

# Paso a paso

## 1) Bajar y agregar al proyecto los siguientes archivos y agregarlos al package “util”:

- DAOException
  - i) Nos permite agregar excepciones (errores) de tipo dao.
- HTTPConnectionManager
  - i) Nos brinda utilidades para las conecciones, como mandarle una url y devolver el string que devuelve el servicio.
- ResultListener
  - i) Nos va a permitir parametrizar los listeners que vamos a implementar.

## 2) Agregar la librería GSON al proyecto en el gradle.

compile 'com.google.code.gson:gson:2.3.1'

# Paso a paso

## 3) Implementar el modelo MVC en el proyecto.

- Crear los siguientes Packages y clases:
  - ✓ controller (Logica de negocio)
    - (1) ProductsController
  - ✓ dao (Aca es donde obtenemos los objetos, ya sea de un archivo local, de una base de datos o a través de un servicio web)
    - (1) ProductDAO
  - ✓ model (aca van los pojos)
    - (1) Product
    - (2) ProductsContainer

# Paso a paso

## 4) Crear en el ProductDAO una clase que extienda de AsyncTask y que realice el pedido a los servicios web.

Como la tarea va a ser asincrónica, java nos ofrece este comportamiento al extender una clase nuestra de la clase AsyncTask. Cuando querramos correr algo en modo asincrónico, simplemente instanciamos un objeto de esa clase y le decimos que se ejecute (execute) y este lo va a hacer de modo asincrónico automáticamente.

Al extender la clase esto nos va a forzar a implementar dos métodos:

- doInBackground
  - Aca va el trabajo pesado que queremos que haga, en nuestro ejemplo, acá es donde hacemos el pedido de productos a los servicios web y parsear el resultado.
- onPostExecute
  - Aca ya teniendo la información tenemos que ver qué hacer con la misma.

# Paso a paso

## 5) Unir todo con listeners

Ya tenemos el objeto asincrónico que nos devuelve los productos. Tenemos que de alguna manera hacer que el controller pueda enterarse cuando este terminó de realizar el parseo. Para esto vamos a implementar un listener que va a unir el Controlador con el DAO y el objeto asincrónico.

En otras palabras, el controlador le va a pasar su listener al DAO, el DAO al objeto asincrónico y este último, cuando termine de armar la lista de productos va a llamar al listener haciendo que la cadena se complete y se ejecute el código del listener que creó el controlador.

Pasos:

- Hacer que el constructor de la tarea asincrónica reciba en su constructor un listener y que lo llame en el `onPostExecute`.
- Hacer que el DAO que crea el objeto asincrónico también reciba un listener (con el que se va a comunicar con el controller) y cree uno para poder comunicarse con objeto asincrónico.
- Hacer que el Controlador que crea el listener se lo mande al DAO por parámetro en el método.