

Universidad tecnológica Metropolitana

CALIDAD DE DESARROLLO DE SOFTWARE



Ing. oSCAR JOSUÉ UH PÉREZ, MTGI

5° “B”

josé luis puc chan

Santiago martinez Chable

30/03/2017

Introducción

En el presente documento vamos a abordar los principales aspectos del proyecto del cálculo del índice de masa corporal (IMC), donde se representará las distintas tablas de la estructura PSP.

Por primer lugar se presentarán el análisis que contiene este proyecto, así como el diagrama de Gantt, sus requerimientos y diagrama de casos de uso.

Por consiguiente, se expone las acciones que cada integrante realizo por medio de las tablas del PSP, en el que el análisis fue individual y el desarrollo lo realizo cada integrante del equipo dependiendo de sus tareas asignadas.

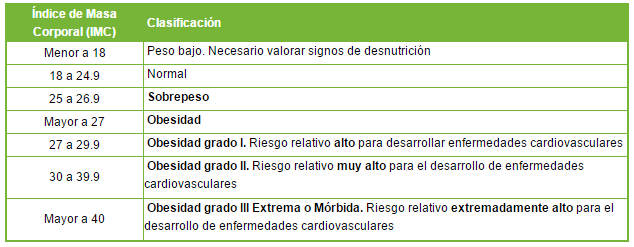
Para el de este documento se tiene el resumen plan de proyecto, donde nos muestra los avances del proyecto basado en tiempos.

Descripción del caso de estudio del IMC

El IMC se calcula de la siguiente forma

                                            http://www.noalaobesidad.df.gob.mx/images/stories/formulaimc.jpg

Desarrolla una aplicación en Android que introduzca dos valores para el peso y la estatura de una persona. El programa debe calcular el IMC y dependiendo del resultado compararlo con la siguiente tabla



Y mostrar un mensaje de acuerdo al resultado obtenido del cálculo del IMC.

Ejemplo:

Hallar el **IMC** de una persona cuyo peso es de 120 kg y estatura de 1.65 metros.

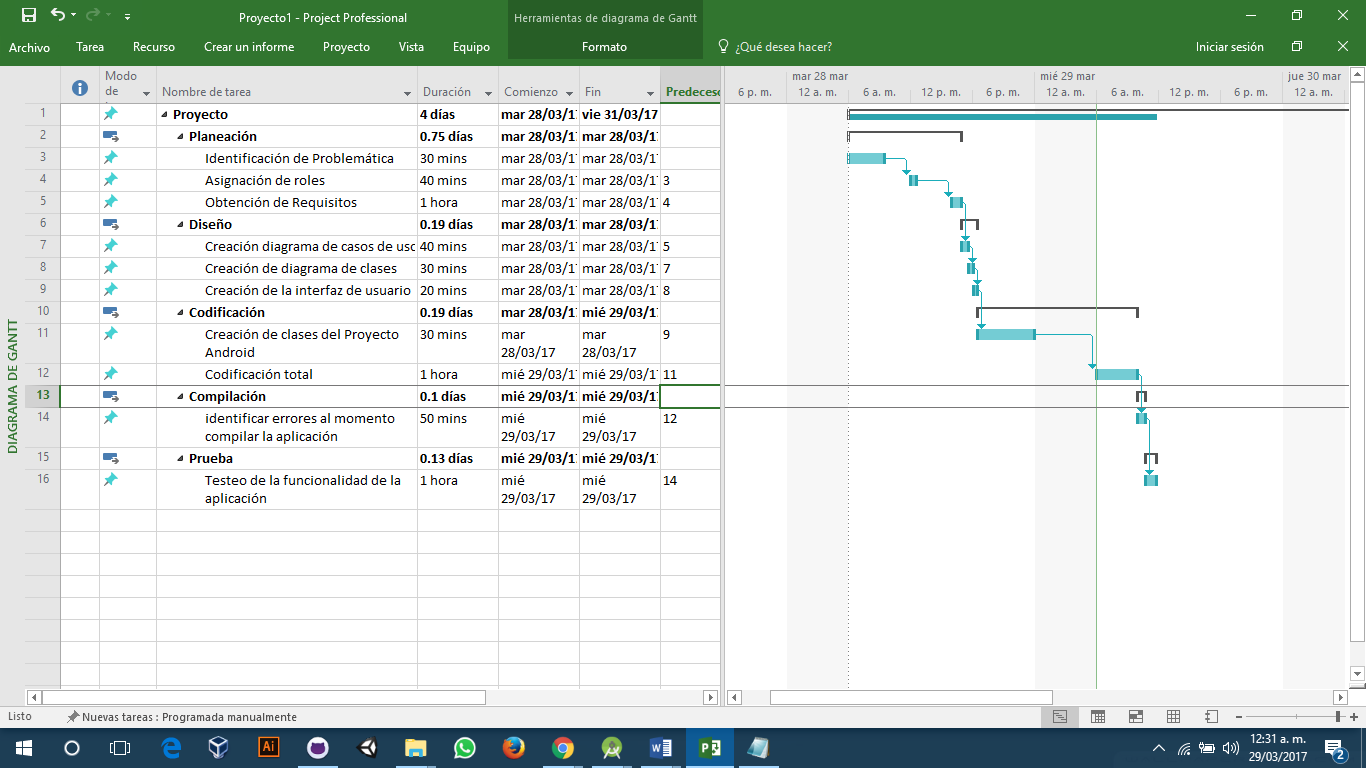
**Peso**   =        120 kg        =     120 kg     = **44**  
                                               **Estatura²**          1.65 x 1.65            2.7225

Y se muestra el siguiente mensaje:

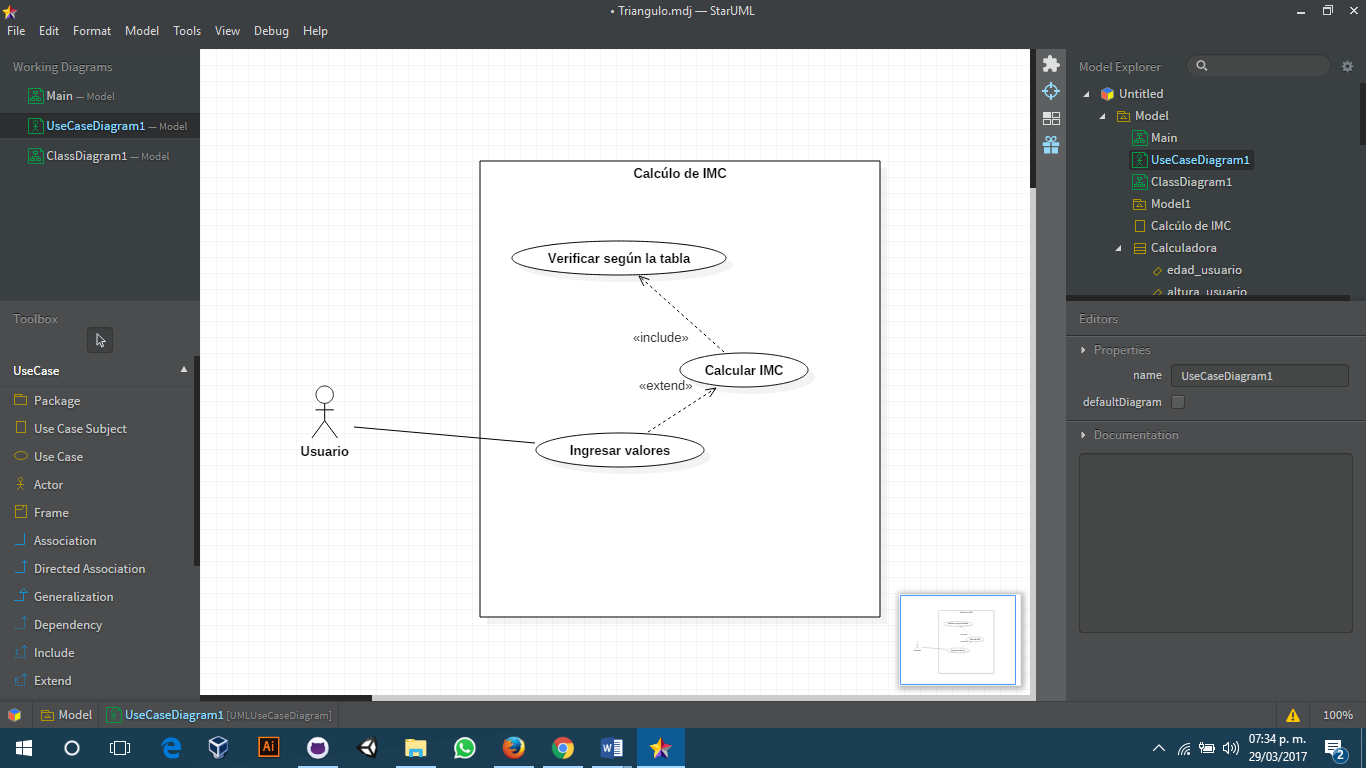
|  |  |
| --- | --- |
| Mayor a 40 | **Obesidad grado III Extrema o Mórbida.** Riesgo relativo **extremadamente alto** para el desarrollo de enfermedades cardiovasculares |

# Planeación

## Diagrama de Gantt:

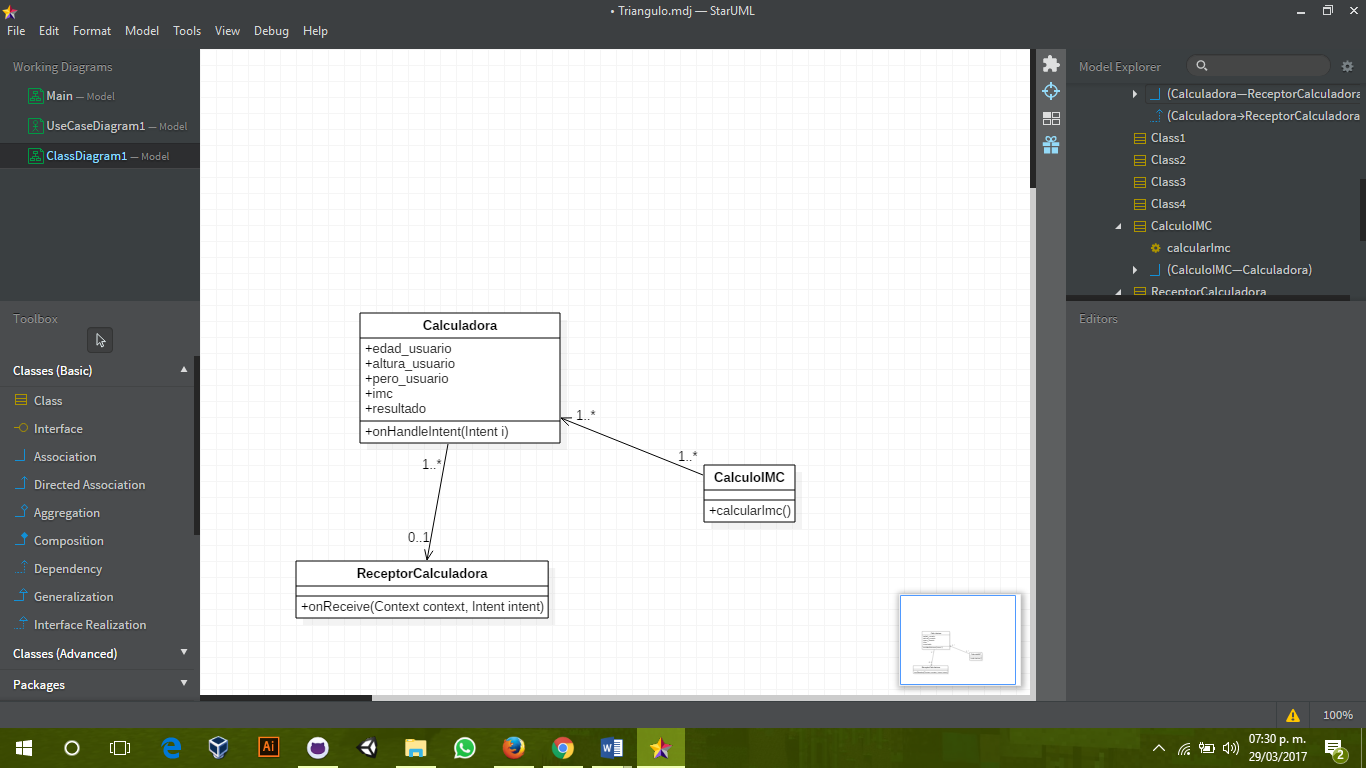


### Diagrama de casos de uso:



# Diseño

## Diagrama de clases:



## Registro de Tiempos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Registro de Tiempo | | | | | | |
| **Fecha** | **Inicio** | **Fin** | **Interrupción** | **Tiempo** | **Actividad** | **Comentarios** |
| 28/03/17 | 11:00 am | 11:10 am |  | 10 min | Creación | Creación del repositorio único para el proyecto. |
| 29/03/17 | 11:10 am | 12:10 am |  | 60 min | Creación | Detección de requisitos. |
| 29/03/17 | 12:10 am | 12:55 am | 5min | 45 min | Creación | Creación del diagrama de casos de uso. |
| 29/03/17 | 1:00 am | 1:10 am |  | 10 min | Creación | Creación del diagrama de clases. |
| 29/03/17 | 1:10 am. | 1:20 am. |  | 10 min | Creación | Creación del proyecto en Android Studio. |
| 29/03/17 | 1:30 am. | 2:00 am. |  | 30 min | Diseño | Implementación del diseño del proyecto. |
| 29/03/17 | 2:00 am | 2:05 am |  | 5 min | Correcciones | Se agregó la portada, tabla de registro de estándares, anexos del repositorio. |
| 29/03/17 | 2:00 am | 2:10 pm |  | 5 min | Codificación | Codificar las clases con sus métodos. |
| 29/03/17 | 2:10 am | 2:40 am |  | 20 min | Codificación | Continuidad en la codificación de las clases. |
| 29/03/17 | 2:40 am | 3:00 am |  | 20 min | Revisión | Revisión de la sintaxis y estructura de las clases. |
| 29/03/17 | 4:22 pm | 4:42 pm |  | 22 min | Cambios | Corrección diagrama de clases. |
| 29/03/17 | 5:15 pm. | 5:54 pm. |  | 39 min | Modificación | Modificación de los requerimientos funcionales y no funcionales. |
| 29/03/17 | 5:54 pm. | 6:18 am. |  | 24 min | Agregación | Estructura de la documentación en Word |
| 29/03/17 | 7:00 pm | 7:20 pm |  | 20 min | Codificación | Codificación del proceso principal del proyecto. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Registro de Tiempo | | | | | | |
| **Fecha** | **Inicio** | **Fin** | **Interrupción** | **Tiempo** | **Actividad** | **Comentarios** |
| 28/03/17 | 11:00 am | 11:10 am |  | 10 min | Creación | Creación del repositorio único para el proyecto. |
| 29/03/17 | 11:10 am | 12:10 am |  | 60 min | Creación | Detección de requisitos. |
| 29/03/17 | 1:10 am. | 1:20 am. |  | 10 min | Creación | Creación del proyecto en Android Studio. |
| 29/03/17 | 2:00 am | 2:05 am |  | 5 min | Correcciones | Se agregó la portada, tabla de registro de estándares, anexos del repositorio. |
| 29/03/17 | 2:00 am | 2:10 pm |  | 5 min | Codificación | Codificar las clases con sus métodos. |
| 29/03/17 | 2:10 am | 2:40 am |  | 20 min | Codificación | Continuidad en la codificación de las clases. |
| 15/02/17 | 07:26 am | 07:49 am |  | 23 min | Codificación | Creación y codificación de la clase Calcular.java |
| 29/03/17 | 5:15 pm. | 5:54 pm. |  | 39 min | Modificación | Modificación de los requerimientos funcionales y no funcionales. |

José Luis Puc Chan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Registro de Tiempo | | | | | | |
| **Fecha** | **Inicio** | **Fin** | **Interrupción** | **Tiempo** | **Actividad** | **Comentarios** |
| 29/03/17 | 12:10 am | 12:55 am | 5min | 45 min | Creación | Creación del diagrama de casos de uso. |
| 29/03/17 | 1:00 am | 1:10 am |  | 10 min | Creación | Creación del diagrama de clases. |
| 29/03/17 | 1:30 am. | 2:00 am. |  | 30 min | Diseño | Implementación del diseño del proyecto. |
| 15/02/17 | 2:40 am | 3:00 am |  | 20 min | Revisión | Revisión de la sintaxis y estructura de las clases. |
| 29/03/17 | 4:22 pm | 4:42 pm |  | 22 min | Cambios | Corrección diagrama de clases. |
| 29/03/17 | 5:54 pm. | 6:18 am. |  | 24 min | Agregación | Estructura de la documentación en Word |
| 29/03/17 | 7:00 pm | 7:20 pm |  | 20 min | Codificación | Codificación del proceso principal del proyecto. |

## **Estándar Tipos de defectos.**

|  |  |  |
| --- | --- | --- |
| **Tipos de defectos** | | |
| **Nº de tipo** | **Nombre del tipo** | **Descripción** |
| 10 | Documentación | Comentarios, mensajes |
| 20 | Sintaxis | Ortografía, puntuación, erratas, formato de las instrucciones |
| 30 | Construir, paquetes | Gestión del cambio, librerías, control de versión |
| 40 | Asignación | Declaración, nombres duplicados, ámbito, límites |
| 50 | Interfaz | Llamadas a procedimientos y referencias, E/S, formatos de usuario |
| 60 | Chequeo | Mensajes de error, chequeos inadecuados |
| 70 | Datos | Estructura, contenido |
| 80 | Función | Lógica, punteros, bucles, recursión, computación, defectos de la función |
| 90 | Sistema | Configuración, temporización, memoria |
| 100 | Entorno | Diseño, compilación, pruebas y otros problemas que soporta el sistema |

## **Resumen Plan del proyecto.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Programa:** IMC | | | | | | | | **Fecha:** 30/03/2017 | | |
| **Descripción:** Software encargado de calcular el índice de masa corporal, por medio del peso la altura y edad del usuario. Está desarrollado en el lenguaje Java en la plataforma Android Studio. | | | | | | | | | | |
| **Resumen** | **Plan** | | | **Real** | | | | **Hasta la fecha** | | |
| Minutos/LOC | 1.6 | | | 1 | | | | 1 | | |
| LOC/Hora | 55 | | | 60 | | | | 60 | | |
| Defectos/KLOC | - | | | 13 | | | | 13 | | |
| Rendimiento | - | | | - | | | |  | | |
| Valoración/Fallo | - | | | - | | | |  | | |
| **Tamaño programa (LOC)** | **Plan** | | | **Real** | | | | **Hasta la fecha** | | |
|  | | |  | | | |  | | |
| Total nuevo & cambiado | 100 | | | 150 | | | | 150 | | |
| Tamaño máximo | 300 | | |  | | | | | | |
| Tamaño mínimo | 100 | | |
| **Tiempo por Fase (min.)** | **Plan** | | **Real** | | | **Hasta la fecha** | | | **% Hasta la fecha** | |
| Planificación | 30 | | 40 | | | 55 | | | 18% | |
| Diseño | 40 | | 50 | | | 55 | | | 21% | |
| Codificación | 100 | | 120 | | | 120 | | | 52% | |
| **Revisión del código** |  | | | | | | | | | |
| Compilación | 1 | | 1 | | | 1 | | | 0.5% | |
| Pruebas | 20 | | 10 | | | 15 | | | 4.25% | |
| Postmorten | 10 | | 10 | | | 10 | | | 4.25% | |
| Total | 201 | | 231 | | | 256 | | |  | |
| Tiempo máximo | 300 | |  | | | | | | | |
| Tiempo mínimo | 150 | |
| **Defectos Introducidos** | **Plan** | **Actual** | | | **Hasta la fecha** | | **% Hasta la fecha** | | | **Def./Hora** |
| Planificación | - | - | | | - | | - | | |  |
| Diseño | - | 5 | | | 5 | | 23% | | |  |
| Codificación | - | 10 | | | 15 | | 68% | | |  |
| Revisión del código |  | | | | | | | | | |
| Compilación | - | - | | | 2 | | 9% | | |  |
| Pruebas | - | - | | | - | | - | | |  |
| Total | - | 15 | | | 22 | | 100% | | |  |
| **Defectos eliminados** | **Plan** | **Actual** | | | **Hasta la fecha** | | **% Hasta la fecha** | | | **Def./Hora** |
| Planificación | - | - | | | - | | - | | |  |
| Diseño | - | 5 | | | 5 | | 23% | | |  |
| Codificación | - | 10 | | | 15 | | 68% | | |  |
| Revisión del código | - | - | | | - | | - | | |  |
| Compilación | - | - | | | 2 | | 9% | | | 1.8 |
| Pruebas | - | - | | | - | | - | | | 2.3 |
| Total | - | 15 | | | 22 | | 100% | | |  |

## **Código fuente según estándar.**

|  |  |
| --- | --- |
| 1. Clases (UpperCamel case) | CalculoIMC |
| 1. Variables (lowerCamel case) | edad |
| 1. Comentarios // | //Asociamos los componentes de la vista |
| 1. Cajas de texto (jText) | txtEdad |
| 1. Botones (jButton) | btnCalcularIMC |

**public class** MainActivity **extends** AppCompatActivity {  
 **private** Button **btnSalir**;  
 **private** Button **btnQueEs**;  
 **private** Button **btnCalcularIMC**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *//Evento para ir a la descripción detallada del programa* **btnQueEs** = (Button) findViewById(R.id.***btnQueEs***);  
 **btnQueEs**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View view) {  
 Intent i = **new** Intent(MainActivity.**this**, Informacion.**class**);  
 startActivity(i);  
 }  
 });  
 *//Evento para calcular* **btnCalcularIMC** = (Button) findViewById(R.id.***btnCalcularIMC***);  
 **btnCalcularIMC**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View view) {  
 Intent i = **new** Intent(getApplicationContext(), CalculoIMC.**class**);  
 startActivity(i);  
 }  
 });  
 *//Evento para salir del sistemas* **btnSalir** = (Button) findViewById(R.id.***btnSalir***);  
 **btnSalir**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View view) {  
 finish();  
 }  
 });  
 }  
}

**public class** Informacion **extends** AppCompatActivity {  
 Button **btnCalcular**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_info***);  
 *//Redirige a la ventana de Calculo de IMC* Button btnCalcular = (Button) findViewById(R.id.***btnCalculo***);  
 btnCalcular.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent i = **new** Intent(getApplicationContext(), CalculoIMC.**class**);  
 startActivity(i);  
 }  
 });  
 }  
}

**public class** CalculoIMC **extends** Activity {  
  
 **public static** TextView *resultados*, *datos*;  
 **public int IMC** = 0;  
 **public** Button **calcular**;  
 **public** ImageView **figura**;  
 **public** EditText **pesoUser**, **alturaUser**, **edadUser**;  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_imc***);  
 *//Asociamos los componentes de la vista  
 resultados* = (TextView) findViewById(R.id.***txtResultado\_calculos***);  
 *datos* = (TextView) findViewById(R.id.***txtdDtos\_usuario***);  
 **calcular** = (Button) findViewById(R.id.***btnCalculadora***);  
 **figura** = (ImageView) findViewById(R.id.***imgSilueta***);  
 **pesoUser** = (EditText) findViewById(R.id.***txtPeso\_usuario***);  
 **alturaUser** = (EditText) findViewById(R.id.***txtAltura\_usuario***);  
 **edadUser** = (EditText) findViewById(R.id.***txtEdad\_usuario***);

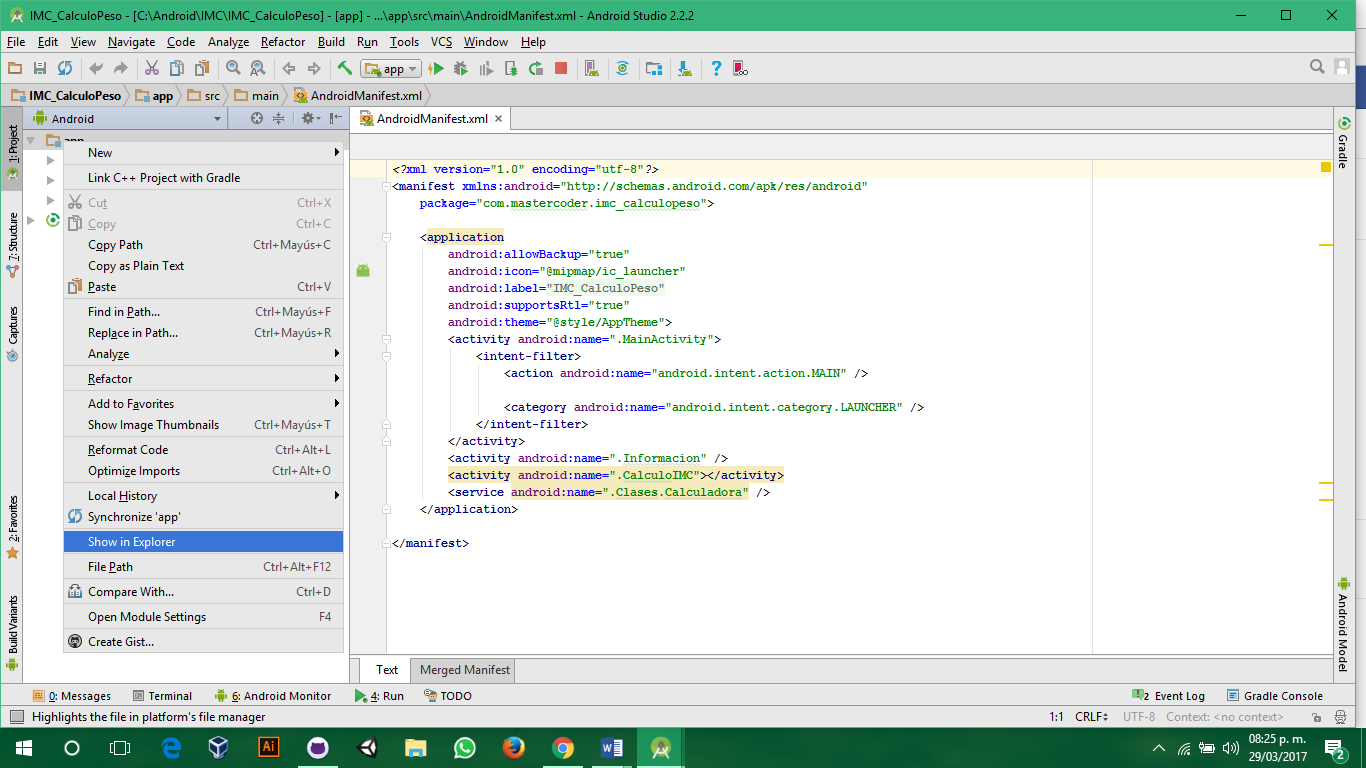
AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 builder.setTitle(R.string.***importante***);  
 builder.setMessage(R.string.***mensaje***);  
 builder.setPositiveButton(**"OK"**, **null**);  
 builder.create();  
 builder.show();  
 *//Ssociamos un anuncio broadcast a nuestro receptor de anuncios* IntentFilter filtro = **new** IntentFilter(ReceptorCalculadora.***ACTION\_RESP***);  
 filtro.addCategory(Intent.***CATEGORY\_DEFAULT***);  
 registerReceiver(**new** ReceptorCalculadora(), filtro);  
 *//Instanciamos el bóton para ejecutar la acción* **calcular**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//ponemos a 0 los datos  
 datos*.setText(**""**);  
 *resultados*.setText(**""**);  
 *//comprobamos que ha escrito algo* String peso\_texto, altura\_texto, edad\_texto;  
 peso\_texto = **pesoUser**.getText().toString();  
 altura\_texto = **alturaUser**.getText().toString();  
 edad\_texto = **edadUser**.getText().toString();  
 *//Si no se ha rellenado alguno saldrá de la función* **if** (peso\_texto.trim().equals(**""**) || altura\_texto.trim().equals(**""**) || edad\_texto.trim().equals(**""**)) {  
 Toast.*makeText*(getApplicationContext(), R.string.***error***, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 *//Guardamos en variables los datos introducidos por el usuario* **int** edad = Integer.*parseInt*(**edadUser**.getText().toString());  
 **int** altura = Integer.*parseInt*(**alturaUser**.getText().toString());  
 *//Convertimos la altura a float* **float** altura\_m = (**float**) (altura / 100.00);  
 **float** peso = Float.*parseFloat*(**pesoUser**.getText().toString());  
 *//Escondemos el teclado* InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  
 inputMethodManager.hideSoftInputFromWindow(**calcular**.getWindowToken(), 0);  
  
 *//Creamos una nueva intenci�n con nuestro contexto y la clase Calculadora* Intent intent = **new** Intent(CalculoIMC.**this**, Calculadora.**class**);  
 *//incluimos los datos que enviaremos al servicio* intent.putExtra(**"edad\_usu"**, edad);  
 intent.putExtra(**"altura\_usu"**, altura\_m);  
 intent.putExtra(**"peso\_usu"**, peso);  
 *//Arrancamos servicio* startService(intent);  
  
 *//Borramos los datos introducidos* limpiarCajasTexto();  
 *datos*.append(**"Edad: "** + edad + **" años\n"** + **"Peso: "** + peso + **" kg.\n"** + **"Altura: "** + altura\_m + **" m.\n"**);  
 }  
 });  
 }  
 **private void** limpiarCajasTexto()  
 {  
 **pesoUser**.setText(**""**);  
 **alturaUser**.setText(**""**);  
 **edadUser**.setText(**""**);  
 }  
 *//Una vez que el servicio ha concluido su trabajo queremos que avise a esta actividad  
 // y le devolva el valor resultante. Lo haremos medio de un anuncio broadcast.* **public class** ReceptorCalculadora **extends** BroadcastReceiver {  
 **public static final** String ***ACTION\_RESP*** = **"com.mastercoder.imc\_calculopeso.action.RESPUESTA\_OPERACION"**;  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 *//recibimos el resultado del calculo* **int** imc\_usuario = intent.getIntExtra(**"imc"**, 0);  
 **int** resultado\_usuario = intent.getIntExtra(**"resultado"**, 3);  
 **if** (resultado\_usuario == 0) {  
 *resultados*.append(**"Su índice de masa corporal es: "** + imc\_usuario + **", y está dentro de los límites normales"**);  
 **figura**.setVisibility(View.***VISIBLE***);  
 **figura**.setImageResource(R.drawable.***normal***);  
 } **else if** (resultado\_usuario == 1) {  
 *resultados*.append(**"Su índice de masa corporal es: "** + imc\_usuario + **", está por debajo de los límites normales. Necesita ganar peso"**);

**figura**.setVisibility(View.***VISIBLE***);  
 **figura**.setImageResource(R.drawable.***infrapeso***);  
 } **else if** (resultado\_usuario == 2) {  
 *resultados*.append(**"Su índice de masa corporal es: "** + imc\_usuario + **", está por encima de los límites normales. Necesita perder peso"**);**figura**.setVisibility(View.***VISIBLE***);  
 **figura**.setImageResource(R.drawable.***sobrepeso***);  
 }  
 }  
 }  
}

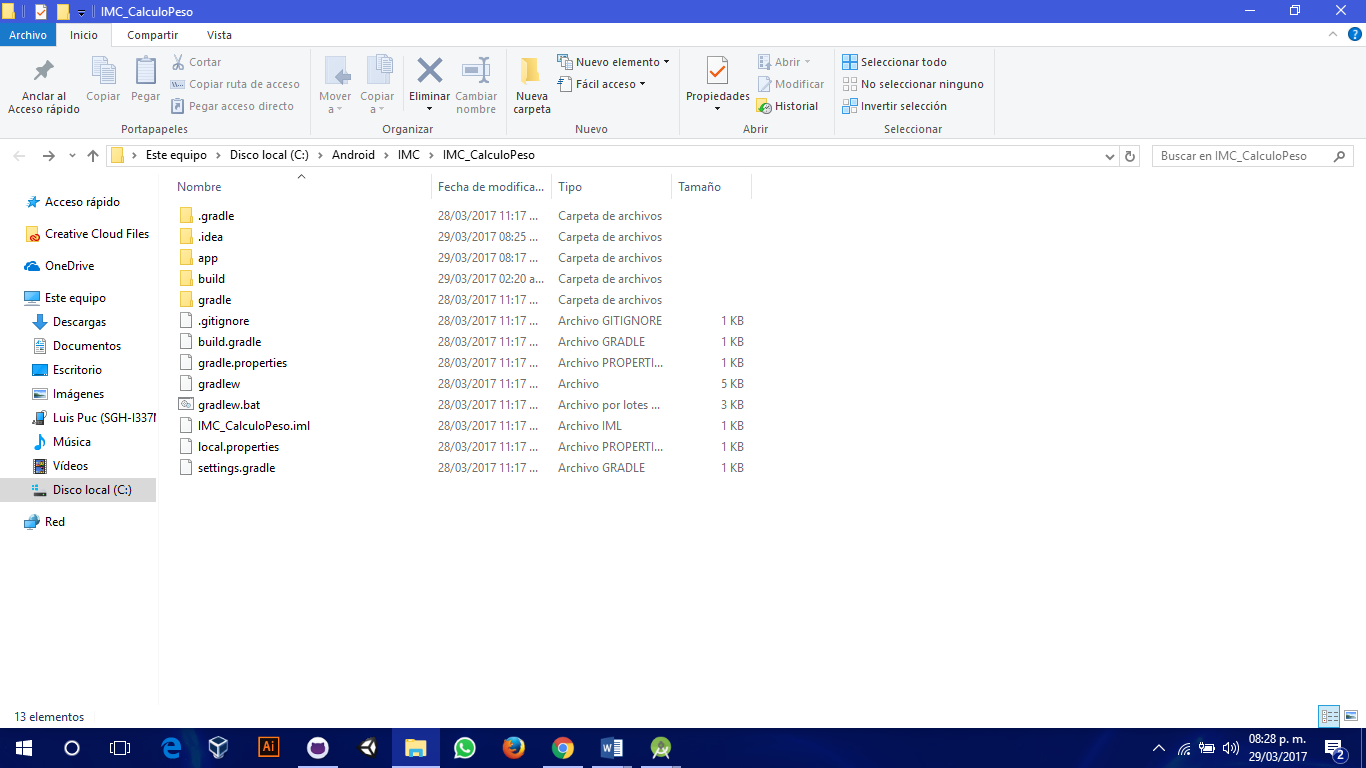
**public class** Calculadora **extends** IntentService{  
 **int edad\_usuario**;  
 **float altura\_usuario**;  
 **float peso\_usuario**;  
 **int imc**;  
 **int resultado**;  
  
  
 **public** Calculadora() {  
 **super**(**"Calculadora"**);  
 *//* ***TODO Auto-generated constructor stub*** }  
  
  
 @Override  
 **protected void** onHandleIntent( ) {  
 **edad\_usuario** = i.getExtras().getInt(**"edad\_usu"**);  
 **altura\_usuario** = i.getExtras().getFloat(**"altura\_usu"**);  
 **peso\_usuario** = i.getExtras().getFloat(**"peso\_usu"**);  
 **imc** = (**int**) (**peso\_usuario**/(**altura\_usuario**\***altura\_usuario**));  
  
 *//Averiguamos si el imc esta dentro de los valores normales según la edad* **if** (**edad\_usuario**>=19&&**edad\_usuario**<=24){  
 **if** (**imc**>=19&&**imc**<=24){  
 **resultado**=0;  
 }  
 **else if** (**imc**<19){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
 **else if** (**edad\_usuario**>=25&&**edad\_usuario**<=34){  
 **if** (**imc**>=20&&**imc**<=25){  
 **resultado**=0;  
 }  
 **else if** (**imc**<20){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
 **else if** (**edad\_usuario**>=35&&**edad\_usuario**<=44){  
 **if** (**imc**>=21&&**imc**<=26){  
 **resultado**=0;  
 }  
 **else if** (**imc**<21){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
 **else if** (**edad\_usuario**>=45&&**edad\_usuario**<=54){  
 **if** (**imc**>=22&&**imc**<=27){  
 **resultado**=0;  
 }  
 **else if** (**imc**<22){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
 **else if** (**edad\_usuario**>=55&&**edad\_usuario**<=64){  
 **if** (**imc**>=23&&**imc**<=28){  
 **resultado**=0;  
 }  
 **else if** (**imc**<23){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
 **else if** (**edad\_usuario**>=65){  
 **if** (**imc**>=24&&**imc**<=29){  
 **resultado**=0;  
 }  
 **else if** (**imc**<24){  
 **resultado**=1;  
 }  
 **else**{  
 **resultado**=2;  
 }  
 }  
  
 *//Enviamos el resultado a la actividad* Intent intent = **new** Intent();  
 intent.setAction(ReceptorCalculadora.***ACTION\_RESP***);  
 intent.addCategory(Intent.***CATEGORY\_DEFAULT***);  
 intent.putExtra(**"imc"**, **imc**);  
 intent.putExtra(**"resultado"**, **resultado**);  
 sendBroadcast(intent);  
 }  
}

## **Aplicación ejecutable e instrucciones para ejecutar.**

### Ejecutable desde dispositivo:

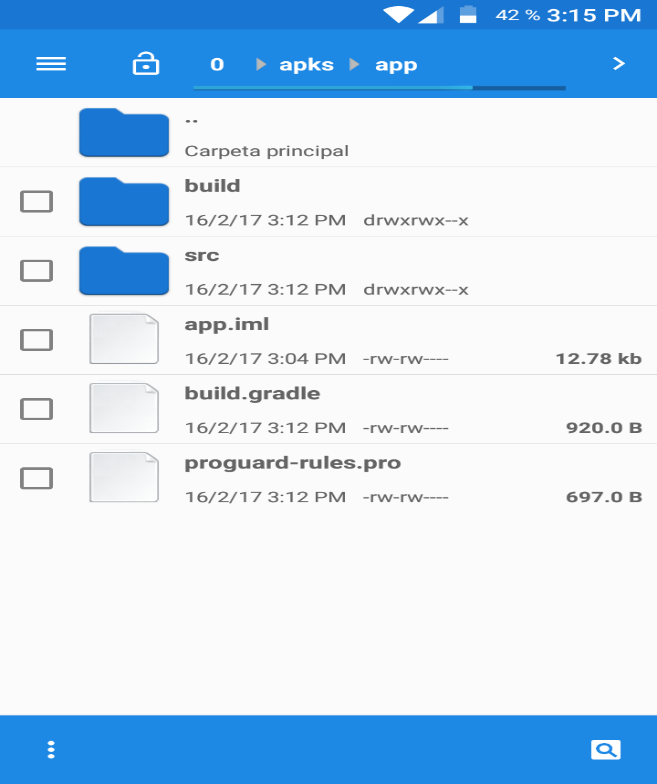


Paso 1: Como primer paso debemos presionar clic derecho a la carpeta app como se muestra en la imagen, seguido de esto aparecerá un menú de opciones donde debemos buscar la sección de “Show in Explorer” y darle clic.

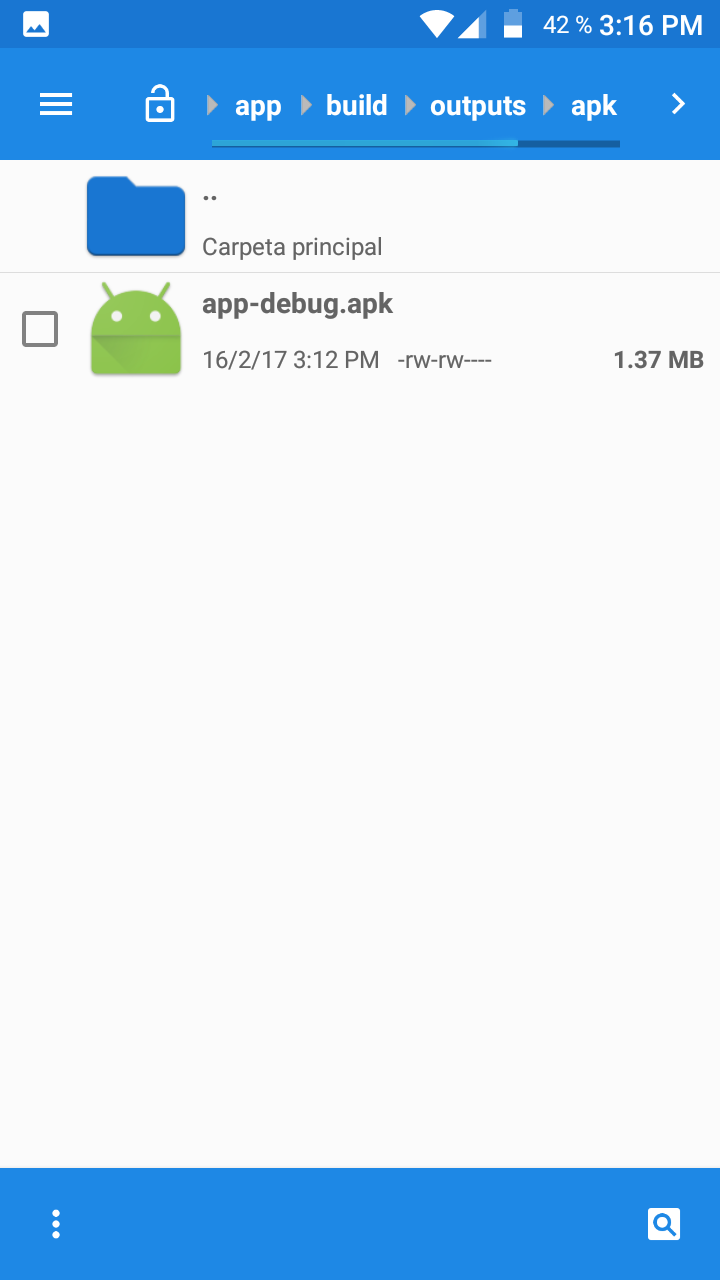


Paso 2: Después del paso 1 se visualizara el explorador de archivos donde se encuentra la carpeta del proyecto, buscamos la carpeta llamada “app” y la copiamos y pegamos en algún lugar que nosotros queremos dentro de la memoria interna del dispositivo Android.

Nota: Debes tener conectado tu teléfono celular a la computadora para realizar el copiado y pegado de la carpeta “app”.

Paso 3: Ya en el dispositivo móvil, debemos encontrar el dónde guardamos la carpeta “app”, una vez encontrada debemos entrar a la carpeta “build” seguido de esto aparecerá una seria de carpeta encontrar la carpeta llamada “outputs” e ingresar a ella.

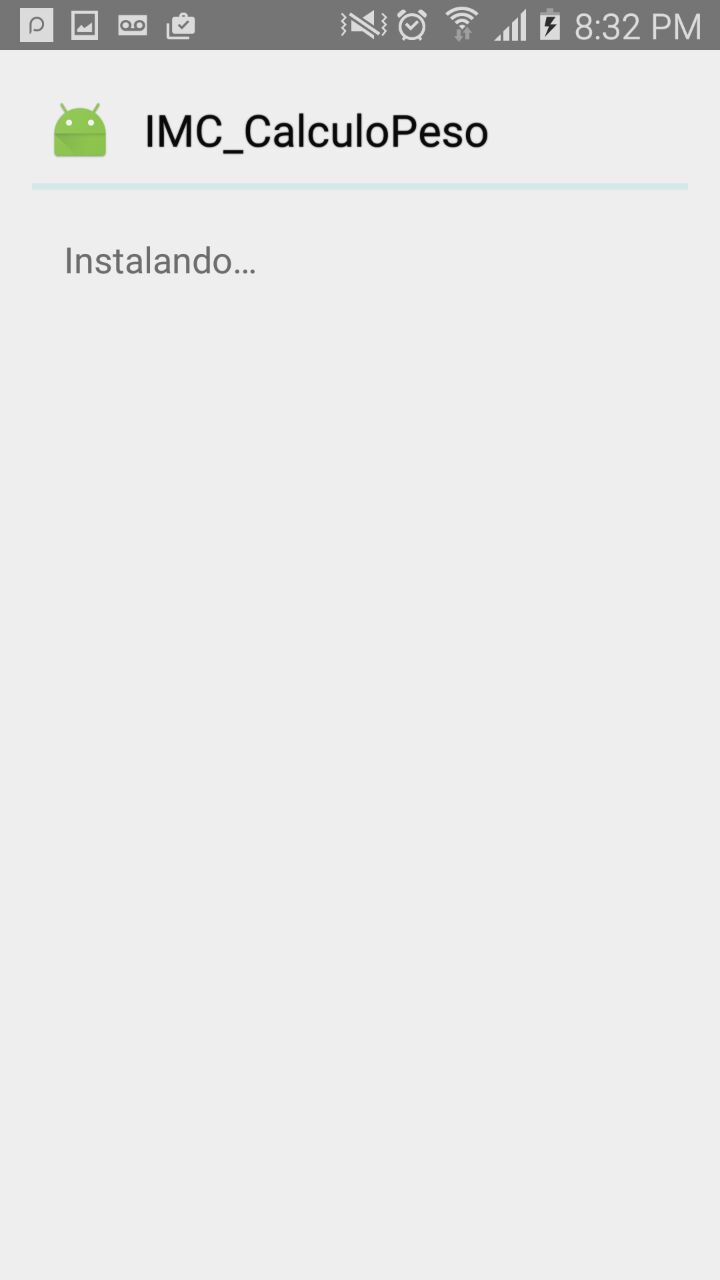
Paso 4: Después de eso aparecerán otras carpetas más nosotros ingresaremos a la carpeta “apk”.

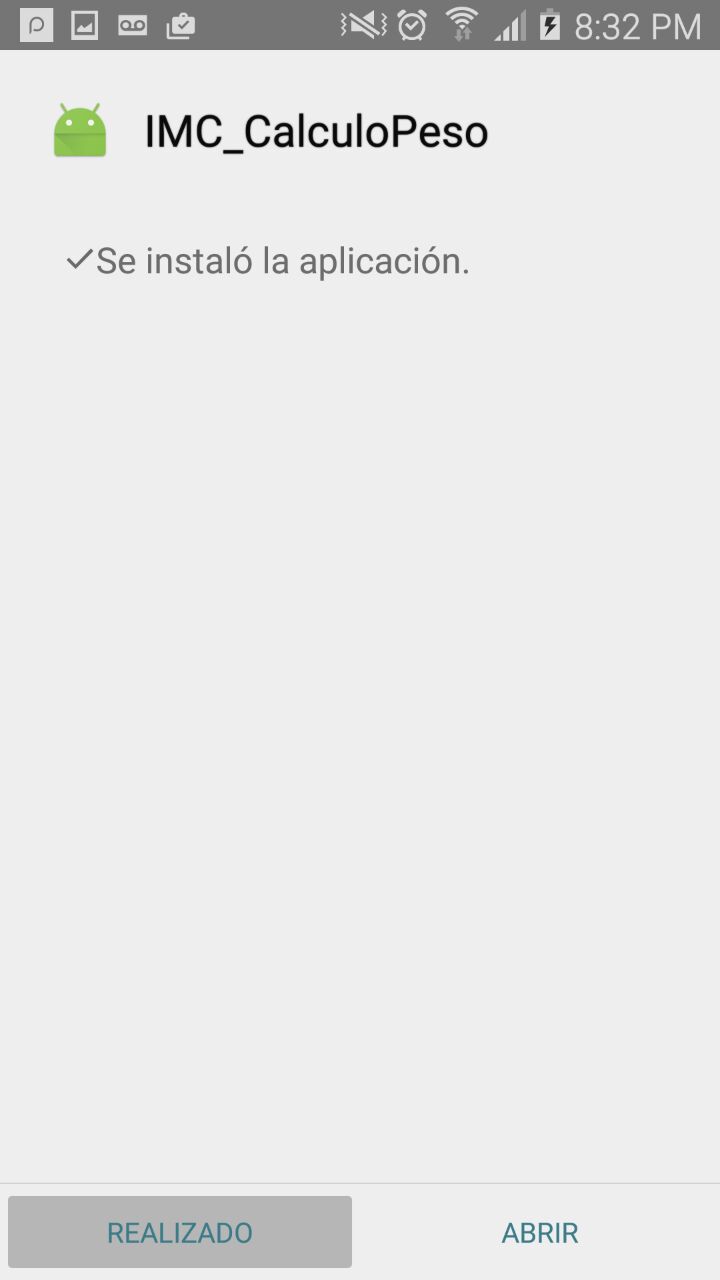


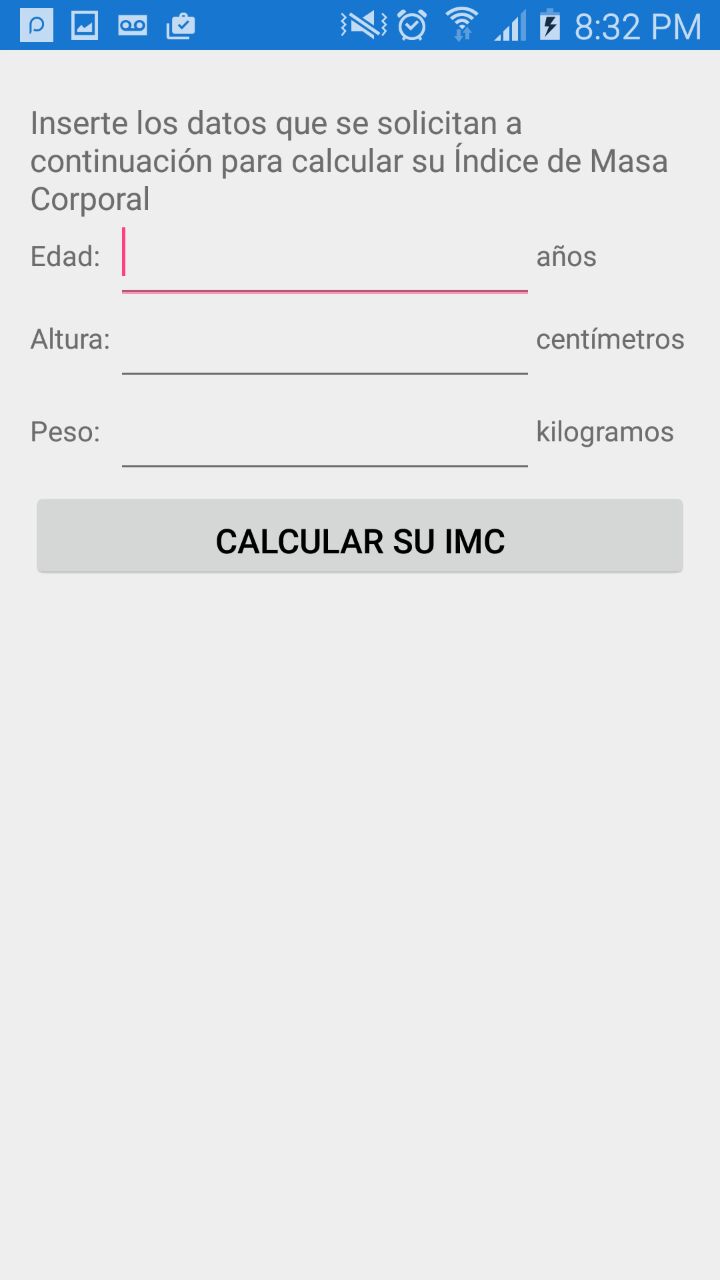
Paso 5: Después de realizar los anteriores pasos de forma adecuada veremos un archivo en dentro de la carpeta “apk” llamado:

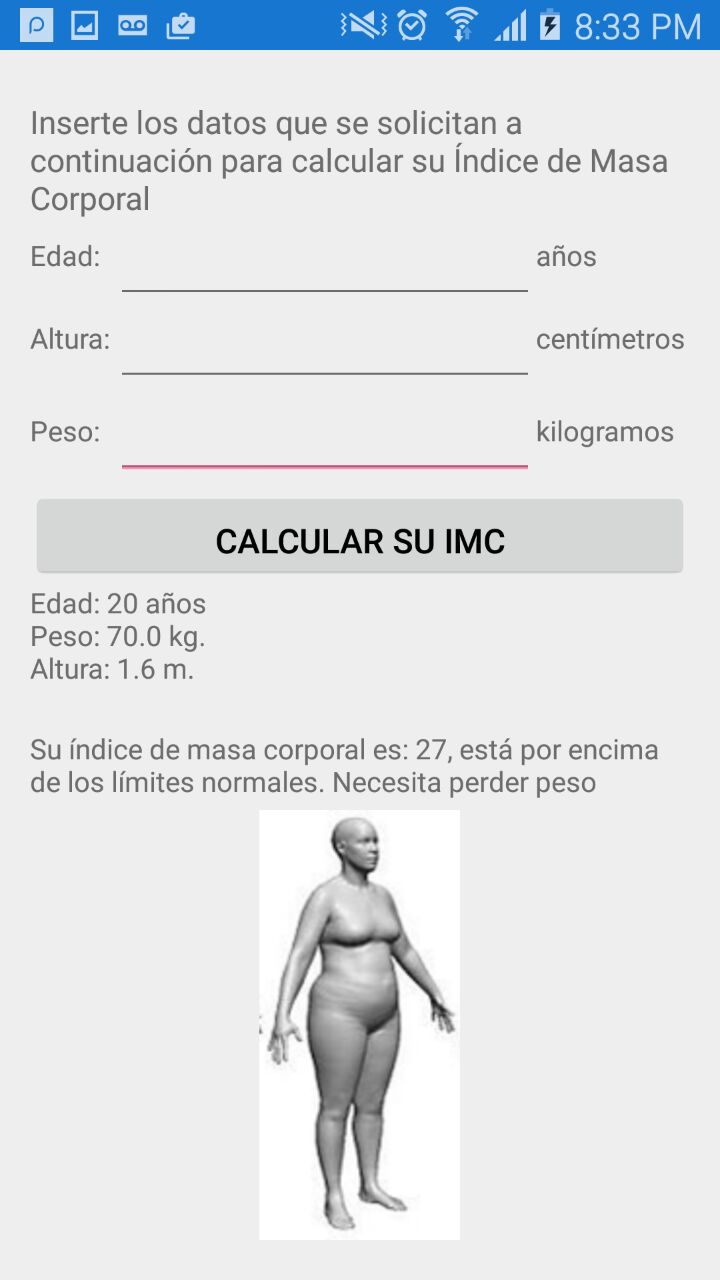
app-debug.apk

Este es el nombre del archivo a instalar en el dispositivo.

Paso 6: Cuando le demos clic al archivo app-debug.apk se visualizará una pantalla como esta (esta varía según la versión de Android y dispositivo que tenga), le damos donde dice instalar y esperamos a que se termine la instalación.

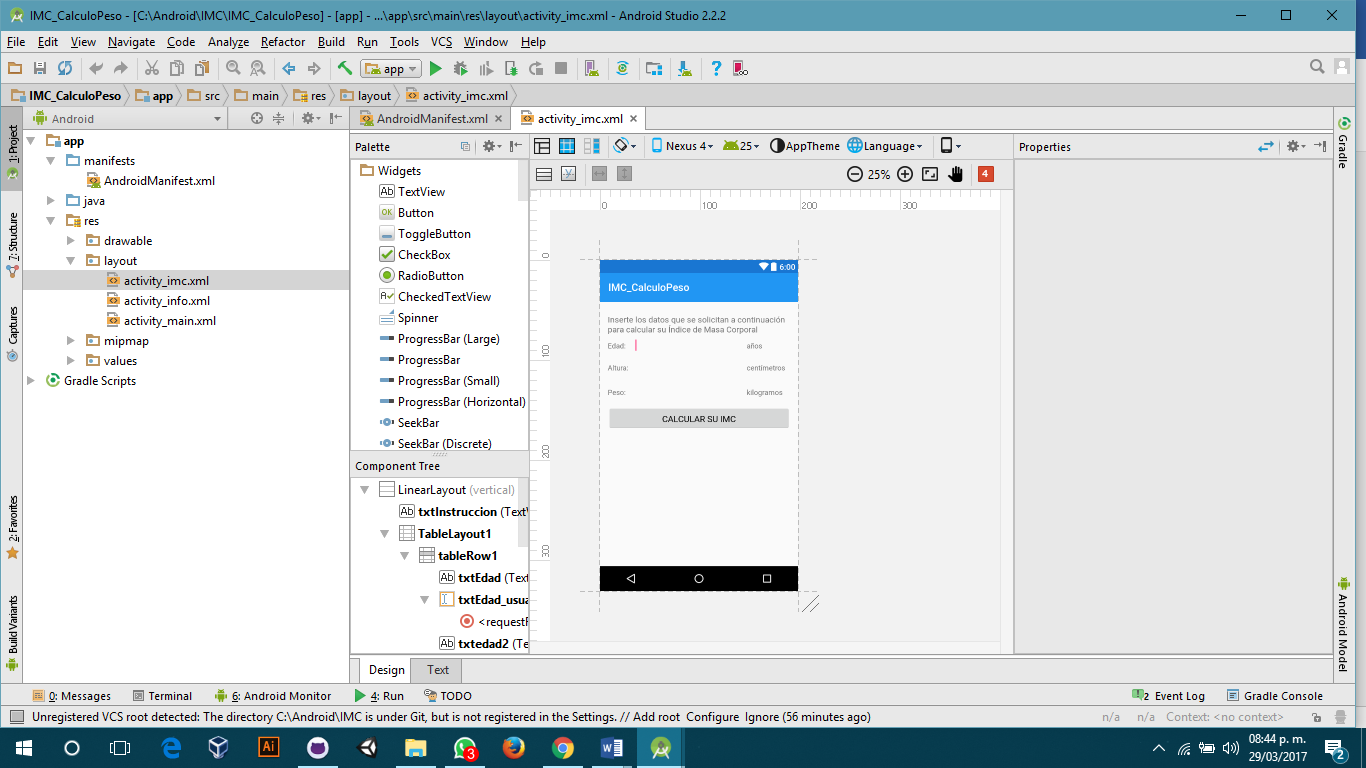
Paso 7: Una vez finalizada la instalación te notificara de que está a sido instalada, luego clicamos en donde dice abrir.

Paso 8: Se muestra la aplicación instalada en el dispositivo. Con el diseño que se le ha otorgado por los desarrolladores.

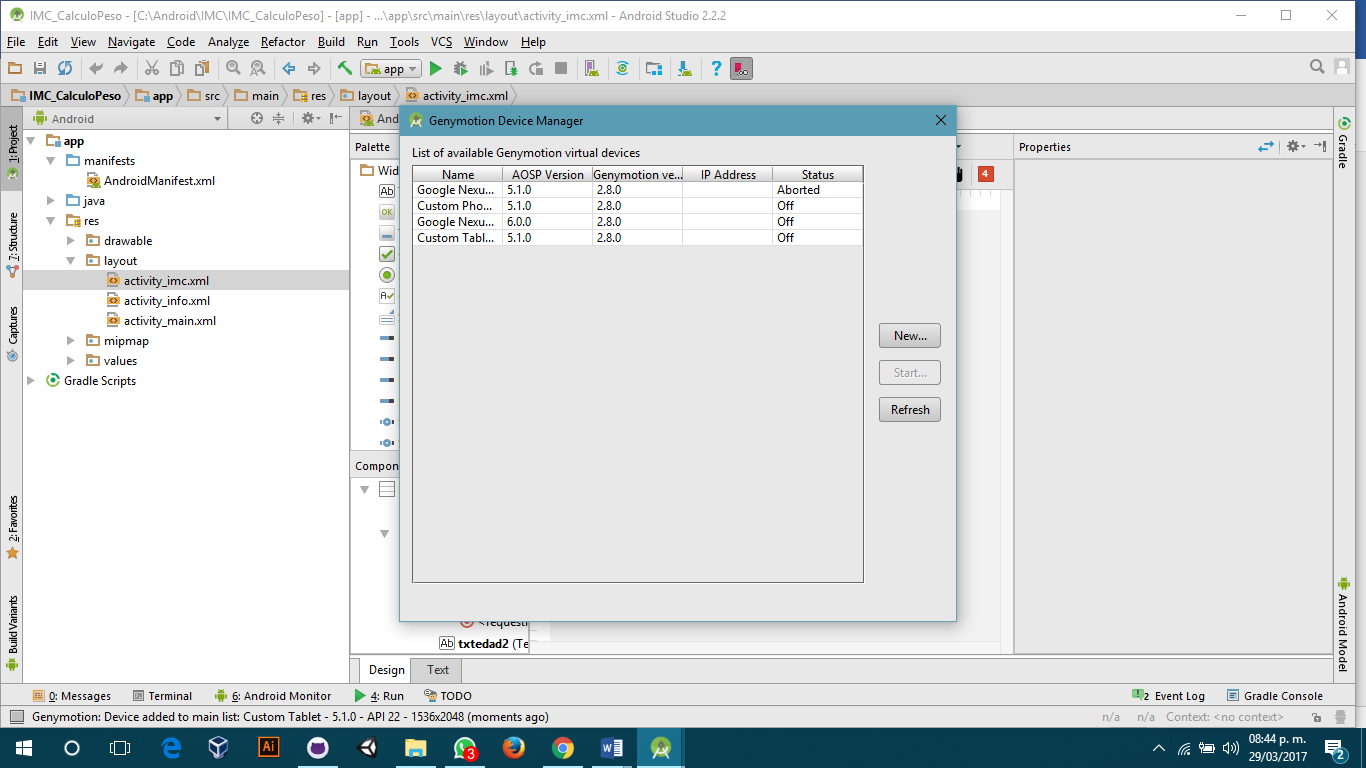


Paso 9: Se prueba la ejecución del programa donde debemos ingresar valores donde nos indica, en este caso se ingresaron los valores mostrados en la imagen. Una vez ingresado los datos le daremos clic al botón llamado “CALCULAR SU IMC” y según el proceso de codificación y especificaciones dadas para este proyecto se mostrará un detalle del Índice de masa corporal.

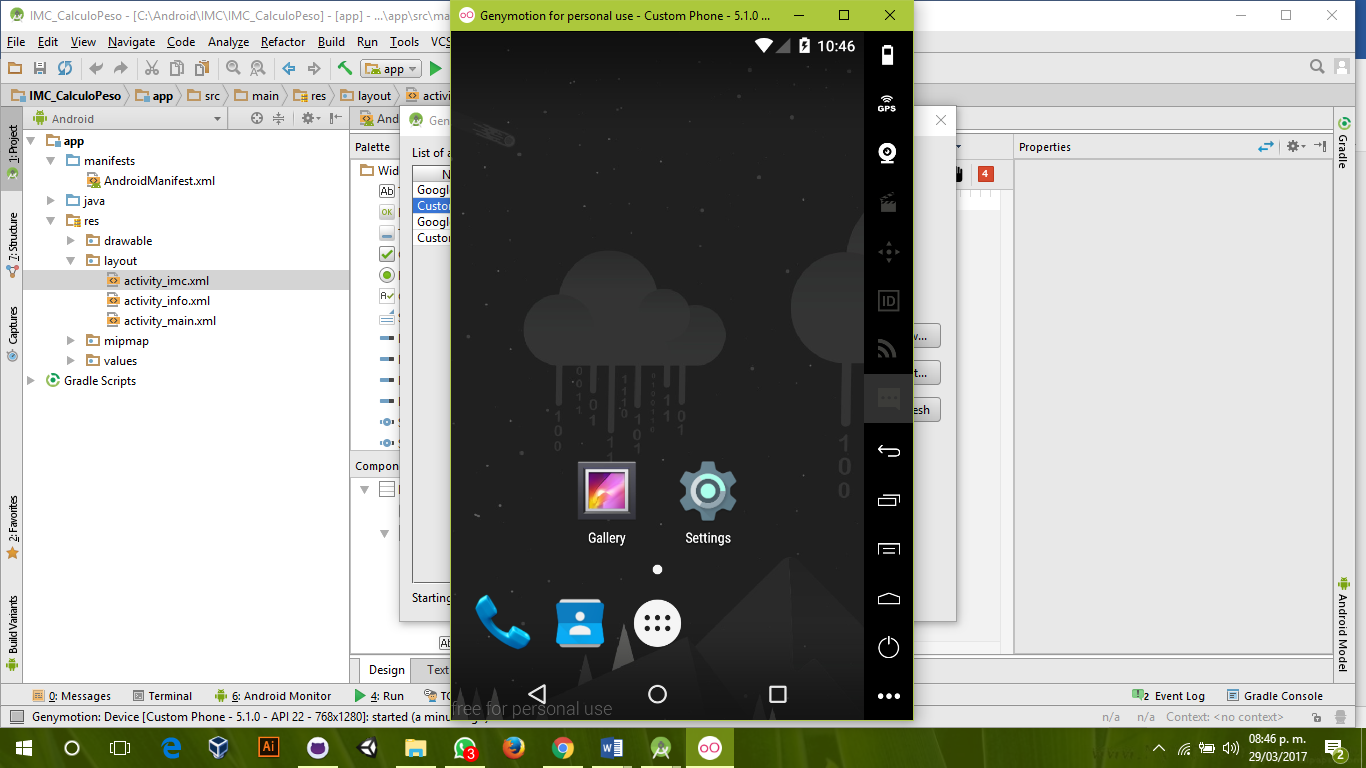
### Ejecutable desde emulador:



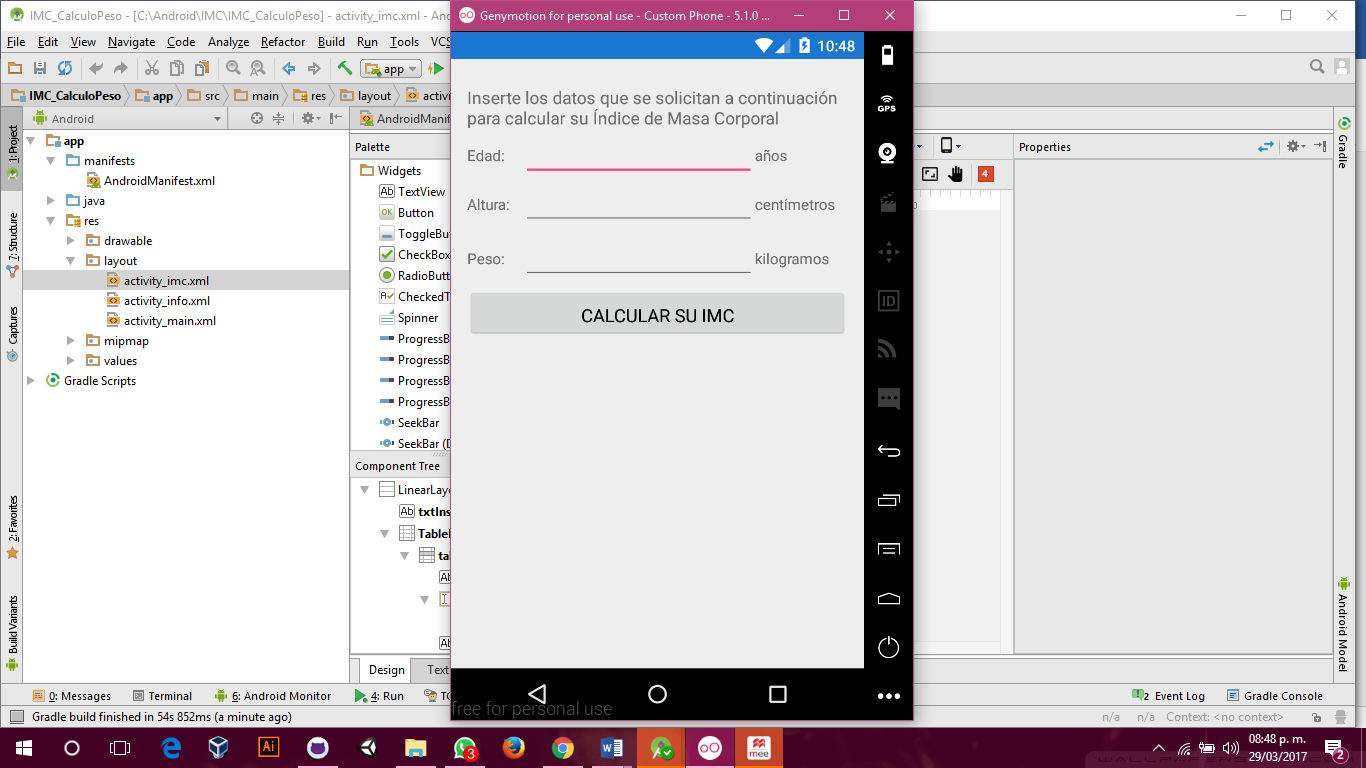
Paso 1: Abrir el Android Studio.



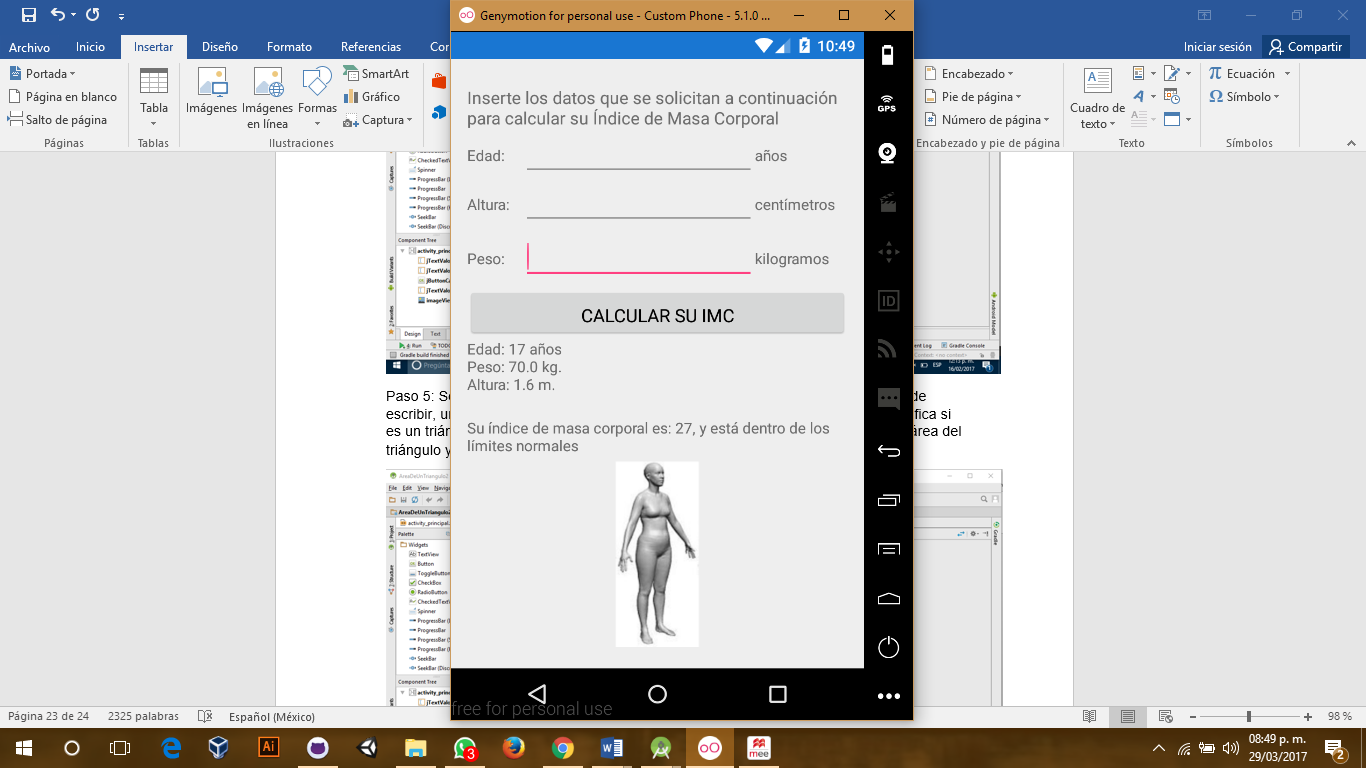
Paso 2: Abrir el emulador que en este caso usamos Genymotion.



Paso 3: Ejecutar el programa y seleccionar el emulador de su preferencia, para este caso se toma el primer emulador que ofrece genymotion.



Paso 4: Una vez realizado esos pasos, se visualiza el dispositivo o emulador que genymotion con la interfaz desarrollada.



Paso 5: Se ingresan valores en las cajas de números que la aplicación te dice donde escribir, una vez ingresado muestra los resultados del cálculo.

# Conclusiones

José Luis Puc Chan

Podemos concluir que la disciplina en el proceso de desarrollo de software es, sin lugar a dudas, uno de los elementos fundamentales para tal propósito debemos comenzar a entenderla y aplicarla, por lo cual la metodología PSP es parte de un proceso cíclico que ayuda al mejoramiento personal de las actividades. Para el caso particular ayudo al proceso desarrollo de la aplicación móvil implementada, pruebas, planeación y herramientas que ayudan a la gestión de todo esto.

Con PSP, el trabajo que requiere de técnicas superiores va siendo más común día a día. Asimismo, exige que el desempeño personal del desarrollador sea cada vez de mayor calidad.

Santiago Martinez

Con los Procesos Personal de desarrollo de software me di cuenta que nos permite mejorar la forma en la que construimos el software tomando en cuenta la planeación, calidad, estimación de costos y de productividad y con esto tenemos el control de nuestros propios proyectos, ya que el PSP nos da un marco de trabajo para desarrollar las habilidades personales y métodos que se necesita para ser un buen Ingeniero de software

Ya que el propósito de este mini proyecto fue el uso del PSP para ver quienes trabajan en el proyecto y el tiempo que lo hizo con el control de versiones y ver nuestros propios errores que teníamos durante la codificación y plasmarlo en el control de los “Logs” y luego poder repararlos.