

# Documentación Arquitectura Software

## Descripción

Documento que tiene como motivación plasmar el diseño arquitectónico del sistema de farmacias que vamos a implementar para la práctica 2 de Desarrollo de Software.

Fecha	Revision	Autores
18/04/2016	v1	Román Arranz Guerrero Hugo Maldonado Cózar
10/04/2016	v2	Román Arranz Guerrero Hugo Maldonado Cózar

# Tabla de Contenidos

1. Introducción
  - 1.1. Proposito del documento
  - 1.2. Contexto
  - 1.3. Definiciones, Acrónimos y Abreviaciones
  - 1.4. Referencias
2. Representación Arquitectónica
3. Objetivos y restricciones arquitectónicas
4. Casos de Uso
  - 4.1. Casos de uso importantes para la arquitectura
  - 4.2. Breve descripción de los casos de uso
5. Lógica del sistema
  - 5.1. Capa de aplicación
  - 5.2. Capa de Dominio
  - 5.3. Capa de fuente de datos
6. Procesos
7. Despliegue
  - 7.1. Cliente de la aplicación web
  - 7.2. Servidor web
  - 7.3. Base de datos
8. Implementación
  - 8.1. Estructura organizativa de los ficheros
  - 8.2. Estructura de directorios del código
  - 8.3. Organización de los archivos de prueba
  - 8.4. Archivos SQL
  - 8.5. Resumen detallado
9. Incidencias de concurrencia
  - 9.1. Gestión optimista de la concurrencia
10. Patrones
  - 10.1. Factoría
  - 10.2. Visitante
  - 10.3. Bridge
  - 10.4. Delegación
  - 10.5. Inmutable
  - 10.6. Fachada
  - 10.7. Singleton
11. Modo de Uso

# 1. Introducción

## 1.1. Propósito del documento

Este documento señala el esquema de la arquitectura del sistema de la aplicación web PharmacyS.

Se han utilizado diferentes vistas de arquitectura para ilustrar los distintos aspectos del sistema.

Este documento también presenta las decisiones arquitectónicas importantes que se realizan en el sistema.

## 1.2. Contexto

El alcance del documento es describir los objetivos y limitaciones arquitectónicas, la vista de casos de uso, la vista lógica, la vista de proceso, la vista de implementación y la vista de implantación. Estas son las vistas basadas en el modelo 4+1 de vistas de Krutchen.

En cuanto al contexto de la aplicación el sistema que se ha solicitado es una aplicación para reservar medicamentos que figuren en el inventario de una farmacia. Se pide incluir geolocalización de las farmacias para seleccionarlas y poder visualizar sus medicamentos, además se pide realizar operaciones de gestión a través del navegador.

## 1.3. Definiciones, Acrónimos y Abreviaciones

- UML - Unified Modeling Language
- WEAA - Web Enterprise Software Architecture Patterns
- AJAX - Asynchronous Javascript and XML
- JSON - Javascript Object Notation
- REST - Representational State Transfer
- GUI - Graphical User Interface
- UX - User Experience
- JS - Javascript
- HTML - HyperText Markup Language
- Visa Arquitectónica - "Vista de la Arquitectura Software de un Sistema dada una perspectiva. Se centra principalmente en la estructura, modularidad, componentes esenciales, y el principal control de los flujos" (Larman).

## 1.4. Referencias

- Larman, Craig, Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd ed, Upper Saddle River: Prentice Hall PTR, 2005.
- Eden, A. and Kazman, R: Architecture Design Implementation: On the Distinction Between Architectural Design and Detailed Design.

## 2. Representación Arquitectónica

La arquitectura para PharmacyS es representada usando las vistas definidas en el modelo “4+1” de Kazman, pero usando la convención de nombres RUP. Las vistas usadas en el documento de la aplicación PharmacyS son:

Vista de Casos de Uso	
<b>Audiencia</b>	Todos los stakeholders
<b>Area</b>	Describe el conjunto de escenarios y/o casos de uso que son críticos para la arquitectura
<b>Artefactos Relacionados</b>	Documento de casos de uso

Vista Lógica	
<b>Audiencia</b>	Los diseñadores de PharmacyS
<b>Area</b>	Requisitos funcionales que proporcionan la operatividad del sistema.
<b>Artefactos Relacionados</b>	Diagrama de Paquetes

Vista de Procesos	
<b>Audiencia</b>	Integradores del Sistema
<b>Area</b>	Requisitos no funcionales que describen el diseño concurrente y aspectos de sincronización. Como se asocian los componentes de la vista lógica a los procesos.
<b>Artefactos Relacionados</b>	Diagrama de Procesos

Vista de Implementación	
<b>Audiencia</b>	Programadores
<b>Area</b>	Componentes software. Describen la estructura organizativa de los archivos, librerías, test unitarios etc...
<b>Artefactos Relacionados</b>	Diseño de la jerarquía de directorios.

Vista de Implantación	
<b>Audiencia</b>	Ingenieros de sistemas
<b>Area</b>	Topología. Describe el mapeo del software en el hardware y muestra aspectos de sistemas distribuidos.
<b>Artefactos Relacionados</b>	Diagrama de Despliegue UML

### 3. Objetivos y restricciones arquitectónicas

Las elecciones arquitectónicas se han hecho considerando los siguientes objetivos y restricciones:

- El objetivo de SOEN 343 y SOEN 344 de la WEAA (por ejemplo, Controlador Frontal, Mapeo de Datos, TDG, Mapa de Identidades, Comandos Frontales) y las vistas arquitectónicas (por ejemplo el modelo 4+1 ). El sistema PharmacyS se ha basado en el diseño propuesto en las transparencias de la asignatura.
- La Autenticación de Usuarios tomará siempre lugar antes que cualquier otro acceso a los datos en PharmacyS. Completando los aspectos de seguridad de los datos a los accesos no autorizados.
- Los requisitos funcionales y no funcionales se mantendrán desde el inicio del proceso de desarrollo con los Casos de Uso especificados.

### 4. Casos de Uso

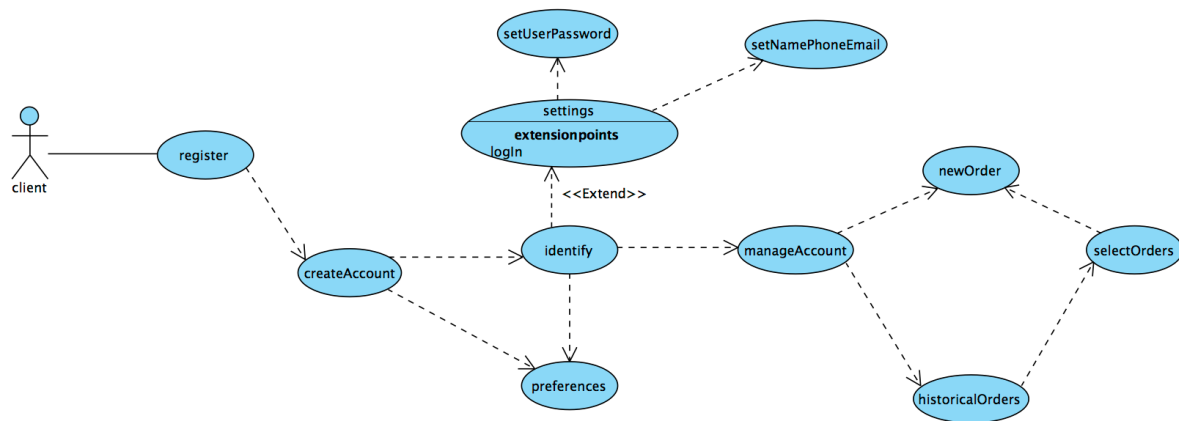
**“La vista de casos de uso describe una vista de la arquitectura del sistema que cubre el comportamiento del sistema en general desde el punto de vista de los actores involucrados”.**

#### 4.1. Casos de uso importantes para la arquitectura

El siguiente conjunto de casos de uso tienen una especial importancia en cuanto a la arquitectura del sistema PharmacyS:

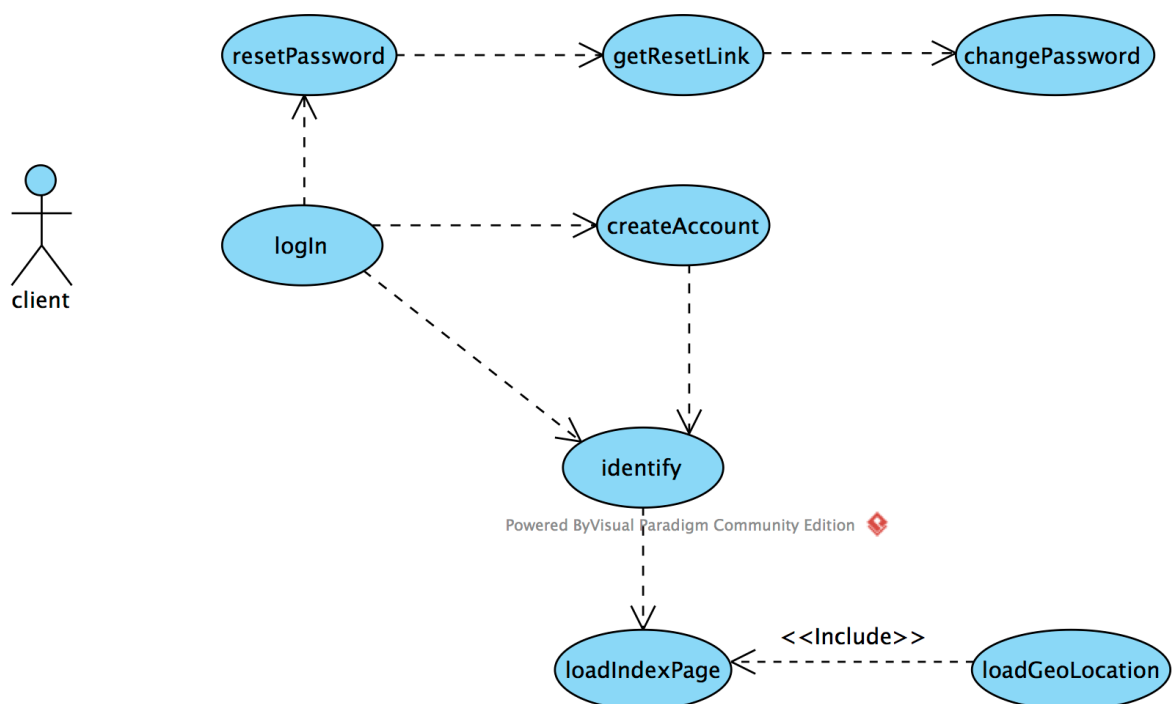
- **UC 4.1** Registrar Usuario
- **UC 4.2** Iniciar Sesión
- **UC 4.3** Cesta
- **UC 4.4** Realizar Pagos por Departamento
- **UC 4.5** Mas

## UC 4.1



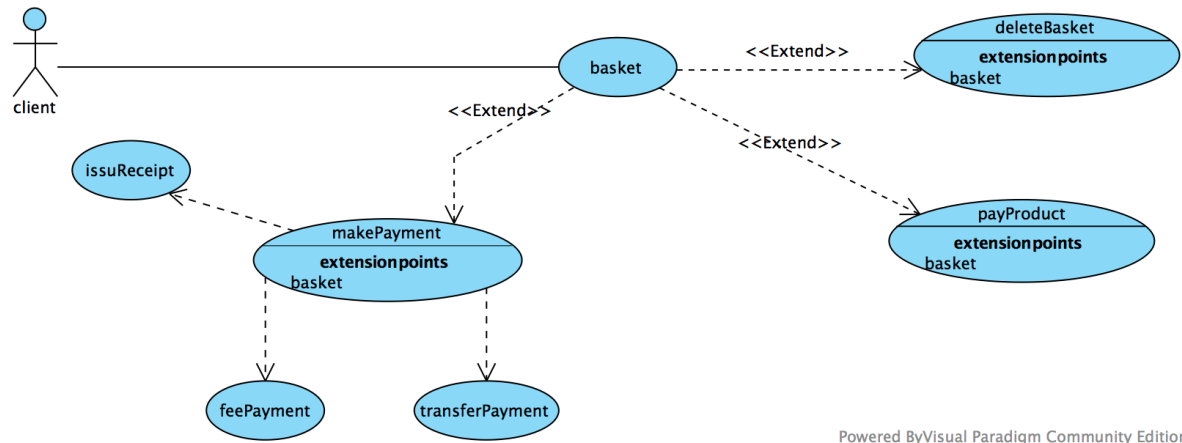
Powered |

## UC 4.2

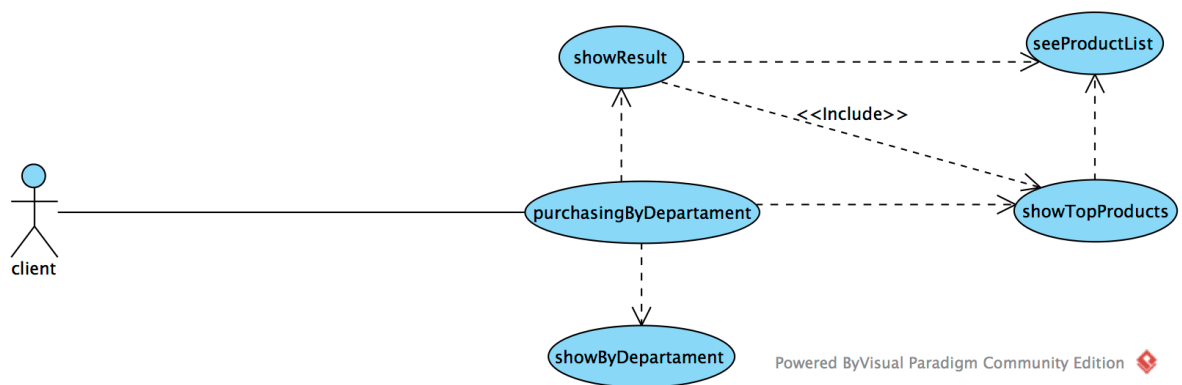


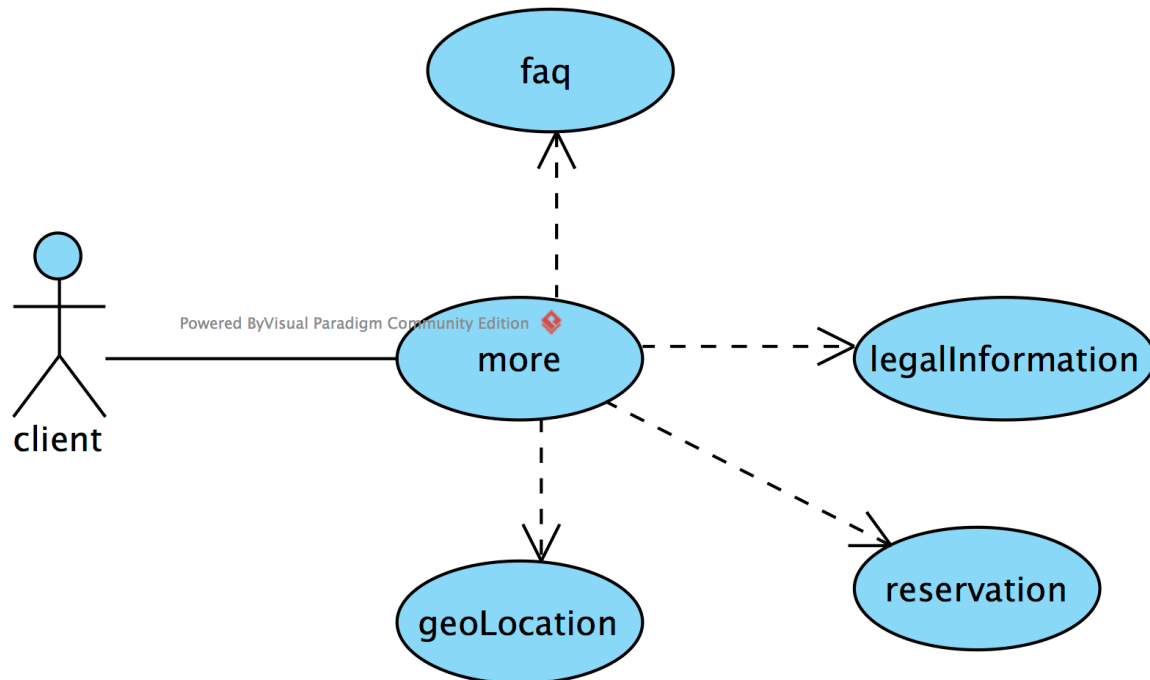
Powered By Visual Paradigm Community Edition

### UC 4.3



### UC 4.4



**UC 4.5**

## 4.2. Breve descripción de los casos de uso

- **Registrar Usuario: UC 4.1**

Un usuario que aún no se haya dado de alta en el sistema podrá registrarse en él rellenando un formulario en el que introducirá sus datos personales (perfil del cliente) así como su email y una contraseña para acceder a la aplicación.

Una vez el usuario pueda acceder al sistema se le ofrecerá la posibilidad de ver su lista de pedidos, configurar su perfil, ver lista de reservas etc...

- **Iniciar Sesión: UC 4.2**

Un usuario que ya se haya registrado previamente podrá acceder a la aplicación a través del proceso de identificación proporcionado por el servidor (aquí interviene una autenticación), en caso de que el usuario no recuerde su contraseña se le proporcionará una forma de reestablecerla enviándole un email con un link. En el caso satisfactorio de que sí consiga acceder se le servirá la página de inicio en la cual se le geolocalizará o bien podrá elegir directamente la farmacia que quiera.

- **Cesta UC 4.3**

Un usuario una vez que haya elegido los productos que quiera comprar de cada una de las farmacias que haya elegido podrá visualizar dichos productos a través de la cesta de compra, en la cual además podrá realizar cambios sobre la cantidad de dichos productos y actualizará de forma automática el precio de la orden de



compra. En el caso desfavorable de que el usuario no este interesado en uno o varios productos se le ofrecerá la forma de eliminarlos de la cesta. En cambio en el caso satisfactorio de que si quiera continuar con el proceso de compra podrá realizar la compra siguiendo los elementos de navegación que ofrece la vista de cesta.

- **Realizar Pagos UC 4.4**

Un usuario identificado podrá acceder a los productos y podrá adquirirlos por categorías. Cada producto adquirido será enviado a la cesta.

- **More UC 4.5**

El usuario podrá visualizar los contenidos de faq y licencia de la aplicación.  
En el lado servidor no seria necesario ya que el acceso es restringido solo a los farmacéuticos.

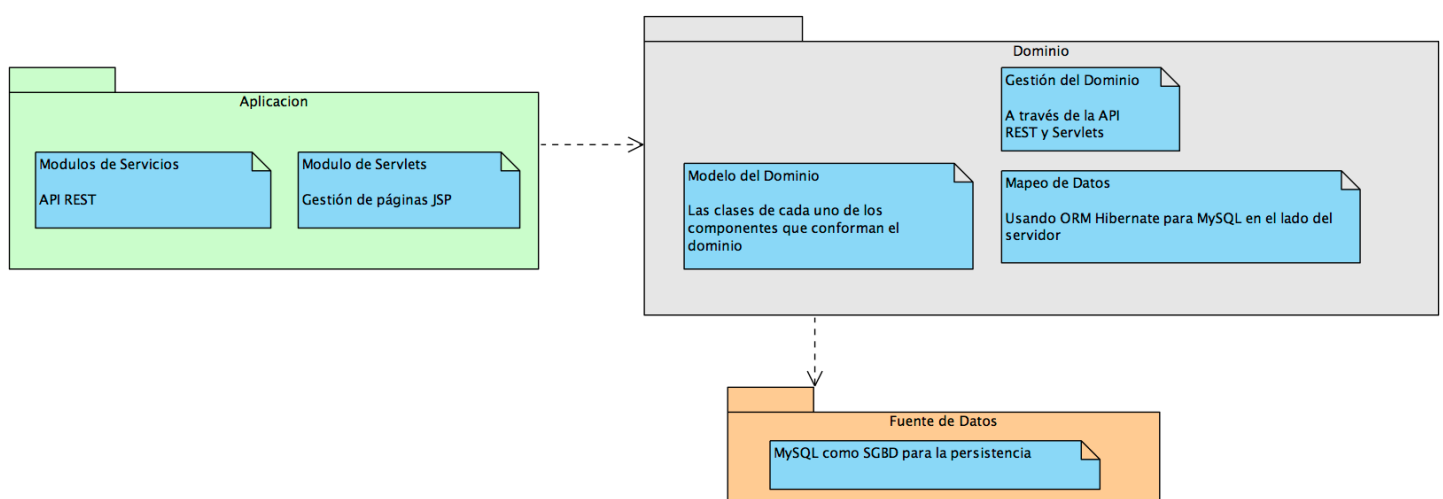
## 5. Vista Lógica

**“La organización conceptual del sistema se presenta en la vista lógica en términos de capas, paquetes, subsistemas y clases”.**

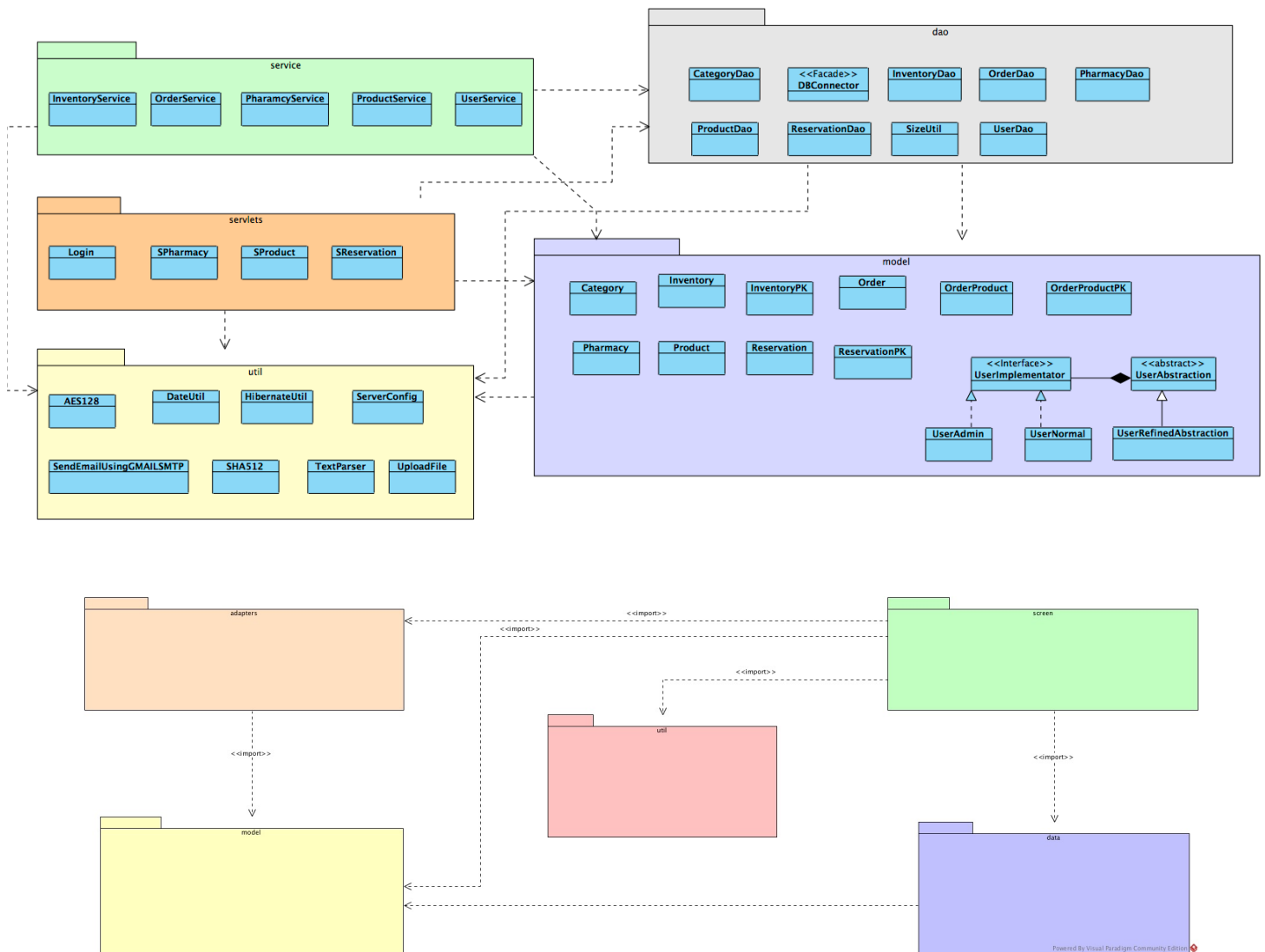
La vista lógica del sistema PharmacyS se separa en 3 capas principales: la capa de aplicación, la capa de dominio y la capa de fuente de datos.

Estas 3 capas principales se estructuran siguiendo la convención arquitectónica de 3 capas:

- La capa de presentación
- La capa de dominio
- La capa de origen de datos.



## Diagrama de Paquetes



### 5.1. Capa de Aplicación

La capa de aplicación en esta aplicación modifica el controlador frontal tradicional con la plantilla de la vista de la organización con componentes AJAX (vistas del servidor) o tareas asíncronas en caso que sean necesarias (en las vistas de la aplicación Android) según se encuentre el usuario farmacéutico en una vista u otra. El controlador frontal ahora se encarga de las solicitudes haciendo referencia a los objetos de domino desde el cliente hacia el servidor y viceversa. Los beneficios de este enfoque permiten un más robusto y más rico GUI para la aplicación, podrá comportarse con más características que una aplicación nativa de escritorio, además, como la mayoría de los datos se almacenan en SQLite de forma local, el posterior resultado se renderiza con más agilidad y le libera de carga al servidor de forma que proporcionamos más UX.

## 5.2. Capa de Dominio

La capa de dominio encapsula los paquetes y las clases para describir las reglas de la lógica del negocio del sistema PharmacyS que controlan y regulan la información del modo de procesamiento. Nos encontramos con el modelo del dominio, el de asignación de datos (dao), el mapa de identidad (model).

El modelo de dominio contiene una clase para cada una de las principales entidades: usuarios, farmacias, productos, inventario, reservas, pedidos...

También contiene una capa para hacer un uso de la unidad genérica de trabajo y un mapa de identidad.

Estos objetos de clase son utilizados por las transacciones de realizadas por el motor ORM Hibernate que hace uso de los objetos mapeados a las correspondientes columnas de las tablas de la base de datos. A su vez también tenemos estructuras de datos principales y auxiliares que nos permiten manejar los objetos y datos con facilidad.

La información acerca de una entidad se hace pasar a través de métodos por los que pasa el objeto de dominio correspondiente como un parámetro. Por cada vez que se quiera gestionar un objeto ( del paquete model) tendremos que pasar por las clases que se encargan de la gestión del objeto correspondiente (DAO's del paquete dao).

Esta conexión se hace siempre a través del DBConnector que es la fachada de todos esos pequeños gestores.

Estos gestores de datos crean una capa de direccionamiento indirecto entre la lógica de dominio y la fuente de datos. Las transacciones no saben acerca de la estructura de la fuente de datos, así como las diferentes consultas para acceder a la información.

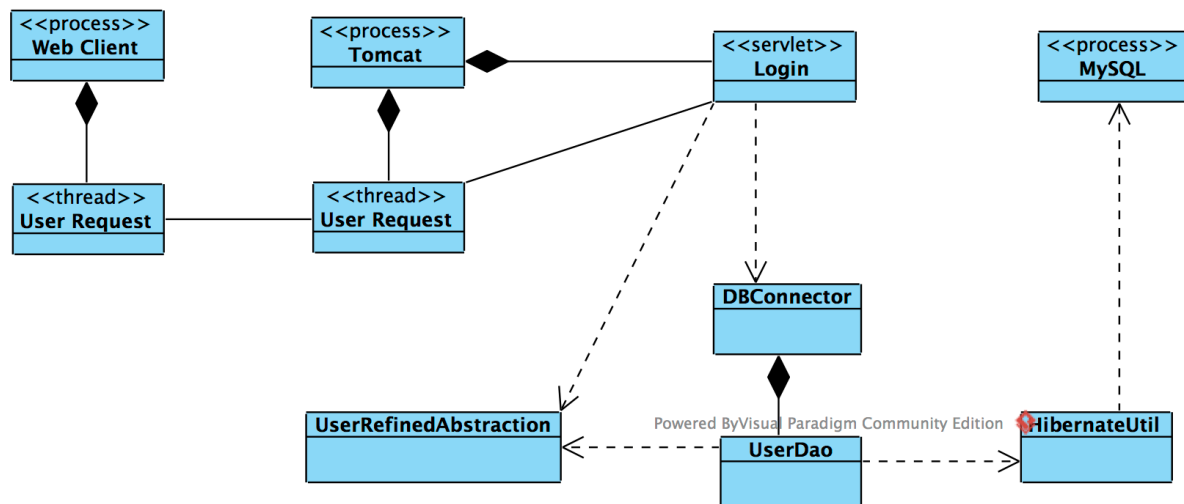
## 5.3. Capa de Fuente de Datos

La capa fuente de datos contiene las clases que definen la infraestructura de servicios técnicos utilizados para almacenar datos persistentes. Se utiliza por la aplicación PharmacyS para interactuar con la base de datos ubicada en el servidor hugomaldonado.ddns.net. El paquete incluye un archivo de mapeo por cada clase que presente en el asignador de datos (dao).

Estos objetos de origen de datos ya vinculados a las tablas son utilizados por los objetos del asignador de datos para acceder a la base de datos directamente para almacenar o recuperar información.

# 6. Vista de Proceso

"El punto de vista del proceso muestra las responsabilidades y colaboraciones entre los procesos y los hilos del sistema", así como la asignación de elementos lógicos a ellos. En esta sección, se describen las tareas, que incluyen los procesos y subprocesos que están involucrados en la ejecución y las interacciones del sistema. El siguiente modelo describe el proceso de la solicitud expresa **Ver Usuario** como procesos ejecutables junto a sus negociaciones.



El diagrama ilustra una petición que he hecho para el farmacéutico identificado en la cual obtiene los detalles de cuenta de usuario específica en el sistema PharmacyS.

El usuario lanza el proceso de su browser, el cual crea un proceso de cliente web.

Una vez que el farmacéutico se ha autenticado, aparece la página principal del usuario.

Al seleccionar la opción de vista de usuario, el proceso de cliente web crea y envía una solicitud al Servlet Login del servidor Tomcat que ha sido y sigue siendo en línea y disponible.

Este hilo petición permanece en la memoria hasta que un resultado se recibe desde el servidor Tomcat. El hilo de solicitud se pasa luego al módulo de servicio adecuado para procesamiento adicional.

## 7. Vista de Implantación

**“La vista de implantación muestra la distribución de los elementos de vista lógica para los nodos de procesamiento físico, y la configuración de red física entre nodos.” El sistema PharmacyS está diseñado como una aplicación web y está destinado a ser utilizado como un cliente / servidor. La aplicación PharmacyS se va a desplegar en el servidor web de una MiniPC GIGABYTE GB-BXBT-2807 alojada en la casa de uno de los integrantes.**

En el diagrama de despliegue de UML que hay a continuación, se presentan los elementos de red físicos involucrados en la implementación del sistema. En la siguiente sección describiremos una descripción de cada elemento de red.

### 7.1. Cliente de la aplicación Web

Los farmacéuticos utilizan un navegador web para gestionar sus farmacias, como Mozilla Firefox, y podrán acceder a un portal de gestión. La comunicación entre el cliente y el servidor está garantizada por el protocolo de red HTTP. El usuario utilizará una interfaz gráfica presentada como una página web a través del cual él / ella puede seleccionar las acciones deseadas.

### 7.2. Servidor Web

La aplicación web PharmacyS está actualmente instalada y operativa desde el servidor “[hugomaldonado.ddns.net](http://hugomaldonado.ddns.net)”. Este servidor permanece bajo el control de Hugo Maldonado Cózar y se encuentra conectado a la red con salida a internet gracias a su ISP. El servidor web es mantenido por Román Arranz Guerrero y su disponibilidad puede decirse que es de 24h salvo los cortes ocasionales producidos por la red eléctrica o por el propio ISP.

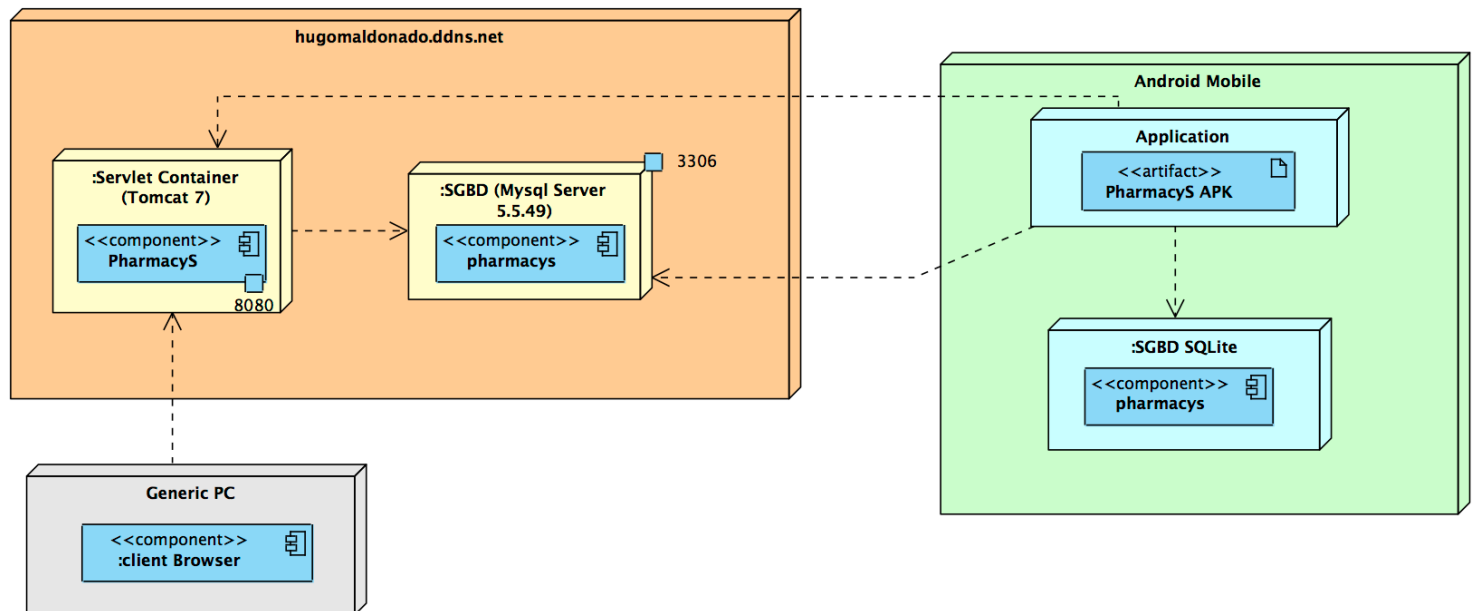
El despliegue de la aplicación Web debe ser realizado por un túnel SSH con el fin de respetar las restricciones de seguridad apropiadas en estos casos. Una aplicación cliente, tales como la FileZilla, se puede utilizar para conectarse al servidor de “[hugomaldonado.ddns.net](http://hugomaldonado.ddns.net)” y realizar las acciones necesarias despliegue de documentación (I & C).

Otras restricciones de seguridad, tales como los siguientes, pueden ocurrir durante la fase de despliegue:

- Conexiones de base de datos se restringen a nivel local (es decir, debe conectarse al servidor local)
- Archivos Servlets lecturas / escrituras se limitan a la creación de políticas de permisos de archivo del propio SO de la máquina.
- Tablas de base de datos se limitan a los archivos de comandos SQL de MySQL

### 7.3. Base de datos

Base de datos La base de datos utilizada por la aplicación web I3EM es MySQL 5. El servidor de base de datos se encuentra en el servidor Stu03 sobre un sistema operativo basado en Linux. la creación de cualquier base de datos está restringido por los archivos de comandos SQL disponibles en el SQL.



Power

Ningún usuario puede crear nuevas tablas a través del sistema I3EM. Sin embargo, cualquier persona que tenga derechos de acceso al servidor de Stu03 puede editar los archivos de comandos SQL actuales para agregar nuevas tablas, crear nuevos archivos de comandos SQL o simplemente acceder a la base de datos MySQL a través de una línea de comandos.

## 8. Vista de Implementación

### 8.1. Estructura organizativa de los ficheros

La organización estructural de los archivos del sistema PharmacyS cuelga de una carpeta raíz llamada “pharmacys” que coincide con el nombre de proyecto JAVA EE para el WebService y “pharmacys” también para la aplicación Android. Todos los archivos fuente de Java se encuentran dentro de la carpeta src para el WebService y para la aplicación Android en app/src/main/java/com.hugoroman.pharmacys.

Todos los archivos SQL para insertar una serie de datos de prueba se encuentran dentro de la carpeta `pharmacys/sql` del WebService.

En el caso de la aplicación Android no hay archivos sql sino que directamente se crean las tablas desde una clase llamada `app/src/main/java/com.hugoroman.pharmacys/data/PharmacySContract`.

## 8.2. Estructura organizativa de los directorios del código

Los archivos de código de la aplicación PharmacyS se encuentran dentro de la aplicación Android en `app/src/main/java/com.hugoroman.pharmacys/`.

Los archivos de código del WebService PharmacyS se encuentran dentro de `pharmacys/src`. Y los archivos jsp de la vista del cliente se encuentran dentro de `pharmacys/WebContent`.

Se deja un listado detallado en el punto 8.5.

## 8.3. Archivos SQL

Las carpetas contienen cada uno de los archivos SQL con los correspondientes comandos SQL. "init.sql" se utiliza para crear el conjunto de tablas de base de datos utilizados por el sistema PharmacyS. Durante la creación de las tablas estas quedan rellenas con una serie de datos de prueba.

## 8.4. Resumen detallado

### Servidor

Recurso	Comentarios
<code>pharmacys</code>	Raíz
<code>pharmacys/sql</code>	Ficheros SQL para la BD
<code>pharmacys/src</code>	Archivos fuente
<code>pharmacys/src/model</code>	El dominio de la logica, las clases
<code>pharmacys/src/dao</code>	Capa de abstracción de interfaz entre los objetos y los datos
<code>pharmacys/src/service</code>	Hooks de la API REST
<code>pharmacys/src/servlets</code>	Servlets que procesan los formularios
<code>pharmacys/src/util</code>	Utilidades usadas en el sistema
<code>pharmacys/target</code>	Classes ejecutables de java y dependencias resueltas por Maven
<code>pharmacys/WebContent</code>	Contenido del cliente web
<code>pharmacys/WebContent/META-INF</code>	Librerías asociadas

pharmacys/WebContent/WEB-INF	Configuración de los Servlets
pharmacys/WebContent/css	Hojas de estilo
pharmacys/WebContent/img	Imágenes propias de la aplicación
pharmacys/WebContent/js	Eventos Javascript
pharmacys/WebContent/assets	Librerías externas usadas en la aplicación
pharmacys/WebContent/data	Carpeta destino de la subida de archivos
pharmacys/WebContent/management	Vistas de administración JSP
pharmacys/WebContent/management/include	Fragmentos de código JSP como módulos para conformar las vistas
pharmacys/WebContent/management/include/heads	Para las cabeceras de las páginas JSP
pharmacys/WebContent/management/include/content	Para el contenido de las páginas JSP
pharmacys/WebContent/management/include/scripts	Para definir los eventos de cada una de las páginas JSP

## Aplicación Android

Recurso	Comentarios
pharmacys/app/src/main/java/com.hugoroman.pharmacys	Raíz
pharmacys/app/src/main/java/com.hugoroman.pharmacys/	Archivos fuente
pharmacys/app/src/main/java/com.hugoroman.pharmacys/model	El dominio de la lógica, las clases
pharmacys/app/src/main/java/com.hugoroman.pharmacys/data	Capa de abstracción de interfaz entre los objetos y los datos, también se incluyen clases para crear las tablas SQLite
pharmacys/app/src/main/java/com.hugoroman.pharmacys/screens	Las vistas de la aplicación android, incluyen actividades y fragments
pharmacys/app/src/main/java/com.hugoroman.pharmacys/adapters	Eventos de la aplicación que usan el patrón ViewHolder para rellenar las listas dinámicamente de productos, farmacias etc.... (todo el contenido dinámico).
pharmacys/app/src/main/res	Todos los recursos que necesita la aplicación, tales como las vistas, los colores, textos, tamaños, animaciones, sonidos, imágenes, logos



## 9. Incidencias de Concurrency

En esta sección se describe cómo se gestiona el uso simultáneo en el sistema portal web PharmacyS.

### 9.1. Gestión Optimista

Vamos en primer lugar a ilustrar con un ejemplo como se podría gestionar un conflicto:

Se tienen dos usuarios conectados con privilegios de Administrador “Admin\_A” y “Admin\_B”, dos administradores tratan de actualizar un dato relacionado con la misma cuenta de usuario. En primer lugar los Administradores se tuvieron que identificar para tener acceso al sistema.

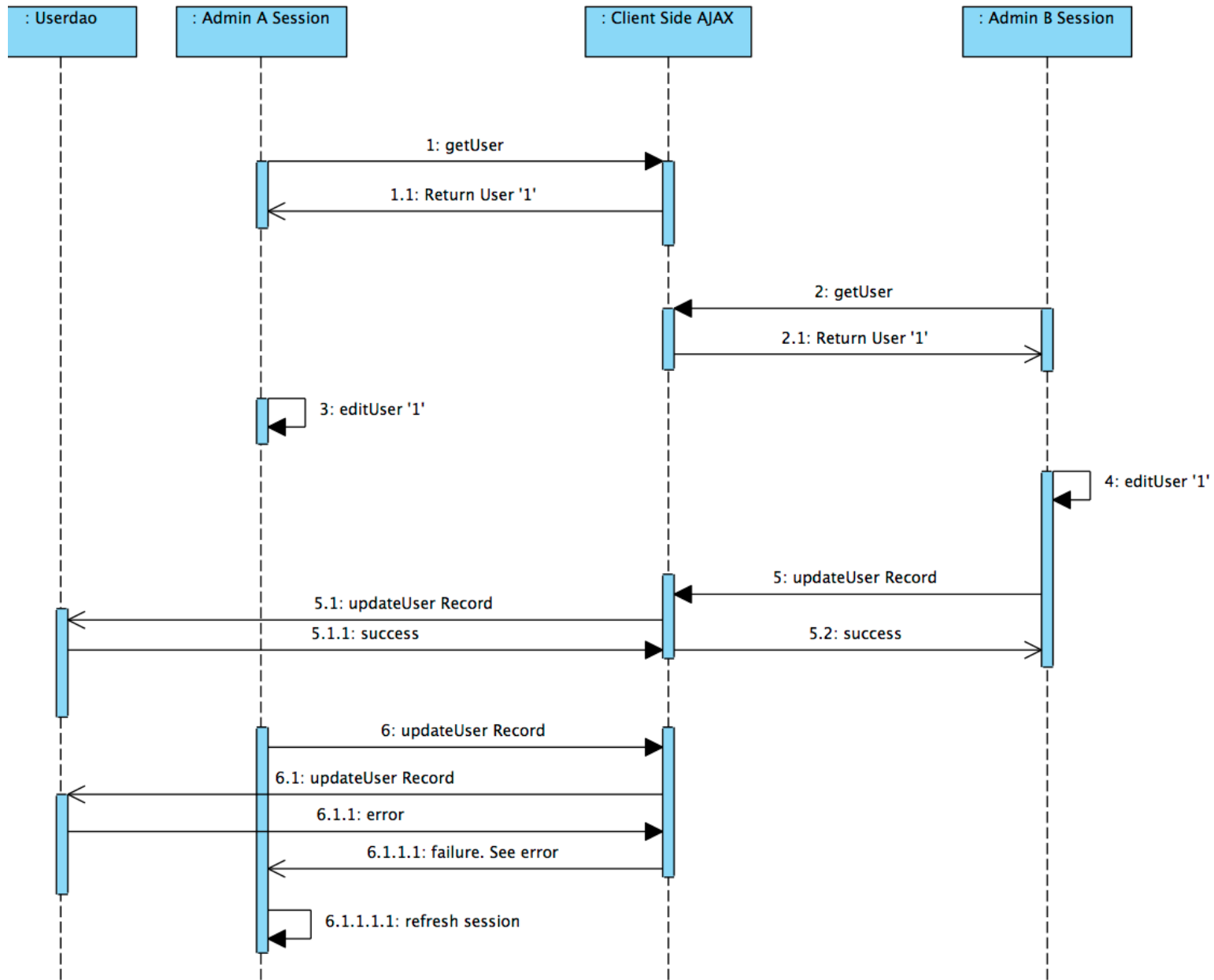
Admin\_A se encuentra administrando las cuentas de usuario y selecciona una de ellas para actualizarla. Por lo tanto Admin\_A tiene una sesión abierta e hizo un cambio sobre un usuario, esto provoca una actualización del contador de versiones en el registro del usuario.

Otro administrador Admin\_B procede a seleccionar el mismo usuario para actualizar casualmente el mismo dato.

Como Admin\_A completó la actualización, la aplicación web PharmacyS incrementa el número de versión y almacena el registro de la cuenta de usuario modificado de nuevo en la BD. Si Admin\_B completa su actualización, el sistema detectará un fallo ya que no coincide el número de la versión de referencia del registro que quiso actualizar.

En consecuencia el sistema devolverá un mensaje de error que se indica un fallo en la versión del registro. Después de esto se procederá a realizar un reajuste en la sesión del administrador.

# Gestión de la Concurrency



## 10. Patrones

### 10.1. Singleton

En la creación de ciertas clases tanto de la aplicación Android como en el servidor.

### 10.2. Bridge

Se ha implementado para las distintas implementaciones de los usuarios administrador y usuario cliente de la aplicación web en el caso de que se hubiera extendido el Web Service a una vista para clientes de consumo y no solo para farmacéuticos. Se deja implementado pero no se llega a usar ya que las vistas de cliente consumidor no se han implementado.

### 10.3. Fachada

Se ha implementado dentro del conector de la Base de Datos, tanto para el conector de la base SQLite de Android como para la del servidor.

### 10.4. Inmutable

Para los productos y las farmacias, que una vez ya creados no pueden cambiar.

### 10.5. View Holder

Para los adaptadores de carga de listas dinámicas. Se guardan en ellos la información de la interfaz de usuario correspondiente a cada elemento de la lista para agilizar el proceso y no tener que ir cargándolo dinámicamente cada vez que se “infla” un elemento de la lista.

### 10.6. Visitante

Para consultar los precios de las distintas farmacias de los distintos productos que tiene una cesta.

### 10.7. Delegación

Se delegan ciertas acciones en otras clases para agilizar el uso. Entre ellas se encuentra el PosyAsyncTask, que además, se ejecuta en segundo plano.

## 11. Modo de Uso

### SERVICIO WEB

El farmacéutico que quiera administrar su farmacia deberá hacer uso del portal de gestión que ofrece el servidor [hugomaldonado.ddns.net](http://hugomaldonado.ddns.net) en el cual deberá identificarse para acceder al panel de administración.

Una vez dentro si no añadió por primera vez su farmacia no podrá configurar su localización ni sus productos y como es obvio no podrá ver ni las reservas que le hicieron sus clientes ni las ordenes de pedido. Lo que si podrá visualizar serán las categorías que dispone su farmacia ya que vienen incluidas por defecto, la posibilidad de modificarlas y añadir nuevas no ha dado tiempo a implementarla.

También tendrá acceso siempre a la configuración de su cuenta en la cual no es necesario que haya añadido su farmacia, dentro de la configuración podrá cambiar la contraseña si lo desea.

Si ya configuró su farmacia las acciones que podrá hacer serán principalmente en cuanto a gestión de productos. Podrá añadir productos, editarlos y eliminarlos usando una sencilla interfaz.

En caso de que se le haya olvidado la contraseña y no consiga acceder al panel de administración podrá hacer clic sobre el enlace de has Forgot your password? para que se le envíe un correo con un enlace de reseteo de contraseña.

### APLICACION ANDROID

En cuanto a la aplicación Android el usuario cliente que quiera adquirir productos de las farmacias cercanas a el podrá hacer uso de la aplicación PharmacyS para desempeñar esta tarea.

Lo primero que se encuentra al abrir la aplicación es una pantalla de identificación, en caso de que el usuario no disponga de una cuenta podrá registrarse en el sistema.

La primera vez que se identifique se guardará su inicio de sesión en el Shared Preferences que Android proporciona y no será necesario volver a identificarse. Pero en el caso en el que el usuario pulse sobre el botón logout del menu se borrará sus datos de inicio de sesión y a la siguiente vez que inicie la aplicación tendrá que volver a identificarse.

Una vez que el usuario está dentro podrá desempeñar las típicas acciones que tendría un cliente en una farmacia.

Por cuestiones de diseño pensamos que tener acceso al mapa de forma ortogonal era prioritario por si el usuario quisiera cambiar de farmacia con agilidad y seleccionar los productos de otras farmacias, por ello, como pantalla principal hemos añadido una serie de accesos rápidos a las zonas de la aplicación más habituales, ordenadas, para que el cliente tenga un acceso rápido a todo lo más importante, entre lo que se encuentra el mapa para poder ver todas las farmacias.

Con esta misma filosofía pensamos que el acceso a las farmacias se podría hacer de forma ortogonal también, en caso de que hubiera muchas se paginarían los resultados (no ha dado tiempo a implementar esa funcionalidad) por lo que mostramos todas.

También tenemos acceso de forma ortogonal a la cesta, a la lista de pedidos que tenemos realizados, a las reservas, a las preferencias de usuario y al log out.

Si nos posicionamos en la vista del listado de pedidos podemos visualizar en formato de lista todas los pedidos que hemos realizados con el numero de orden de pedido y la fecha en la que los realizamos. Si pulsamos en uno de ellos visualizamos la lista de productos que conforman el pedido así como la cantidad, y si pulsamos dentro de cada producto podremos ver la información del producto, precio, descripción etc...

Si nos posicionamos en la vista a de reservas podremos visualizar una lista de reservas al igual que en el listado de pedidos, pulsando una reserva podremos ver el listado de productos que solicitamos, la cantidad que solicitamos y que no estuvo en stock en ese momento. También pulsando en el producto podremos ver la información del producto.

Las reservas hay opción de realizarlas únicamente cuando no hay stock de ese producto. Cuando pase 1 mes la reserva expirará (aun por implementar).

En cuanto al proceso de pago cuando ya hemos rellenado la cesta con los productos que queremos visualizaremos el total que tenemos que pagar y la cantidad de productos que hemos seleccionado en cada farmacia. Una vez que efectuemos el pago se nos notificara enviando PharmacyS un correo a nuestro email con el resumen de la compra.

## 12. Seguridad

Hemos implementado en la aplicación Android el acceso a los datos de la BD del Servidor mediante llamadas MySQL directamente a la BD para desacoplar aun mas a cada una de nuestras entidades, es decir, añadir como medida de seguridad para impedir que algún posible atacante pueda capturar los paquetes por la red y reconocer con facilidad el diseño del sistema. Además esto favorece en mayor medida la descentralización de los datos así como aumenta la capacidad de distribución de los componentes del sistema.

Para ello hemos utilizado el ya conocido jdbc como conector de BD.

También hemos realizado llamadas a la API REST para cumplimentar otros servicios y siempre usando POST de forma asíncrona en cuanto a manejo y ocultación de datos.

Las contraseñas siempre van cifradas con SHA512.