

Diseño Arquitectónico

Software de Farmacias

Emilio Chica Jiménez

19/04/2016

El presente documento intenta realizar una descripción del diseño arquitectónico de una aplicación que gestiona el consorcio de farmacias de una localidad. Dicha aplicación presenta una versión para dispositivos con el SO Android de Google y una versión web que puede ser usada por los gestores de cada farmacia para gestionar su catálogo.

Introducción al diseño arquitectónico

Inicialmente tras un análisis de los requisitos del sistema se va a necesitar una serie de componentes software que van a estar distribuidos y con un estilo arquitectónico centrado en datos (Blackboard). Con esto conseguimos que el centro de la arquitectura sea una pizarra y que otros componentes tengan acceso a él para actualizar, agregar, eliminar o consultar sus datos. Sin embargo cada componente va a requerir un estilo arquitectónico distinto.

Vamos a tener 3 componentes de software principales, el primero de ellos es una aplicación Android que tiene un estilo arquitectónico basado en capas ya que tiene una presentación, una lógica de aplicación y una consulta a datos. El segundo componente es una página web que está basada también en el mismo modelo basado en capas. Por último tenemos otro componente software que hará las veces de pizarra de datos como repositorio pasivo que será una aplicación en un servidor creada con javaREST.

A la hora de elegir el patrón arquitectónico, el software que se va a presentar en este documento está orientado con un patrón arquitectónico de cliente-servidor. En secciones posteriores se detallarán las necesidades que han llevado a que este software tenga un patrón de arquitectura tipo cliente-servidor.

Motivación del diseño arquitectónico

La elección del estilo arquitectónico basado en pizarra es debido a varios factores:

- Facilita la integración pues los componentes son independientes.
- Se puede pasar datos entre componentes a través del almacén de datos.
- Los componentes software de los que disponemos no necesitan comunicarse entre sí salvo con la pizarra.

La elección del patrón arquitectónico cliente-servidor es debido a que tenemos un componente software javaREST que está alojado en un servidor y hace las veces de proveedor de servicios que necesita el cliente, en este caso son datos de una base de datos.

Los clientes a su vez realizan peticiones al servidor de manera continuada, como clientes podemos identificar al componente software aplicación de Android y al componente software página web.

Diagrama de contexto inicial

El diagrama de la figura 1 no sigue ningún tipo de notación UML ni otro lenguaje específico, podría asemejarse a un diagrama de despliegue en el que se muestra la estructura en tiempo de ejecución del sistema, es decir, la configuración del hardware y cómo el software se distribuye en él. Se adjunta este diagrama para hacer una representación gráfica de las conexiones y elementos que intervienen en el sistema para tenerlos en cuenta a la hora de las conexiones y posibles elementos.

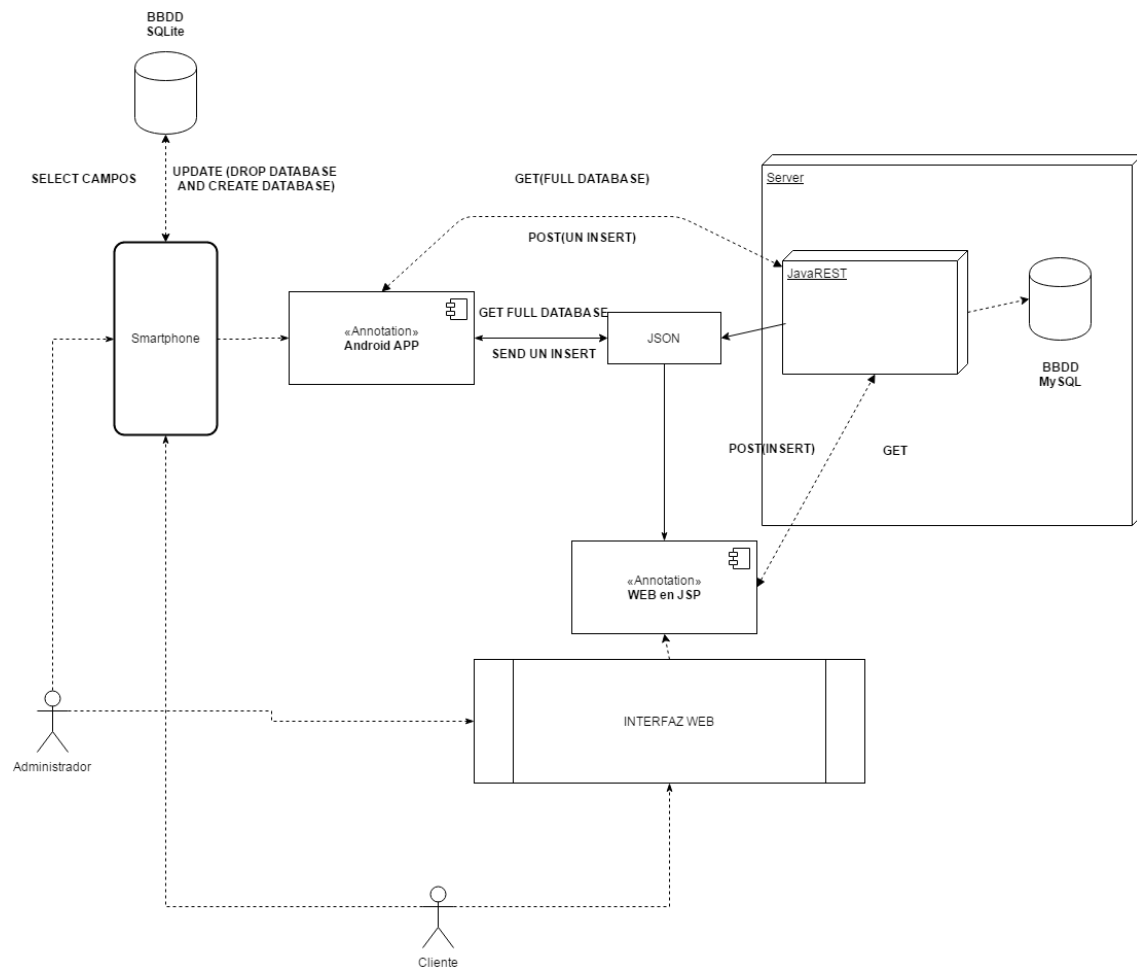


Figura 1: Diagrama de contexto inicial

Diagramas de clases

A continuación para mostrar la estructura de los componentes software vamos a utilizar diagramas de clases UML divididos en dos componentes principales la aplicación en Android y los diagramas del servidor en javaREST. Figura 2 y 3 respectivamente.

Diagrama de clases Farmacia ANDROID

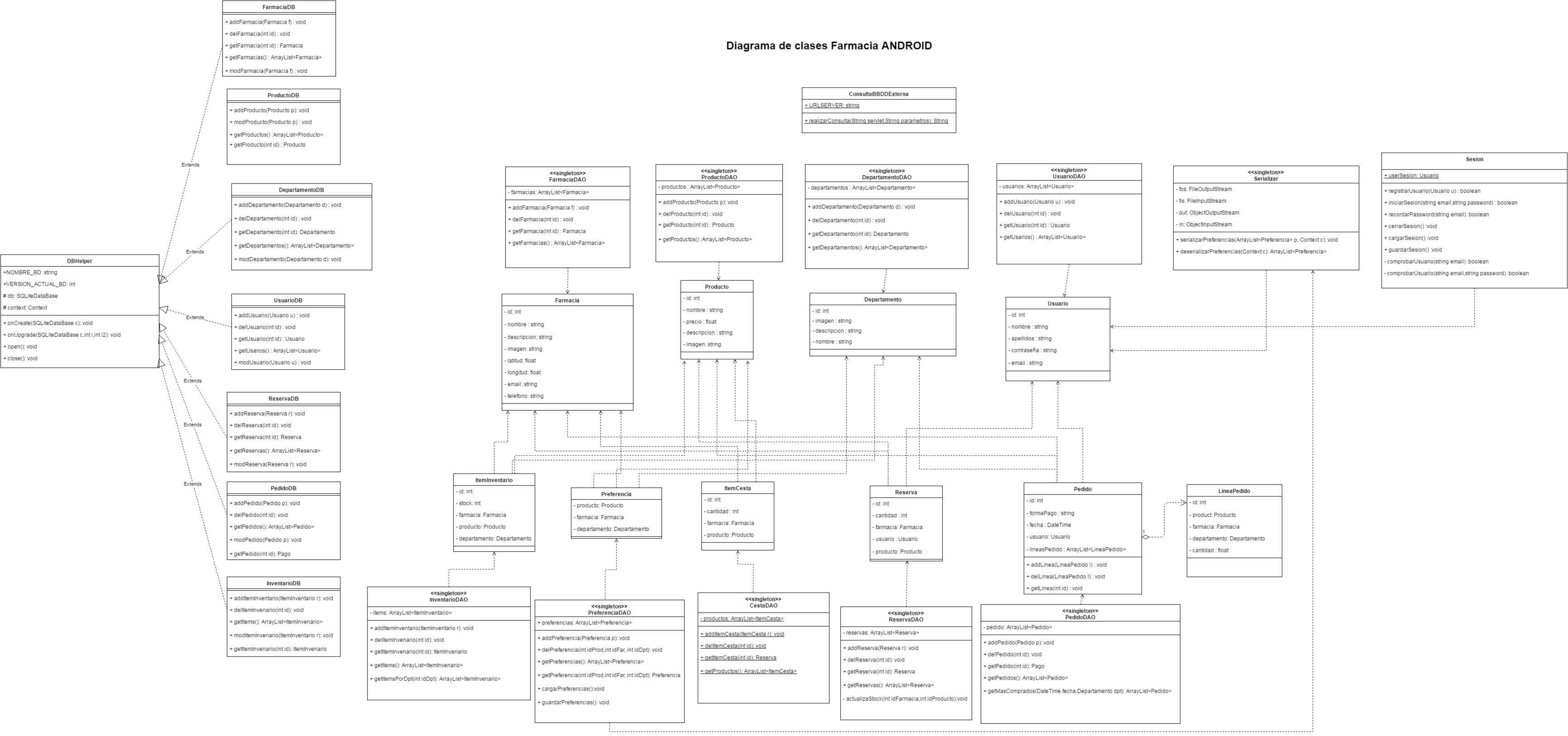


Diagrama de clases Farmacia SERVIDOR

