

ActivityMain.java

```
1 package com.jmsr.extreme.mrnumero;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9
10 public class ActivityMain extends Activity
11 {
12     private Button calcular;
13     private EditText numero;
14
15     public void onCreate(Bundle savedInstanceState)
16     {
17         super.onCreate(savedInstanceState);
18         this setContentView(R.layout.main);
19
20         calcular = (Button) this.findViewById(R.id.calcular);
21         numero = (EditText) this.findViewById(R.id.numero);
22         calcular.setOnClickListener(new View.OnClickListener()
23         {
24             public void onClick(View v)
25             {
26                 calcular(Integer.parseInt(numero.getText().toString()));
27             }
28         });
29     }
30
31     private void calcular(int numero)
32     {
33         //Empieza la cuenta
34         long startTime = System.currentTimeMillis();
35         //Preparamos el futuro vector de números primos y lo llenamos a true
36         boolean[] esPrimo = new boolean[numero];
37         for (int i = 2; i < numero; i++)
38             esPrimo[i] = true;
39
40         //Criba de Eratóstenes para encontrar los números primos
41         for (int i = 2; i * i < numero; i++)
42         {
43             if (esPrimo[i])
44             {
45                 for (int j = i; i * j < numero; j++)
46                     esPrimo[i*j] = false;
47             }
48         }
49         //Contamos los primos
50         int primos = 0;
51         for (int i = 2; i < numero; i++)
52             if (esPrimo[i])
53                 primos++;
54
55         //Guardamos los primos en una lista
56         int[] lista = new int[primos];
57         int n = 0;
58         for (int i = 0; i < numero; i++)
59             if (esPrimo[i])
60                 lista[n++] = i;
61
```

M<sub>n</sub> Numero

ActivityMain.java

```

62      //Comprobamos si el número puede componerse como la suma de dos
        primos
63      int descomposiciones = 0;
64      int izquierda = 0;
65      int derecha = primos-1;
66
67      int primeraIzquierda = 0;
68      int primeraDerecha = 0;
69      int ultimaIzquierda = 0;
70      int ultimaDerecha = 0;
71
72      while (izquierda <= derecha)
73      {
74          if(lista[izquierda] + lista[derecha] == numero)
75          {
76              if(descomposiciones == 0)
77              {
78                  primeraIzquierda = lista[izquierda];
79                  primeraDerecha = lista[derecha];
80              }
81              ultimaIzquierda = lista[izquierda];
82              ultimaDerecha = lista[derecha];
83              descomposiciones++;
84              izquierda++;
85          }
86          else if(lista[izquierda] + lista[derecha] < numero)
87              izquierda++;
88          else derecha--;
89      }
90      //Terminamos la cuenta
91      long endTime = System.currentTimeMillis();
92      //Escribimos resultados
93      TextView descomposicionesView = (TextView)
        this.findViewById(R.id.descomposiciones);
94      descomposicionesView.setText("Descomposiciones: "+descomposiciones);
95      TextView primeraView = (TextView) this.findViewById(R.id.primera);
96      primeraView.setText(primerasIzquierda+" "+primeraDerecha);
97      TextView ultimaView = (TextView) this.findViewById(R.id.ultima);
98      ultimaView.setText(ultimaIzquierda+" "+ultimaDerecha);
99      TextView tiempoView = (TextView) this.findViewById(R.id.tiempo);
100     tiempoView.setText("Tiempo: "+(endTime-startTime)+" milisegundos");
101 }
102 }

```