

Práctica de Análisis y Diseño Orientado a Objetos

Mashup

Miguel Ángel Castillo Bellagona

Miguel Ángel López Cabana

02/05/2015

La práctica consiste en la aplicación de tecnologías de Servicios Web (REST y SOAP) y técnicas de diseño por capas para el desarrollo de una aplicación Web de tipo “Mashup” y un pequeño conjunto de servicios.

Índice

1. Introducción	3
2. Diseño.....	3
2.1 Arquitectura Global	3
2.2 Módulos	4
2.2.1 mashup-internalserver	4
2.2.2 mashup-productnews	4
2.2.3 mashup-virtualstore	5
2.2.3.1 productprovider	6
2.2.3.1.1 internalprovider	6
2.2.3.1.2 ebayprovider	6
2.2.3.2 reviewprovider	6
3. Compilación.....	7
4. Problemas conocidos	7

1. Introducción

La práctica consiste en la aplicación de tecnologías de Servicios Web (REST y SOAP) y técnicas de diseño por capas para el desarrollo de una aplicación Web de tipo “Mashup” y un pequeño conjunto de servicios.

Los servicios creados son un cliente rest para obtener las valoraciones de Facebook, un servidor rest para ofrecer al público un servicio ATOM con los últimos productos. Para nuestro servicio de productos interno hemos creado una interfaz SOAP y para obtener la información de los productos de EBay nos conectamos a su servidor SOAP.

2. Diseño

2.1 Arquitectura Global

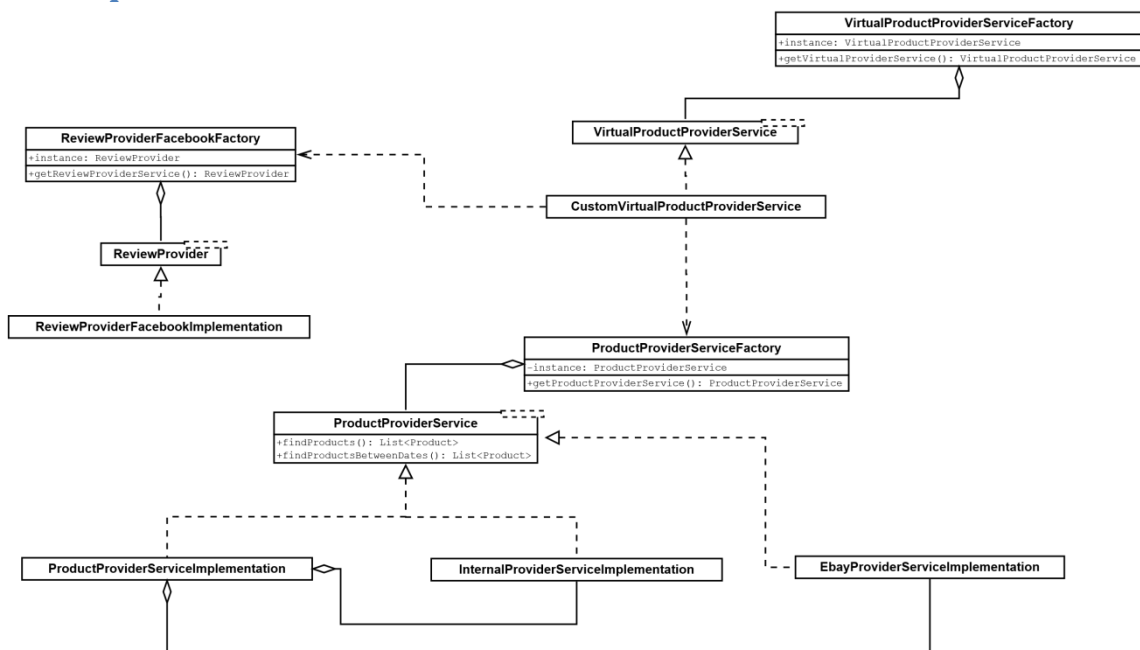
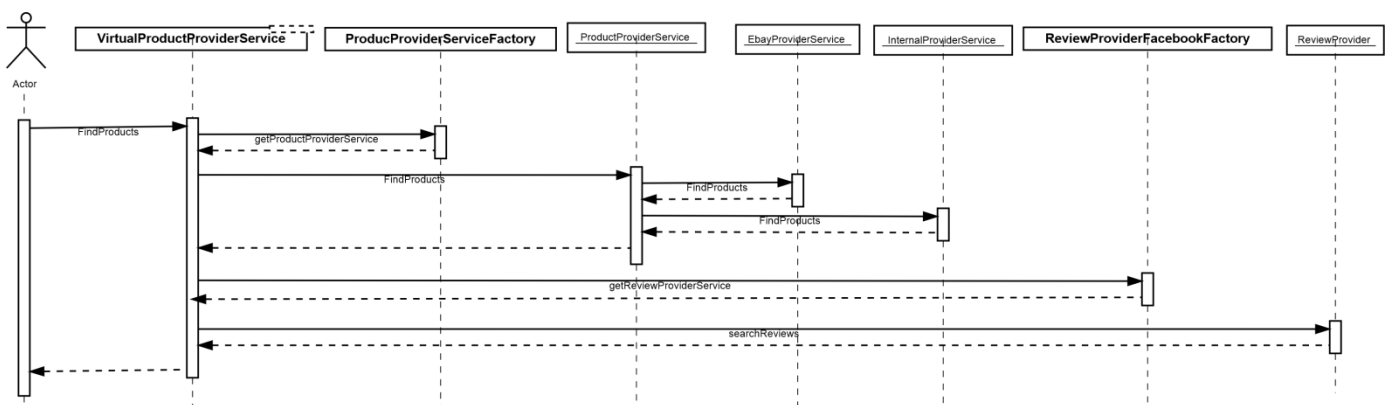
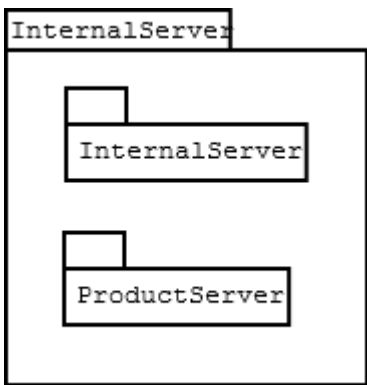


Diagrama de secuencia de la llamada a findproducts: Nos muestra como obtiene los productos y sus valoraciones.

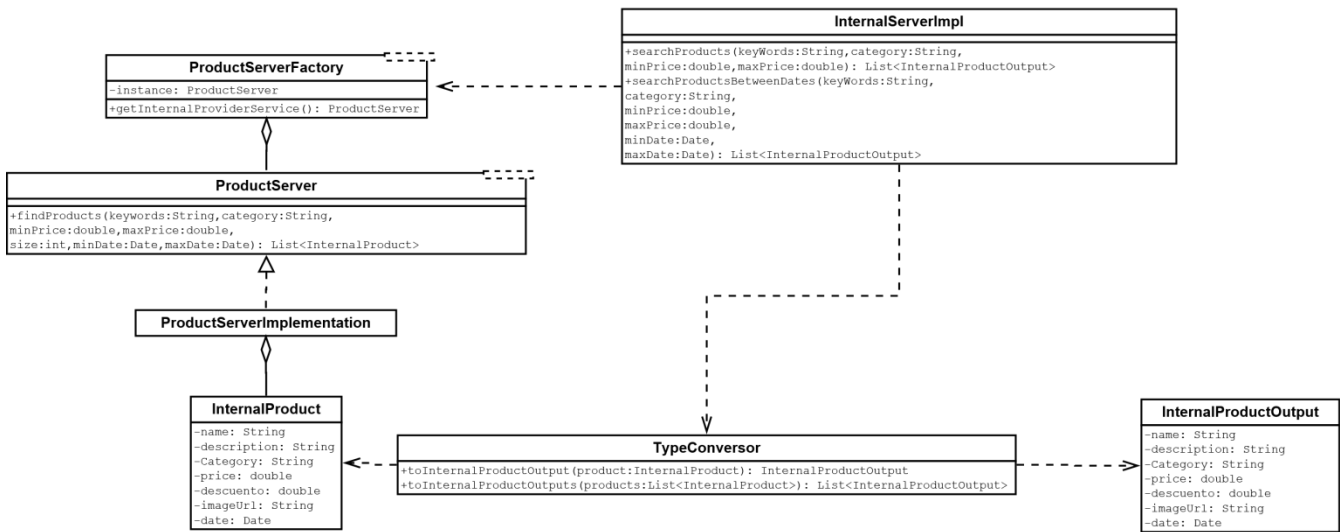


2.2 Módulos

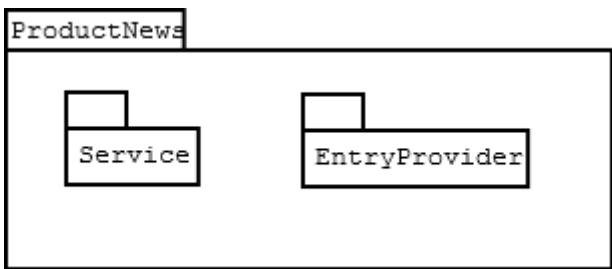
2.2.1 mashup-internalserver



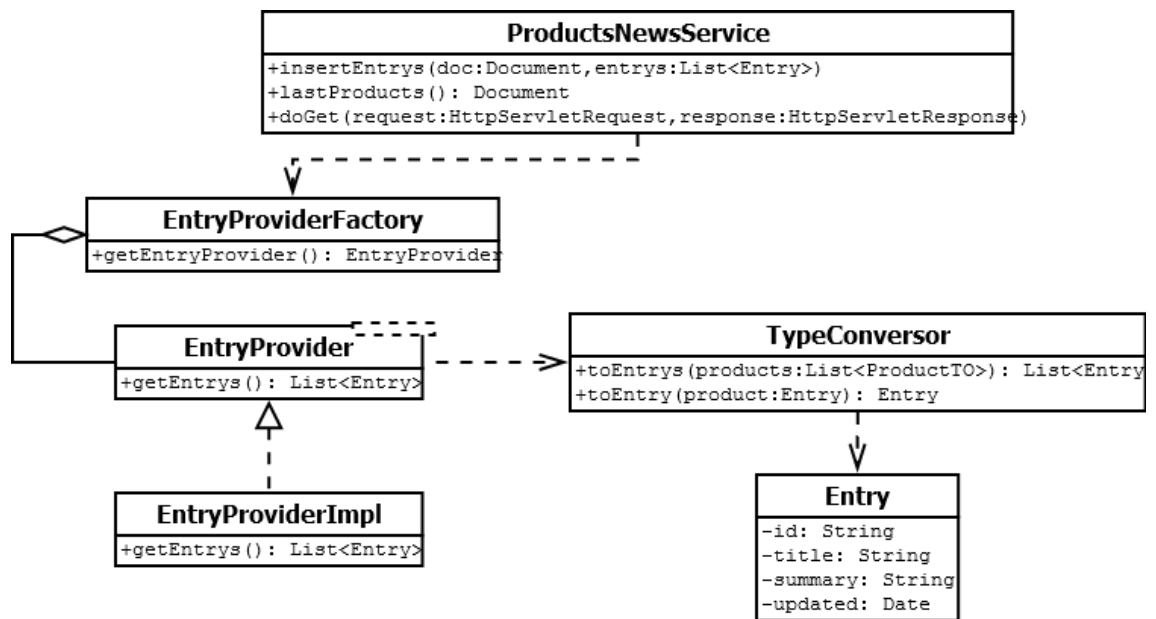
Módulo que implementa el servidor de productos interno y ofrece un servicio SOAP para hacer búsquedas en sus productos.



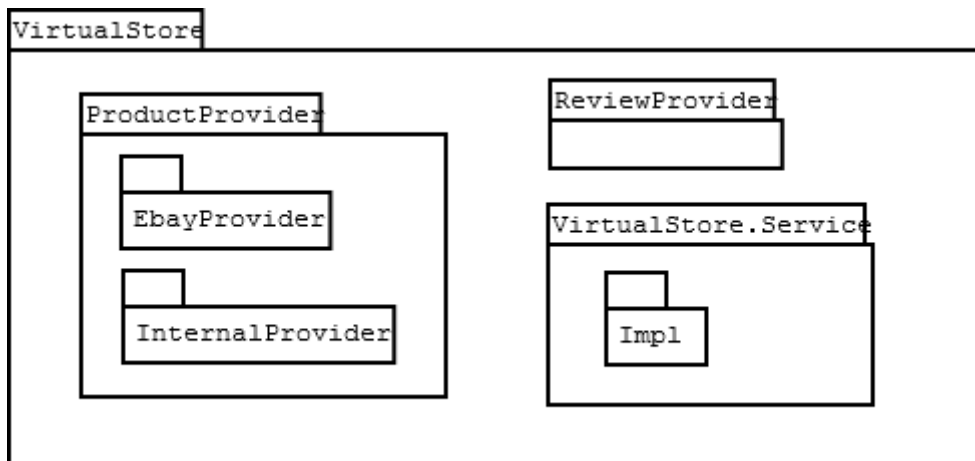
2.2.2 mashup-productnews



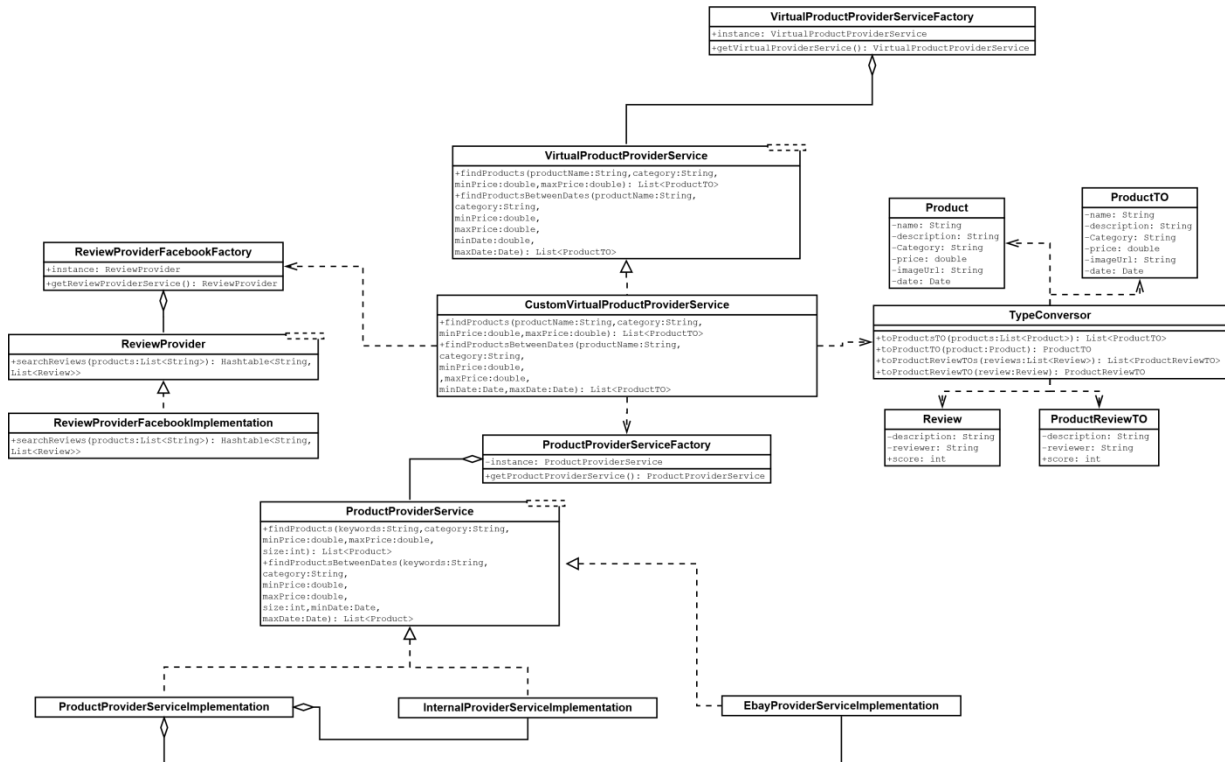
Módulo que implementa un servidor REST que ofrece un servicio ATOM con los últimos productos añadidos. Para obtener los productos hace una llamada a mashup-virtualstore. El subpaquete Service contiene la definición de la clase Entry que se usa en el servicio ProductNewsService para generar el xml del feed ATOM. En el subpaquete EntryProvider se encuentran el resto de clases que obtienen elementos externos y transforman a Entrys para devolvérseles al servicio y que trabaje con ellos.



2.2.3 mashup-virtualstore



Módulo principal del programa, es el que se encarga de obtener los productos y valoraciones desde los diferentes servicios y unificarlos.



2.2.3.1 productprovider

Módulo que se encarga de obtener los productos desde los distintos servicios y transfórmalos a un mismo tipo, esté módulo une los providers que se especifican en el archivo de configuración (independientemente de su número).

2.2.3.1.1 internalprovider

Obtiene los productos del servidor interno.

2.2.3.1.2 ebayprovider

Obtiene los productos del servidor de EBay

2.2.3.2 reviewprovider

Módulo que obtiene las valoraciones de los productos haciendo una petición REST a Facebook.

3. Compilación

El entorno Maven está configurado para que la aplicación pueda que sus diferentes módulos puedan ser ejecutados directamente mediante el uso de Jetty; en este caso los servicios mashup-internalserver, mashup-productnews y mashup-ui funcionan en puertos distintos ya que cada uno utiliza una instancia distinta del servidor de aplicaciones Jetty. Están configuradas para que se ejecuten en los puertos 8081, 8082 y 8080 respectivamente.

También se puede compilar para usarse en Tomcat, para ello hacemos un maven compile o maven install para generar los .war. Copiamos los .war en la carpeta wwapps de Apache Tomcat (en este caso se ha utilizado la versión 8.0.15 de Apache Tomcat que ya carga automáticamente dichos módulos). Si se va a ejecutar de esta forma hay que tener en cuenta que el puerto donde se ejecuta mashup-internalserver puede cambiar y que por ello puede que tengamos que cambiar la configuración de mashup-virtualstore indicándole la nueva dirección donde se encuentra ese servicio.

4. Problemas conocidos

El código desarrollado en esta práctica no tiene ningún error conocido. Dado que no era el objetivo de esta práctica y para no tener que añadir muchos productos, el servidor interno de productos se han simplificado y devuelve siempre los mismos productos, sin realizar ningún tipo de búsqueda.

El único problema conocido es que Facebook nos pide confirmación de usuario mediante SMS para confirmar que no es un robot tras múltiples usos de la clave OAuth para acceder a su servicio REST, denegando mientras el acceso a todos sus servicios. Más tarde incluso pide confirmación de identidad mediante DNI o similar, lo cual no podemos realizar al ser la página de una práctica. Por lo tanto, la parte de la práctica que accede al servicio REST de Facebook está correctamente implementada pero no funciona debido a que Facebook no nos deja acceder su servicio.