

Release notes

<https://github.com/Lifka/Servicio-reserva-en-farmacias>

Izquierdo Vera, Javier - javierizquierdovera@gmail.com

Gallardo Morales, Juan carlos - jcgallardomorales@gmail.com

Cómo probar el sistema (sistema desplegado):

1. Descargar la App de Play Store:
<https://play.google.com/store/apps/details?id=com.red.lifka.sisfarmaapp>
2. En la App indicar como servidor:
<http://farmaciaweb.azurewebsites.net/FarmaciaServer/>
3. O también desde el cliente web:
<http://farmaciaweb.azurewebsites.net/FarmaciaWeb/>

La entrega incluye:

Android:

Cliente Android, proyecto Android Studio. Dentro se encuentra la APK firmada → Android/SisFarmaApp/app/app-release.apk.

La APP también se encuentra en la Play Store →

<https://play.google.com/store/apps/details?id=com.red.lifka.sisfarmaapp>

Al iniciar la App pide que se introduzca la URL del servidor. Por defecto viene establecida la URL de un servidor Tomcat 8.0.33 en Azure, el cual puede utilizarse para probar la funcionalidad del sistema → <http://farmaciaweb.azurewebsites.net/FarmaciaServer/>

FarmaciaWebGlassfish:

Aplicación web realizada mediante JSP y Servlets. Contiene la web principal de la farmacia, dando la posibilidad de descargar la aplicación para Android, consultar el catálogo de productos, consultar el catálogo de farmacias y administrar tanto productos como farmacias. Los servlets (paquete src / org / farmaciaWeb / servlets) son los encargados de acceder al servidor mediante llamadas REST para obtener toda la información e introducir nueva

información. Sin embargo permite la administración a cualquier usuario, debido a la falta de tiempo para controlar el acceso de un administrador (funcionalidad incompleta).

FarmaciaWebTomcat:

Proyecto similar a FarmaciaWebGlassfish pero adaptado a un servidor Tomcat. Esto se ha realizado debido a la falta de recursos para encontrar un servicio gratuito que permita implantar un proyecto con Glassfish. Es necesario tener instalado en la máquina local un servidor Tomcat (en nuestro caso hemos utilizado la versión 8.0.33). Contiene múltiples modificaciones y dependencias que Glassfish no requería. Muchos tipos de datos como los de la biblioteca `java.sql.*` dejan de funcionar y son reemplazados por bibliotecas de `java.util.*`. Las peticiones al servidor se hacen mediante el uso de `URLConnection` y el manejo de los datos se hace mediante el uso de la biblioteca JSON.

Podemos acceder a la versión implantada en Microsoft Azure (con el archivo `FarmaciaWeb.war` de la carpeta `war_files`) mediante la siguiente URL:

<http://webfarmacia.azurewebsites.net/FarmaciaWeb>

ServidorGlassfish:

Servidor restful implantado sobre Glassfish (versión 4.0, contiene todo lo necesario sobre Jersey). Únicamente está disponible para montarlo en local. Para su correcto funcionamiento debemos instalar un servidor Glassfish, abrir el proyecto y asignárselo al servidor instalado. (Todo puede hacerse desde Eclipse). Además debemos instalar una base de datos 'MySQL' ya sea local como de forma remota (los parámetros de conexión están en el fichero `'src / org / farmacia / restful / db / Conexion.java'`). Contiene toda la conexión a la base de datos, un modelo, un paquete de recursos (detectan las URIs) y un paquete de servicios el cual usará el paquete de recursos.

FarmaciaServerTomCat:

Proyecto equivalente a ServidorGlassfish adaptado al uso del servidor TomCat (v8.0.33). Múltiples correcciones respecto a usos de datos (`java.sql` deja de funcionar) y añadidas nuevas dependencias para que funcione en el servidor TomCat y para que funcione en el servidor TomCat que ofrece Microsoft Azure (necesita nuevas dependencias respecto al TomCat instalado de forma local). Se modifican asimismo las consultas a la base de datos, las cuales necesitan de una sintaxis más estricta a la utilizada de forma local. Algunas funcionalidades no han terminado de ser corregidas debido a la falta de tiempo, tales como la administración.

Podemos acceder a la versión implantada en Microsoft Azure (con el archivo FarmaciaServer.war de la carpeta war_files) mediante la siguiente URL: <http://webfarmacia.azurewebsites.net/FarmaciaServer/api/recurso>". Por ejemplo, para obtener los productos: <http://webfarmacia.azurewebsites.net/FarmaciaServer/api/productos>

Diagramas:

Se incluye el proyecto de Visual Paradigm (Versión 13.0 - Build 20160203) con los diagramas de clases y de casos de uso. También se han exportado a PDF y PNG, pero incluyen la marca de agua en consecuencia de utilizar la versión de evaluación de Visual Paradigm.

MySQL_server:

Carpeta que contiene la base de datos utilizada por el servidor en formato .sql. Además contiene un archivo leeme.txt con los parámetros de conexión a dicha base de datos implantada sobre Microsoft Azure.