos. Luego devuelve un Map con 5 Listas:
 1. Productos Agregados (nuevos).
 2. Productos que subieron de precio.
 3. Productos que bajaron de precio.
 4. Productos que se habilitaron.
 5. Productos que se deshabilitaron.

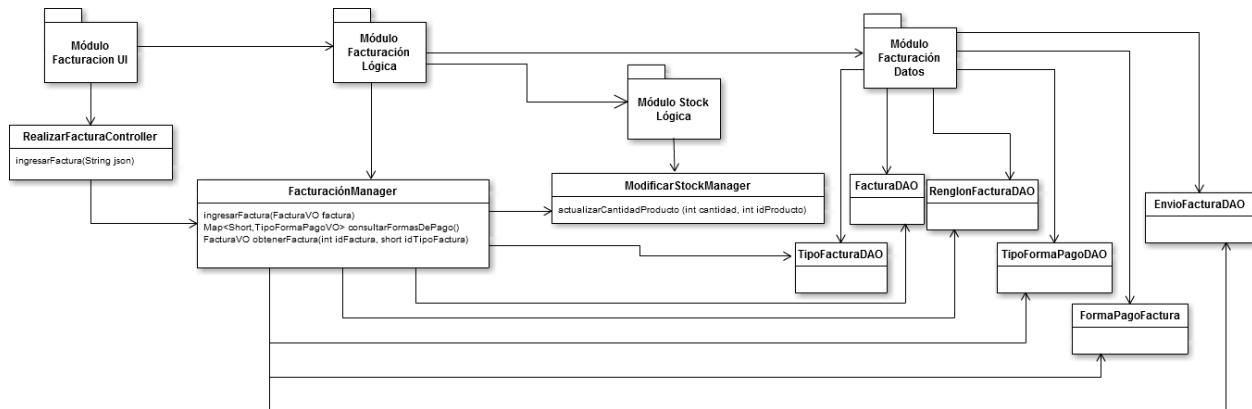
VOs (Objetos de Valor):

- **DataInfoProductoVO:** Creado provisoriamente mientras los servidores de DUSA no funcionan (que contiene lo mismo que el DataInfoProducto proporsionado por DUSA).
 - int numeroArticulo: id del producto.
 - String idLaboratorio: id del laboratorio.
 - String idLineaLaboratorio: id de la línea del laboratorio.
 - String clave1
 - String clave2
 - String clave3
 - String descripcion: descripción del producto.
 - String tipoIVA: descripción del tipo de IVA.
 - BigDecimal precioVenta: precio al cuál DUSA le vende a la farmacia (corresponde con precioCompra del producto en nuestra base de datos)
 - BigDecimal precioOferta
 - BigDecimal precioPublico: precio al cuál hay que vender el producto a los clientes (corresponde con precioVenta del producto en nuestra base de datos)
 - int habilitado: representa si el producto está habilitado para la venta.
 - BigDecimal porcentajeDescuentoOferta
 - boolean hasCodigoBarra: representa si tiene código de barra.
 - String codigoBarra: código de barra del producto.
 - String codigoLaboratorio: código del laboratorio.
 - XMLGregorianCalendar fechaUltimoPrecio
 - XMLGregorianCalendar fechaUltimaActualizacion

- int idProductoNoritel
- int idPresentacionNoritel
- **ProductoVO:** Contiene la información de un producto:
 - int idProducto
 - String descripcion
 - double precioCompra
 - double precioVenta
 - int idTipoIva
 - boolean habilitado
 - long idTransaccion

1.5 Caso de Uso Ajustar Stock

1.6 Caso de Uso Realizar Factura



Controladores:

- RealizarFacturaController

Manejadores:

- FacturacionManager
- ModificarStockManager

DAOs (Data Access Object):

- FacturaDAO
- RenglonFacturaDAO
- TipoFacturaDAO
- TipoFormaPagoDAO
- FormaPagoFacturaDAO
- EnvioFacturaDAO
- StockDAO

VOs (Objetos de Valor):

- RenglonFacturaVO
- FacturaVO
- TipoFacturaVO
- TipoFormaPagoVO
- FormaPagoFacturaVO
- EnvioFacturaVO

2. Subsistemas

2.1 [Subsistema 1]

[Se indica para el subsistema 1 la correspondencia con el Modelo de Diseño, una descripción del mismo y de los subsistemas y componentes que lo integran.]

2.2 [Subsistema 2]

...

3. Componentes

3.1. [Componente 1]

[Se indica la correspondencia con el Modelo de Diseño y se especifica como están organizados los componentes de acuerdo a la estructura y mecanismos de modularización disponibles en el ambiente de implementación y en el(los) lenguaje(s) de programación, y cuales son las dependencias entre los distintos componentes. Pueden existir dependencias de compilación entre componentes, denotando cuales componentes son requeridos para la compilación de un componente específico.]

3.2. [Componente 2]

...

4. Interfases

4.1 [Interfase 1]

[Se detalla la interfase y se traza la correspondencia con el Modelo de Diseño]

4.2 [Interfase 2]

...