# PORTADA

Instituto Tecnológico de la GAM.

Desarrollo de aplicaciones móviles.

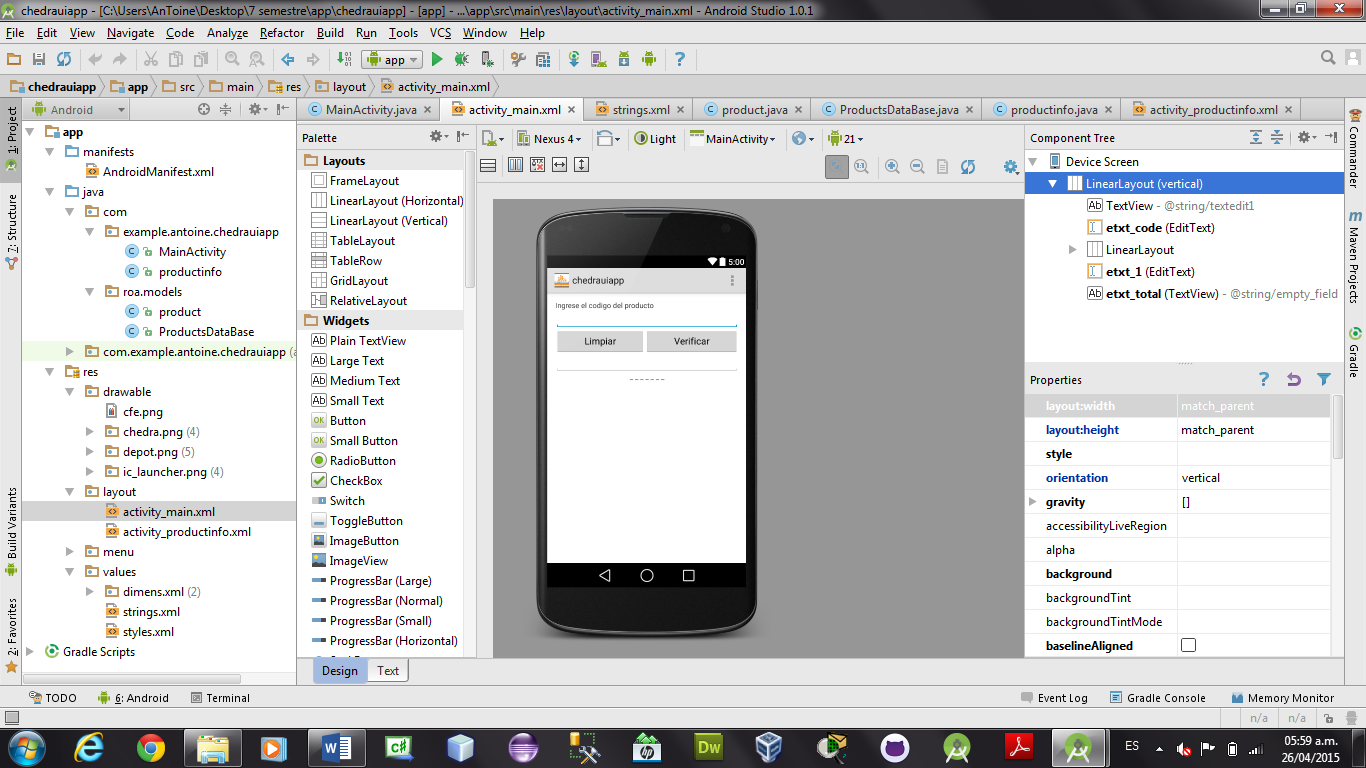
Tutor de la materia:

Jorge Iván Rivalcoba

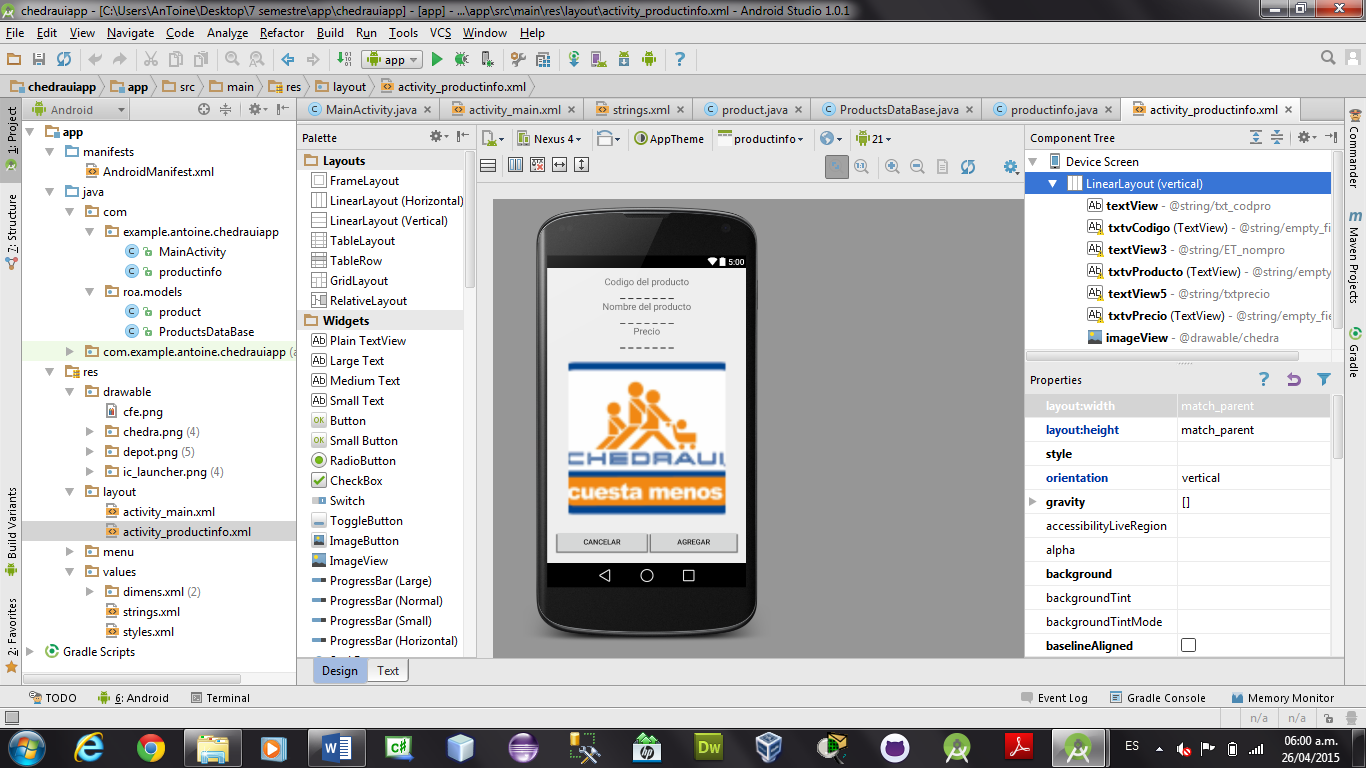
LUIS ANTONIO ROA J. FRIAS 121130033

**APP CHEDRAUI**

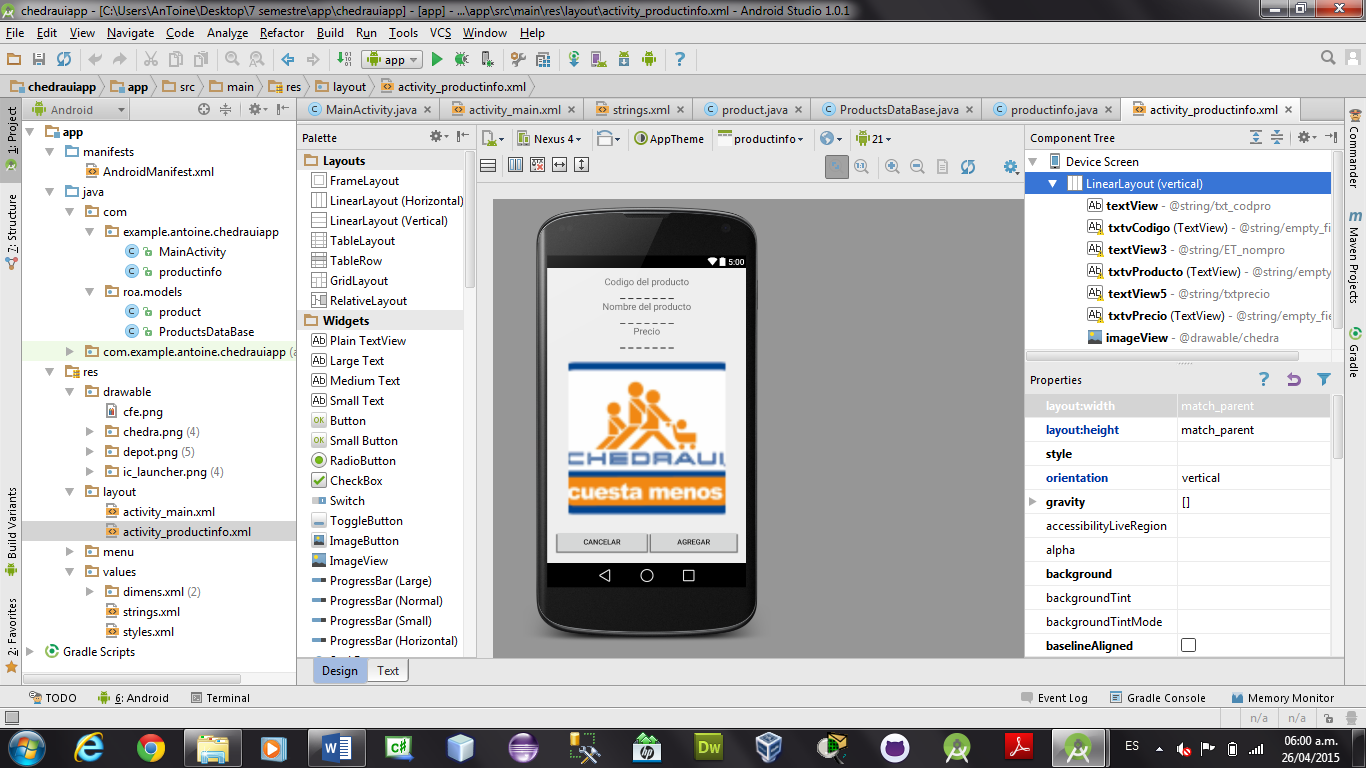
Se crea una vista en la primera actividad como la siguiente, dos botones, dos edittext



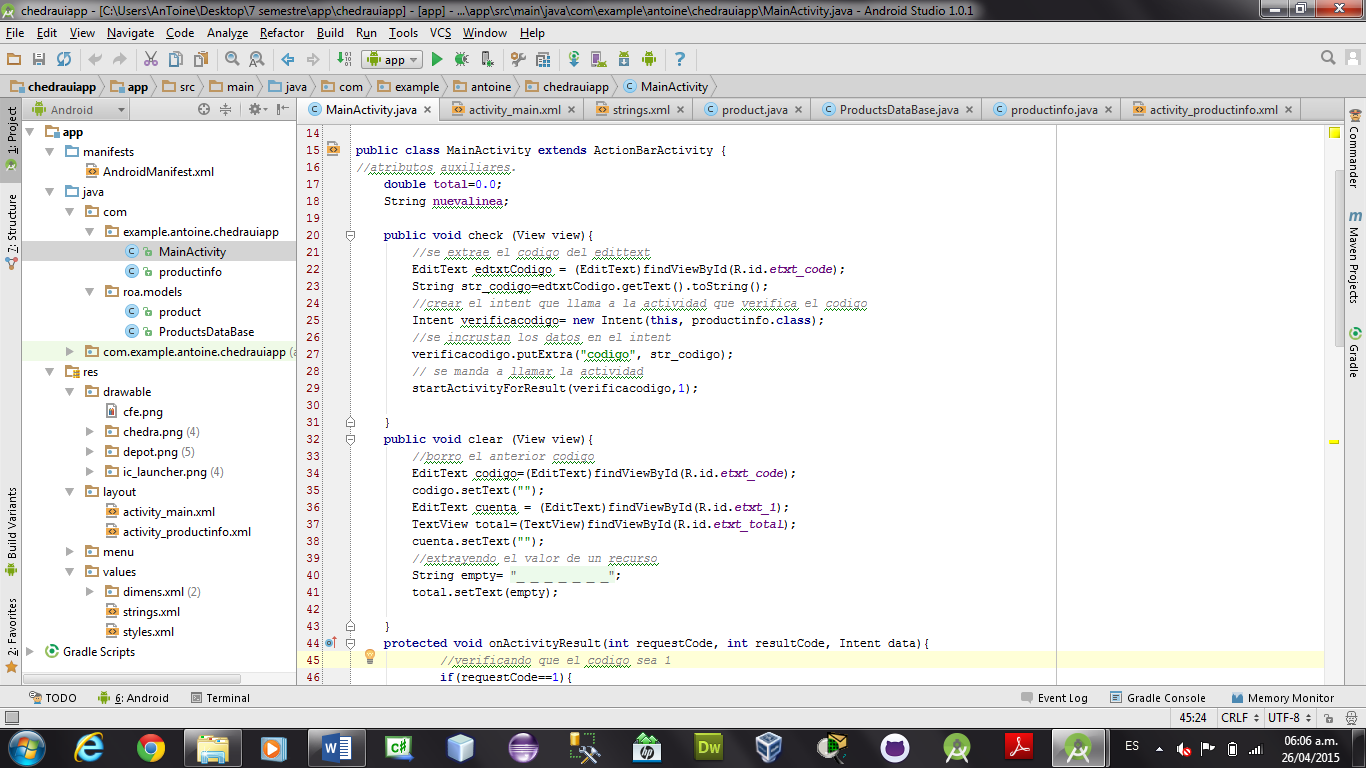
En la segunda actividad creamos una vista igual a la siguiente imagen.



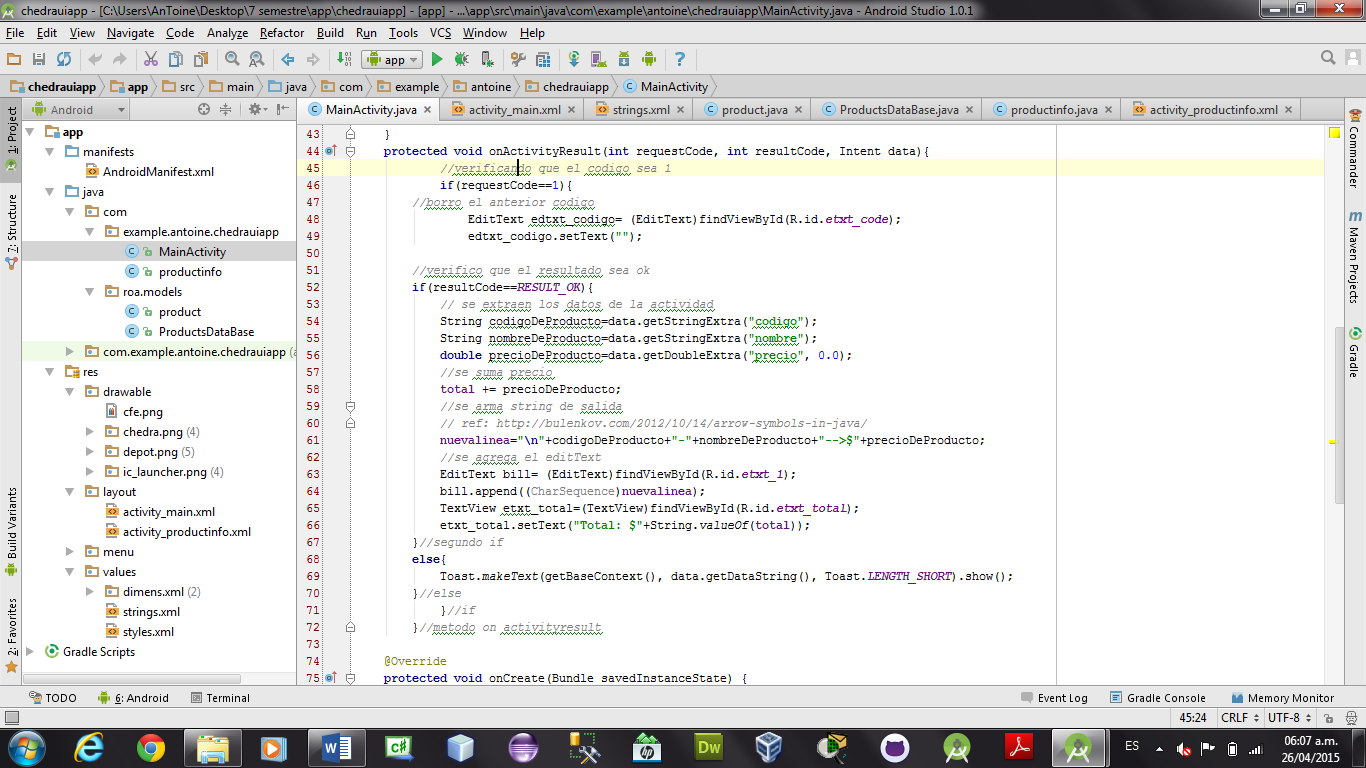
Se agregan imágenes en los recursos, imágenes de los productos y marca de Chedraui, posteriormente agregamos un packet ubicado en java y le nombramos roa.models el cual ahí se agregaran dos clases.



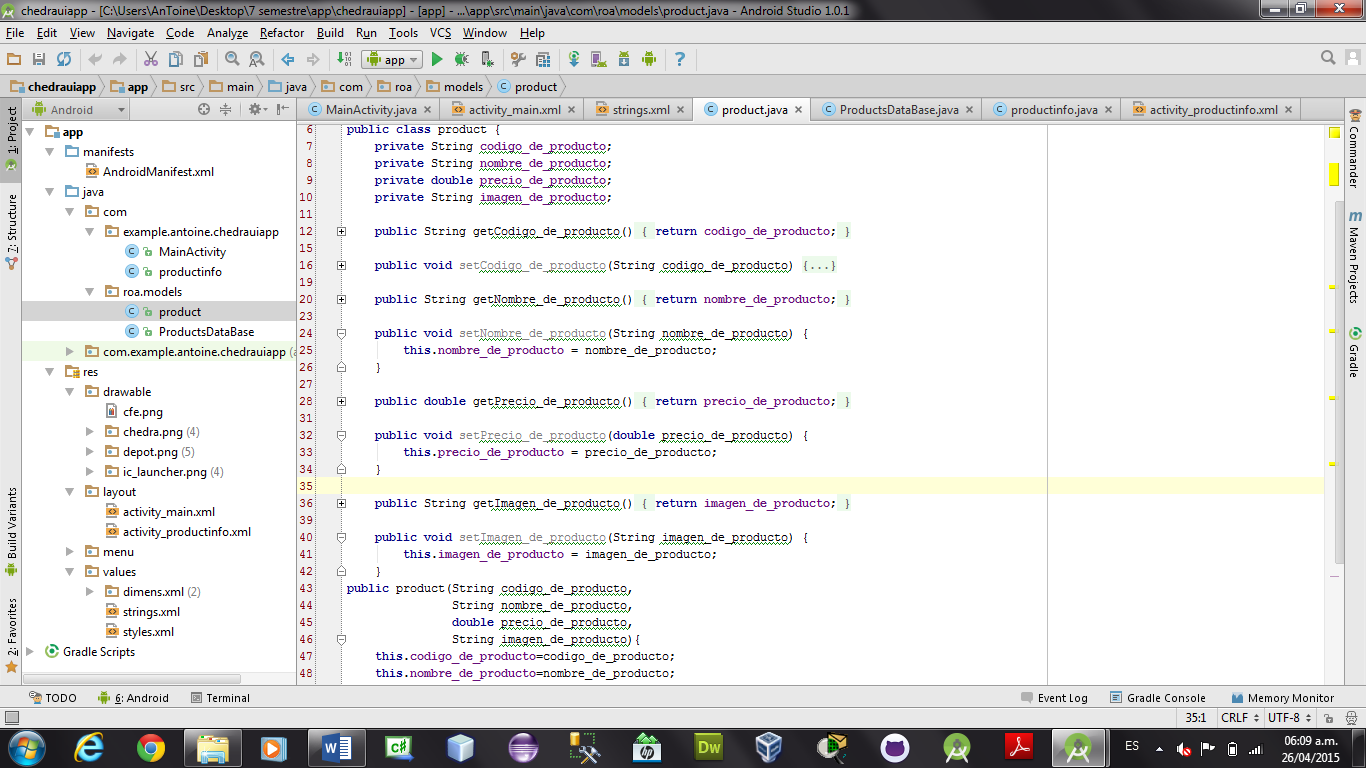
Después configuramos la clase java de la actividad principal, en el botón limpiar se le agrega el método clear, que solo se encarga en mandar por un set text un vacío para que borre los valores mandados de la otra actividad. En el botón verificar se le agrega el método check que se encarga de abstraer el string insertado en el edittext y lo manda a la segunda actividad para verificar dicho código.



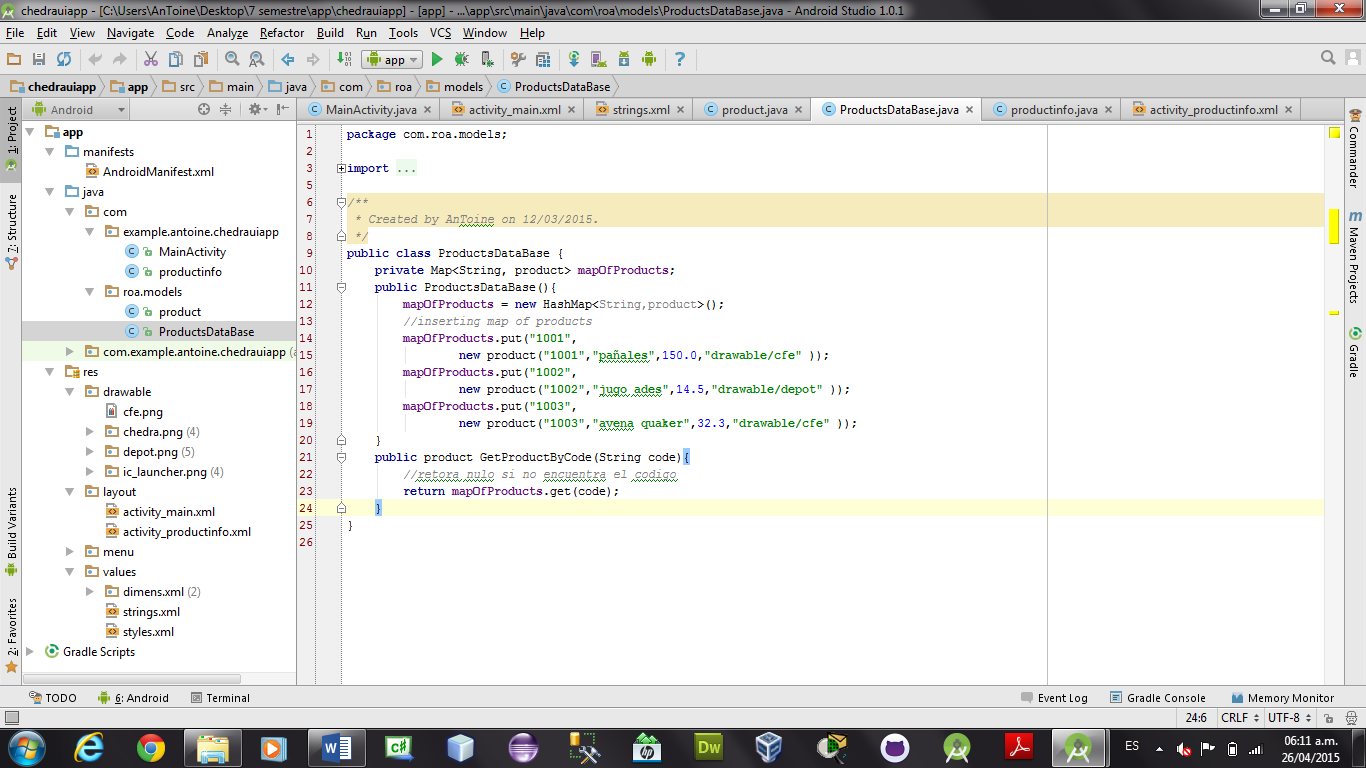
En la misma clase del main, agregamos un método onActivityResult que se da al regresar la actividad y espera un resultado o respuesta de la otra actividad, en ello se configuro que si encontraba un valor lo imprimiera en el editText e hiciera la operación de la suma de los productos encontrados. En caso contrario mandaría un toast o mensaje que no se encontró



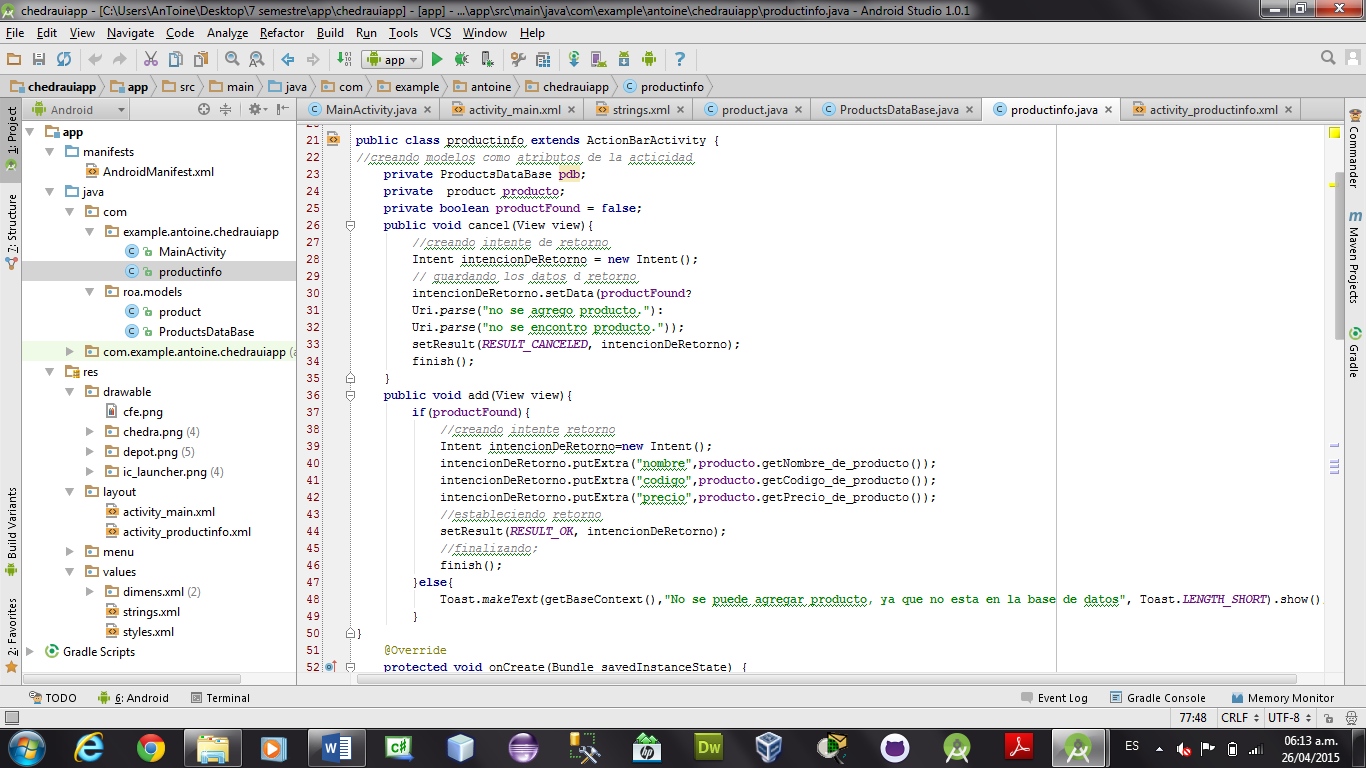
En la clase producto generamos variables para cada concepto como codigo, nombre, precio, imagen que abstraeran un valor y mandaran un set o estableceran, se hacen privados y esperan dicho valor en el otro publico.



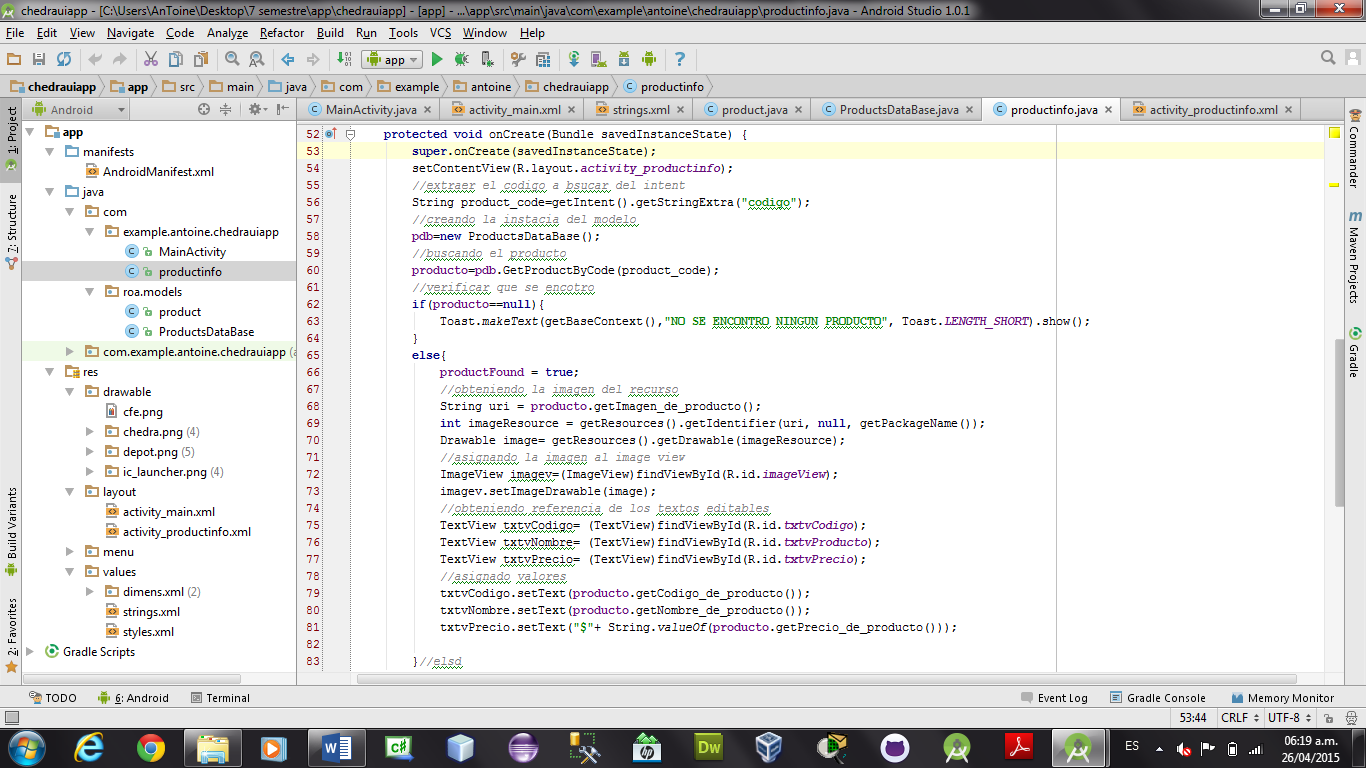
En la clase productdatabase se crea una base de datos de los productos, no se utiliza un gestor de base de datos como tal, pero se almacena en un mapOfProducts y se crea la estructura de dichos productos.



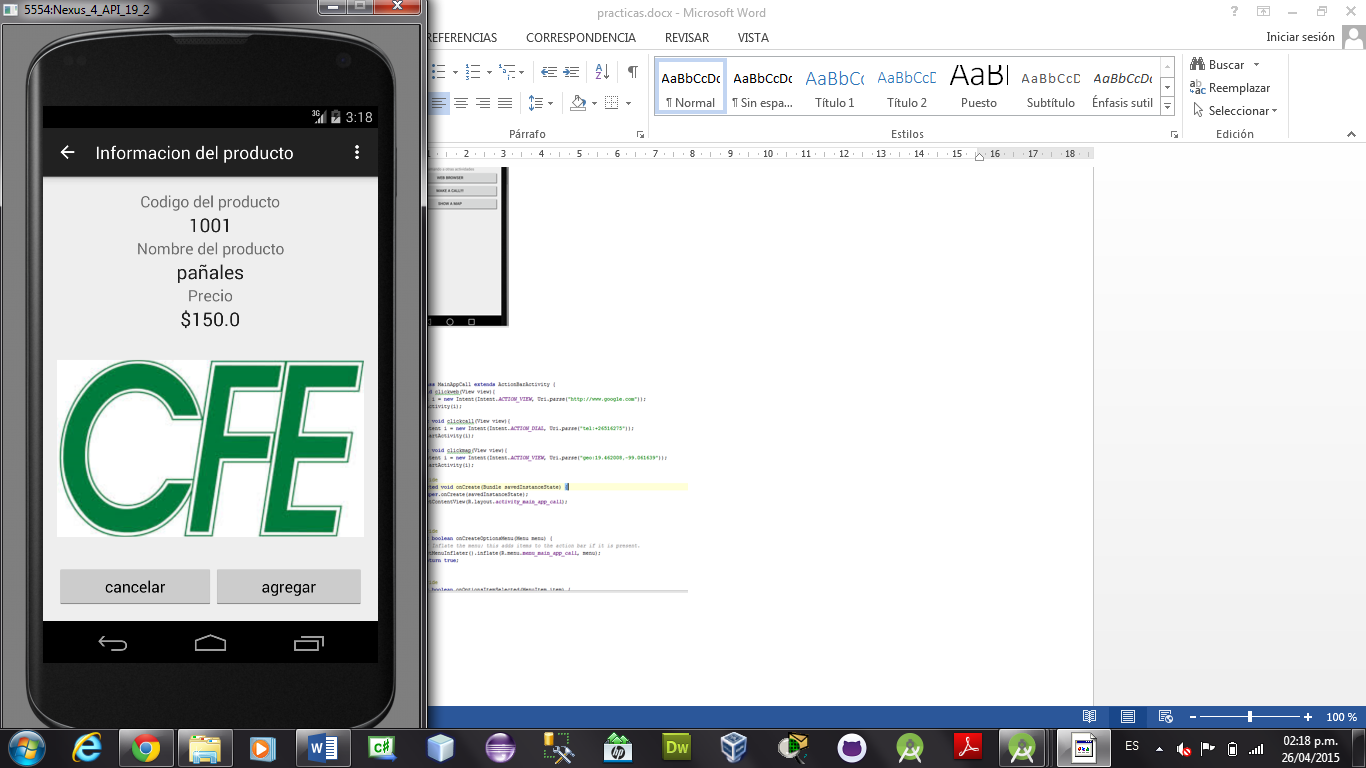
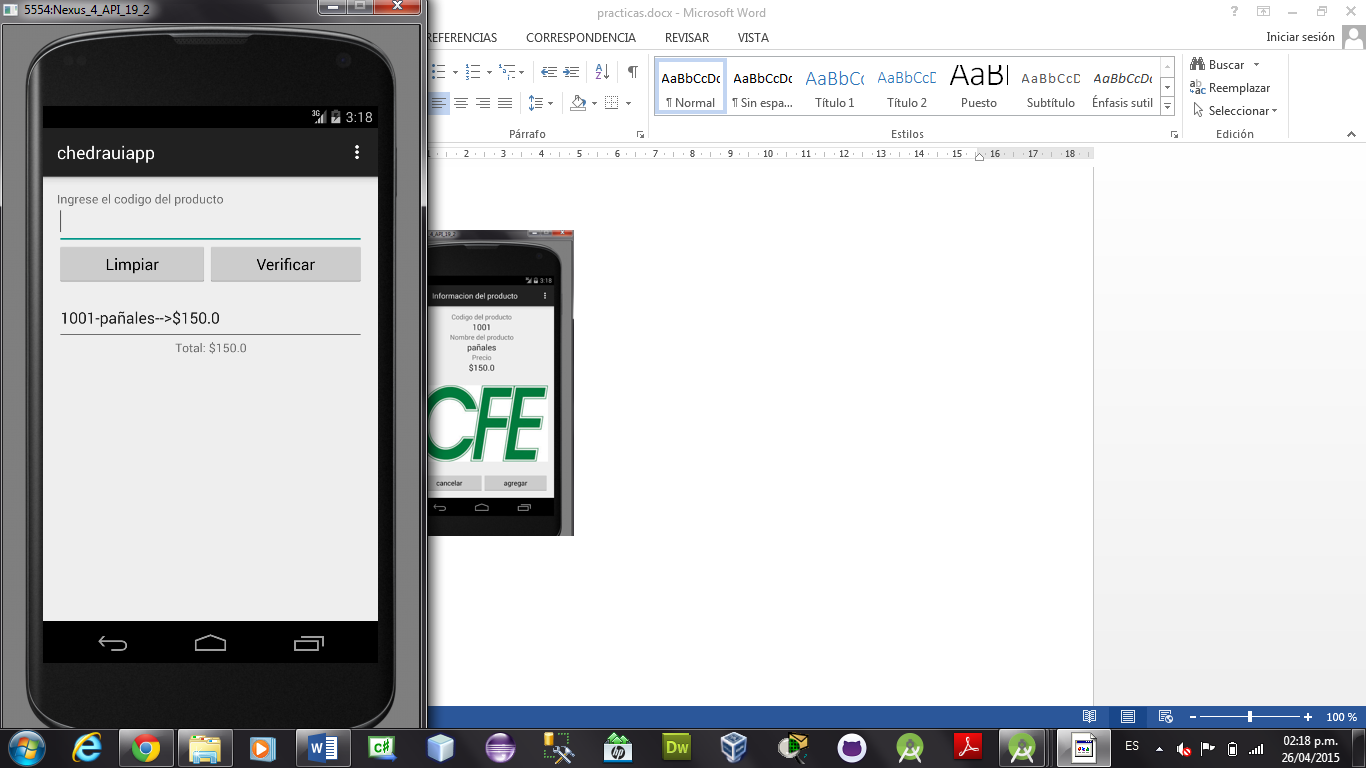
En la clase de la segunda actividad se genera el código necesario para mandar a llamar las dos clases que se crearon como product y productdatabase; y se genera dos métodos uno de cancel y otro de add, que son para los botones agregar y cancelar, en el botón cancelar su método es verificar por un boolean si encontró o no encontró el producto y dependiendo de ello mandaría un mensaje de que lo encontró pero fue cancelado o no se encontró dicho código. En el botón agregar su método consiste en utilizar la clase producto para abstraer dicha información que se encontró en dicho método oncreate que se utiliza para ubicar de la base de datos dicha información y este lo encapsula en un putExtra para darle como respuesta a la actividad principal. En lo contrario manda un mensaje de que no se encontró el producto y no se puede agregar.



En el método onCreate se genera la interfaz de la actividad y agregamos el código necesario para que visualice el producto encontrado en la base de datos y muestre su información como la imagen estableciendo los valores en el text view y en el imageview.



Al final tendremos un resultado como este, el producto encontrado y haciendo la suma y el total, y segunda actividad muestra la visualizacion del producto del codigo a peticion que se le envia.

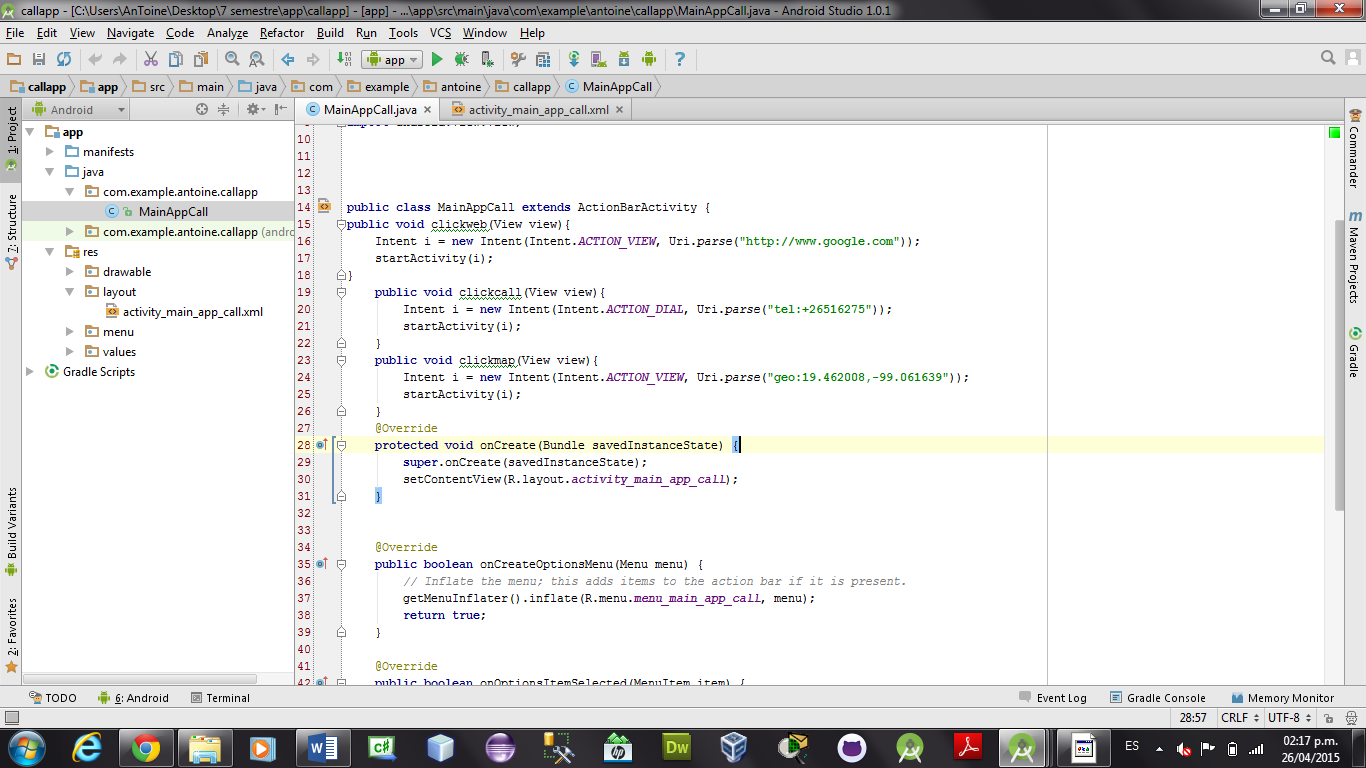


**CALLAPP**

Se crea una interfaz igual de la siguiente imagen que muestra la actividad.



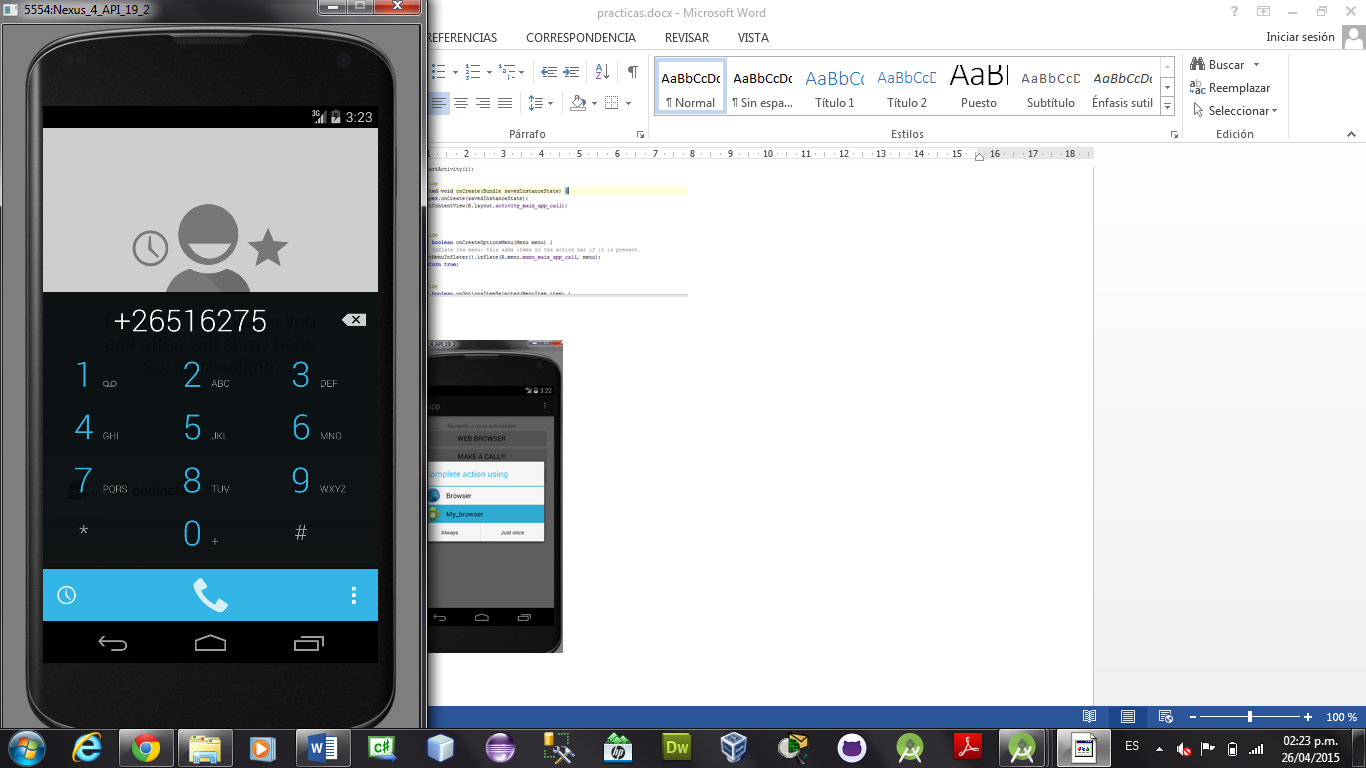
El codigo de esta actividad es sencillo para cada boton se le crea un metodo, para el boton web browser se le crea el metodo clickweb que con una intencion por medio de un ACTION\_VIEW manda a llamar dicho explorador instalado al celular y con un referencia web de tipo uri manda a llamar la pagina de google. En el segundo metodo se utiliza ACTION\_VIEW pero se utiliza ACTION\_DIAL que funciona igual pero en este caso el uri manda a llamar un numero telefonico. En el tercer metodo pasa lo mismo pero en este se genera una geo localizacion estableciendo las coordenadas.



Al final tenemos un resultado como la siguiente foto al dar click al web explorarador, y nos da opciones de como abrir cierta pagina web.

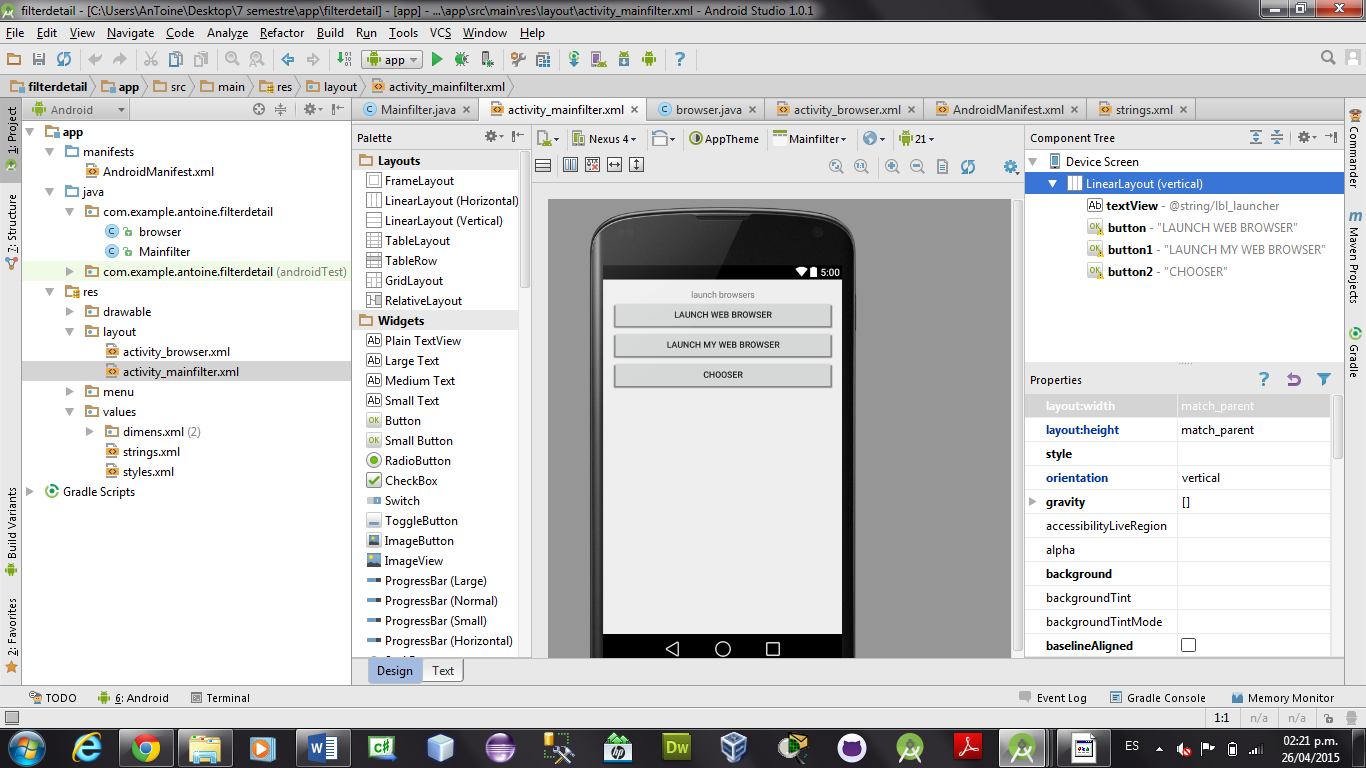


En el segundo boton tenemos un ejemplo como manda llamar una llamada al darle en el uri dicho numero.

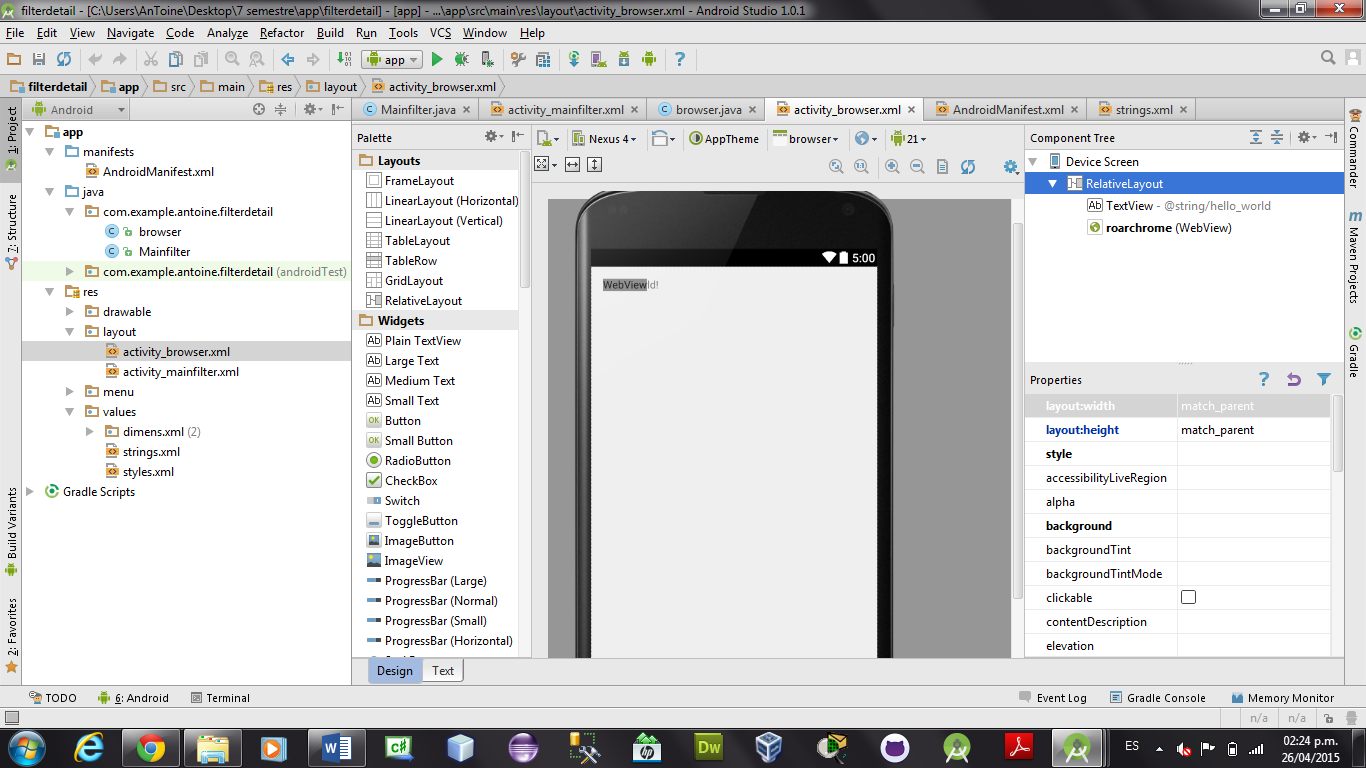


**FILTERAPP**

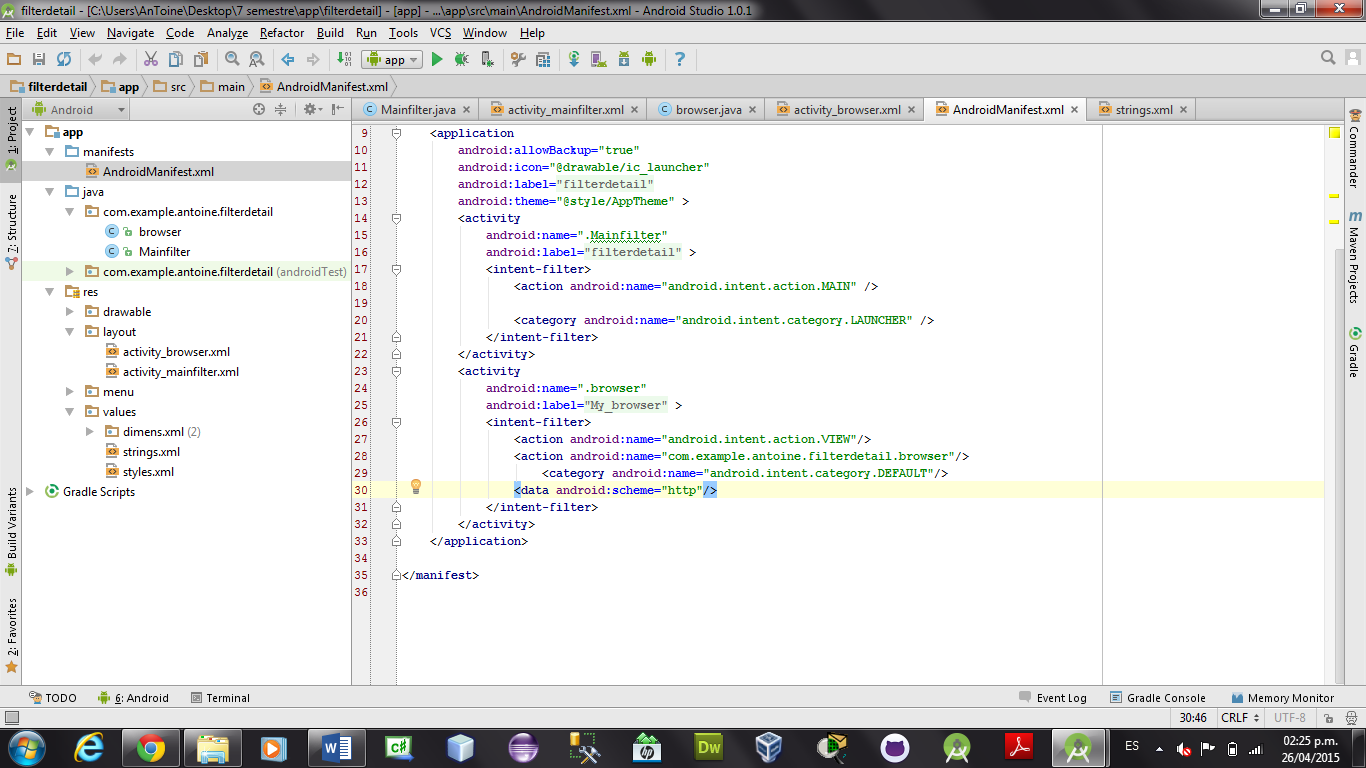
Generamos una interfaz igual a la siguiente foto.



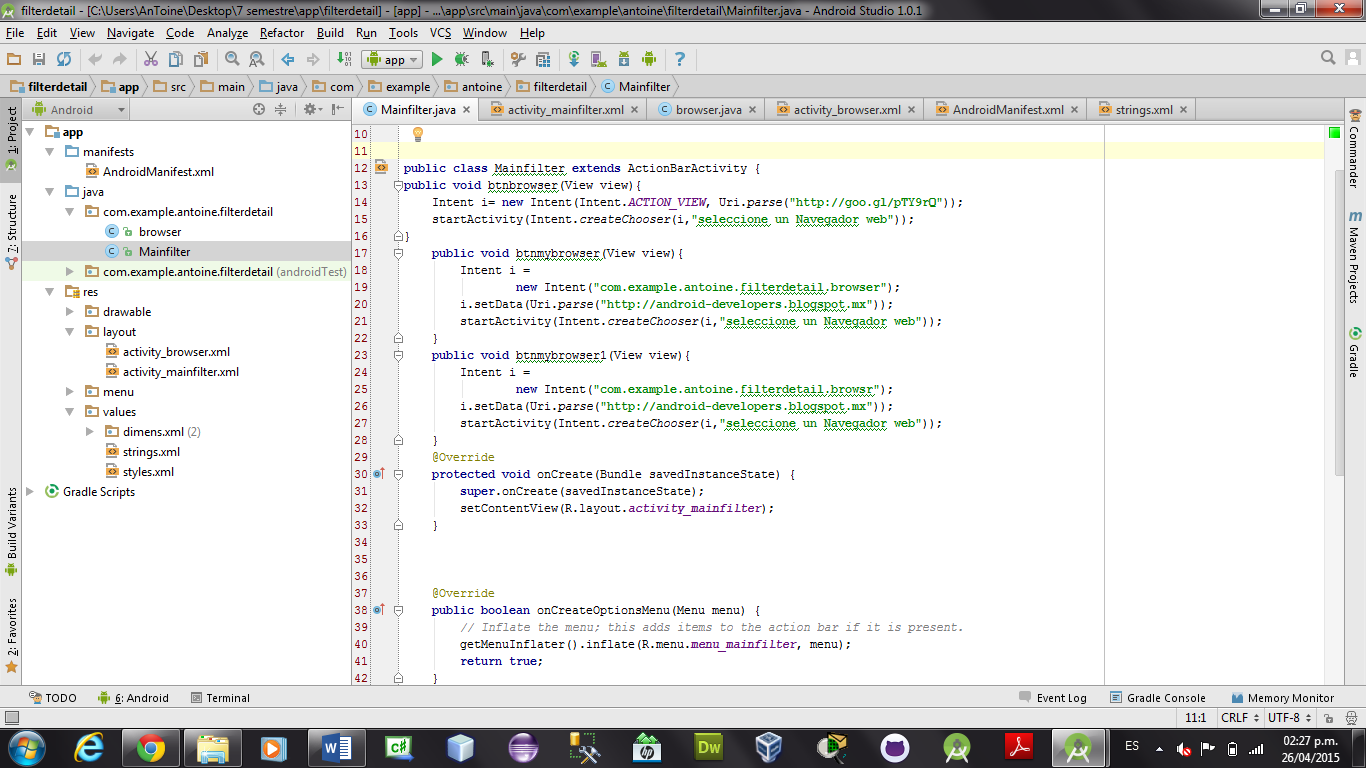
En la segunda actividad solo insertamos un webview, donde visualizaremos nuestra webBrowser



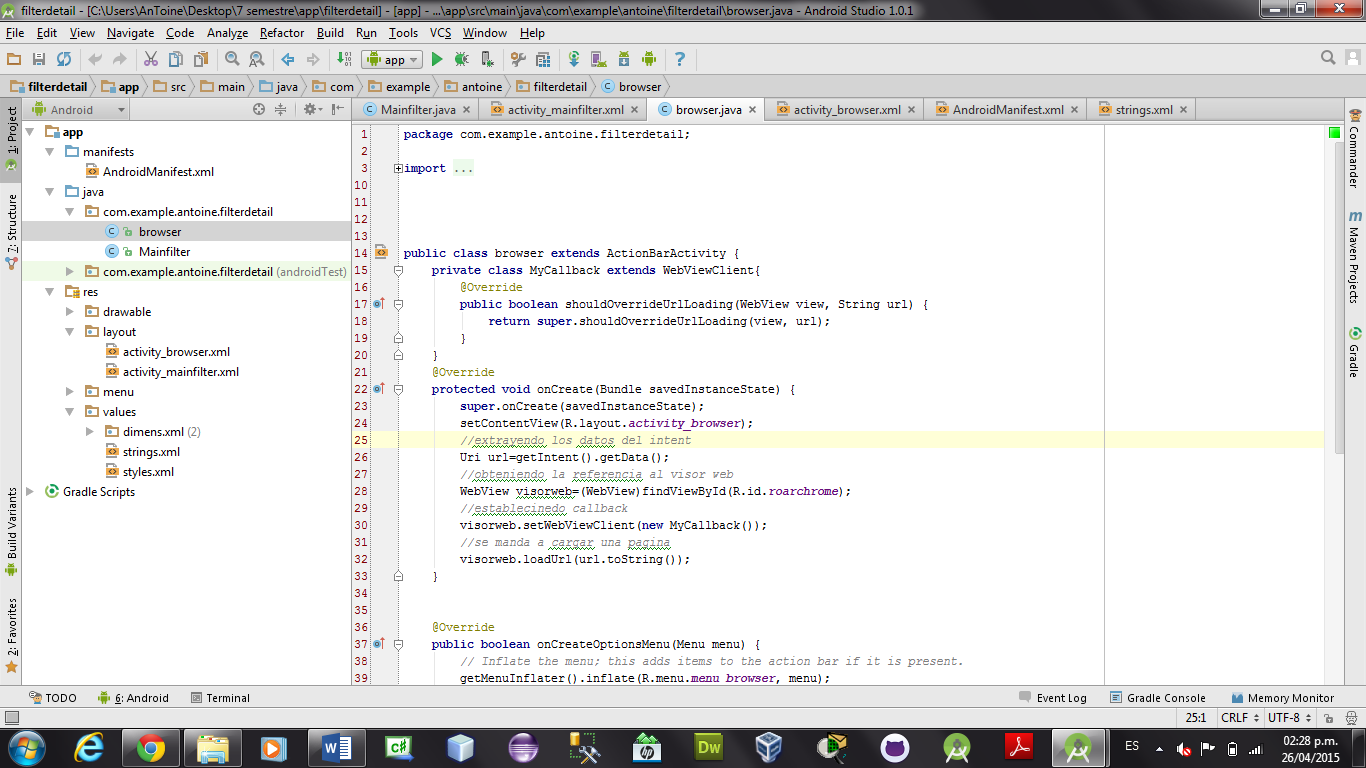
En el codigo del manifest agregamos un filter en la segunda actividad que tenga un category.default y un action.view, y dicho filtro es filterdetail.browser.



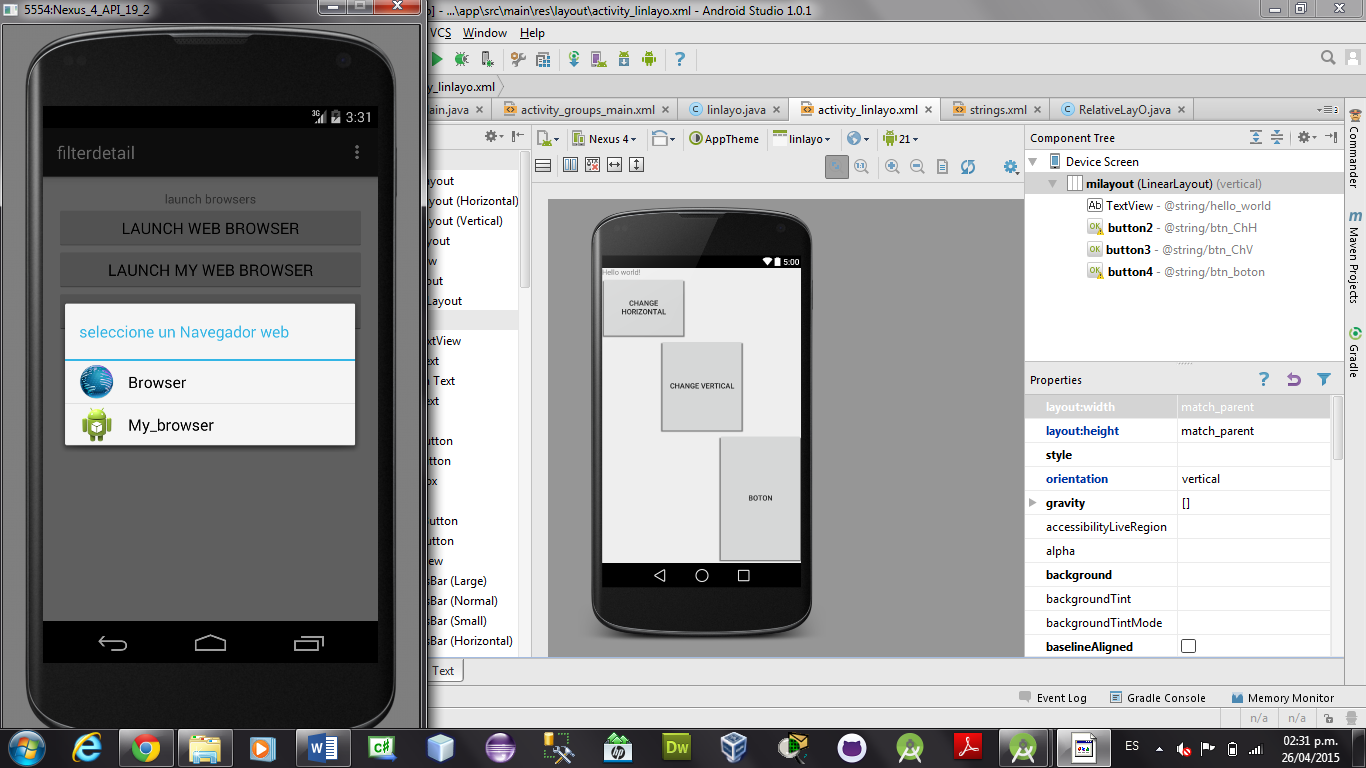
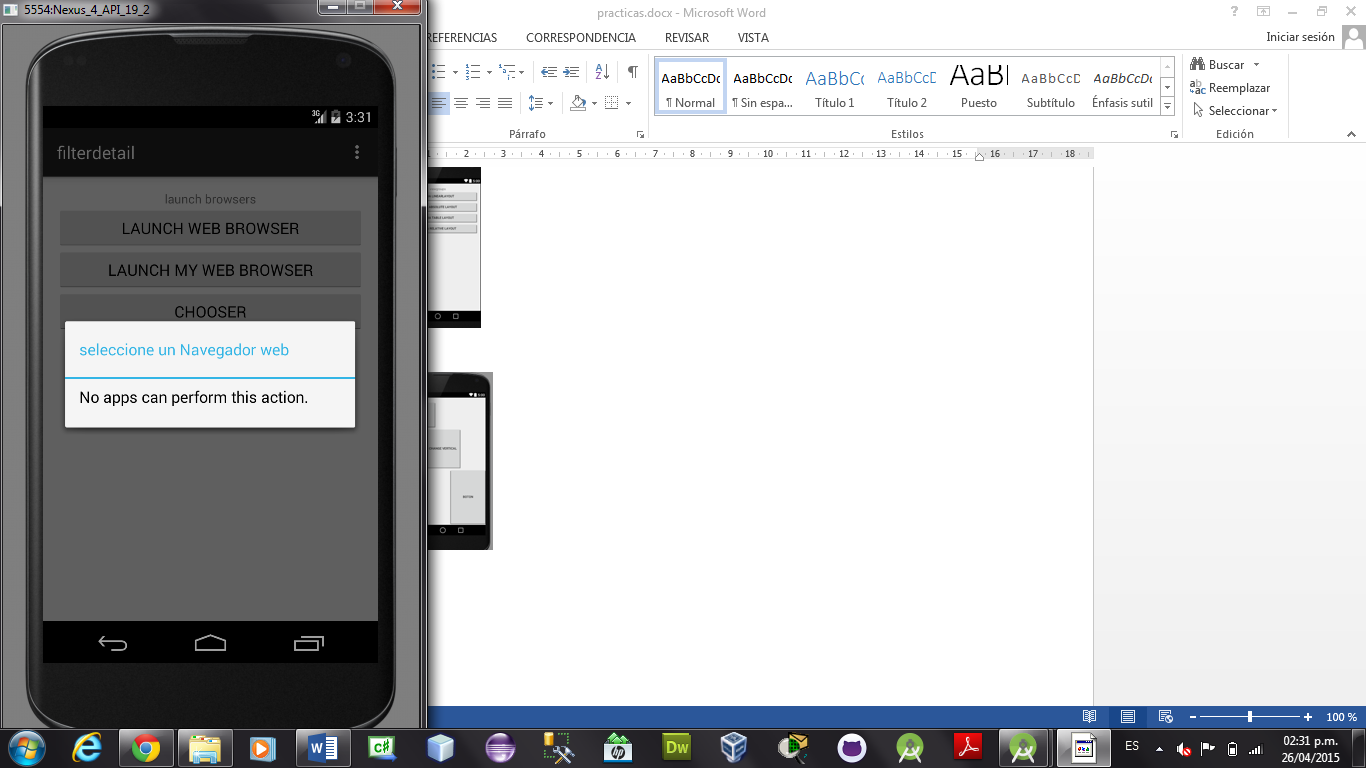
En la clase mainfilter vamos agregar tres metodos para su respectivos botones en el primer metoco mandamos a llamar como tal dicha web como en la anterior practica, agregando un URI y dicha pagina. En el segundo metodo generamos un intent implicito que se le da el filtro para determinar que nuestro explorador se va abrir. Y el tercer metodo es igual al segundo pero en este caso esta mal determinado el filtro, y al darle click se cierra la aplicación, y para eso generamos un chooser para que no se rompa la aplicación y genere solo un mensaje de notificacion de error.



En la clase de la segunda actividad simplemente abstraemos en un uri la pagina almacenada en el data y en el intenet el cual posterirormente lo mandaremos a llamar en el visorweb mandandole la url almacenada en uri el cual lo visualizara en el recurso roarchrome que es l webview.

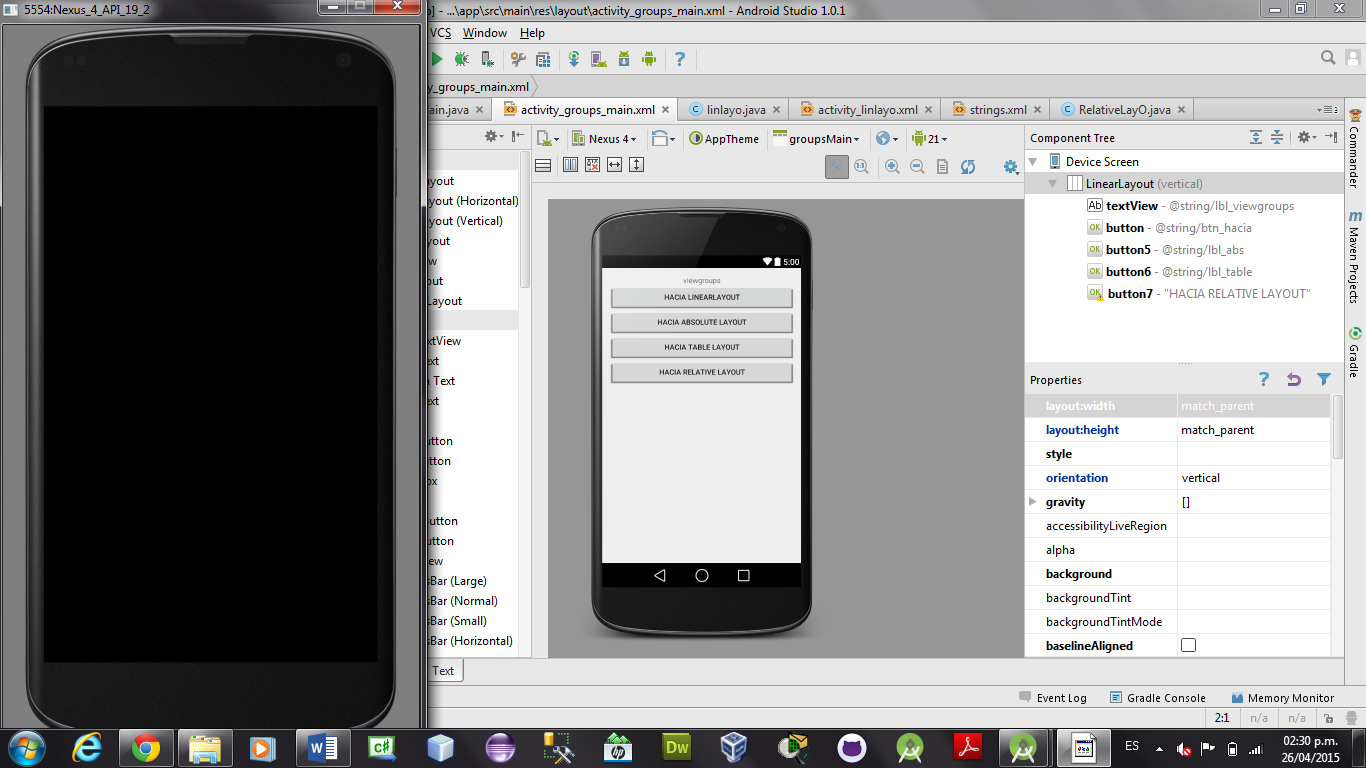


Al final tendremos estos resultados, en el chooser en el tercer botón vemos que nos manda un mensaje sobre que hubo un error en el código y no encontró dicho valor establecido, mientras en el primer botón nos manda la opción de que explorador deseamos abrir la web, mientras con el tercer botón mandamos a llamar directamente nuestro explorador abriendo dicha web.



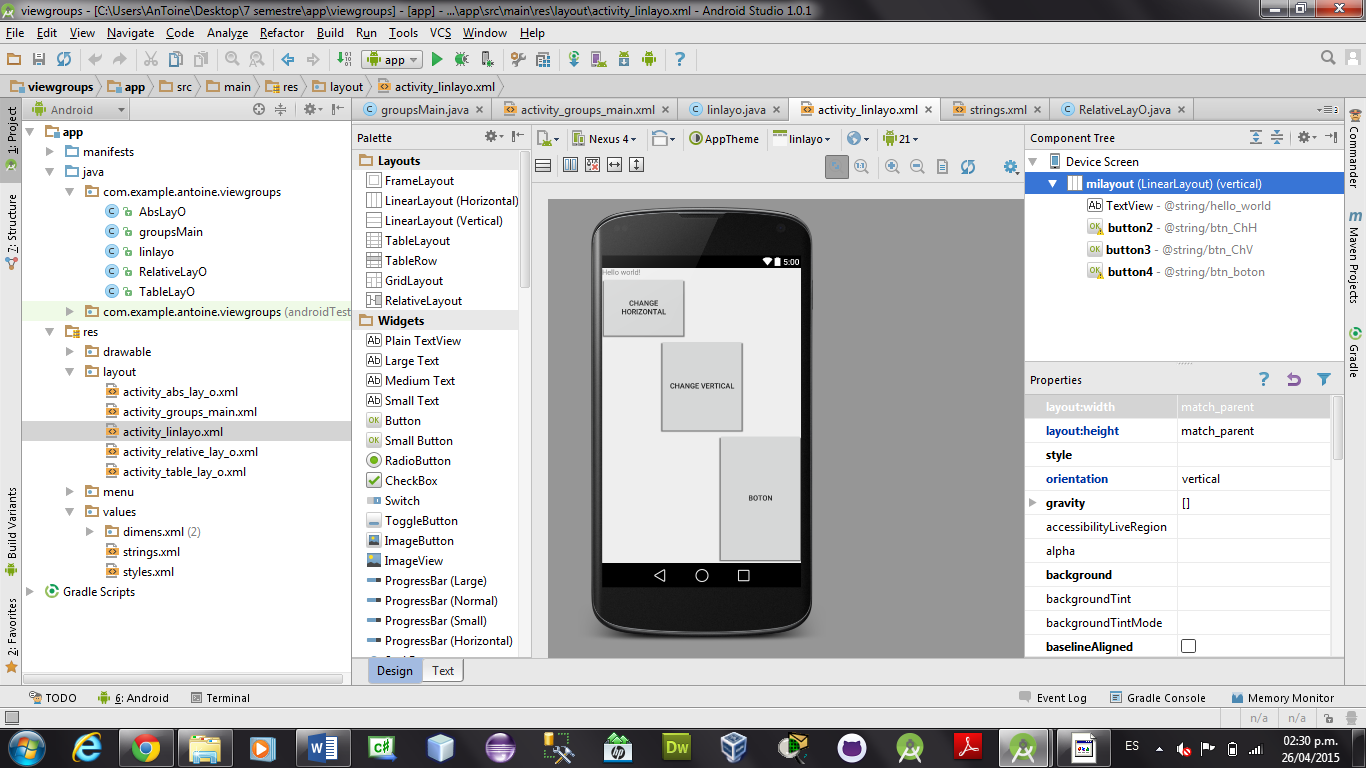
**APP VIEWGROUP**

Se crea una interfaz igual a la siguiente foto con 4 botones llamando dichos métodos para llevar a sus respectivas actividades.

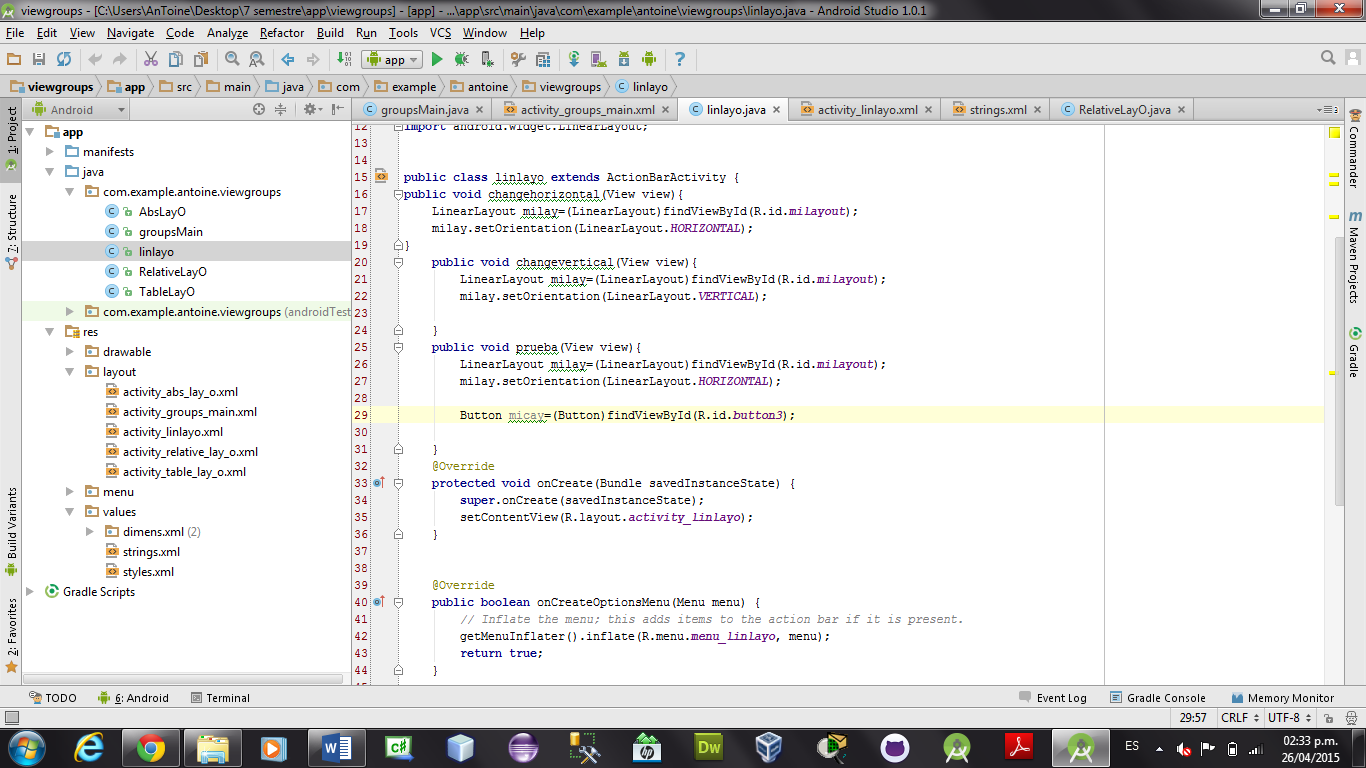


En el primer botón manda a llamar dicha interfaz o actividad que debemos establecer por medio de LINEARLAYOUT vertical con sus respectivas alineaciones y pesos.

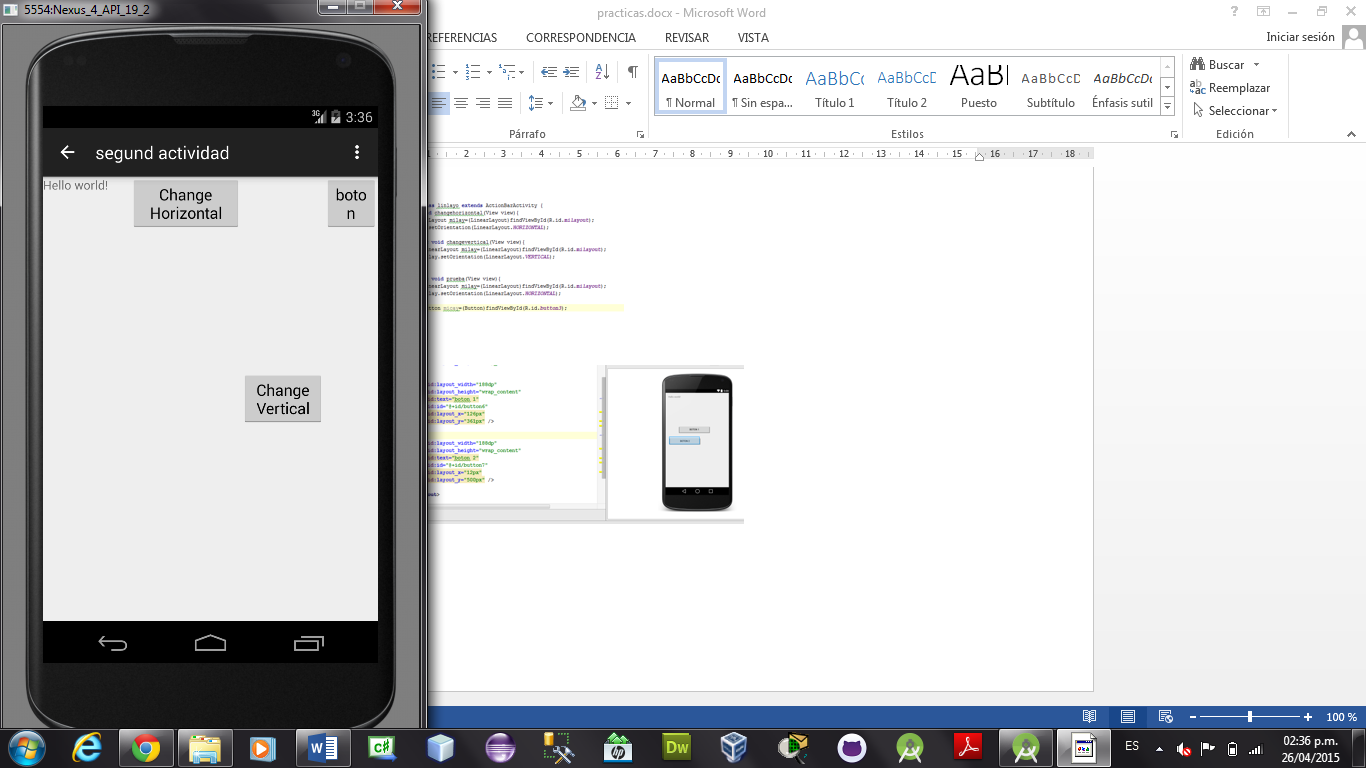
En esta actividad se genera un codigo que cambia la interfaz de dicha vista que tenemos.



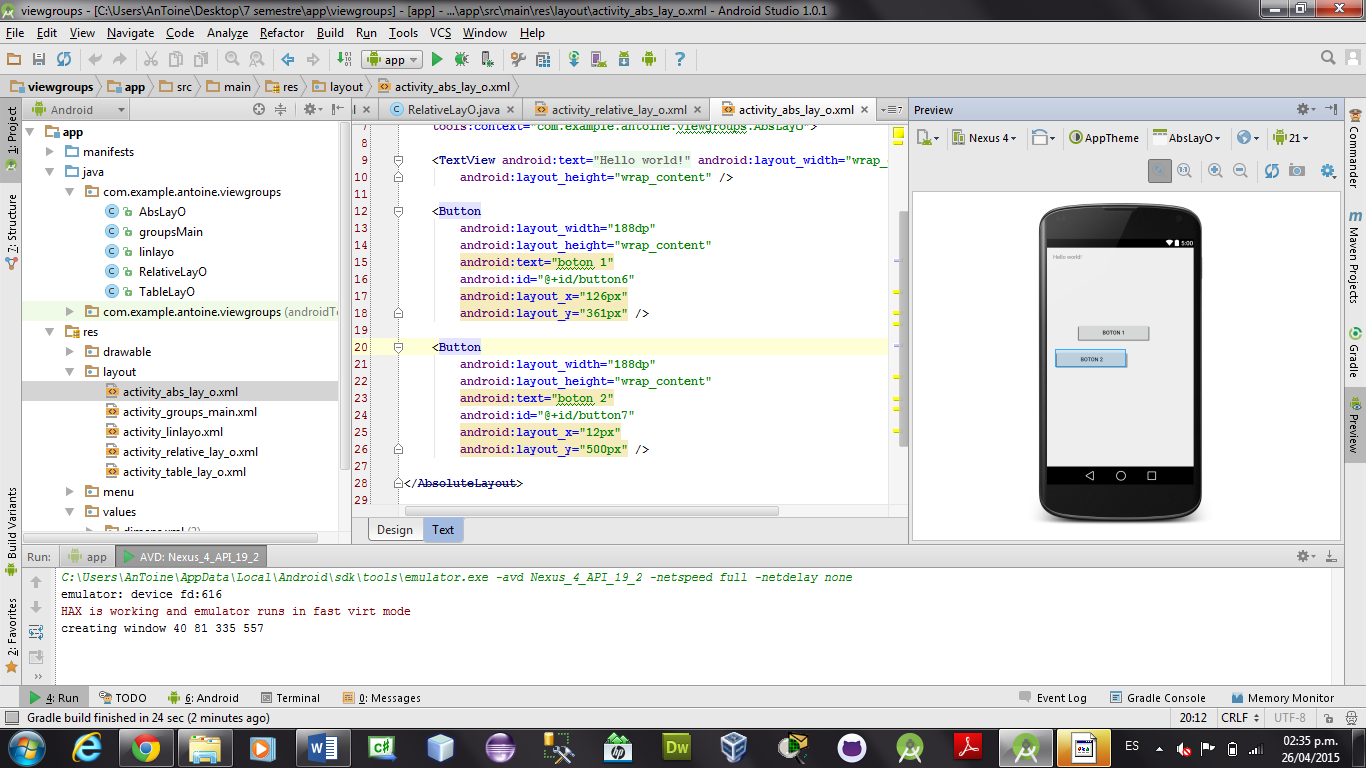
En la clase de linlayo que es linearlayout generamos dos métodos, uno para el horizontal y otro para vertical, en el método vertical genera la orientación de manera vertical. Mientras en el horizontal establece la orientación del layout en horizontal y dicho layout lo busca por su id y lo establece como linearlayout y se le da la propiedad de orientacion



Tendremos un resultado igual a este al darle click al boton horizontal, y al darle vertical nos restablecera la vista anterior que teniamos.



Para el segundo botón el absolutLayout se genera una vista de grupos de manera absoluta y se le establece propiedades a los botones para que tengamos un diseño como en la siguiente foto.



En el tercer botón nos manda los que es una vista de manera de tabla, una tabla invisible que lo ordenaremos por filan, el cual nos muestra una vista como la siguiente foto, el cual muestra como acomodamos por fila dicha distribución de los text view, editText y las propiedades para tener una vista igual a la que se muestra.

