PATRONES USADOS EN ANDROID

Control de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Modificación** | **Fecha** | **Autor** | **Versión** |
| Versión Inicial. | 02/06/2014 | Mayra Jumbo | 1.0 |
|  |  |  |  |

Índice y Contenidos

[**1.** **PATRONES DE DISEÑO UTILIZADOS EN ANDROID** 4](#_Toc389487714)

[1.1.Adapter 4](#_Toc389487715)

[1.1..1.Estructura 4](#_Toc389487716)

[1.1..2.Aplicación en el proyecto 4](#_Toc389487717)

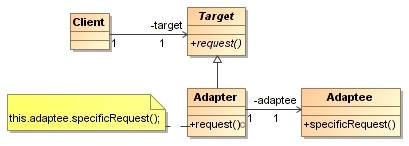
# **PATRONES DE DISEÑO UTILIZADOS EN ANDROID**

En el desarrollo de la aplicación Android se han utilizado el siguiente patrón de diseño:

# Adapter

El patrón de diseño adapter se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda.

# Estructura



Donde :

Target:- Define la interfaz específica del dominio que Client usa.

Client:- Colabora con la conformación de objetos para la interfaz Target

Adaptee:- Define una interfaz existente que necesita adaptarse

Adapter:- Adapta la interfaz de Adaptee a la interfaz Target

# Aplicación en el proyecto

Se ha utilizado el patrón adapter, para trabajar con listas (listview) en las siguientes secciones:

* Despensa: Para mostrar las listas de los productos de la despensa.
* Lista de la compra: Para mostrar los productos añadidos a una lista.
* Recetario: Para mostrar las listas de recetas.

Un adapter gestiona el modelo de datos y lo adapta a las filas individuales en la vista de una lista. El adapter extiende la clase BaseAdapter.

Cada línea en la vista de la lista consta de una disposición que puede ser tan complejo como queramos. Una línea típica en una vista de una lista tiene una imagen en el lado izquierdo y dos líneas de texto en el medio como se muestra en el siguiente gráfico:



El adapter inflará el diseño para cada fila en su método getView() y asignará los datos a las vistas individuales en la fila.

El adapter se asigna al objeto ListView con el método setAdapter().