

LIBRO DE RECETAS

Interacción y diseño de interfaces



Nombre: German Eduardo Mora Macias

Fecha: 09 de Enero de 2016

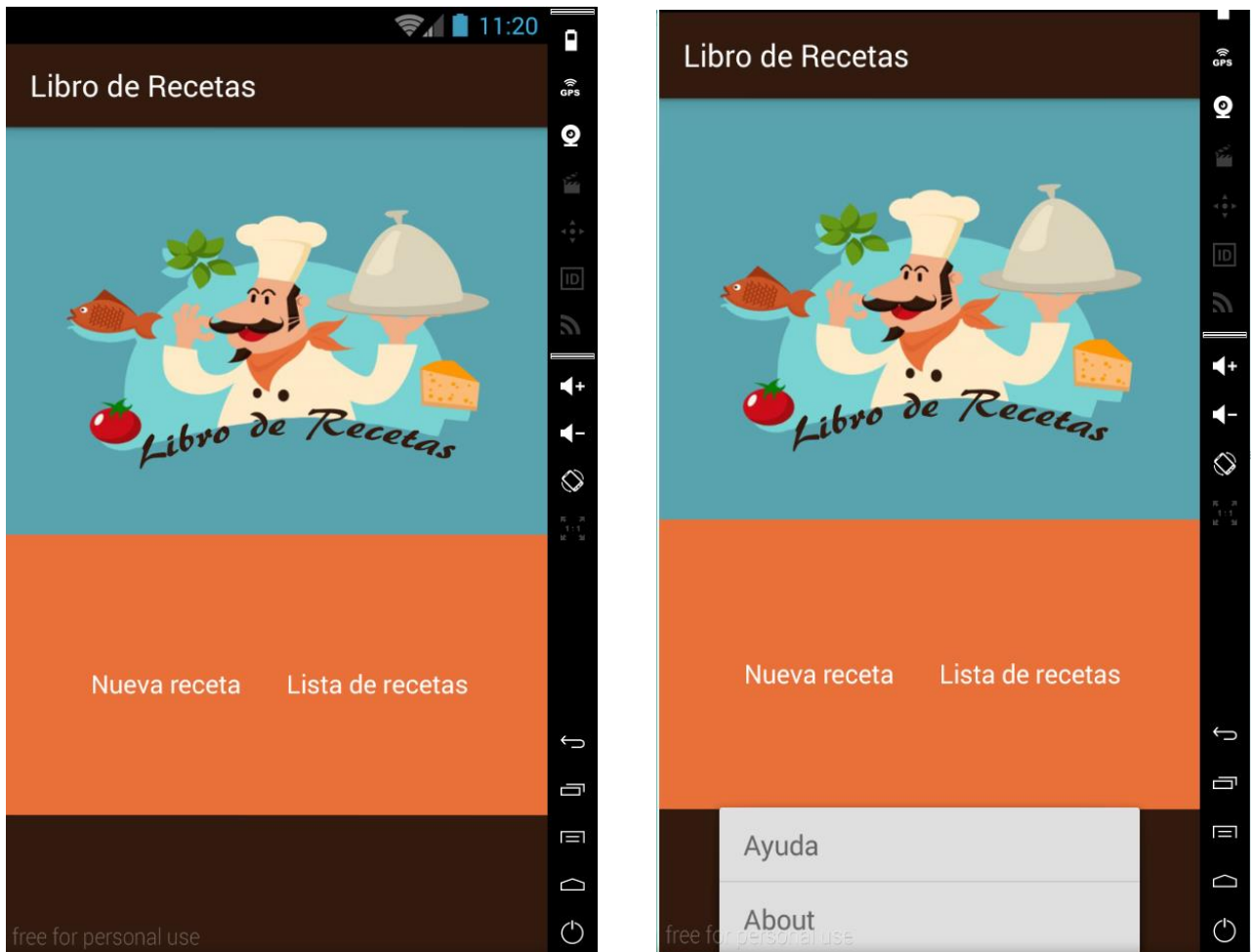
Índice

1. Descripción de la interfaz	5
1.1. Pantalla principal	5
1.2. Nueva receta	6
1.3. Listar recetas	7
1.4. Consultar recetas	8
2. Código Fuente	9
2.1. MainActivity.java	10
2.2. NuevaReceta.java	10
2.3. ListarReceta.java	11
2.4. ConsultarReceta.java	11
2.5. DbHelper.java	11
2.6. SQLiteControlador.java	12
2.7. Ingrediente.java	13
2.8. Receta.java	13
2.9. Item.java	13
2.10. ListViewAdapter.java	14

1. Descripción de la interfaz

La aplicación está constituida por un total de cuatro pantallas, cada una de estas pantallas realiza una tarea específica, en función de la actividad que realice el usuario.

1.1 Pantalla principal:



La pantalla principal, o main, está constituida por dos botones. El primer botón nos direcciona a la pantalla en la que podemos crear una nueva receta. No obstante, el otro botón, nos permite acceder a la pantalla en la que podemos listar todas las recetas que tenemos almacenadas en la aplicación. En el caso de querer consultar la ayuda o el about, es necesario acceder al menú que nos proporciona el emulador.

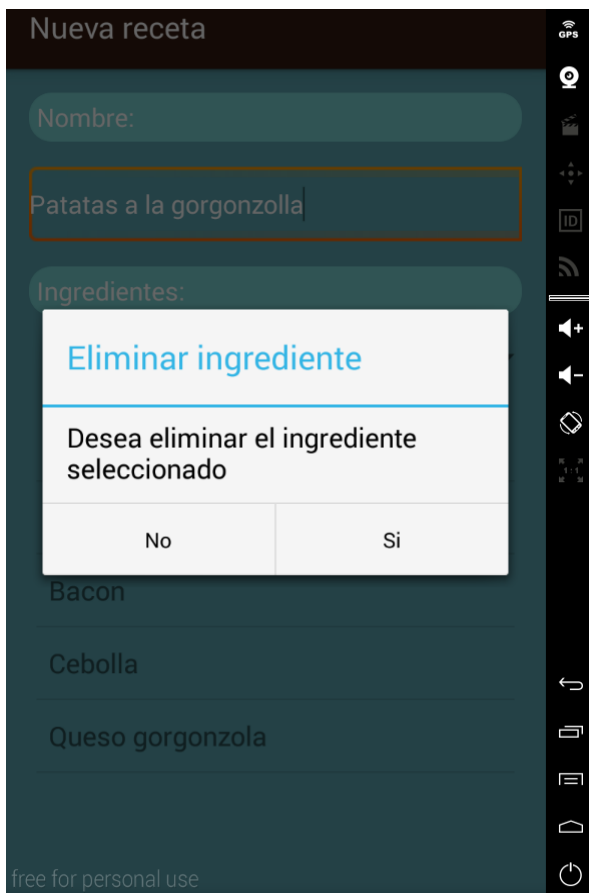
1.2 Nueva receta:

The screenshot shows the 'Nueva receta' (New recipe) screen. At the top, there's a title bar 'Nueva receta'. Below it, a light blue rounded rectangle contains the label 'Nombre:'. Underneath, a text input field is highlighted with an orange border and contains the text 'Patatas a la gorgonzolla'. Below the input field is another light blue rounded rectangle with the label 'Ingredientes:'. Underneath this, there's a list of ingredients: 'Queso gorgonzolla', 'Queso', 'Queso gorgonzola', 'Queso mozzarella', 'Sal', 'Tomate', 'Tomate frito', and 'Vino blanco'. A vertical toolbar on the right side contains various icons for navigation and functionality.

La ventana nueva receta nos permite dar de alta o registrar una nueva receta. Para poder registrar una receta es imprescindible añadir el nombre de la receta, seleccionar los ingredientes específicos de esta, la preparación de la receta y el tipo de comida. Opcionalmente podemos agregar la imagen de la receta, que la podremos seleccionar de la galería o incluso insertarla al realizar la fotografía.

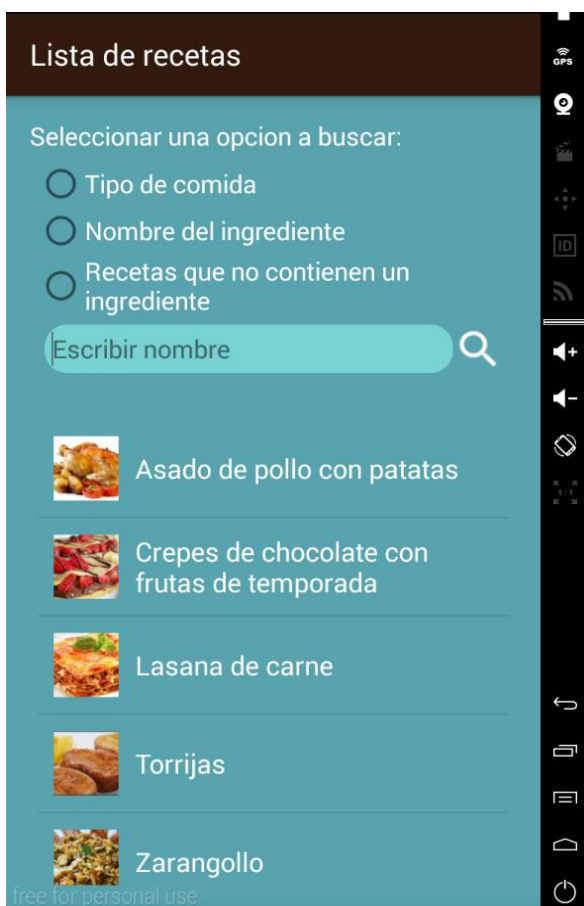
The screenshot shows the 'Nueva receta' (New recipe) screen. At the top, there's a title bar 'Nueva receta'. Below it, a light blue rounded rectangle contains the label 'Nombre:'. Underneath, a text input field is highlighted with an orange border and contains the placeholder text 'Escribir nombre'. Below the input field is another light blue rounded rectangle with the label 'Ingredientes:'. Underneath this, there's a button labeled 'Seleccionar ingredientes'. At the bottom of the screen, there's a white rounded rectangle containing three buttons: 'Guardar', 'Cancelar', and 'Ayuda'. A vertical toolbar on the right side contains various icons for navigation and functionality.

Una vez completados todos los campos de la nueva receta se deberá clicar en el botón menú y pulsar en guardar. Si no se quiere almacenar la receta, podemos cancelar todo el proceso de creación, en este caso, se restablecen todos los campos.



En el caso de haber añadido un ingrediente que no es el correcto y no se desea, tan solo debemos cliclar sobre este y nos aparecerá un mensaje en el que podemos confirmar que no deseamos ese ingrediente.

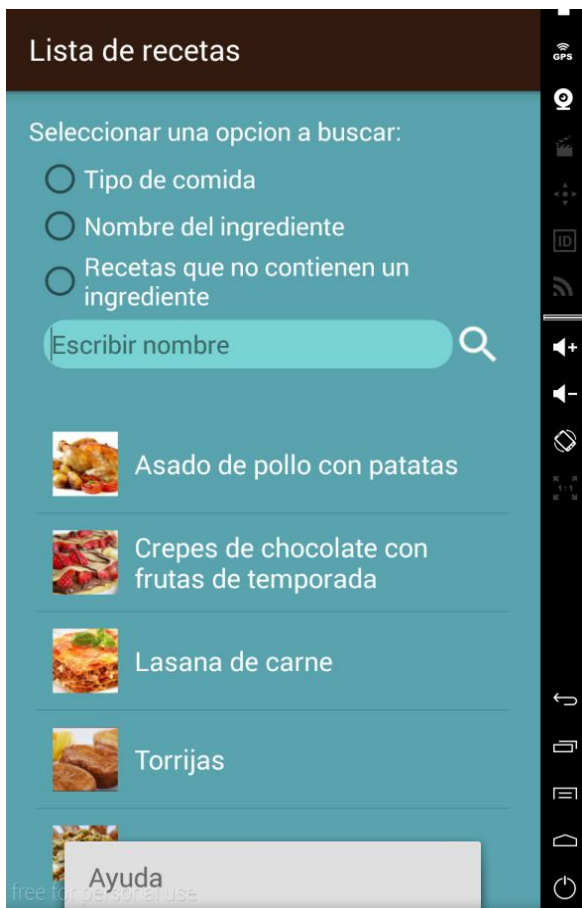
1.3 Lista de recetas:



La ventana de listar recetas, nos permite visualizar todas las recetas que tenemos registradas en el sistema.

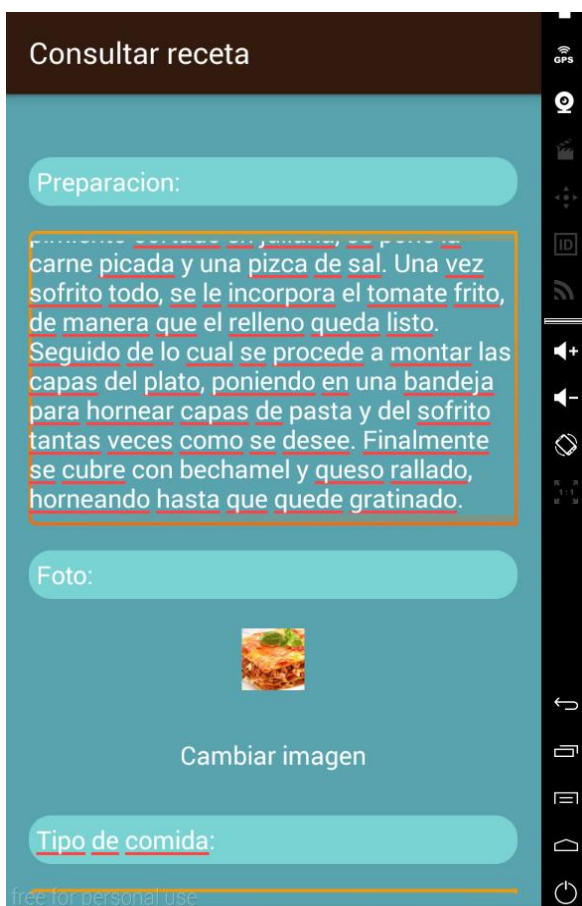
En caso de querer listar las recetas por un tipo de comida determinada, se deberá seleccionar la opción “Tipo de comida”, escribir el nombre de dicho tipo y clicar en buscar.

Analogamente debemos hacemos lo mismo para listar las recetas existentes por un ingrediente especifico, o por las recetas que no contienen dicho ingrediente.

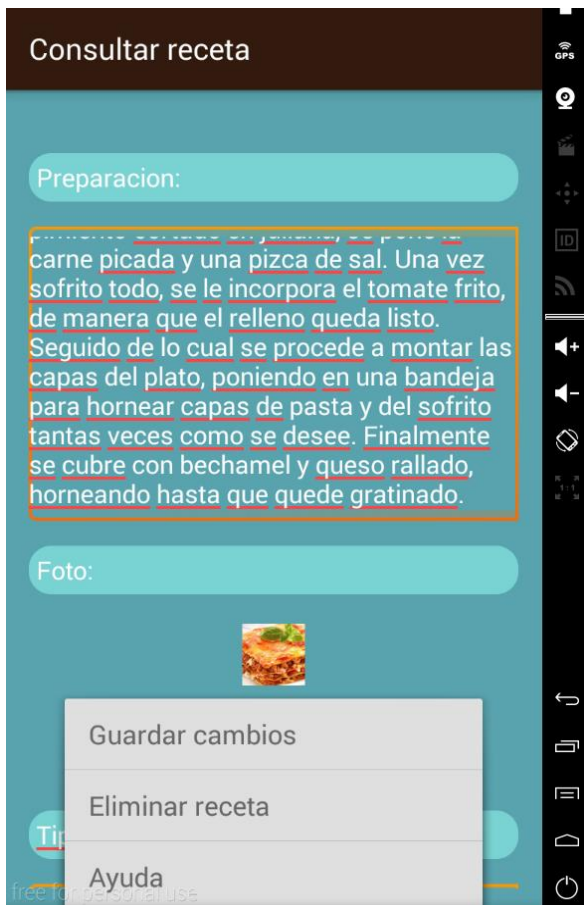


Si se necesita consultar la ayuda es necesario acceder al menú, y nos aparecerá la opción para seleccionarla.

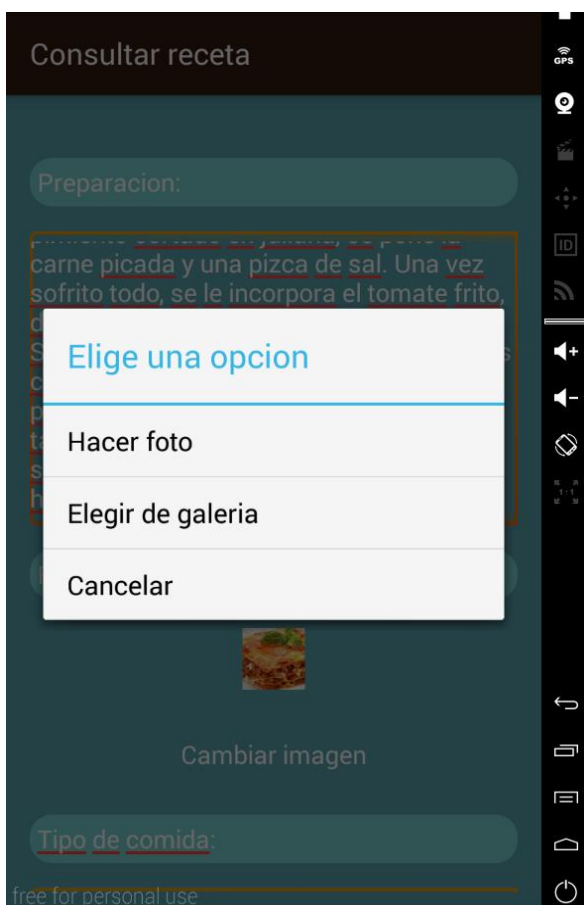
1.4 Consultar receta:



Esta pantalla nos permite visualizar y consultar todos los campos que componen una receta específica.



En el caso de haber modificado alguno de los diferentes campos de la receta, se tendrá que acceder al menú para poder guardar los cambios que se hayan realizado.



De la misma manera que al crear una nueva receta, podemos añadir o sustituir la imagen que de esta clicando el botón Cambiar imagen.

2. Código Fuente

Especificación de las clases empleadas para el desarrollo de la aplicación.

2. 1 MainActivity.java:

protected void onCreate(Bundle savedInstanceState);

public boolean onCreateOptionsMenu(Menu menu);

public boolean onOptionsItemSelected(MenuItem item):

Permite al usuario elegir una de las opciones, Ayuda o About.

public void clickNuevaR(View v):

Funcion que es llamada por el botón Nueva receta, y nos redirecciona al activity

NuevaReceta.

public void clickListarR(View v):

Funciona que es llamada por el botón Lista de recetas, y nos redirecciona al activity

ListarRecetas.

2. 2 NuevaReceta.java:

protected void onCreate(Bundle savedInstanceState);

private void listarIngredientesSeleccionados(final ArrayList<String> seleccionados):

Añade a la listView la lista de ingredientes seleccionados para mostrárselos al usuario.

private void anadirIngredientesLista():

Añade los ingredientes que han sido seleccionados del snnipper, a un ArrayList.

public boolean onCreateOptionsMenu(Menu menu);

public boolean onOptionsItemSelected(MenuItem item):

Permite al usuario elegir una de las tres opciones existentes, Guardar, Cancelar o Ayuda.

public void makePicture(View v)

Permite elegir entre tres opciones, abrir la galería, hacer una foto nueva o salir.

protected void onActivityResult(int requestCode, int resultCode, Intent data);

private void decodeBitMap(String dir):

Descodificamos el String dir, que contiene el path de la imagen, para añadirlo en el ImageView del activity.

private void openCamera():

Funcion que permite abrir desde el activity la cámara, para poder realizar una foto.

private void dissmis():

Deshace todos los cambios realizados durante la creación de la receta.

2. 3 ListarReceta.java:

protected void onCreate(Bundle savedInstanceState);

private void listarTodasRecetas():

Añade todas las recetas existentes en la base de datos, a un listView, para poder mostrárselas al usuario.

public void clickBuscarR(View v):

Permite seleccionar una de las tres opciones al usuario para poder, realizar la búsqueda en la base de datos, en función de la opción de los radioButtons.

public boolean onCreateOptionsMenu(Menu menu);

public boolean onOptionsItemSelected(MenuItem item);

2. 4 ConsultarReceta.java:

protected void onCreate(Bundle savedInstanceState);

public boolean onCreateOptionsMenu(Menu menu);

public boolean onOptionsItemSelected(MenuItem item);

public void guardarCambiosEfectuados():

Guarda todos los cambios realizados en la receta que se está consultando.

public void volverListarRecetas():

Permite volver al activity que ha llamado a esta activity, es decir, volvemos a Listar Recetas.

public void makePicture(View v):

Ídem que en Nueva Receta.

protected void onActivityResult(int requestCode, int resultCode, Intent data);

private void decodeBitMap(String dir):

Ídem que en Nueva Receta.

private void openCamera():

Ídem que en Nueva Receta.

2. 5 DbHelper.java:

Clase que extiende de SQLiteOpenHelper y permite crear la base de datos.

public DbHelper(Context context);

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion);

private void insertReceta1(SQLiteDatabase db):

Creación de la primera receta por defecto en la base de datos.

private void insertReceta2(SQLiteDatabase db):

Ídem.

private void insertReceta3(SQLiteDatabase db):

Ídem

private void insertReceta4(SQLiteDatabase db):

Ídem

private void insertReceta5(SQLiteDatabase db):

Ídem

2. 6 SQLControlador.java:

public SQLControlador(Context c);

public SQLControlador abrirBaseDatos() throws SQLException;

public void cerrar();

Función que cierra el acceso a la base de datos.

private ContentValues valoresReceta(Receta r):

Devuelve un contentValues con los parámetros de la receta r, para poder insertarla en la base de datos.

public boolean existeReceta(String nombre):

Comprobamos que no exista una receta con ese nombre en la base de datos.

public boolean insertarDatos(Receta r):

Funcion que inserta la receta r en la base datos.

public ArrayList<String> leerReceta(int idR):

Permite leer los parámetros que contiene una receta con el identificador idR.

public ArrayList<Item> listarRecetasIngrediente(String nombreI)

Función que devuelve un ArrayList con la lista de recetas que contienen el ingrediente nombreI.

public ArrayList<Item> listarRecetasNoIngrediente(String nombreI)

Función que devuelve un ArrayList con la lista de recetas que no contienen el ingrediente nombreI.

public ArrayList<Item> listarRecetasTipo(String nombreT)

Función que devuelve un ArrayList con la lista de recetas que se clasifican por e tipo de comida nombreT.

public ArrayList<Item> listarRecetas();

Lista todas las recetas de la base de datos.

public int _idReceta(String nombreR)

Función que devuelve el identificador de la receta con nombre nombreR.

public void eliminarReceta(int idR):

Función que elimina una receta con identificador idR, de la base de datos.

public int actualizarReceta(int idR, Receta newr):

Función que actualiza todos los parámetros de una receta con identificador idR, de la base de datos.

private ContentValues valoresIngrediente(Ingrediente i):

Función que crea un contentValues del objeto i, de la clase ingrediente, para poder insertarlo en la base de datos.

public void insertarIngredientes(Ingrediente i):

Función que te permite insertar un ingrediente en la base datos.

public ArrayList<String> leerIngredientesReceta(int idR):

Función que devuelve una lista de ingrediente que pertenece a una recete con identificador idR.

2.7 Ingrediente.java:

Esta clase se emplea para crear un ingrediente, que contiene el nombre del ingrediente y el identificador de la receta al que pertenece.

Ingrediente(String Nombre, int iDR):

public int getId()

public int getIdR()

public void setIdR(int iDR)

public String getNombre()

public void setNombre(String newNombre)

2.8 Receta.java:

Esta clase se emplea para crear una receta, que contiene el nombre de la receta, la preparación, el path o dirección de la imagen y el tipo de comida.

Receta (String Nombre,String Preparacion, String Path, String Tipo)

public Integer getIdR()

public String getNombre()

public void setNombre(String nombre)

public String getPreparacion()

public void setPreparacion(String preparacion)

public String getPath()

public void setPath(String path)

public String getTipo()

public void setTipo(String tipo)

2.9 Item.java:

Esta clase se emplea para poder obtener los datos más característicos de la receta, como el nombre, el identificador y el path o dirección en la que se almacena esta.

```
public Item()  
public Integer getIdR()  
public String getNombreR()  
public void setIdR(Integer idR)  
public void setNombreR(String nombreR)  
public void setPathR(String pathR)  
public String getPathR()
```

2. 10 ListViewAdapter.java:

Esta clase se emplea para poder mostrar la imagen de la receta y el nombre de esta, en el activity ListarReceta.

```
public ListViewAdapter(Context context, ArrayList<Item> receta)  
public int getCount()  
public Object getItem(int position)  
public long getItemId(int position)  
public String getItemName(int position)  
public View getView(int position, View convertView, ViewGroup parent)
```