

Sistema de Administración de Farmacias

Prototipo Evolutivo

Versión 1.4

Historia de revisiones

Fecha	Versión	Descripción	Autor
04/09/2014	1.0	Redacción del documento.	Mª José Rabaza
05/09/2014	1.1	Agregado prototipo Web Services.	Manuela Viola
11/09/2014	1.2	Cambios en el documento.	Mª José Rabaza
12/09/2014	1.3	Agregado prototipo Creación PDF.	Manuela Viola
13/09/2014	1.4	Agregando comentarios de presentación.	Santiago Nieves
13/09/2014	1.4	Agregado objetivo del prototipo	Guillermo López
13/09/2014	1.4	Verificación del documento	Juan Pablo Sierra
14/09/2014	1.4	Revisión.	SQA

Contenido:

1. Información general	4
1.1 Requerimientos	4
1.2 Conceptos utilizados.....	4
1.2.1 Spring	4
1.2.2 Lucene	4
1.2.3 Knockout	4
2. Objetivos del Prototipo	5
2.1 UI.....	5
2.1.1 Requerimientos	5
2.1.2 Descripción.....	5
2.2 Seguridad.....	5
2.3 Buscador utilizando Lucene.....	6
2.3.1 Requerimientos	6
2.3.2 Descripción.....	6
2.4 Conexión WS DUSA	7
2.4.1 Requerimientos	7
2.4.2 Descripción.....	7
2.5 Creación del Pdf para imprimir Factura	8
2.5.1 Requerimientos	8
2.5.2 Descripción.....	8
3. Resultados Obtenidos y conclusiones	9
3.1 UI- Conclusiones Spring MVC	9
3.2 Seguridad.....	9
3.3 Buscador utilizando Lucene.....	9
3.4 Conexión WS DUSA	10
3.5 Creación del Pdf para imprimir Factura	10
4. Verificación	11
4.1 Verificar	11

4.1.1 Presentación y Spring MVC	11
4.1.2 Seguridad	11
4.1.3 Buscador utilizando Lucene.....	11
4.1.4 Conexión WS DUSA	11
4.1.5 Creación del PDF para imprimir Factura	11
4.2 Planificación	12
4.3 Ejecución de la verificación	12
4.3.1 Preparación de la base de datos	12
4.3.2 Ejecución de las pruebas	13
4.4 Conclusión	16
4.5 Consideraciones adicionales.....	16
4.6 Entorno utilizado para las pruebas	16
5. Referencias	17
6. Descargas	17

1. Información general

1.1 Requerimientos

Se debe instalar MySQL en una versión 5.6.20 y, descargar el driver o conector a la base de datos de MySQL y la librería Lucene en su versión 4.9.0. Para este proyecto utilizaremos además Netbeans 8.0 y Apache Tomcat 8.0.11. Ver sección Descargas.

1.2 Conceptos utilizados

1.2.1 Spring

Spring es un framework para el desarrollo de aplicaciones, el mismo está integrado por varios módulos. En particular se usarán los módulos de **Modelo Vista Controlador** y **Autenticación**.

1.2.2 Lucene

Lucene es una librería open source (con licencia de Apache) que nos permite integrar funciones de indexación y búsquedas de información textual dentro de nuestros proyectos.

1.2.3 Knockout

Knockout es una biblioteca JavaScript que ayuda a crear interesantes pantallas y editor de interfaces, que responden con un modelo de datos subyacente limpio. Cada vez que se tenga secciones de la interfaz de usuario que se actualizan de forma dinámica (por ejemplo, el cambio de función de las acciones del usuario o cuando cambia la fuente de datos externos), KO ayuda a implementarla de manera más simple. Por más información ver referencia [1].

2. Objetivos del Prototipo

El objetivo principal de este prototipo es construir los pilares básicos del sistema a desarrollar acorde a los requerimientos planteados [4], así como mitigar los riesgos referentes a el uso de tecnologías no conocidas.

Este producto deberá ser desarrollado en un plazo no mayor a 2 semanas, a partir del comienzo de la segunda iteración de la fase inicial. De esta manera se contaría con una base para el comienzo de la implementación del sistema en la próxima fase.

Se cubrirán los aspectos solicitados en los requerimientos: UI modular, seguridad, búsqueda y selección de productos, consulta a servidor externo y generación de archivos PDF.

2.1 UI

2.1.1 Requerimientos

Además de los especificados en la sección anterior, se utiliza los Framework bootstrap y jquery, los cuales ya están incluidos en el prototipo.

2.1.2 Descripción

Esta sección tiene 2 objetivos:

1. Investigar el funcionamiento del Framework Spring MVC.
2. Diseñar pautas y estructura base de la capa de presentación.

Por más información ir al documento de **Prototipo Evolutivo Capa de Presentación**. Ver referencia [3].

2.2 Seguridad

[Implementar la funcionalidad de autenticación de usuarios, esto incluye además, ir viendo cómo podemos implementar el acceso a funcionalidades en función de los roles de un usuario.]

Se entrega en la siguiente semana.

2.3 Buscador utilizando Lucene

2.3.1 Requerimientos

Además de los especificados en la sección 1.1, se necesitará descargar knockout-3.0.0.js y jquery-1.11.1.min.js. Ver sección Descargas.

2.3.2 Descripción

Esta sección tiene tres objetivos:

1. Investigación de la librería Knockout.
2. El objetivo principal es la búsqueda rápida y eficiente de productos sobre una base de datos de prueba. Al igual que en el prototipo "Prototipo de Uso de Lucene" se cargó una base de datos de prueba utilizando la planilla de excel que nos proporcionó el cliente.
3. Si se termina esta parte del prototipo antes del plazo establecido se agregarán alguna de las siguientes funcionalidades:
 - Paginado. Se busca mejorar la muestra de los resultados obtenidos. En caso de que éstos sean muy numerosos se realizará un paginado para no sobrecargar la página.
 - Ordenar los resultados obtenidos. El objetivo es que el cliente pueda ordenar los resultados por determinados campos para poder localizar el producto en cuestión más fácilmente.
 - Filtrar la búsqueda. Con el filtrado se busca ser más específicos sobre el campo por el cual se realizará la búsqueda. Con esto reduciremos los resultados a mostrar.

A diferencia del prototipo ya mencionado en el punto 2, éste tiene como objetivo la implementación y prueba de una aplicación web. Se seguirá usando Lucene para realizar la búsqueda.

En un comienzo se diseñará una presentación simple: un campo input para ingresar el nombre del producto a buscar, y una tabla donde se mostrarán los resultados.

El cargado de la tabla y la búsqueda del producto será de forma dinámica utilizando la librería Knockout y ajax.

Se implementará un Servlet que se encargará de la conexión a la base de datos y de la indexación y búsqueda utilizando Lucene.

Si se termina con la implementación del punto 2 antes del plazo establecido se seguirá desarrollando las funcionalidades del punto 3. Para ello:

- Los resultados deberán mostrarse paginados. Para utilizar dicha funcionalidad se agregará un sistema de flechas para poder moverse entre las páginas generadas.

- Se agregará a la página un menú desplegable para poder ordenar los resultados obtenidos. Deberá darse la opción de ordenar los resultados ascendentemente o descendientemente por nombre, precio y laboratorio.
- Luego de hablar con el cliente se decidió que la búsqueda por defecto se deberá hacer sólo por nombre y descripción. Se dará la opción de realizar la búsqueda por alguno de los siguientes campos: laboratorio, código, droga y presentación. Además se le podrá dar la opción de que busque por todos los campos ya mencionados a la vez.

En la tabla proporcionada por el cliente no se encuentran los datos de: nombre, droga y presentación, por lo cual en este prototipo sólo se buscará por código, descripción y laboratorio (de forma individual por cada campo, o todos a la vez).

Se deberá evitar la duplicación de los resultados a mostrar en el caso de la búsqueda que involucran varios campos.

Se agregará a la página web diseñada hasta el momento un menú desplegable con los campos para filtrar la búsqueda.

2.4 Conexión WS DUSA

2.4.1 Requerimientos

Además de los especificados en la sección 1.1 (Ver sección Descargas), se necesitará tener los datos de los servicios web ofrecidos por DUSA. Dichos servicios son:

- Consulta de stock: http://dev.dusa.com.uy/ws_consulta_stock?wsdl
- Consulta de comprobantes: http://dev.dusa.com.uy/ws_consulta_comprobantes?wsdl
- Pedidos: http://dev.dusa.com.uy/ws_pedidos?wsdl

Además todas las operaciones requieren de un usuario y contraseña que será distribuido a los implementadores via e-mail.

2.4.2 Descripción

En este prototipo se creará una conexión con los servicios web externos ofrecidos por DUSA con los siguientes objetivos:

1. Obtener una idea de cómo se modelan los datos e investigar qué funciones, parámetros y tipos se pueden obtener.
2. Asegurarse que se puede realizar la conexión sin ningún problema y obtener todos los datos queridos sin problemas, verificando que todos los métodos disponibles funcionen.

El prototipo consistirá primero en realizar la conexión con los servidores de DUSA para luego acceder a ellos y desplegar los datos obtenidos.

Para la primera parte, se deberán crear en el proyecto tres Web Service Client, uno por cada dirección de los web services ofrecidos por DUSA (mencionados en la sección anterior de requerimientos).

Luego, para obtener los datos se deberán investigar las operaciones que fueron agregadas por los web services y acceder a ellas a través de distintas opciones disponibles en la página principal (Por ejemplo Ver Stock, Listar Laboratorios, etc). Es importante acceder a todas las operaciones disponibles, ya que uno de los objetivos del prototipo es verificar que todos los accesos se pueden realizar con éxito.

Finalmente, para desplegar los datos obtenidos, se mostrarán en forma de tabla o de lista, según sea conveniente. En caso de que falle la obtención de datos, se desplegará el mensaje de error correspondiente.

2.5 Creación del Pdf para imprimir Factura

2.5.1 Requerimientos

Además de los especificados en la sección 1.1, se necesitará descargar la librería itextpdf-5.3.5.jar. Ver sección Descargas.

2.5.2 Descripción

El objetivo de este prototipo es investigar cómo se realiza un archivo pdf a partir de información obtenida dinámicamente y cómo se le brinda al cliente dicho pdf.

Para crear el pdf se utilizará la librería mencionada en la sección anterior y para la obtención de los datos dinámicos se utilizarán los datos brindados por el Web Service de Consultar Stock brindado por DUSA (en particular brindado por la función getLaboratorio).

La creación del archivo pdf se encontrará dentro del prototipo de la sección anterior (Web Services) y se utilizarán dos formas distintas de creación y visión del mismo.

El primer modo es dentro de Listar Laboratorios, cuando se ofrece la información de todos los laboratorios existentes en forma de tabla, hay una columna llamada "impresión" de forma que si se selecciona dicha columna se creará un archivo físico pdf dentro del proyecto en la carpeta pdf y se abrirá dicho archivo. El problema de esta modalidad es que la dirección brindada para crear el pdf es local a cada

computadora donde se ejecute el prototipo y se deberá cambiar adecuadamente, con lo cuál se decidió probar la segunda modalidad.

La segunda modalidad consiste de un botón en la página principal que dice "PruebaPDF" y al presionarlo se recarga la página ofreciendo el pdf (el contenido de dicho pdf se creará con la información de un laboratorio particular a elección). Para la creación de dicho pdf no se creará un archivo físico dentro del proyecto sino que simplemente se utilizará un espacio de memoria utilizando OutputStream y se mostrará directamente en el navegador.

3. Resultados Obtenidos y conclusiones

3.1 UI- Conclusiones Spring MVC

Spring MVC, además de ser un Framework muy completo facilita por completo la implementación del patrón MVC, haciéndolo extremadamente simple.

La capa de presentación quedó armada y de una forma modular para que en futuras iteraciones evolucione al producto.

3.2 Seguridad

Se entrega en la siguiente semana.

3.3 Buscador utilizando Lucene

Se decidió que al buscar, el usuario deberá ingresar como mínimo 3 caracteres. Con esta decisión reducimos los resultados obtenidos de la búsqueda a mostrar. Otro punto en la implementación relacionado a la decisión anterior fue demorar los pedidos que realizamos con ajax al servlet, con esto logramos poder capturar más caracteres de búsqueda para luego realizarla.

Se pudieron completar las funcionalidades adicionales mencionadas en el tercer objetivo de la sección 2.3.2.

Los resultados obtenidos fueron los deseados. Los filtros funcionaron bien y el sistema de ordenación funcionó correctamente. Además se comprobó nuevamente la rápida búsqueda utilizando Lucene.

La herramienta investigada (Knockout) resultó ser muy útil a la hora de implementar, simplificando la carga dinámica de la tabla.

3.4 Conexión WS DUSA

Los resultados obtenidos han mejorado un poco respecto a la semana pasada. Se ha descubierto que todas las funciones del servicio de Consultar Stock funcionan correctamente menos dos (getPreciosRecetas y getOfertas) que devuelven el error "504: Gateway Time-Out" (aunque getOfertas a veces funciona y devuelve la lista de ofertas). Además la función getStockUpdates devuelve una lista vacía de productos con lo cuál no se pudo probar la función getStock ya que se precisa el identificador de algún producto y no se encuentra ninguno.

Todas las funciones del servicio de Consultar Comprobantes y la del servicio Pedidos devuelven el mensaje de error "Not Supported Yet". De todas formas, si bien no es lo ideal, los clientes ya habían anunciado que no estaba pronto aún (estaba incompleto), pero que de todas formas servía para cumplir el objetivo 1 mencionado en el apartado anterior. Se comprobó con los Clientes que efectivamente solo estas funciones están habilitadas al momento de realizar este documento.

Se puede obtener como conclusión que el prototipo sirvió para cumplir su primer objetivo de ver cómo se modelan los datos y además se pudo comprobar parte del segundo objetivo al haber podido acceder a algunos datos ofrecidos por DUSA (se pudo verificar que se realizó la conexión con ellos exitosamente). Se quedará a la espera de que el resto de las operaciones queden habilitadas.

3.5 Creación del Pdf para imprimir Factura

Se notó que la primera modalidad mencionada en la sección anterior no es buena ya que no debería depender de cambiar el path donde se guarda el archivo pdf además de que en realidad no es necesario guardarlo sino simplemente mandarlo a imprimir.

La segunda modalidad tuvo resultados positivos ya que simplemente se muestra el pdf en el navegador y queda a disposición del cliente la opción de guardarlo donde lo desee o simplemente imprimirlo. El único problema que se notó con esta modalidad es que por la forma en cómo se realizó el prototipo no se pudo realizar de la misma forma que para la primera modalidad (a partir de hacer click en la tabla de laboratorios) ya que en el script utilizado se usaba el XMLHttpRequest quien no podía devolver un archivo pdf para abrir sino que tenía que devolver una respuesta HTTP (ver referencia [2]). Es por esto mismo que ésta modalidad se probó con un laboratorio en particular. De todas formas no creo que esto sea un problema ya que para este prototipo no se utilizó el framework SpringMVC que se va a utilizar para realizar el proyecto.

4. Verificación

4.1 Verificar

4.1.1 Presentación y Spring MVC

El responsable de verificación deberá comprobar el correcto funcionamiento de la capa de presentación con hasta 3 versiones anteriores a la actual del navegador Firefox y Chrome.

Comprobar el funcionamiento del patrón MVC implementado con spring, como se detalla en el documento **Prototipo Evolutivo Capa de Presentación**. Ver referencia [3]

4.1.2 Seguridad

Se entrega en la siguiente semana.

4.1.3 Buscador utilizando Lucene

El responsable de verificación deberá comprobar el correcto funcionamiento del módulo así como asegurar que los tiempos de búsqueda se adecuan a los requerimientos de performance solicitados. El cliente no nos supo dar una métrica para validar el requisito de búsquedas ágiles. Lo primordial es que la experiencia del cliente de farmacia sea la mejor, por lo cual la búsqueda debe ser rápida. Según los resultados obtenidos en este prototipo establecemos que la búsqueda deberá demorar a lo sumo 1 segundo.

Se deberá comprobar el correcto paginado, filtrado y ordenamiento de los resultados.

4.1.4 Conexión WS DUSA

El responsable de verificación deberá comprobar que se pudo realizar la conexión con los datos ofrecidos por DUSA y que efectivamente solo funcionan las operaciones mencionadas. También deberá comprobar que todas las funcionalidades ofrecidas han sido probadas (a pesar de que no funcionen).

4.1.5 Creación del PDF para imprimir Factura

El responsable de verificación deberá probar las dos formas de realizar el pdf y deberá comprobar que se crean y se muestran adecuadamente.

4.2 Planificación

Las pruebas correspondientes a la verificación de cada uno de los tres aspectos (Presentación, funcionamiento de búsquedas mediante AJAX y consultas a Web Services) se realizarán sobre el final de la iteración.

4.3 Ejecución de la verificación

4.3.1 Preparación de la base de datos

Para contar con el entorno necesario, lo que implica un conjunto de datos apropiados, tanto en volumen como en calidad, se solicitó a los responsables del desarrollo de los prototipos que faciliten un volcado de la base de datos con la cual han llevado a cabo la construcción de los mismos. El volcado del mismo se realiza utilizando la herramienta ya incluida en la distribución de MySQL; mysqldump.

Para importar este volcado, brindado en la forma de un archivo con extensión sql, se realiza la siguiente invocación:

```
mysqldump -u [usuario] -p prueba_lucene < archivo_de_volcado.sql
```

Se verifica que las credenciales que se utilizan en los prototipos y las que se encuentran configuradas en el equipo de pruebas son coincidentes.

4.3.2 Ejecución de las pruebas

Buscador AJAX y Lucene

Para el buscador AJAX se realizan las siguientes pruebas:

- Búsquedas con entradas vacías
- Búsquedas con entradas de menos de tres caracteres
- Búsquedas con tres caracteres
- Búsquedas con más de tres caracteres

Que se combinan con

- Selección de filtro "Código"
- Selección de filtro "Laboratorio"
- Selección de filtro "Droga"
- Selección de filtro "Presentación"
- Selección de filtro "Todo"

Que además se combinan con el uso de dos navegadores Web:

- Chrome 37
- Firefox 32

Por otra parte, en el caso que aplique, también se realiza avance y retroceso de página.

Se observa

1. El avance y retroceso de página genera que la selección del artículo actual avance en una posición.
2. Existe una cierta inconsistencia en la obtención de los resultados. Probar "ml." vs "ml. asdad"
3. Se ignora el criterio de ordenación si se altera la o las palabras claves utilizadas.

Se agrega

1. Utilizar las flechas del teclado ocasiona que la página realice un avance ("scroll") si la misma no cabe en la ventana actual.
2. No se indica el número de páginas ni de resultados obtenidos, lo que puede suponer un problema de usabilidad
3. Cambiar el criterio de filtro no actualiza los resultados, lo que puede ser negativo para la usabilidad
4. Se asumió que el filtro "Droga" hacía referencia al campo "Descripción".

Se considera

- Que las iteraciones de los dos navegadores citados son lo suficientemente breves como para no haber introducido cambios que pudieran afectar al prototipo, por lo que verificar en dos versiones adicionales de Chrome y Firefox no hubieran supuesto una mejora de la calidad de la verificación.
- Que las observaciones realizadas no son aspectos claves para los objetivos que el presente prototipo persigue.

Por lo que entonces, se decide aprobar el prototipo.

Prototipo de consumo de WebServices

Este prototipo tiene como finalidad validar que se puede establecer un vínculo con los servicios web que la droguería ha facilitado, a tal punto que sea posible:

- Establecer una comunicación con los WebServices, accediéndolos por la ubicación de recurso brindada por la droguería.
- Realizar la consulta de información, obteniendo los datos necesarios
- Comunicar al usuario los valores transmitidos en un formato que éste pueda comprender

Las pruebas que se desarrollarán sobre este prototipo incluyen:

- Listar laboratorios, que muestra una lista de laboratorios e información asociada a los laboratorios presentes en la base de datos de D.U.S.A.
- Listar ofertas, que brinda información acerca de las combinaciones para obtener descuentos
- Precios recetas, que debería retornar la información sobre los precios de las recetas, pero que se indica no estar disponible.
- Ver stock, que se indica que no será posible verificarlo en esta ocasión dado que se aclara que tampoco se encuentra funcional el servicio
- Ver tipos IVA, que muestra los distintos porcentajes de IVA
- Consultar un laboratorio, para obtener información detallada sobre el mismo.
- Exportación a archivo PDF, que permite consultar información de un laboratorio en el mencionado formato.

Se observa

- La opción de listado de ofertas, que se reporta como funcional, falló en la primer invocación con un código de error 504. Sin embargo finalizó exitosamente la segunda vez que se seleccionó.

- Los WebServices reportados como no funcionales efectivamente retornan error.
- El resto de las funcionalidades se ejecutan adecuadamente
- Los tiempos de respuesta son aceptables
- El formato de visualización es apropiado

Se agrega

- Que la ruta de generación del archivo PDF se encontraba fija en el código, y siendo la misma dependiente del usuario actual (dado que hacía referencia a un cierto usuario bajo el directorio Users), causaba que todo el proceso terminará de forma errónea. Una vez ajustada la misma al equipo de pruebas, el proceso de exportación a PDF se desarrolló satisfactoriamente.

Se considera

- Que una fracción importante del conjunto de funcionalidades no pudo verificarse
- Que la causa de las incidencias encontradas son ajenas al grupo de desarrollo.
- Que la importancia del adecuado funcionamiento de la interacción con los sistemas de la droguería son críticos

Por lo que entonces, se sugiere solicitar a la parte responsable de los sistemas de servicios web para que se realicen los ajustes pertinentes y se puedan llevar a cabo las pruebas en su completitud.

De esta forma, se rechaza el prototipo.

4.4 Conclusión

Es necesario volver a validar el prototipo en cuanto se resuelvan las incidencias reportadas.

4.5 Consideraciones adicionales

La solución de las incidencias no depende únicamente del equipo de proyecto, por lo que será necesario determinar estrategias de comunicación con la parte responsable y considerar ese aspecto dentro de la planificación.

4.6 Entorno utilizado para las pruebas

Servidor MySQL con el conjunto de datos *conjunto_datos_prototipo1.0.sql*, ejecutando sobre Windows 7.

5. Referencias

- [1]: <http://knockoutjs.com/documentation/introduction.html>
- [2]: <http://stackoverflow.com/questions/4750461/xmlhttprequest-to-open-pdf-in-browser>
- [3]: Prototipo Evolutivo Capa de Presentación
https://docs.google.com/document/d/1BzIUgxpI_7S4UKreB8nCM5JszcF5IDSmG7v2b2FFIMI/edit
- [4]: Especificación de Requerimientos de Software para Prototipo Evolutivo
<https://docs.google.com/file/d/0BwfEyVAXgpzMTzctbmpMb3h4VU0/edit>

6. Descargas

- Lucene 4.9.0
<http://download.nextag.com/apache/lucene/java/4.9.0/>
- MySQL Community Server 5.6.20
<http://dev.mysql.com/downloads/mysql/>
- Netbeans 8.0 (Versión de 210 MB)
<https://netbeans.org/downloads/>
- knockout-3.0.0.js
<http://ajax.aspnetcdn.com/ajax/knockout/knockout-3.0.0.js>
- jquery-1.11.1.min.js
<http://code.jquery.com/jquery-1.11.1.min.js>
- Apache Tomcat 8.0.11
<http://tomcat.apache.org/download-80.cgi>
- itextpdf-5.3.5.jar.zip
<http://www.java2s.com/Code/JarDownload/itextpdf/itextpdf-5.3.5.jar.zip>
- Spring Framework 4.0.1
(Incluido en la librerías estándar de netbeans 8.x)