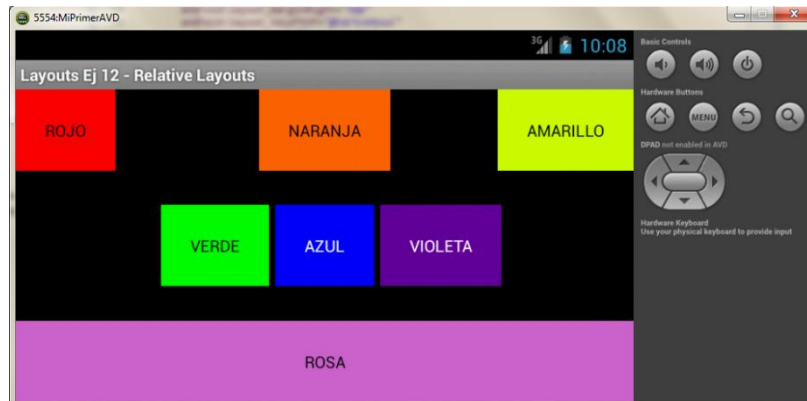


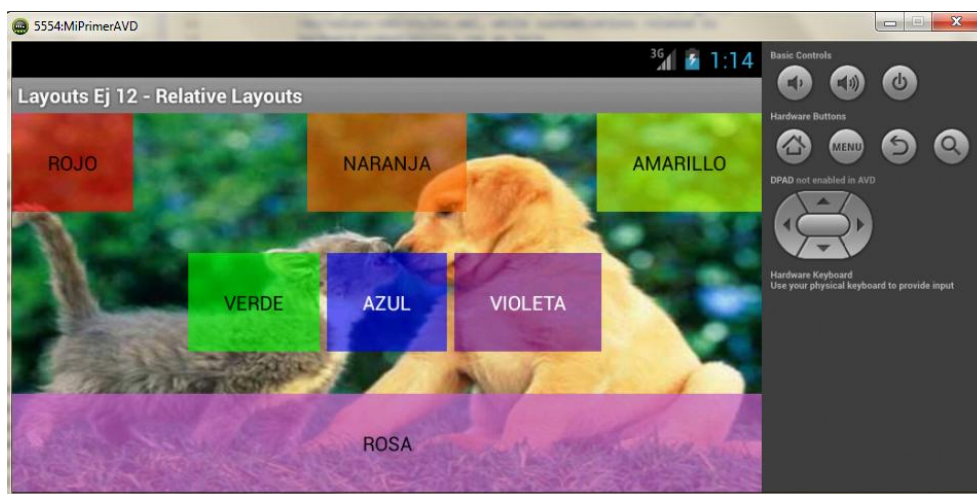
## INTERFAZ DE USUARIO: LAYOUTS

### **Ej. 12: RelativeLayout y su creación desde código Java y fondos**

1. Crear un nuevo proyecto Android, y definir un layout relativo (vale usar el que viene por defecto en el proyecto) de forma que quede igual que en la imagen, mediante el uso de TextViews. Se debe crear el layout usando únicamente reglas y propiedades del layout (no usar la interfaz gráfica para colocar los elementos).



2. Una vez hecho esto, intentar conseguir lo siguiente. La imagen puede ser cualquiera (podemos descargarla). La imagen deberemos ponerla en el lugar correspondiente y acceder a ella como un recurso. Para información sobre cómo especificar en canal alfa (la transparencia) buscar en la documentación de recursos tipo Color.



3. Una vez obtenido el layout, crear el layout desde código. En lugar de llamar al layout con setContentView especificando el id del layout, lo crearemos desde Java directamente  
Cosas a tener en cuenta:  
Debemos crear un objeto RelativeLayout. Para asignarle propiedades debemos crear un objeto LayoutParams y luego asignarlo al Layout:

```
RelativeLayout rl = new RelativeLayout(getApplicationContext());  
LayoutParams theLayoutParams = new LayoutParams(LayoutParams.MATCH_PARENT,  
                                                LayoutParams.MATCH_PARENT);  
rl.setLayoutParams(theLayoutParams);
```

Debemos crear un objeto que sea subtipo de View por cada elemento del layout. En este caso crearemos varios TextViews.

## INTERFAZ DE USUARIO: LAYOUTS

### **Ej. 12: RelativeLayout y su creación desde código Java y fondos**

Por ejemplo, para cada TextView debemos declarar un nuevo textview con `new TextView()`, pasando los parámetros correspondientes.

A cada vista que creemos debemos pasarle un objeto **RelativeLayout.LayoutParams**, porque si no no tendrá las propiedades necesarias para representarse en el RelativeLayout y no funcionará.

Este objeto es una clase interna de RelativeLayout (se puede ver por la sintaxis con el punto) y se puede crear pasándole por ejemplo un valor para la anchura y otro para la altura. En este ejemplo le pasamos dos constantes ya definidas (en realidad su valor es -1) que ya conocemos:

```
RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams (LayoutParams.MATCH_PARENT,
                                                                    LayoutParams.WRAP_CONTENT);
```

Una vez que tengamos el objeto creado, se le pueden cambiar propiedades y añadir reglas (consultar la documentación). Para asignar los LayoutParams al TextView hay que usar el método correspondiente.

Hay que darse cuenta de que hay propiedades de la propia vista y otras del layout, así que hay que buscarlas en el sitio correspondiente.

A cada textview le podemos asignar Ids (consultar cuál es el método correspondiente). En este caso, los Ids no van a estar en R, lógicamente, así que podemos definir constantes de tipo int en la actividad.

Acordarse de que los Id deben ser únicos.

Una vez creado cada objeto, hay que añadirlo al layout con el método (del layout) `addView`.