Pedidos Online - DUSA
Modelo de Diseño
Versión 4.1

Historia de revisiones

Fecha	Versión	Descripción	Autor
13/09/2013	1.0	Versión Inicial	Joana Miller
15/09/2013	1.1	Modificaciones Generales, agregado DSS de Analistas y Paquetes	Analistas y Joana Miller
15/09/2013	1.2	Revisión Responsable SQA	Verónica Gamarra
29/09/2013	2.0	Actualización Diseño DSS, agregación de DCD, DC.	Joana Miller
30/09/2013	2.1	Revisión Responsable SQA	Verónica Gamarra
20/10/2013	3.0	Agregado CU de Iteración 2 de Fase elaboración	Joana Miller
21/09/2013	3.1	Revisión Responsable SQA	Verónica Gamarra
3/11/2013	4.0	Agregado CU VerCarrito, Enviar comprobante, restablecer contraseña. Actualizacion DCD Logica, DCD WEB, manejo de excepciones.	Joana Miller
3/09/2013	4.1	Revisión Responsable SQA	Verónica Gamarra

Índice

1. Introducción 5
   1.1. Propósito 5
   1.2. Alcance 5
   1.3. Definiciones, siglas y abreviaturas. 5
   1.4. Referencias 5
2. Diseño de Casos de Uso 6
   2.1. Diseño del Caso de Uso - Iniciar Sesión 6
      2.1.1. Diagrama de paquetes 6
      2.1.2. Diagrama de Interacción 7
      2.1.3. Diseño de Flujo de eventos 7
      2.1.4. Requerimientos especiales o de implementación 7
      2.1.5. Diagrama de comunicación 7
   2.2. Diseño del Caso de Uso - Finalizar Sesión 9
      2.2.1. Diagrama de paquetes 9
      2.2.2. Diagrama de Interacción 9
      2.2.3. Diseño de Flujo de eventos 9
      2.2.4. Requerimientos especiales o de implementación 9
      2.2.5. Diagrama de comunicación 10
   2.3. Diseño del Caso de Uso - Buscar Producto 10
      2.3.1. Diagrama de paquetes 10
      2.3.2. Diagrama de Interacción 11
      2.3.3. Diseño de Flujo de eventos 11
      2.3.4. Requerimientos especiales o de implementación 11
      2.3.5. Diagrama comunicación 12
   2.4. Diseño del Caso de Uso - Agregar Producto a Pedido 12
      2.4.1. Diagrama de paquetes 12
      2.4.2. Diagrama de Interacción 13
      2.4.3. Diseño de Flujo de eventos 13
      2.4.4. Requerimientos especiales o de implementación 13
      2.4.5. Diagrama de comunicación 14
   2.5. Diseño del Caso de Uso - Asignar Reparto 15
      2.5.1. Diagrama de paquetes 15
      2.5.2. Diagrama de Interacción 15
      2.5.3. Diseño de Flujo de eventos 15
      2.5.4. Requerimientos especiales o de implementación 15
      2.5.5. Diagrama de comunicación 16
   2.6. Diseño del Caso de Uso - Asignar Forma Pago 16
      2.6.1. Diagrama de paquetes 16
      2.6.2. Diagrama de Interacción 17
      2.6.3. Diseño de Flujo de eventos 17
      2.6.4. Requerimientos especiales o de implementación 17
      2.6.5. Diagrama de comunicación 17
   2.7. Diseño del Caso de Uso - Confirmar Pedido 18
      2.7.1. Diagrama de paquetes 18
      2.7.2. Diagrama de Interacción 19
      2.7.3. Diseño de Flujo de eventos 19

2.7.4. Requerimientos especiales o de implementación 19
2.7.5. Diagrama de comunicación 20
2.8. Diseño del Caso de Uso - Cancelar Pedido 20
2.8.1. Diagrama de paquetes 20
2.8.2. Diagrama de Interacción 21
2.8.3. Diseño de Flujo de eventos 21
2.8.4. Requerimientos especiales o de implementación 21
2.8.5. Diagrama de comunicación 21
2.9. Diseño del Caso de Uso - Modificar Pedido 22
2.9.1. Diagrama de paquetes 22
2.9.2. Diagrama de Interacción 22
2.9.3. Diseño de Flujo de eventos 23
2.9.4. Requerimientos especiales o de implementación 23
2.9.5. Diagrama de comunicación 23
2.10. Diseño del Caso de Uso - Ver Detalle Producto 24
Este caso de uso particular, se diseña dentro del buscar producto, es decir al desplegar el tootltip con la información de la oferta del producto seleccionado se agrega la infromación restante a desplegar, como ser: 24
2.11 Diseño del Caso de Uso - Quitar Producto a Pedido 24
2.11.1. Diagrama de Paquetes 24
2.11.2. Diagrama de Interacción 24
2.11.3. Diseño de Flujo de eventos 24
2.11.4. Requerimientos especiales o de implementación 24
2.11.5. Diagrama de comunicación 25
2.12. Diseño del Caso de Uso - Ver Carrito 25
2.12.1. Diagrama de Paquetes 25
2.12.2. Diagrama de Interacción 25
2.12.3. Diseño de Flujo de eventos 26
2.12.4. Requerimientos especiales o de implementación 26
2.12.5. Diagrama de comunicación 26
2.13. Diseño del Caso de Uso - Enviar comprobante pedido via e-mail 26
2.13.1. Diagrama de Paquetes 26
2.13.2. Diagrama de Interacción 27
2.13.3. Diseño de Flujo de eventos 27
2.13.4. Requerimientos especiales o de implementación 27
2.13.5. Diagrama de comunicación 27
2.14. Diseño del Caso de Uso - Restablecer contraseña 28
2.14.1. Diagrama de Paquetes 28
2.14.2. Diagrama de Interacción 28
2.14.3. Diseño de Flujo de eventos 28
2.14.4. Requerimientos especiales o de implementación 28
2.14.5. Diagrama de comunicación 28
3. Diseño de Objetos 29
4. Diseño de Subsistemas 29
4.1. LogicServer 29
4.1.1 Diseño de Objetos 29
4.2. WEB-APP 31
4.2.1 Diseño de Objetos 31
5. Diseño Interfaz de Usuario 34

5.1. Framework Backbone.....34
5.2. Formatos individuales de interfaz de pantalla.....34
5.3. Controles y elementos de diseño de interfaz de pantalla.....34
5.4. Forma de navegación de interfaz de pantalla 34
5.5. Formatos de impresión.....34
6. Diseño de Datos 34
7. Manejo de Excepciones.....34
8. Diseño Configuraciones para Administrador Sistema.....36
9. Solución Problemas Concurrentes Acceso a Base de Datos 36

1. Introducción

En este documento se presenta el Modelo de Diseño del sistema, que servirá como apoyo a la descripción de la Arquitectura del sistema.

1.1. Propósito

Este documento tiene como finalidad presentar el diseño del Sistema, los subsistemas, clases, interfaces que lo componen y la forma de interacción entre los mismos. Presenta la estructura del Sistema en base al Modelo de Dominio, extendiéndolo para contemplar los casos de uso identificados en el Modelo de Casos de Uso.

1.2. Alcance

En este documento se construye parte del diseño del sistema, por lo que el documento de Descripción de Arquitectura se verá afectado ante alguna modificación de este documento. Se pretende lograr a lo largo de la Fase de Elaboración una refinación tal del diseño que permita a los implementadores desarrollar los casos de uso con consistencia y siguiendo las buenas prácticas de programación.

1.3. Definiciones, siglas y abreviaturas.

Se recomienda recurrir al Glosario del Proyecto, de toda forma la siguiente tabla describe los términos propios de este documento.

Concepto	Descripción
aplicación WEB	Referencia al front end del Sistema
LogicServer	Referencia al Servidor de Lógica de Negocio

1.4. Referencias

En la siguiente tabla se listan nombre y descripción de los documentos que son referenciados en este documento

Nombre	Descripción
DSARQG03v3.1.pdf	Descripción de Arquitectura
RQMODG03v6.8	Modelo de Casos de Uso
RQOOMDOG03v2.2	Modelo de Dominio
DPPDEG03v5.1.pdf	Plan de Desarrollo
DSOOMDAG03v4.1.pdf	Modelo de Datos

2. Diseño de Casos de Uso

En esta sección se presenta el diseño de los casos de uso que según el Plan de Desarrollo se han implementado hasta Iteración 1 de Fase Construcción.

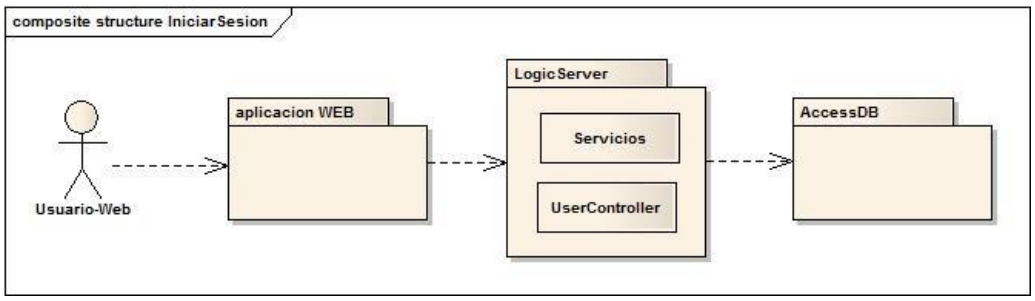
En las secciones siguientes se presenta diseño para los siguientes casos de uso:

ID	Nombre CU
01	Iniciar sesión
02	Finalizar sesión
14	Buscar producto
07	Iniciar pedido
11	Agregar producto a pedido
20	Asignar Reparto
24	Asignar Forma de Pago
10	Confirmar Pedido
09	Cancelar Pedido
08	Modificar Pedido
13	Ver Detalle de Producto
12	Quitar producto a pedido
25	Ver Carrito
26	Enviar comprobante pedido via e-mail
06	Restablecer contraseña

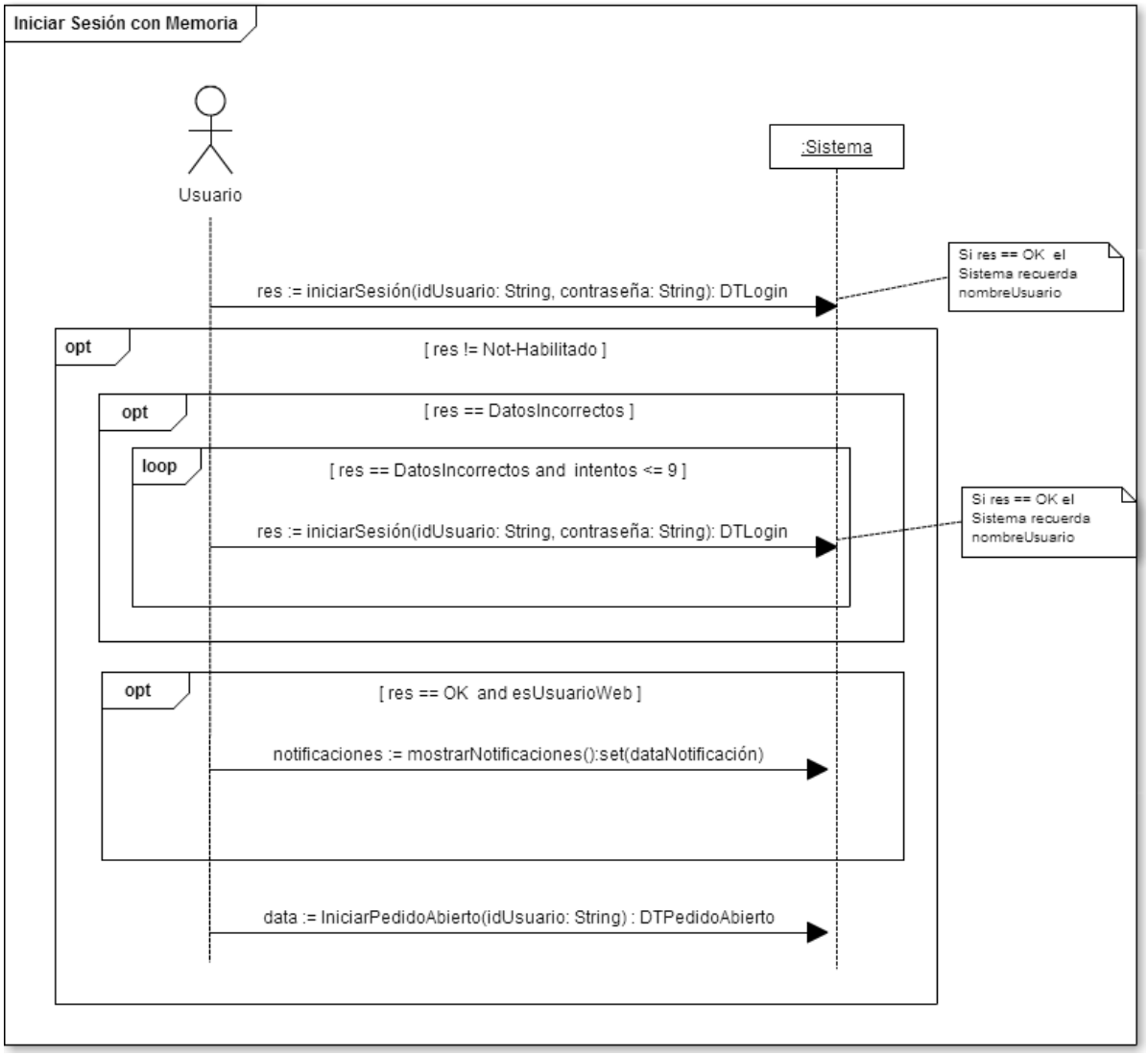
Para cada caso de uso se presenta el diagrama de paquetes, para conocer los componentes de software que estarán participando de la solución, el diagrama de interacción (DSS), diagrama de comunicación (DC - de las operaciones del DSS), diagrama de flujo (si corresponde) y restrincciones especiales (si aplica alguna)

2.1. Diseño del Caso de Uso - Iniciar Sesión

2.1.1. Diagrama de paquetes



2.1.2. Diagrama de Interacción



2.1.3. Diseño de Flujo de eventos

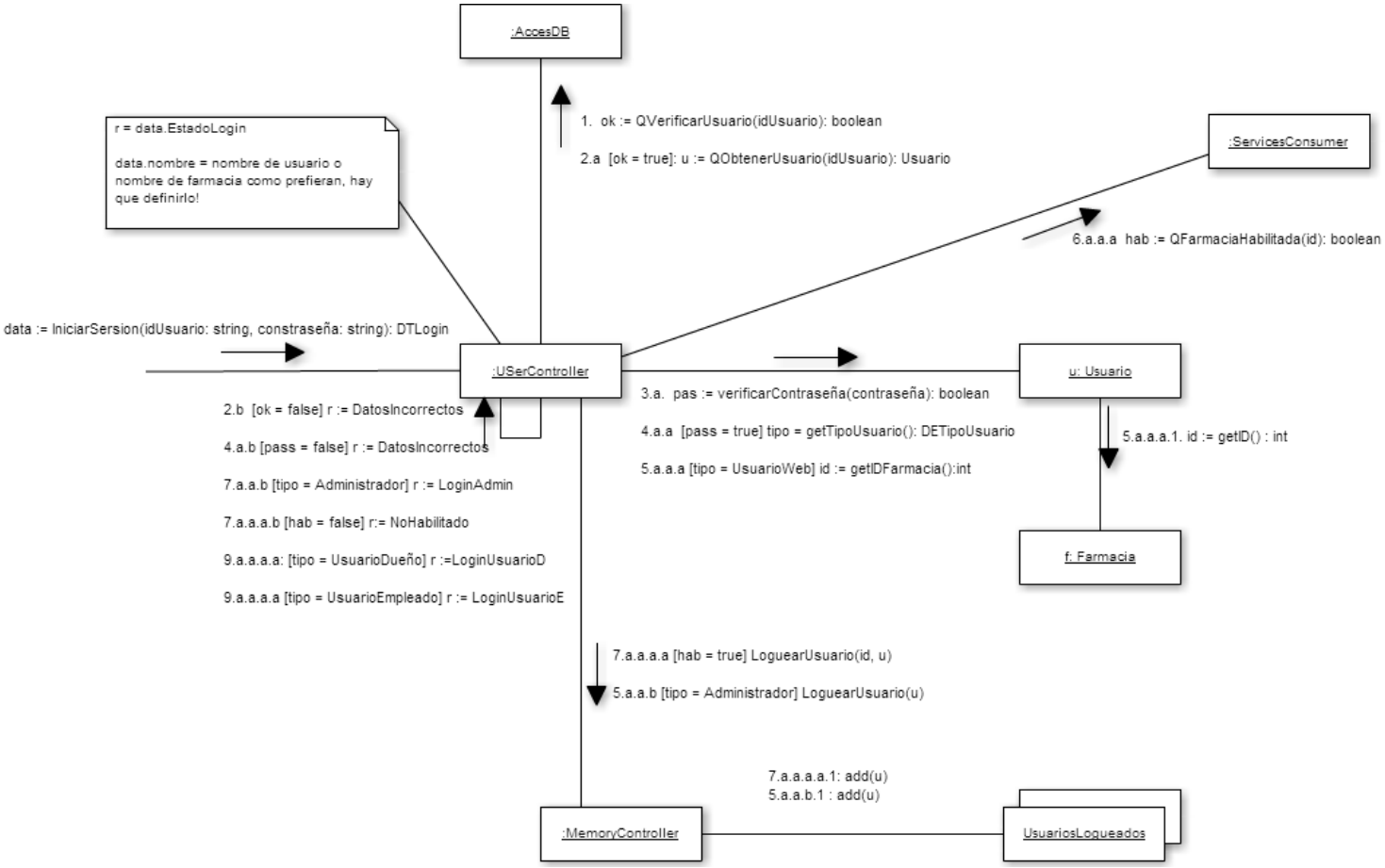
No aplica a esta entrega.

2.1.4. Requerimientos especiales o de implementación

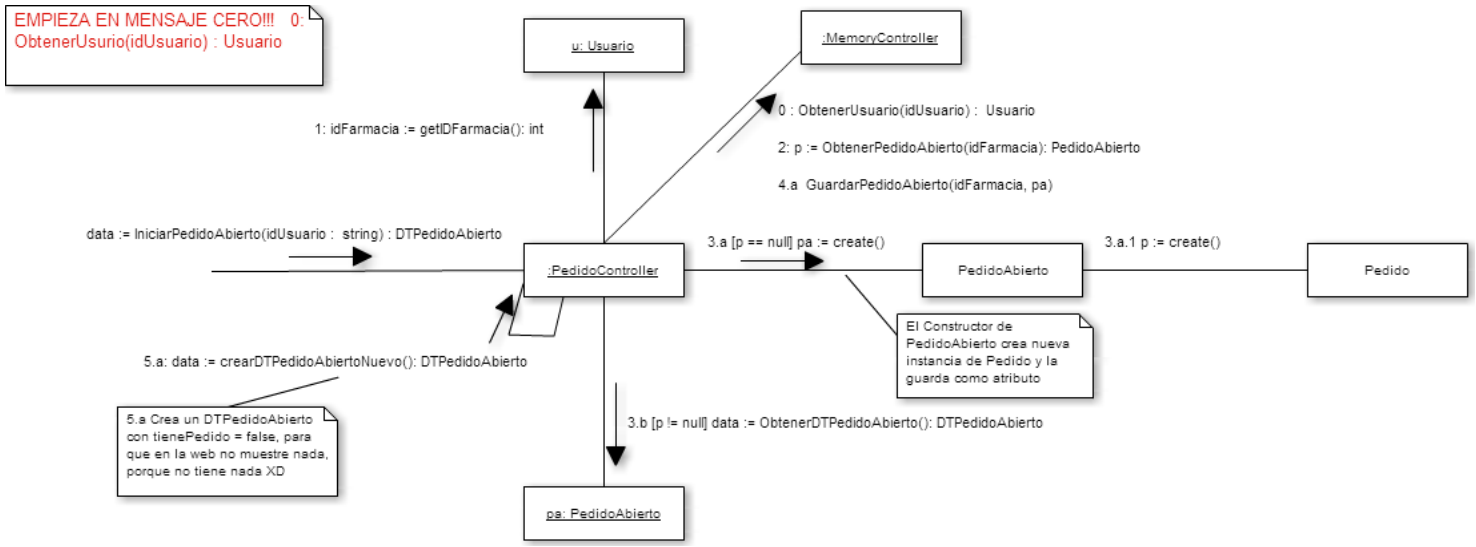
No hay.

2.1.5. Diagrama de comunicación

Operación Iniciar Sesión

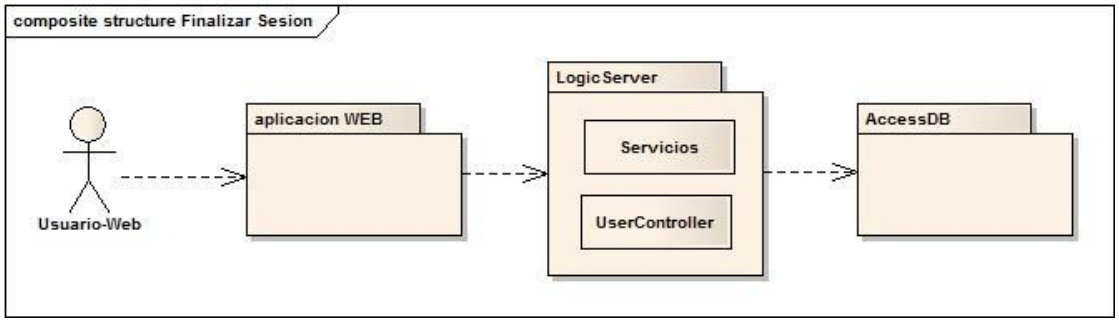


Operación Iniciar Pedido Abierto

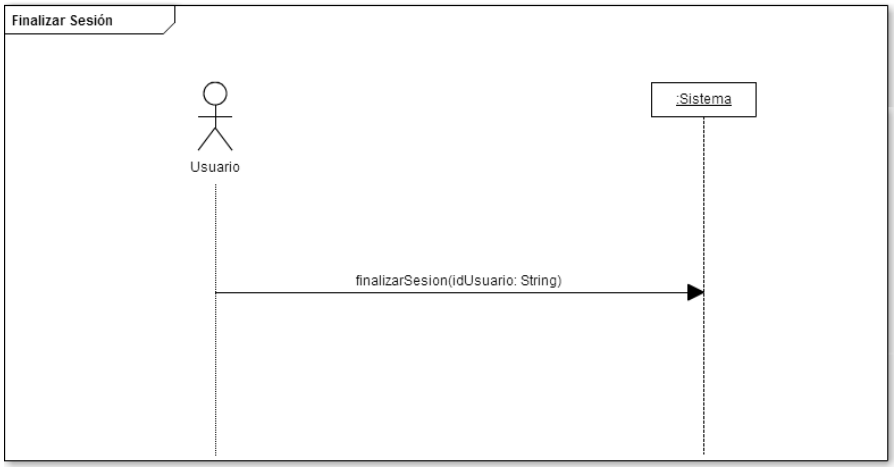


2.2. Diseño del Caso de Uso - Finalizar Sesión

2.2.1. Diagrama de paquetes



2.2.2. Diagrama de Interacción



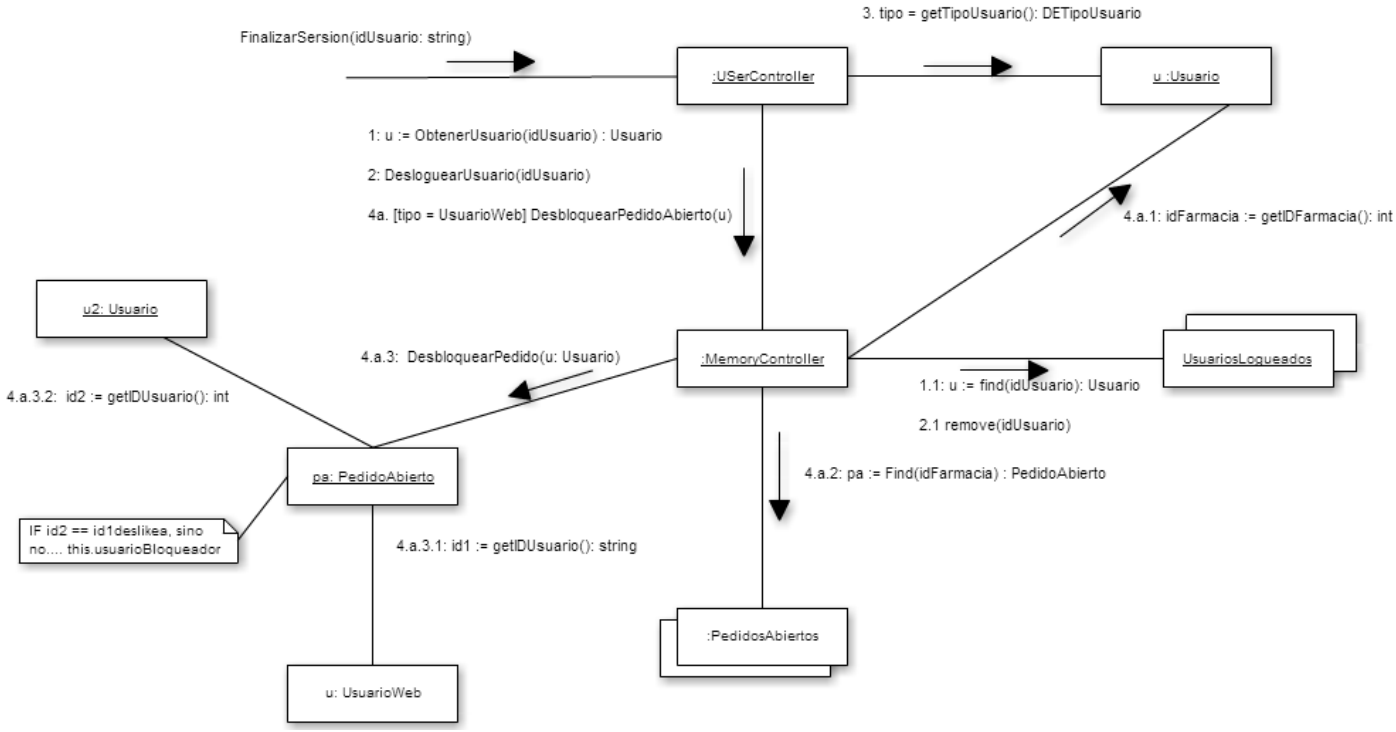
2.2.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.2.4. Requerimientos especiales o de implementación

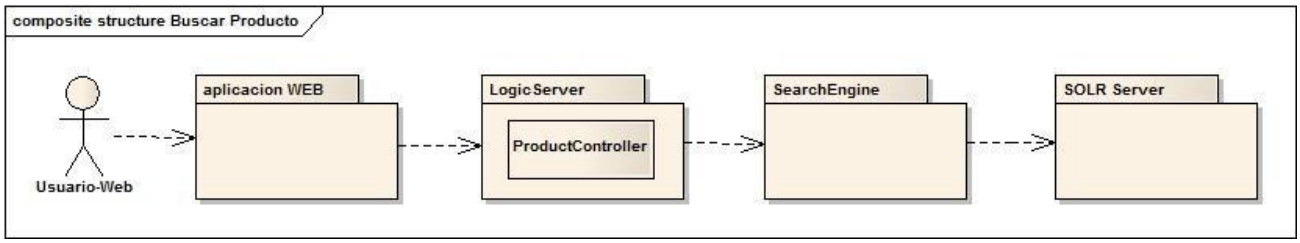
No hay.

2.2.5. Diagrama de comunicación

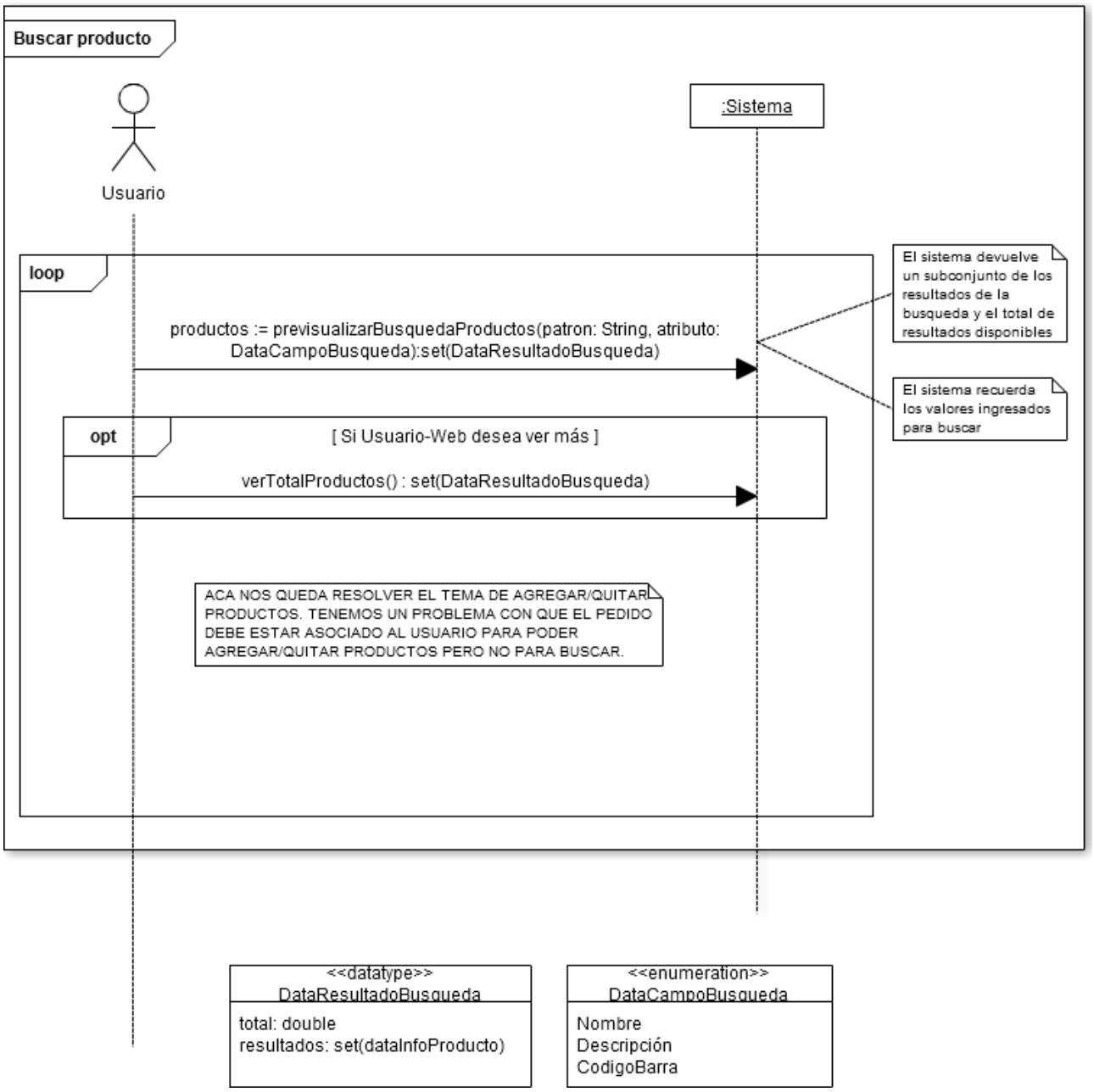


2.3. Diseño del Caso de Uso - Buscar Producto

2.3.1. Diagrama de paquetes



2.3.2. Diagrama de Interacción



2.3.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.3.4. Requerimientos especiales o de implementación

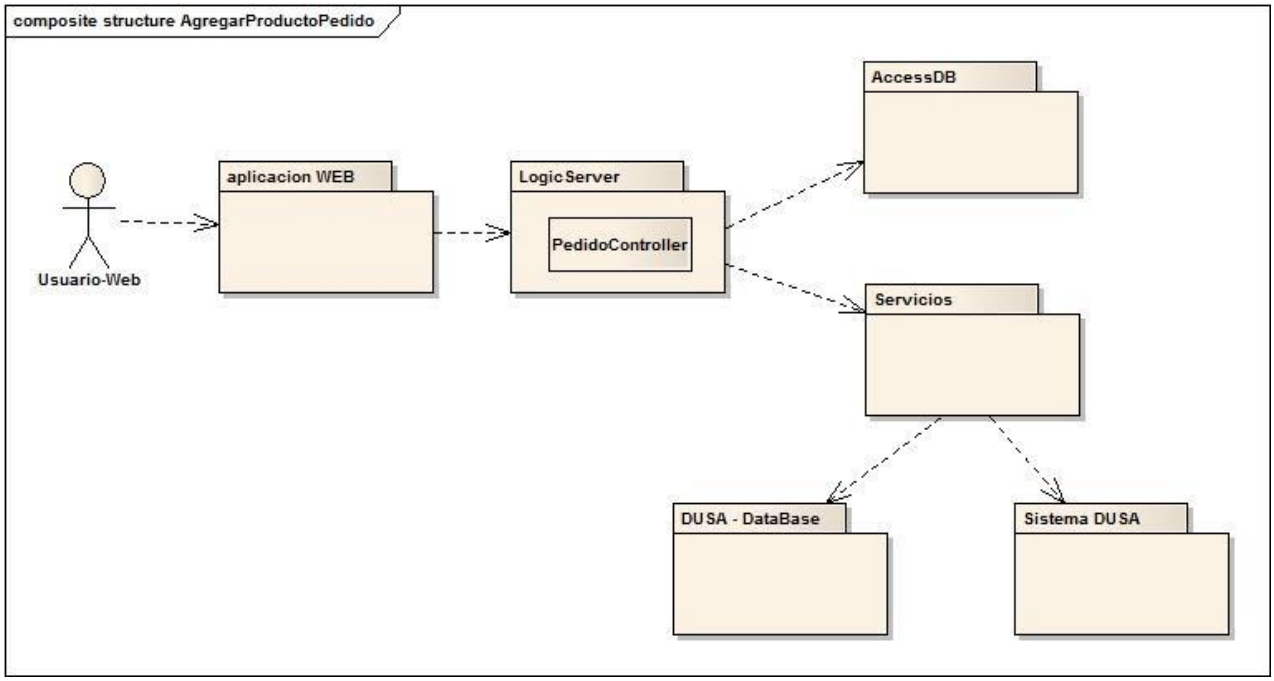
El cliente insistió en la performance que debe tener la búsqueda de un producto, ser ágil, dinámica. Condiciona una de las formas de implementación a evitar son las conexiones a base de datos innecesarias, u optimizarlas.

2.3.5. Diagrama comunicación

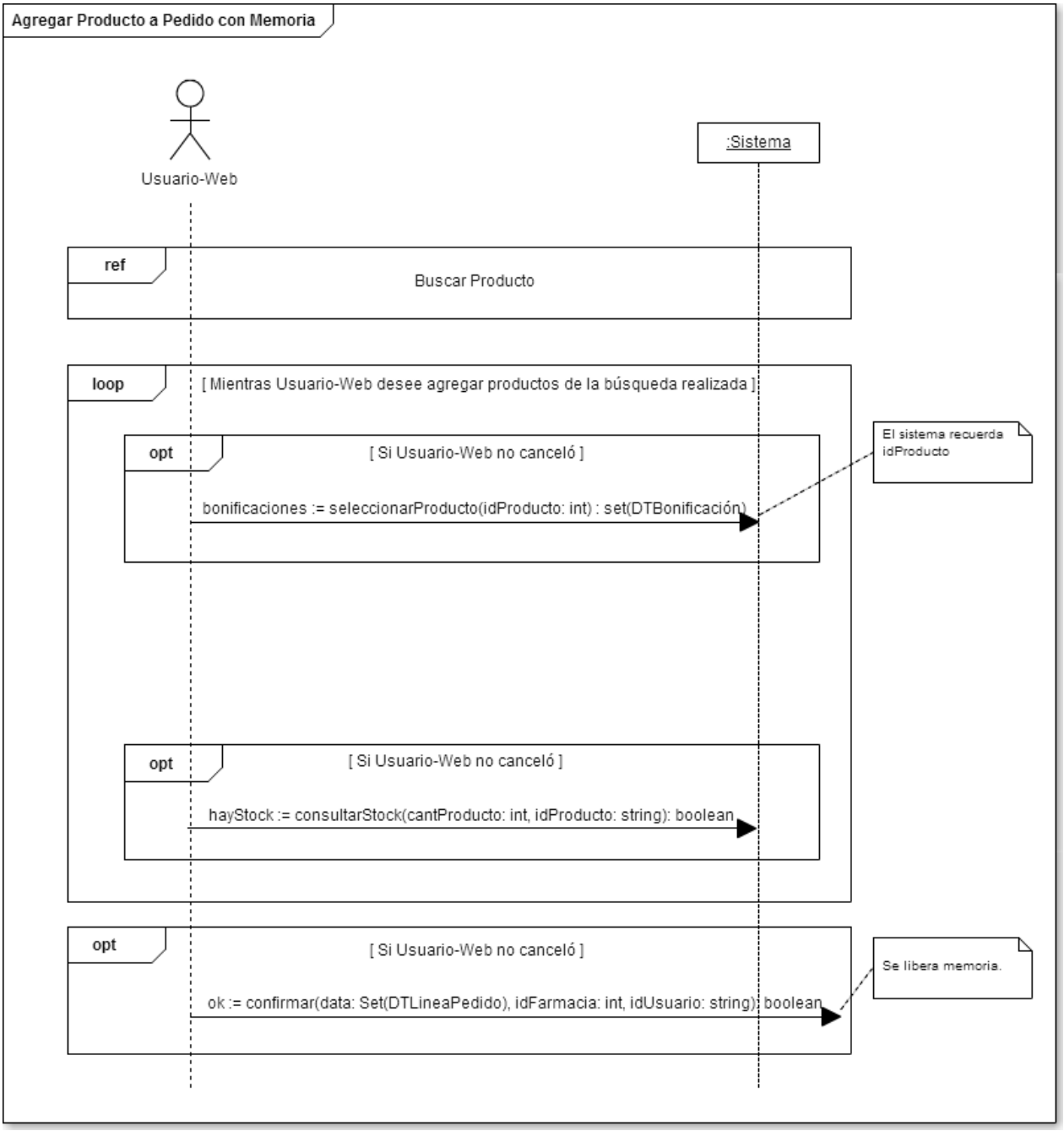
Se adjuntará en una próxima entrega.

2.4. Diseño del Caso de Uso - Agregar Producto a Pedido

2.4.1. Diagrama de paquetes



2.4.2. Diagrama de Interacción



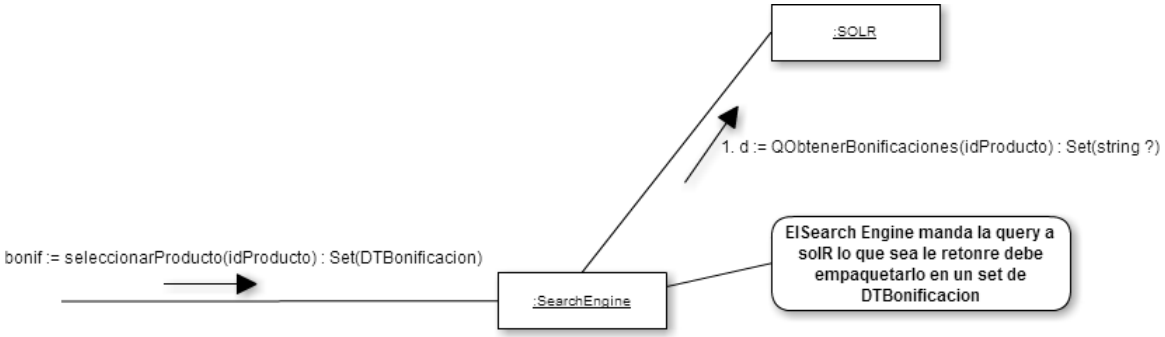
2.4.3. Diseño de Flujo de eventos

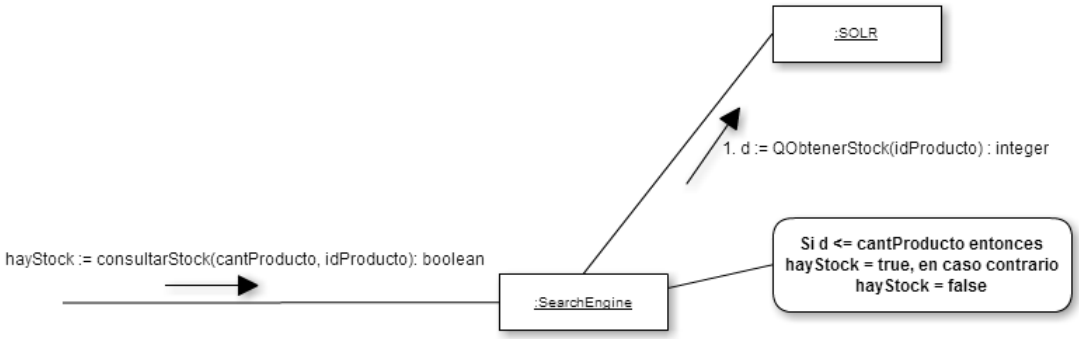
No aplica a esta entrega.

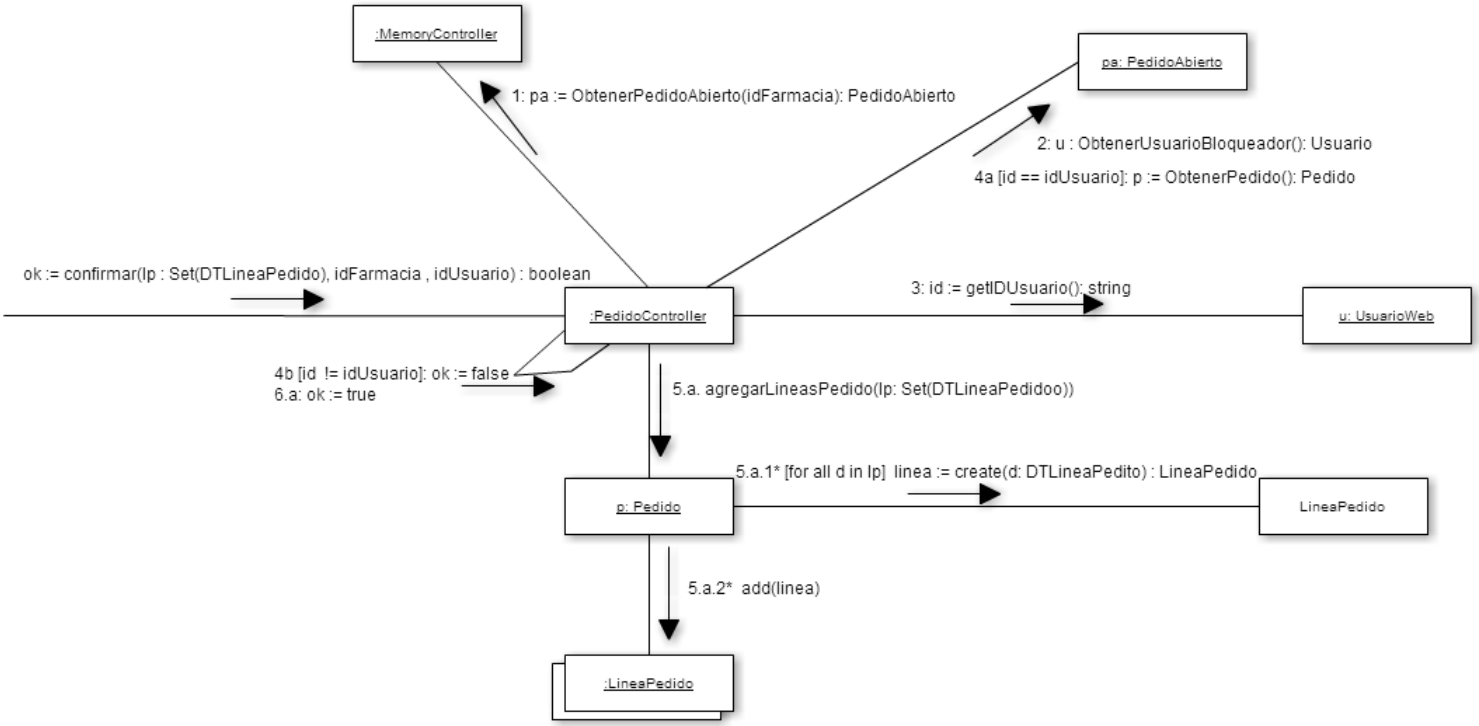
2.4.4. Requerimientos especiales o de implementación

No hay.

2.4.5. Diagrama de comunicación

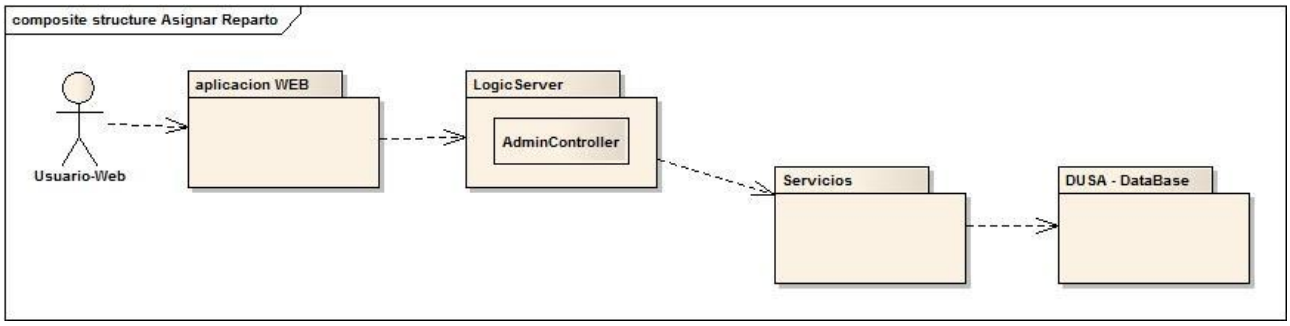




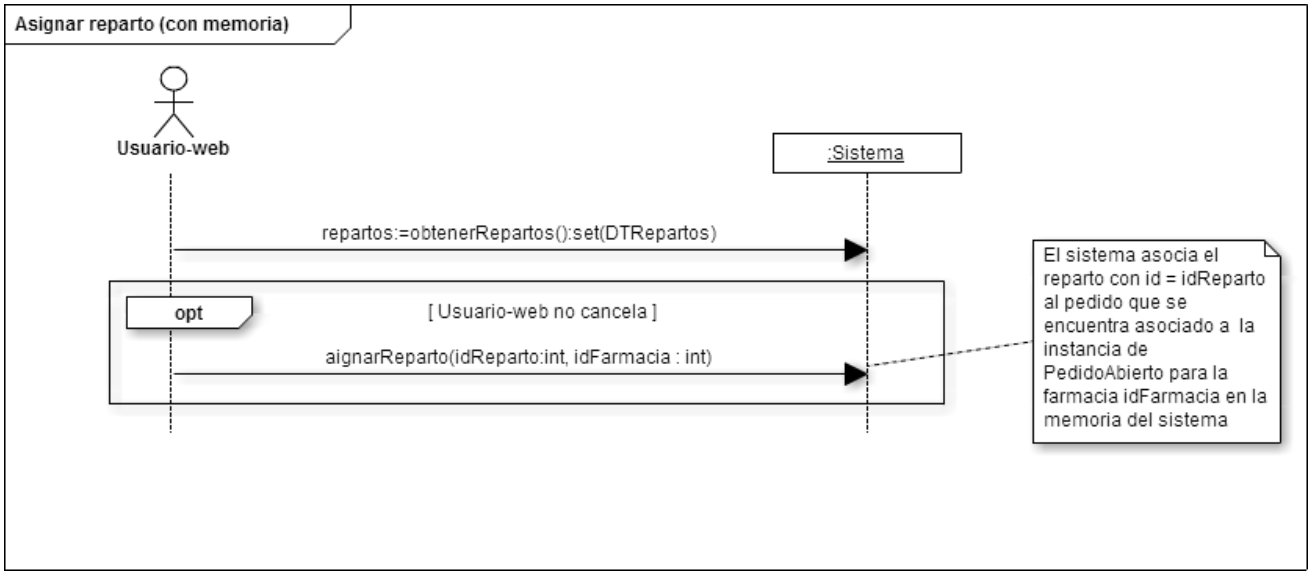


2.5. Diseño del Caso de Uso - Asignar Reparto

2.5.1. Diagrama de paquetes



2.5.2. Diagrama de Interacción



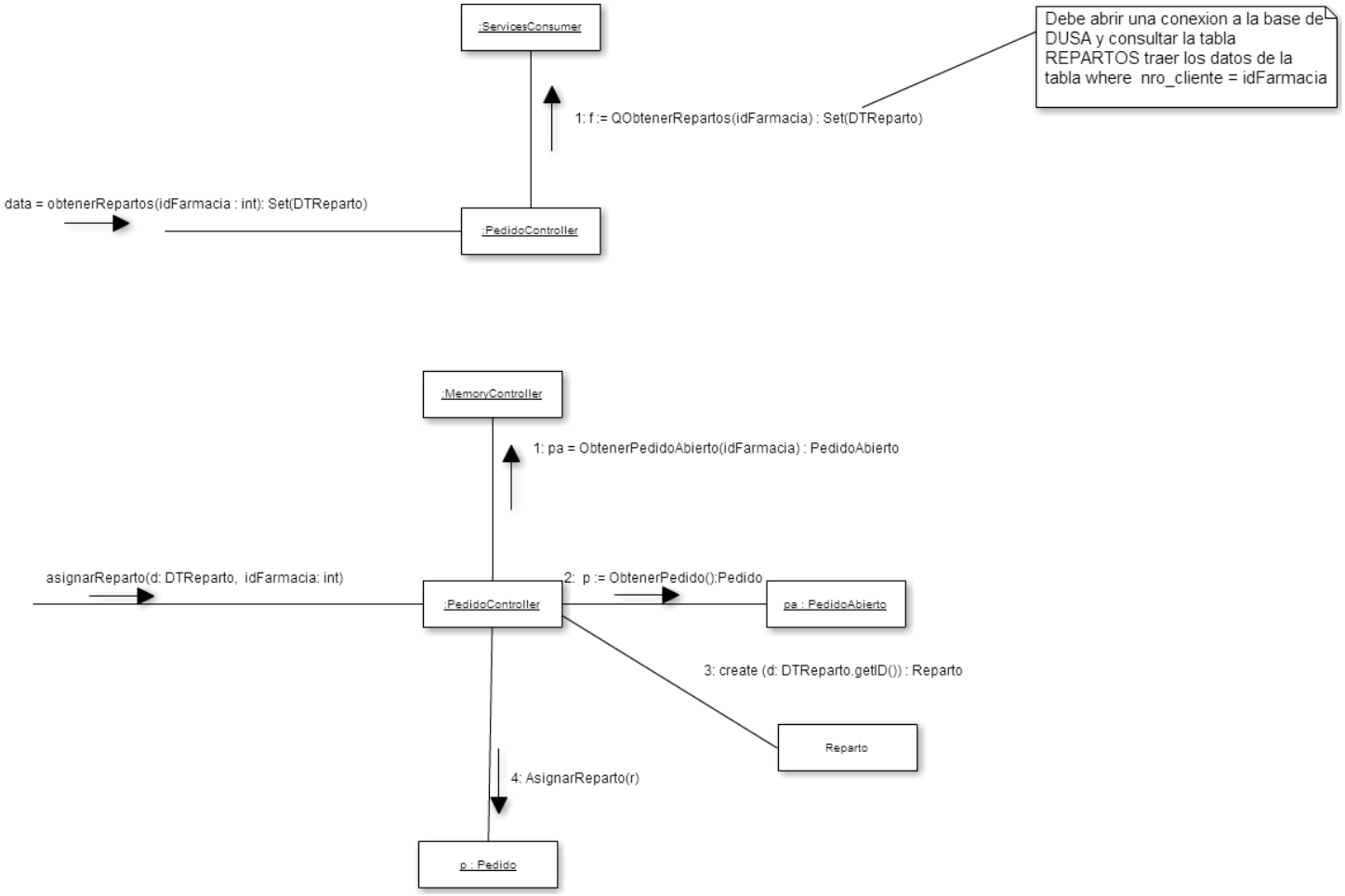
2.5.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.5.4. Requerimientos especiales o de implementación

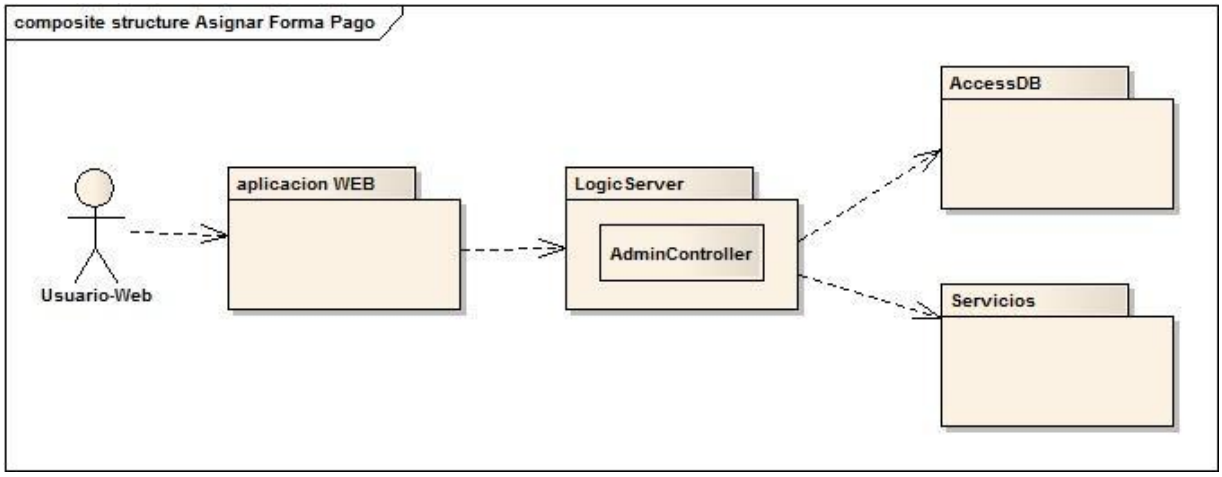
No hay.

2.5.5. Diagrama de comunicación

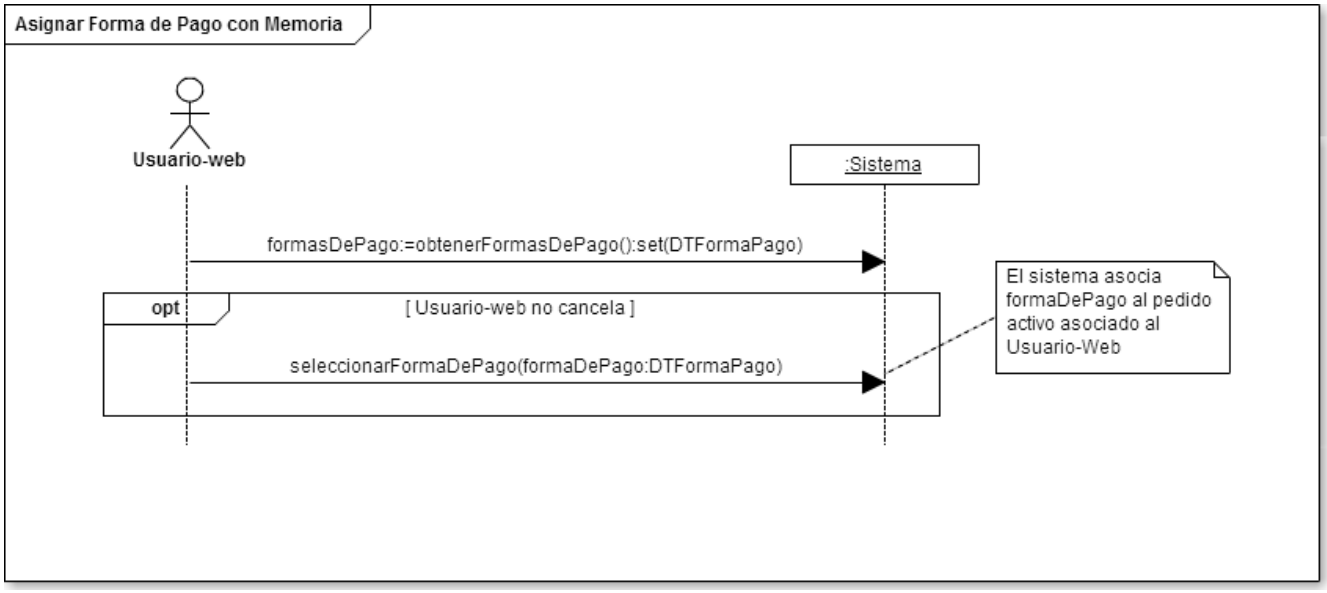


2.6. Diseño del Caso de Uso - Asignar Forma Pago

2.6.1. Diagrama de paquetes



2.6.2. Diagrama de Interacción



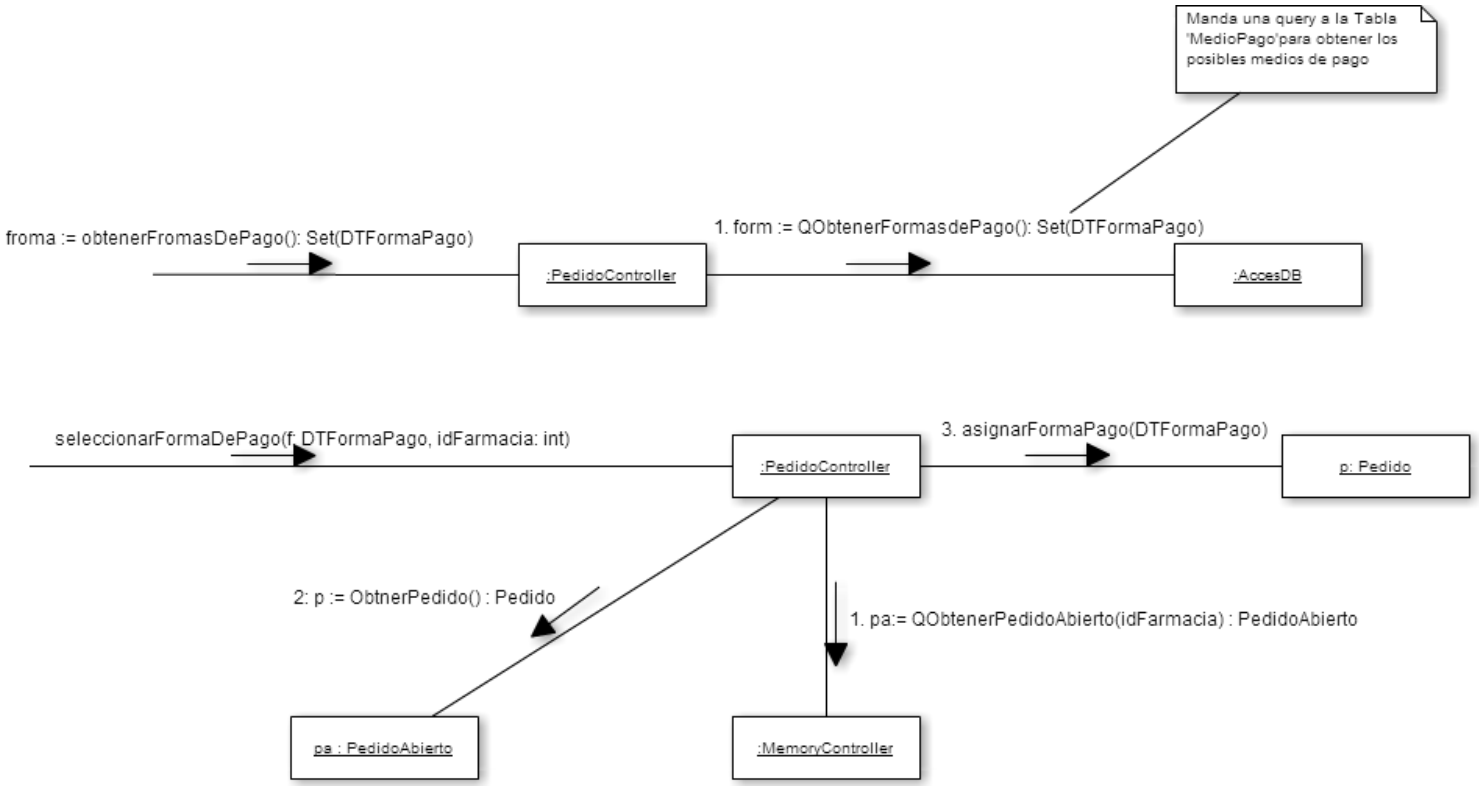
2.6.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.6.4. Requerimientos especiales o de implementación

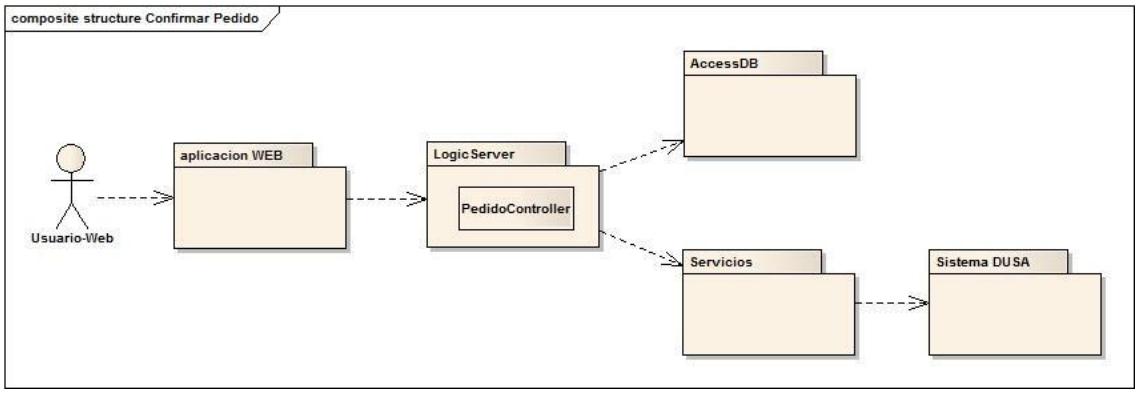
No hay.

2.6.5. Diagrama de comunicación

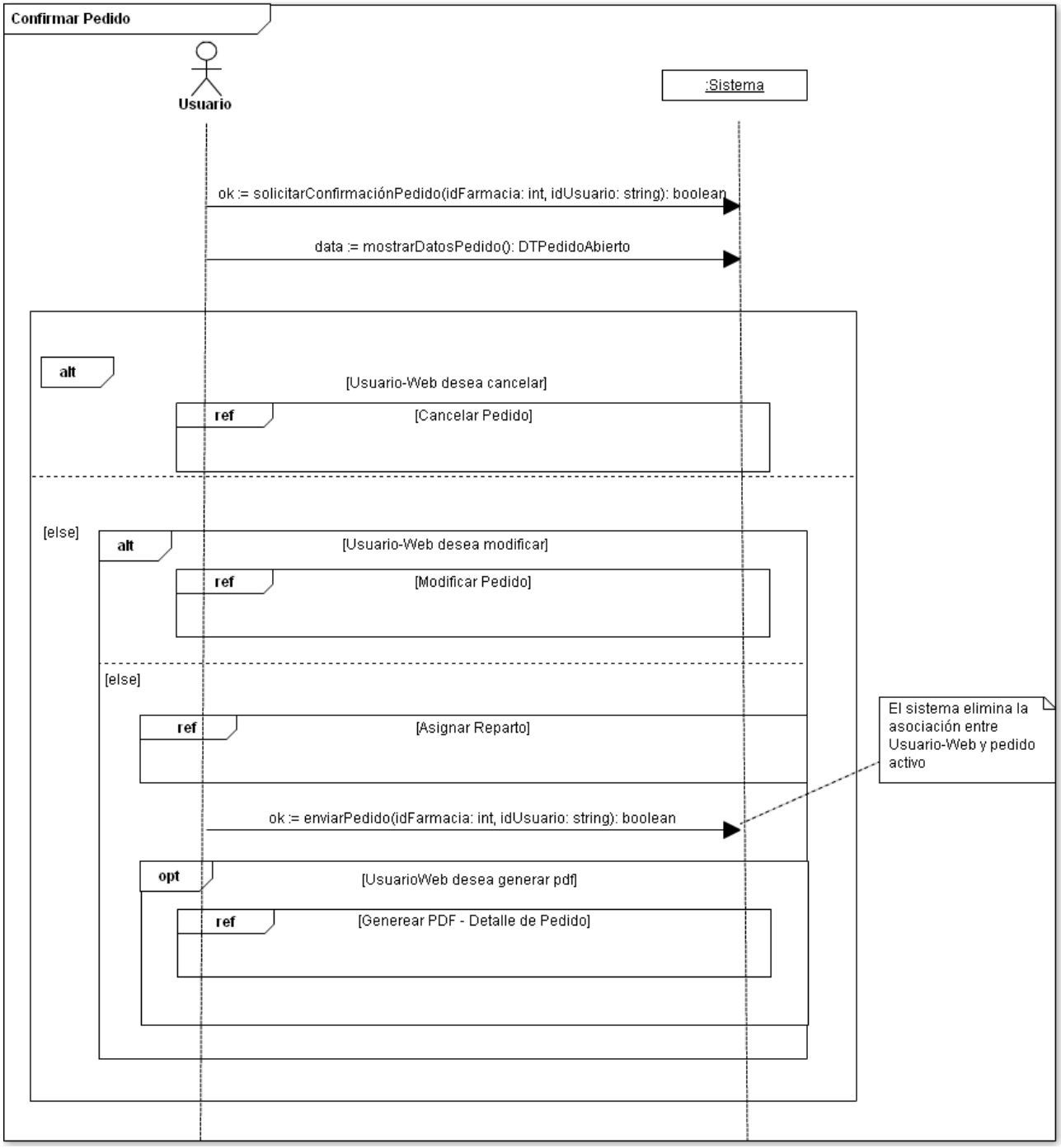


2.7. Diseño del Caso de Uso - Confirmar Pedido

2.7.1. Diagrama de paquetes



2.7.2. Diagrama de Interacción



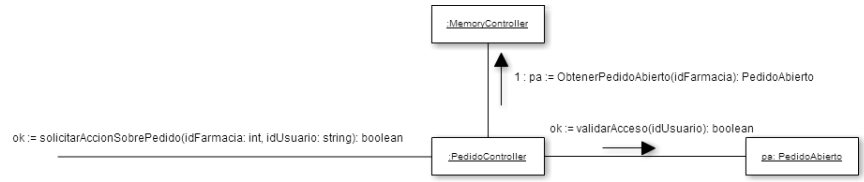
2.7.3. Diseño de Flujo de eventos

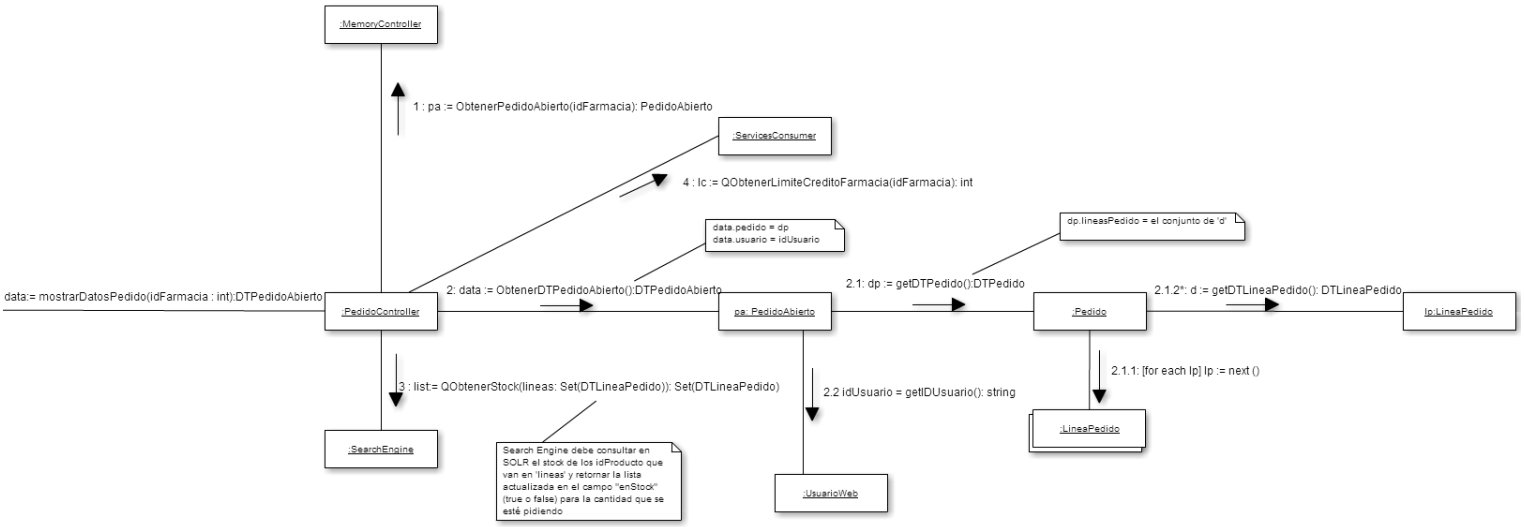
No aplica a esta entrega.

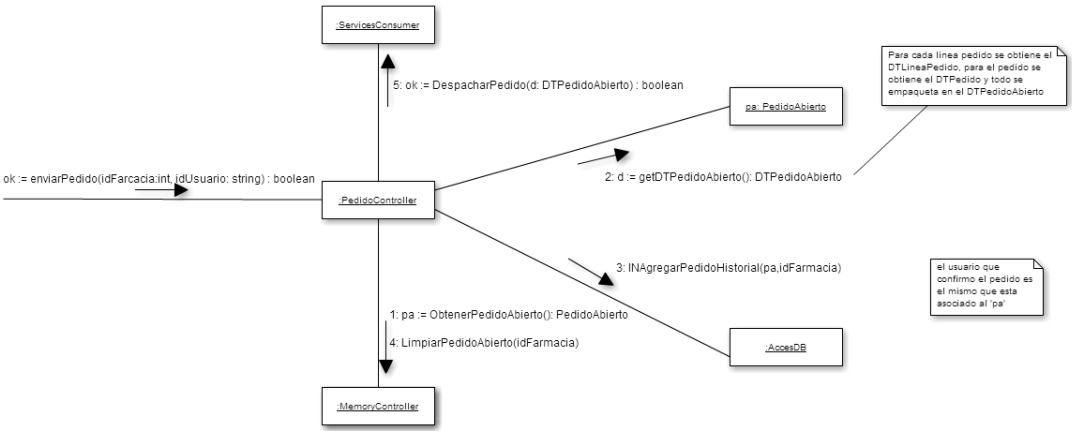
2.7.4. Requerimientos especiales o de implementación

No hay.

2.7.5. Diagrama de comunicación





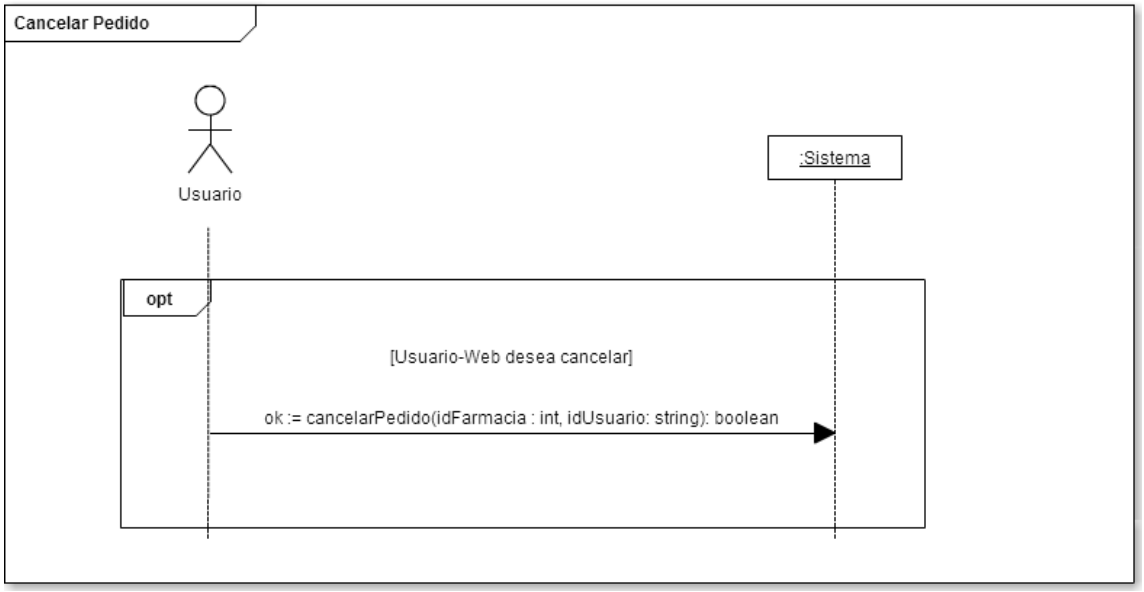


2.8. Diseño del Caso de Uso - Cancelar Pedido

2.8.1. Diagrama de paquetes

No aplica describirlo se implementa con los mismos paquetes de software del caso de uso Finalizar Sesión.

2.8.2. Diagrama de Interacción



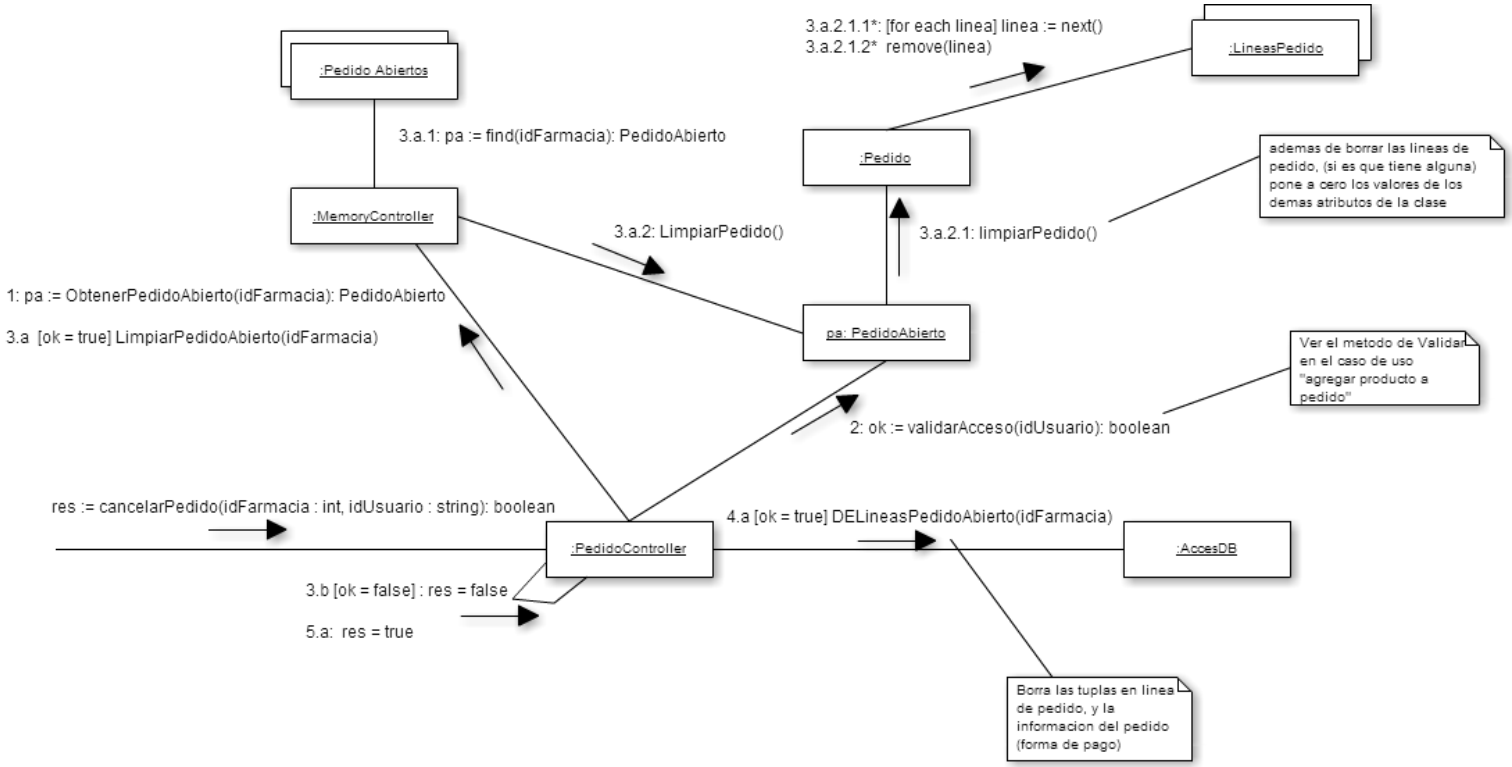
2.8.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.8.4. Requerimientos especiales o de implementación

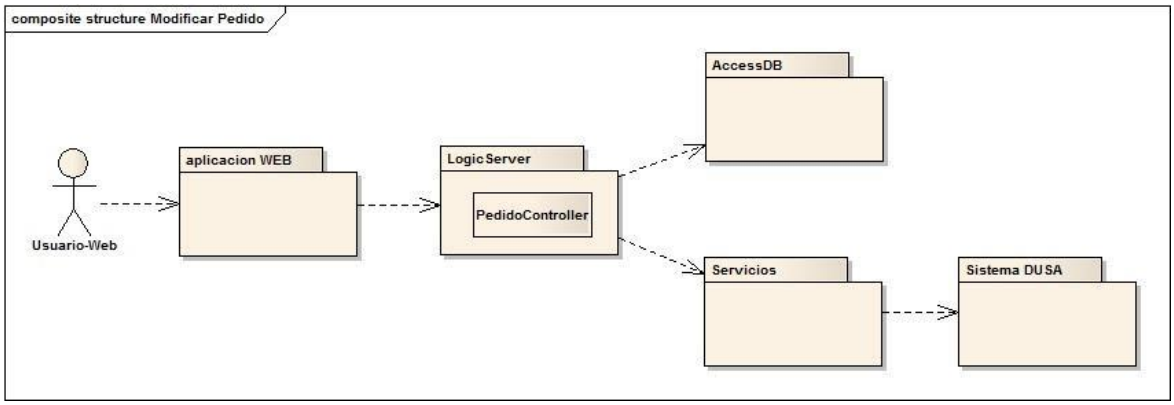
No hay.

2.8.5. Diagrama de comunicación

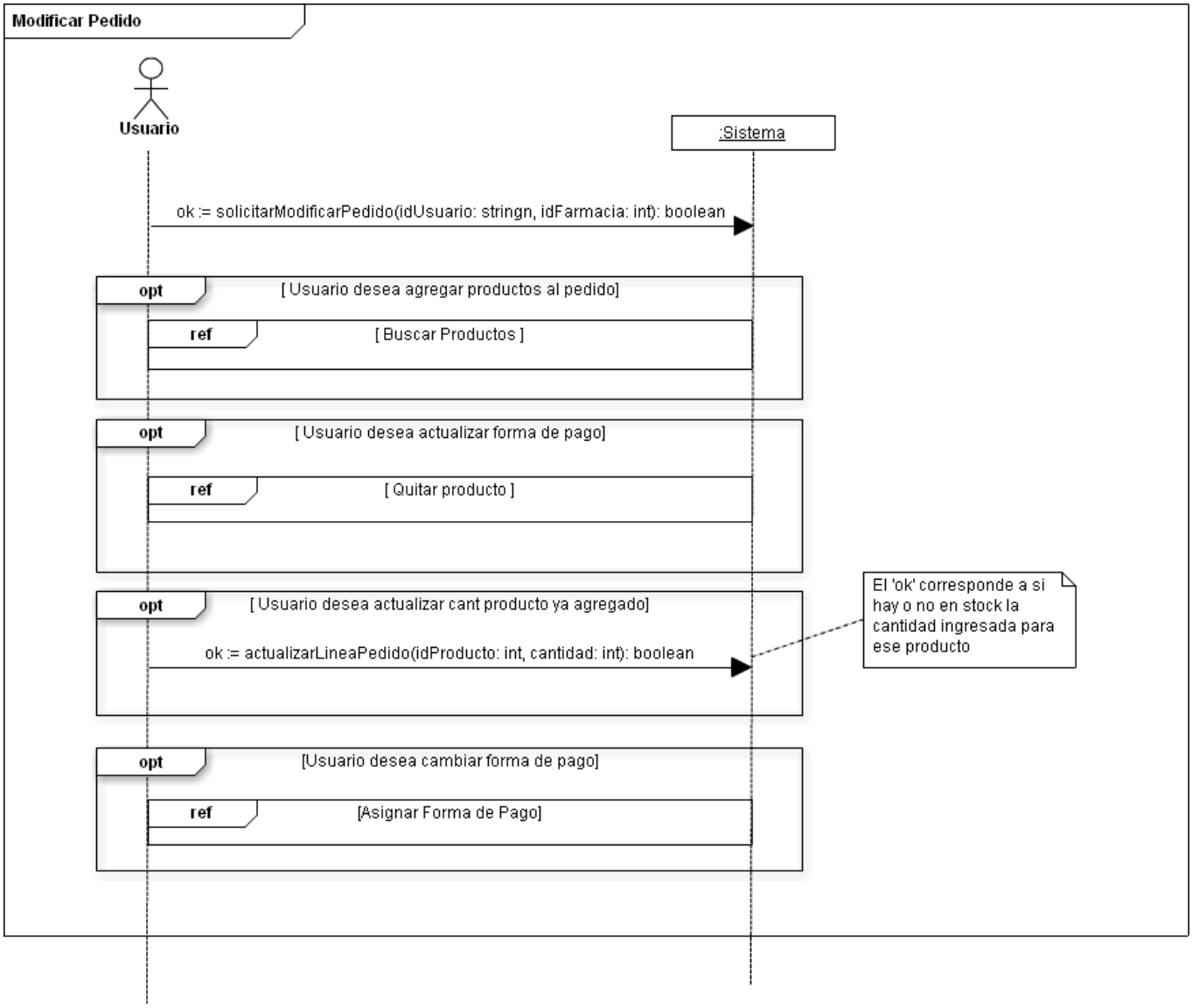


2.9. Diseño del Caso de Uso - Modificar Pedido

2.9.1. Diagrama de paquetes



2.9.2. Diagrama de Interacción



2.9.3. Diseño de Flujo de eventos

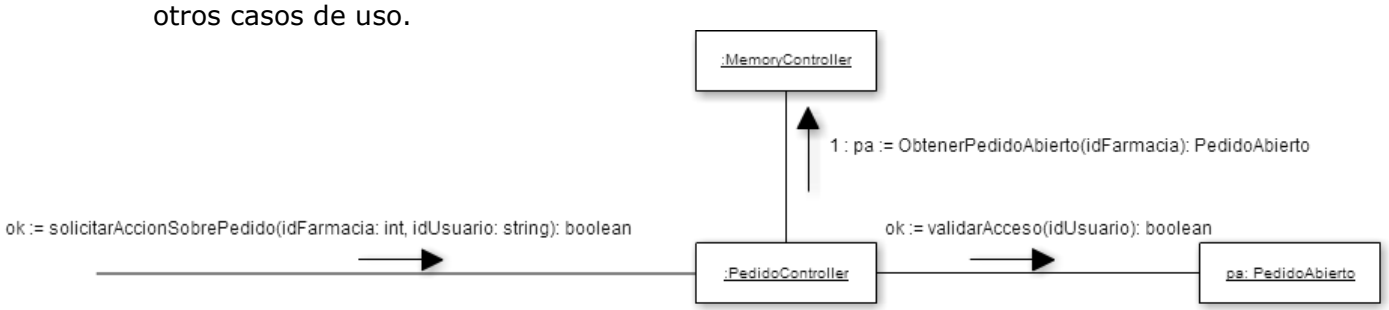
No aplica a esta entrega.

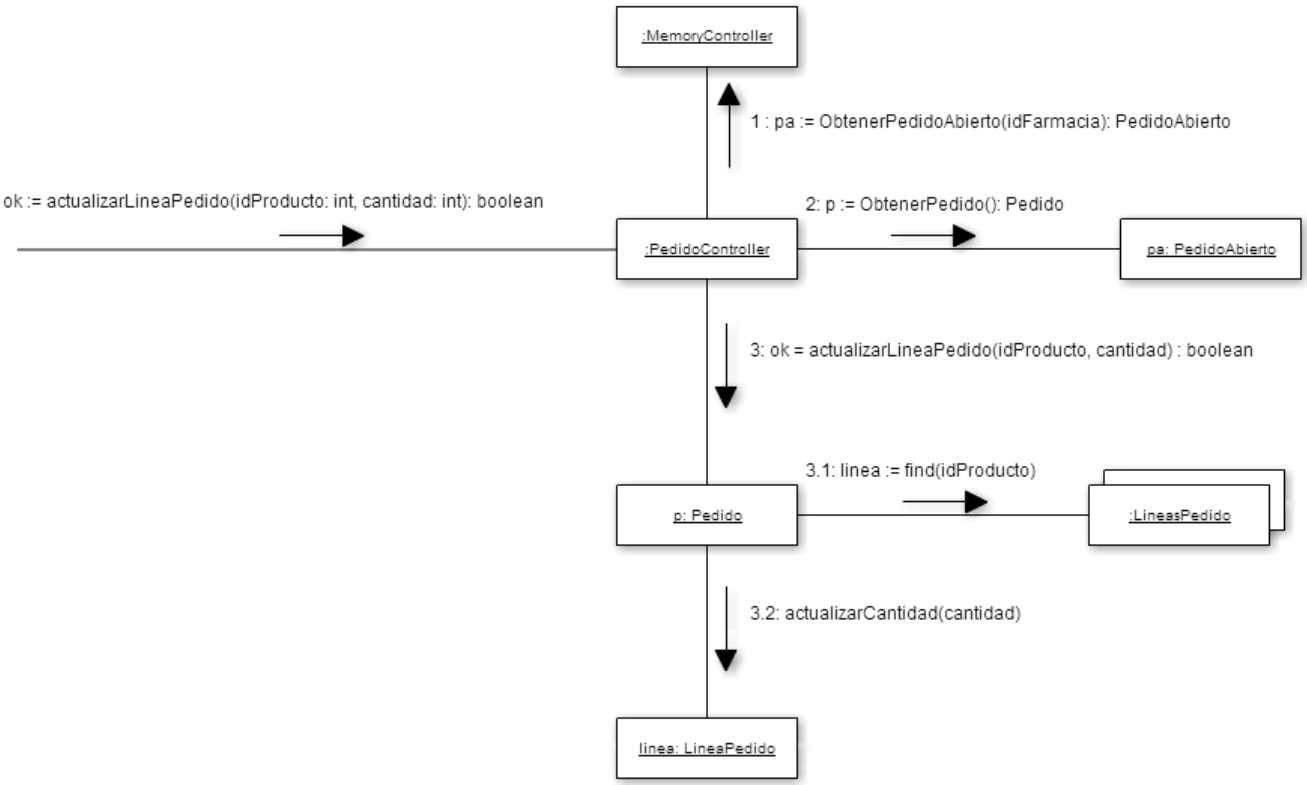
2.9.4. Requerimientos especiales o de implementación

No hay.

2.9.5. Diagrama de comunicación

Dado que el caso de uso Modificar incluye varios casos de uso ya descriptos se presetan los diagramas de comuniación de los métodos que no son utilizados en otros casos de uso.





2.10. Diseño del Caso de Uso - Ver Detalle Producto

Este caso de uso particular, se diseña dentro del buscar producto, es decir al desplegar el tootltip con la información de la oferta del producto seleccionado se agrega la infromación restante a desplegar, como ser:

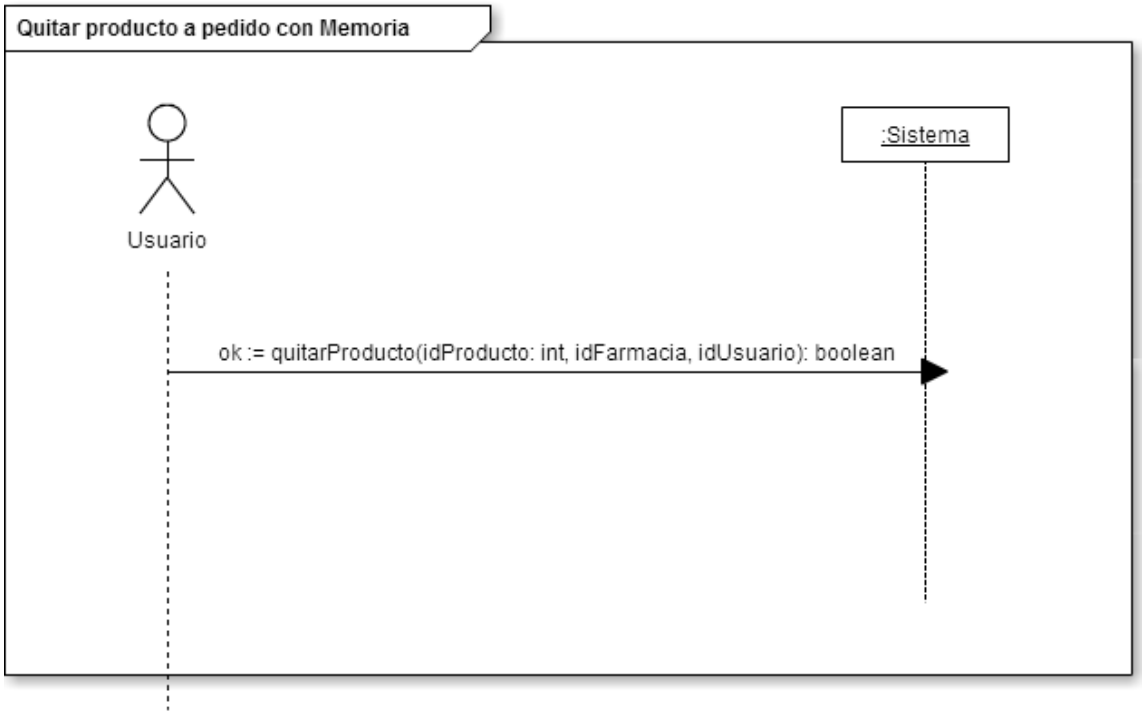
- Laboratorio
- Descripción producto
- Presentación

2.11 Diseño del Caso de Uso - Quitar Producto a Pedido

2.11.1. Diagrama de Paquetes

No aplica describirlo se implementa con los mismos paquetes de software del caso de uso Finalizar Sesión.

2.11.2. Diagrama de Interacción



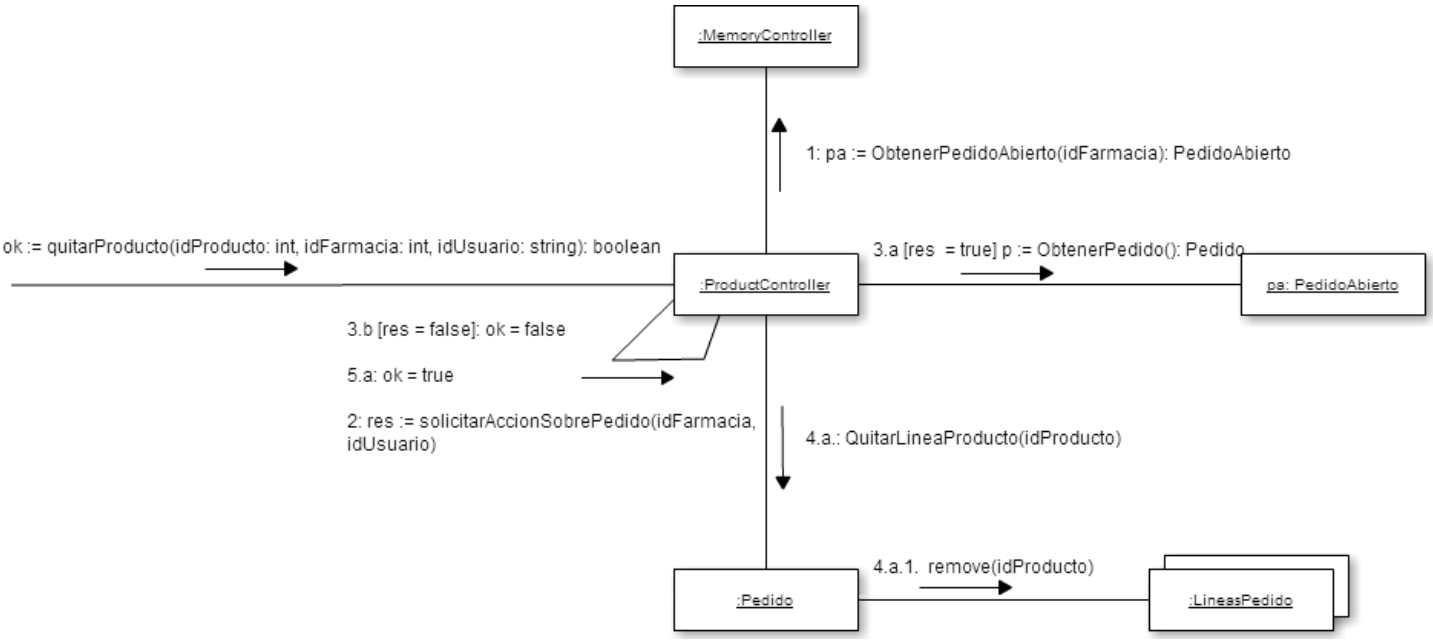
2.11.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.11.4. Requerimientos especiales o de implementación

No hay.

2.11.5. Diagrama de comunicación

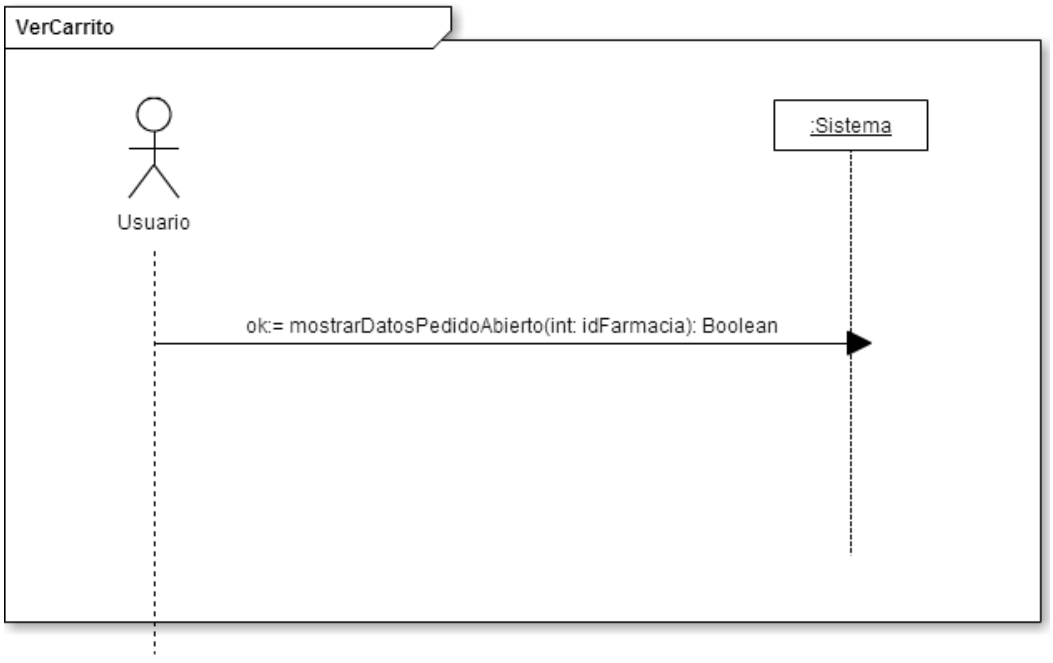


2.12. Diseño del Caso de Uso - Ver Carrito

2.12.1. Diagrama de Paquetes

No aplica describirlo se implementa con los mismos paquetes de software del caso de uso Asignar Reparto.

2.12.2. Diagrama de Interacción



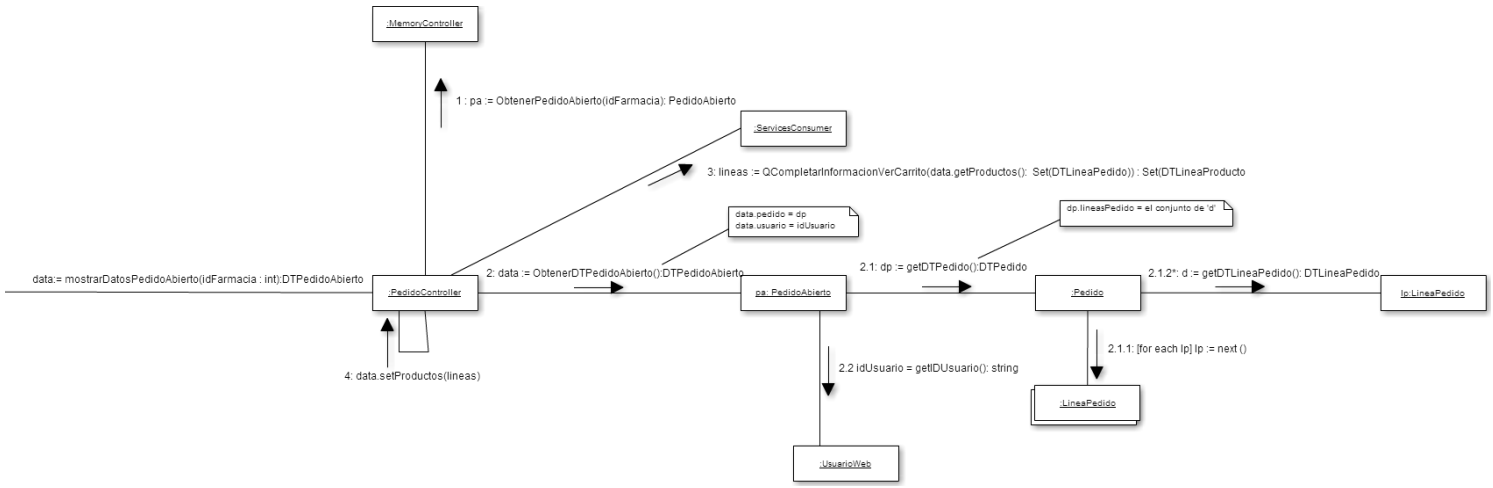
2.12.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.12.4. Requerimientos especiales o de implementación

No hay.

2.12.5. Diagrama de comunicación

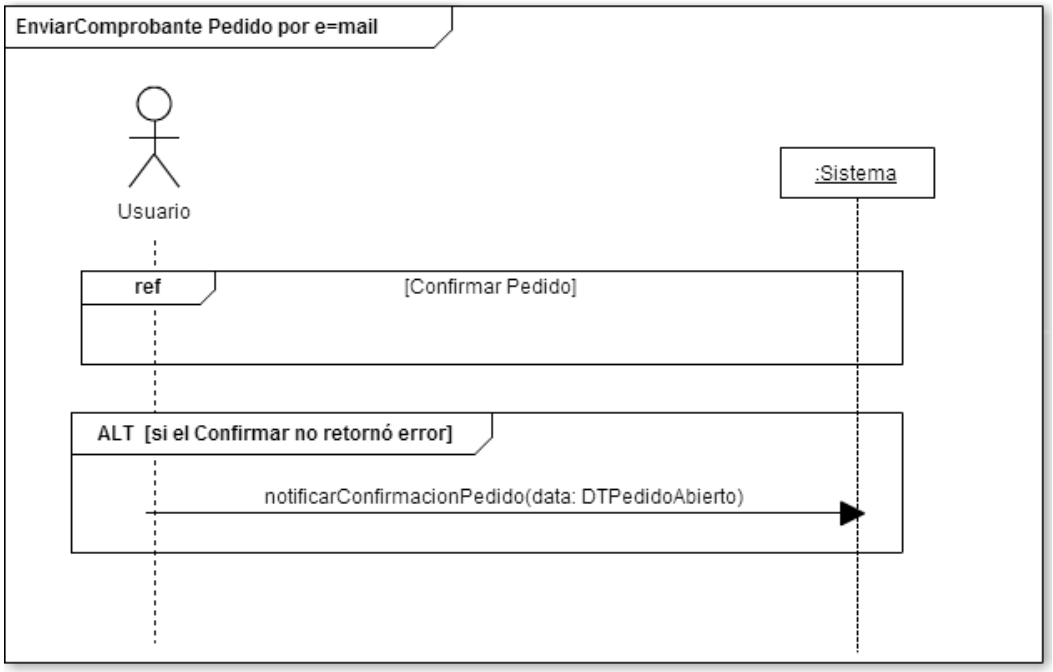


2.13. Diseño del Caso de Uso - Enviar comprobante pedido via e-mail

2.13.1. Diagrama de Paquetes

No aplica describirlo se implementa con los mismos paquetes de software del caso de uso Asignar Reparto.

2.13.2. Diagrama de Interacción



2.13.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.13.4. Requerimientos especiales o de implementación

No hay.

2.13.5. Diagrama de comunicación

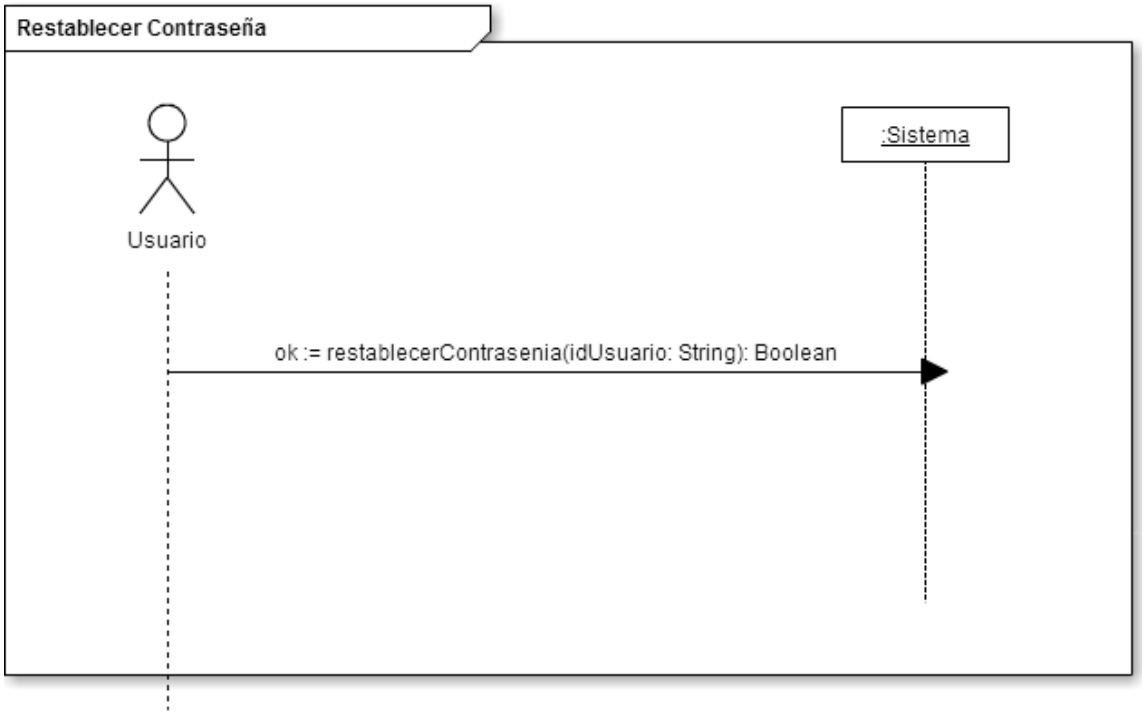
2.14. Diseño del Caso de Uso - Restablecer contraseña

2.14.1. Diagrama de Paquetes

No aplica describirlo se implementa con los mismos paquetes de software del caso de uso Iniciar Sesión.

2.14.2. Diagrama de Interacción

Se adjuntará en la próxima entrega del documento.



2.14.3. Diseño de Flujo de eventos

No aplica a esta entrega.

2.14.4. Requerimientos especiales o de implementación

No hay.

2.14.5. Diagrama de comunicación

Se adjuntará en la próxima entrega del documento.

3. Diseño de Objetos

Es de interés presentar en las siguientes secciones el diseño de objetos que componen el Sistema Pedidos Online DUSA. Para una mayor claridad se presenta el diseño de objetos mediante diagrama de clases de diseño separado por subsistemas. En el documento de Descripción de la Arquitectura para este sistema se contemplaron integrar 3 subsistemas, de los cuales 2 son implementados en su totalidad por el equipo de desarrollo (subsistema Logica y WEB) el diseño de objetos de cada subsistema se describen las secciones siguientes.

4. Diseño de Subsistemas

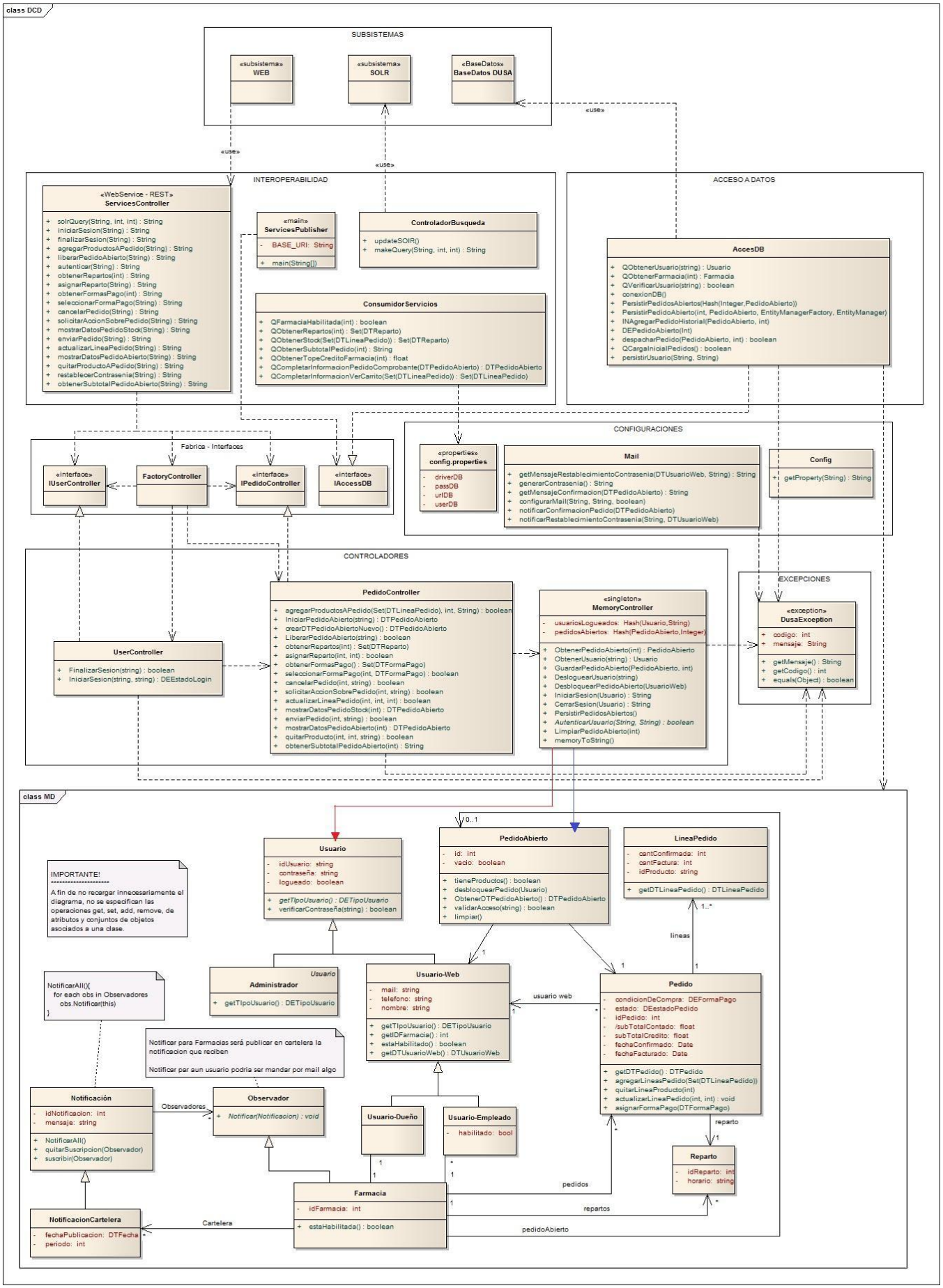
En esta sección es de interés presentar el diseño de los subsistemas definidos para la solución de todo el sistema. Para cada subsistema se presenta el diseño de objetos que lo componen.

4.1. LogicServer

4.1.1 Diseño de Objetos

En el DCD que se presenta en la siguiente página se pueden identificar capas (de abajo hacia arriba):

- MD - corresponde a las clases que surgieron del modelo de dominio.
- Controladores - corresponde a las clases Controller que implementan las operaciones de los DSS de los casos de usos del sistema, y el controlador Memory que oficia de memoria del sistema (siendo una clase singleton)
- Una capa de interfaces y fábrica donde se exponen aquellos métodos de los controladores que interesa que sean publicados como web-services para ser consumidos desde la aplicación web.
- Un paquete de clases de configuración que sirven para administrar las conexiones del subsistema tanto con la base de datos de DUSA, como con el servidor de correo provisto por DUSA.
- La capa de interoperabilidad, que reúne las clases para interactuar con otros subsistemas, propios a la solución como externos:
    - AccesDB, para acceder a las tablas definidas en el documento Modelo de Datos
    - Paquete de clases "Servicios" (engloba las clases ServiceConsumer para consumir servicios de dusa y leer de su base de datos, ServiceController para armar los web-services de las operaciones conocidas por medio de las interfaces de controladores y ServicePublisher para publicar los web-services definidos por ServiceController)

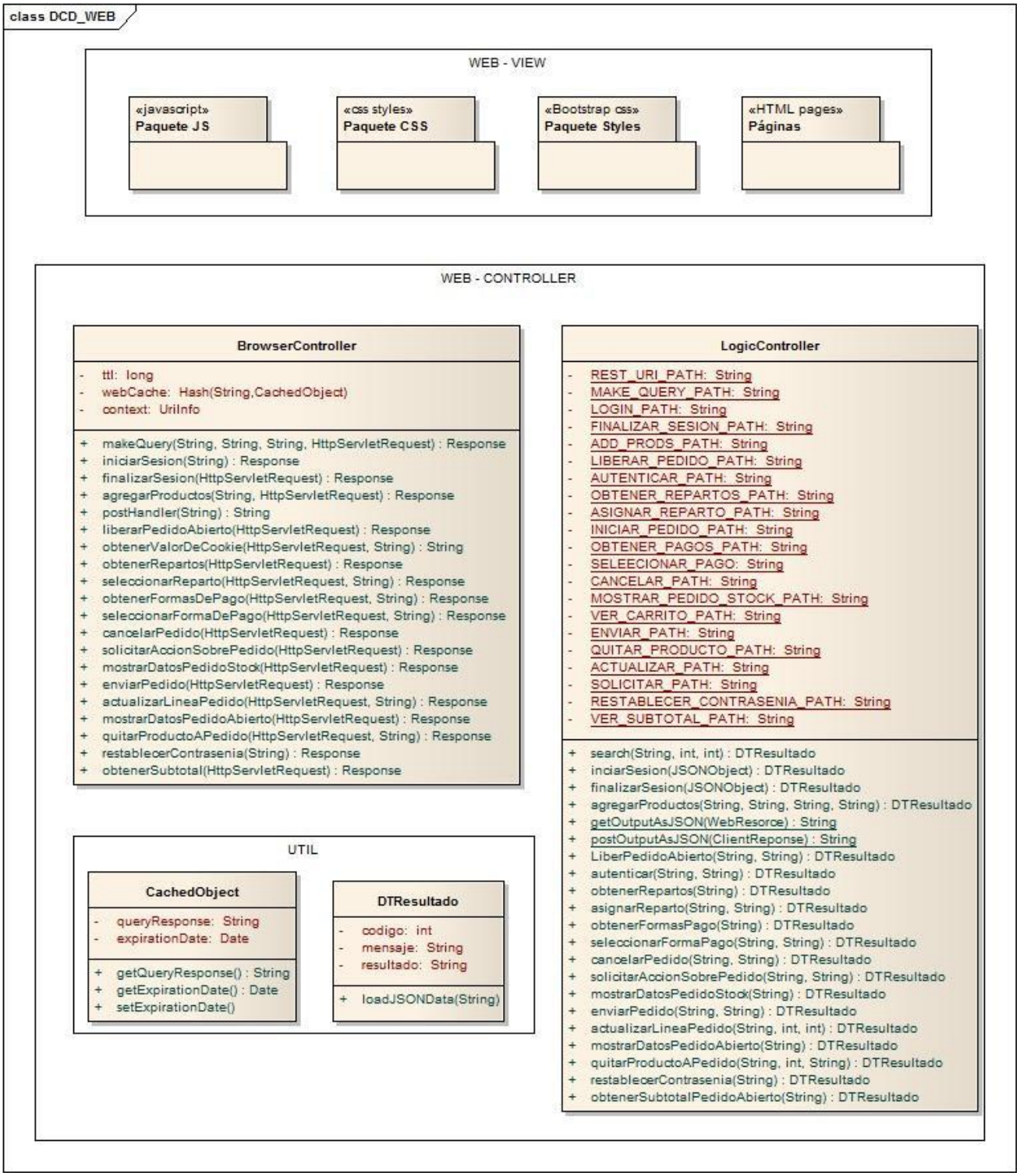

class DCD
SUBSISTEMAS
«subsistema» WEB
«subsistema» SOLR
«BaseDatos» BaseDatos DUSA
«use»
INTEROPERABILIDAD
«WebService - REST» ServicesController
+ solrQuery(String, int, int) : String
+ iniciarSesion(String) : String
+ finalizarSesion(String) : String
+ agregarProductosAPedido(String) : String
+ liberarPedidoAbierto(String) : String
+ autenticar(String) : String
+ obtenerRepartos(int) : String
+ asignarReparto(String) : String
+ obtenerFormasPago(int) : String
+ seleccionarFormaPago(String) : String
+ cancelarPedido(String) : String
+ solicitarAccionSobrePedido(String) : String
+ mostrarDatosPedidoStock(String) : String
+ enviarPedido(String) : String
+ actualizarLineaPedido(String) : String
+ mostrarDatosPedidoAbierto(String) : String
+ quitarProductoAPedido(String) : String
+ restablecerContrasenia(String) : String
+ obtenerSubtotalPedidoAbierto(String) : String
«main» ServicesPublisher
- BASE_URI: String
+ main(String[])
ControladorBusqueda
+ updateSOIR()
+ makeQuery(String, int, int) : String
ConsumidorServicios
+ QFarmaciaHabilitada(int) : boolean
+ QObtenerRepartos(int) : Set(DTReparto)
+ QObtenerStock(Set(DTLineaPedido)) : Set(DTReparto)
+ QObtenerSubtotalPedido(int) : String
+ QObtenerTopeCreditoFarmacia(int) : float
+ QCompletarInformacionPedidoComprobante(DTPedidoAbierto) : DTPedidoAbierto
+ QCompletarInformacionVerCarrito(Set(DTLineaPedido)) : Set(DTLineaPedido)
ACCESO A DATOS
AccesDB
+ QObtenerUsuario(string) : Usuario
+ QObtenerFarmacia(int) : Farmacia
+ QVerificarUsuario(string) : boolean
+ conexionDB()
+ PersistirPedidosAbiertos(Hash(Integer,PedidoAbierto))
+ PersistirPedidoAbierto(int, PedidoAbierto, EntityManagerFactory, EntityManager)
+ INAgregarPedidoHistorial(PedidoAbierto, int)
+ DEPedidoAbierto(int)
+ despacharPedido(PedidoAbierto, int) : boolean
+ QCargaInicialPedidos() : boolean
+ persistirUsuario(String, String)
Fabrica - Interfaces
«interface» IUserController
FactoryController
«interface» IPedidoController
«interface» IAccessDB
CONFIGURACIONES
«properties» config.properties
- driverDB
- passDB
- urlDB
- userDB
Mail
+ getMensajeRestablecimientoContrasenia(DTUsuarioWeb, String) : String
+ generarContrasenia() : String
+ getMensajeConfirmacion(DTPedidoAbierto) : String
+ configurarMail(String, String, boolean)
+ notificarConfirmacionPedido(DTPedidoAbierto)
+ notificarRestablecimientoContrasenia(String, DTUsuarioWeb)
Config
+ getProperty(String) : String
CONTROLADORES
PedidoController
+ agregarProductosAPedido(Set(DTLineaPedido), int, String) : boolean
+ IniciarPedidoAbierto(string) : DTPedidoAbierto
+ crearDTPedidoAbiertoNuevo() : DTPedidoAbierto
+ LiberarPedidoAbierto(string) : boolean
+ obtenerRepartos(int) : Set(DTReparto)
+ asignarReparto(int, int) : boolean
+ obtenerFormasPago() : Set(DTFormaPago)
+ seleccionarFormaPago(int, DTFormaPago) : boolean
+ cancelarPedido(int, string) : boolean
+ solicitarAccionSobrePedido(int, string) : boolean
+ actualizarLineaPedido(int, int, int) : boolean
+ mostrarDatosPedidoStock(int) : DTPedidoAbierto
+ enviarPedido(int, string) : boolean
+ mostrarDatosPedidoAbierto(int) : DTPedidoAbierto
+ quitarProducto(int, int, string) : boolean
+ obtenerSubtotalPedidoAbierto(int) : String
«singleton» MemoryController
- usuariosLogueados: Hash(Usuario,String)
- pedidosAbiertos: Hash(PedidoAbierto,Integer)
+ ObtenerPedidoAbierto(int) : PedidoAbierto
+ ObtenerUsuario(string) : Usuario
+ GuardarPedidoAbierto(PedidoAbierto, int)
+ DesloguearUsuario(string)
+ DesbloquearPedidoAbierto(UsuarioWeb)
+ IniciarSesion(Usuario) : String
+ CerrarSesion(Usuario) : String
+ PersistirPedidosAbiertos()
+ AutenticarUsuario(String, String) : boolean
+ LimpiarPedidoAbierto(int)
+ memoryToString()
EXCEPCIONES
«exception» DusaException
+ codigo: int
+ mensaje: String
+ getMensaje() : String
+ getCodigo() : int
+ equals(Object) : boolean
UserController
+ FinalizarSesion(string) : boolean
+ IniciarSesion(string, string) : DEEstadoLogin
class MD
IMPORTANTE!
A fin de no recargar innecesariamente el diagrama, no se especifican las operaciones get, set, add, remove, de atributos y conjuntos de objetos asociados a una clase.
Usuario
- idUsuario: string
- contraseña: string
- logueado: boolean
+ getTipoUsuario() : DETipoUsuario
+ verificarContraseña(string) : boolean
PedidoAbierto
- id: int
- vacio: boolean
+ tieneProductos() : boolean
+ desbloquearPedido(Usuario)
+ ObtenerDTPedidoAbierto() : DTPedidoAbierto
+ validarAcceso(string) : boolean
+ limpiar()
LineaPedido
- cantConfirmada: int
- cantFactura: int
- idProducto: string
+ getDTLineaPedido() : DTLineaPedido
0..1
1..*
lineas
Usuario
Administrador
+ getTIpoUsuario() : DETipoUsuario
Usuario-Web
- mail: string
- telefono: string
- nombre: string
+ getTIpoUsuario() : DETipoUsuario
+ getIDFarmacia() : int
+ estaHabilitado() : boolean
+ getDTUsuarioWeb() : DTUsuarioWeb
usuario web
Pedido
- condicionDeCompra: DEFormaPago
- estado: DEestadoPedido
- idPedido: int
- /subTotalContado: float
- subTotalCredito: float
- fechaConfirmado: Date
- fechaFacturado: Date
+ getDTPedido() : DTPedido
+ agregarLineasPedido(Set(DTLineaPedido))
+ quitarLineaProducto(int)
+ actualizarLineaPedido(int, int) : void
+ asignarFormaPago(DTFormaPago)
reparto
NotificarAll(){
for each obs in Observadores
obs.Notificar(this)
}
Notificar para Farmacias será publicar en cartelera la notificacion que reciben
Notificar par aun usuario podria ser mandar por mail algo
Notificación
- idNotificacion: int
- mensaje: string
+ NotificarAll()
+ quitarSuscripcion(Observador)
+ suscribir(Observador)
Observadores
Observador
+ Notificar(Notificacion) : void
Usuario-Dueño
Usuario-Empleado
- habilitado: bool
Reparto
- idReparto: int
- horario: string
Farmacia
- idFarmacia: int
+ estaHabilitada() : boolean
pedidos
repartos
pedidoAbierto
NotificacionCartelera
- fechaPublicacion: DTFecha
- periodo: int
Cartelera

Observación:

Para una mayor claridad del DCD se adjunta la imagen al final de este documento.Para mayores detalles respecto a este subsistema se sugiere ver la descripción del mismo en el documento de Descripción de Arquitectura, donde se expresan las razones de su existencia, diseño y principales funcionalidades.

4.2. WEB-APP

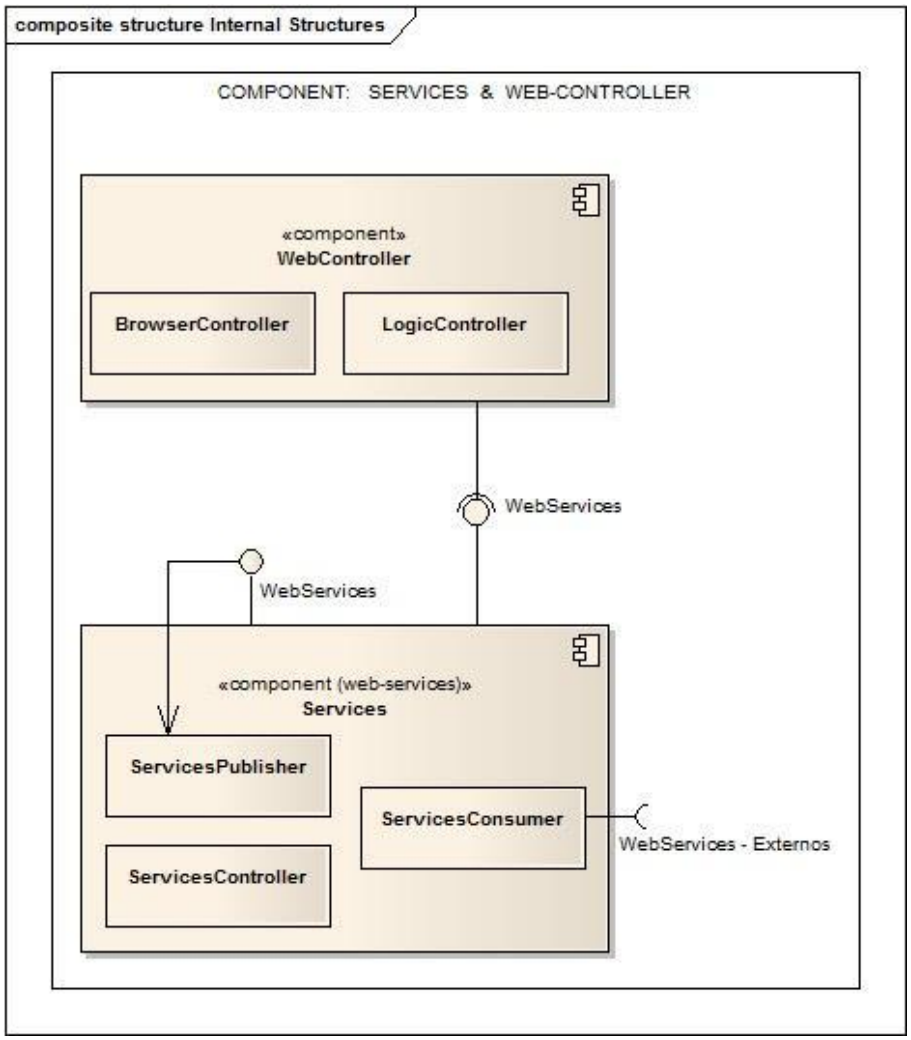
4.2.1 Diseño de Objetos



En el DCD se pueden identificar capas:

- WEB-VIEW - contiene los paquetes de software que implementan la vista de las distintas páginas que visualizará el usuario final. Dentro de este conjunto de paquetes se identifican:
    - Paquete JS, paquete de clases javascript que son utilizadas bajo el modelo de framework Backbone para renderizar las distintas vistas a presentar al usuario, mediante captación de eventos se renderizan las páginas y se invocan los servicios publicados por el subsistema Logica.
    - Paquete CSS, contiene los estilos CSS que son utilizados en el fron-end
    - Paquete Styles, contiene estilos css propios de Bootstrap, el framwork seleccionado para las págias del front-end
    - Páginas, es el paquete que contiene los html que son renderizados por backbone en respuesta a los distintos eventos captados.
- WEB-Controller, comprende dos clases que actuan como controladores facade:
    - BrowserController, capta los request del navegador, para cada request ejecuta un método del LogicController, obtiene el resultado y lo retorna como response a la llamada del navegador.
    - LogicController, realiza las invocaciones sobre los servicios publicados por el subsistema Logica, es utilizado como fachada donde se encuentran todos los servicios que pueden ser consumidos desde el BrowserController, para responder las solicitudes enviadas desde el navegador por el usuario.
- Un par de clases utiliatarias, que son necesarias por los controladores facade de la capa Web-Controller.

A modo de complementar los diagramas que se ilustran en el documento de Descripción de Arquitectura para estos sub-sistemas, se presenta un diagrama de componentes internas para los componentes: WebController (del WEB-APP) y Services (del LogicServer)



En la siguiente tabla se describen las principales características de las componentes internas ilustradas en el diagrama anterior.

Subsistema	Component		Descripción
WEB-APP	Web-Controller	Browser Controller	Es el que envía los mensajes hacia el navegador.
		Logic Controller	Es el que invoca los web services publicados por ServicesPublisher, se comunica hacia el LogicServer
Logic Server	Services	Services Publisher	Es el que publica los web services asociados los métodos de los controladores del LogicServer
		Services Controller	Es el que realiza las anotaciones REST sobre los métodos de la Lógica que serán publicados.
		Services Consumer	Es la componente que consumirá los servicios de Sistemas externos (en caso que

			hayan sistemas externos que interactúen con el Sistema "Pedidos Online -DUSA")

Al igual que el caso anterior este subsistema también fue presentado en el documento Descripción de Arquitectura, se sugiere en cuanto a diseño y funcionalidades remitirse al documento referenciado.

5. Diseño Interfaz de Usuario

Se complementará en la próxima entrega del documento.

5.1. Framework Backbone

5.2. Formatos individuales de interfaz de pantalla

No aplica a esta entrega.

5.3. Controles y elementos de diseño de interfaz de pantalla

No aplica a esta entrega.

5.4. Forma de navegación de interfaz de pantalla

No aplica a esta entrega.

5.5. Formatos de impresión

No aplica.

6. Diseño de Datos

A modo de evitar una redundancia de información se sugiere considerar el documento de Modelo de Datos, para esta sección, en el mismo se encuentra el DCD correspondiente a las clases que serán persistidas, Modelo entidad relación y diagrama de las tablas definidas.

7. Manejo de Excepciones

Para el manejo de errores en el sistema se optó por definir una excepción DusaExcepcion, con código y mensaje asociado, de forma tal que para cada código se corresponde un único mensaje. De esta forma ante cualquier error, de validación, de conexión con los sistemas externos (Base Datos D.U.S.A., Servidor de Mail de D.U.S.A.) el sistema sea capaz de responder apropiadamente al usuario.

En la siguiente tabla se detalla la lista parejas (código, mensaje) de DusaExcepción que son utilizadas en el sistema:

CODIGO	MSJ - ERROR ó ALERTA
1	El Usuario no se encuentra en la base datos del Sistema
2	La contraseña ingresada es incorrecta
3	La Farmacia asociada al usuario no se encuentra en el Sistema
4	La Farmacia no esta habilitada
5	Error de autenticación
6	No se encuentra el Pedido Abierto para la Farmacia
7	El Pedido esta siendo modificado por otro usuario
8	Error al despachar el pedido
9	No existe un pedido abierto para cancelar
10	Ha ocurrido un error en el Sistema, disculpe las molestias ocasionadas
11	Ha ocurrido un error en la Base de Datos del Sistema, disculpe las molestias ocasionadas
12	Su farmacia no posee repartos designados, por favor comuniquese con D.U.S.A.
13	No es posible confirmar su pedido, debe de asignarle un reparto
14	No es posible confirmar su pedido, debe agregar al menos un producto
15	El usuario ingresado no posee una dirección de correo
16	Ha ocurrido un error con el envío de correo electrónico, disculpe las molestias ocasionadas
17	Error al obtener subtotales

8. Diseño Configuraciones para Administrador Sistema

En la próxima entrega de este documento, se redondeará los archivos de configuración para el Administrador del Sistema, es de interés dejar constancia que estos archivos sean contemplados en la próxima versión de este documento.

9. Solución Problemas Concurrentes Acceso a Base de Datos

En la próxima entrega de este documento, finalizada Fase de Construcción se anexará en esta sección un detalle de la solución a los accesos concurrentes a la Base de Datos, donde se optó por utilización de semáforos y threads adicionales.