



# CREACIÓN DE UNA **ANDROID** APP 2

# FASE 4

Básicamente, en la fase 3 hemos resuelto la lógica de nuestra app, ahora, vamos a mejorar el planteamiento para que tengamos una mejor distribución y organización del código.

Comenzamos por crear una nueva clase de java, que será la encargada de alojar las frases. Yo la he llamado MisFrases

Dentro de la clase principal, crea un nuevo método y llámalo `public String cargarFrase()`

```
public String cargarFrases () {  
  
}
```

copiamos el array del MainActivity, así como el método que genera un número al azar y lo pegamos dentro. Luego devolvemos con `return` la frase aleatoria. (ver página siguiente)

# FASE 4

Cortamos:

```
String[] frases = {  
    "Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un  
    esfuerzo total es una victoria completa.\n Mahatma Gandhi",  
    "Si ya sabes lo que tienes que hacer y no lo haces entonces estás peor  
    que antes.\n Confucio",  
    "Buscando el bien de nuestros semejantes, encontramos el nuestro.\n  
    Platon",  
    "Es mejor ser rey de tu silencio que esclavo de tus palabras.\n William  
    Shakespeare"  
};  
  
String frase = "";  
Random alAzar = new Random();  
int numeroAlAzar = alAzar.nextInt(frases.length);  
  
frase = frases[numeroAlAzar];
```

y lo pegamos dentro del nuevo método.

# FASE 4

Finalmente añadimos:

```
return frase;
```

Para devolver la frase. No te preocupes si aparece ahora un error en el mainactivity, vamos a instanciar este nuevo objeto creado en el mainactivity. A l ahora de crear el objeto, es muy importante plantearlo de forma que se cargue una vez y dure la vida completa de la app, de lo contrario, si el objeto es muy grande, podemos tener problemas de rendimiento.

Usamos dentro de la clase principal y antes de cualquier método:

```
private MisFrases mMisFrases = new MisFrases();
```

Ahora que tenemos acceso a los métodos de esa clase, completamos el método onClick() del mainActivity con lo siguiente:

```
public void onClick(View v) {  
    String frase = mMisFrases.cargarFrases();  
    frasePortada.setText(frase);  
}
```

Los errores deben desaparecer y el main activity queda así:

# FASE 4

The screenshot displays the Android Studio IDE interface. The top toolbar includes menus for File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: Frases > app > src > main > java > com > example > david > frases > MainActivity.

The left sidebar contains the Project view, showing the directory structure of the 'app' module, including manifests, java, resources (res), and Gradle Scripts. The right sidebar shows the Maven Projects and Gradle tool windows.

The main editor window displays the following Java code for MainActivity.java:

```
package com.example.david.frases;

import ...

public class MainActivity extends AppCompatActivity {

    private MisFrases mMisFrases = new MisFrases();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final TextView frasePortada = (TextView) findViewById(R.id.frasePortada);
        Button botonPortada = (Button) findViewById(R.id.botonPortada);

        botonPortada.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String frase = mMisFrases.cargarFrases();

                frasePortada.setText(frase);
            }
        });
    }
}
```

The bottom of the IDE features the Android Monitor, which shows the emulator 'Emulator Nexus\_4\_API\_23' and the application 'com.example.david.frases (7278) [DEAD]'. The status bar at the very bottom indicates the connection status and the time '16:46' on '25/11/2015'.

# FASE 4

Esto funciona perfectamente, pero aun le podemos dar un retoque más. Vamos a convertir el array, en una propiedad del objeto, así, si en el futuro queremos añadir más elementos, como el autor... podemos hacerlo sin mucho esfuerzo. Ahora, nuestro código se ve así:

```
public class MisFrases {

    String[] frases = {

        "Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa.\n Mahatma Gandhi",

        "Si ya sabes lo que tienes que hacer y no lo haces entonces estás peor que antes.\n Confucio",

        "Buscando el bien de nuestros semejantes, encontramos el nuestro.\n Platon",

        "Es mejor ser rey de tu silencio que esclavo de tus palabras.\n William Shakespeare"

    };

    public String cargarFrases(){

        String frase = "";

        Random alAzar = new Random();

        int numeroAlAzar = alAzar.nextInt(frases.length);

        frase = frases[numeroAlAzar];

        return frase;

    }

}
```

# FASE 4

Vamos a tratar a la variable como tal y le añadimos un public. Además las variables de los objetos deben ser tratadas como variable miembros, por eso le hacemos un refactor y queda así:

```
public class MisFrases {  
  
    public String[] mFrases = {  
        "Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo  
total es una victoria completa.\n Mahatma Gandhi",  
        "Si ya sabes lo que tienes que hacer y no lo haces entonces estás peor que antes.\n Confucio",  
        "Buscando el bien de nuestros semejantes, encontramos el nuestro.\n Platon",  
        "Es mejor ser rey de tu silencio que esclavo de tus palabras.\n William Shakespeare"  
    };  
  
    public String cargarFrases(){  
        String frase = "";  
        Random alAzar = new Random();  
        int numeroAlAzar = alAzar.nextInt(mFrases.length);  
        frase = mFrases[numeroAlAzar];  
        return frase;  
    }  
}
```

# FELICIDADES

¡Hemos conseguido crear una app nativa para Android desde 0!

Puedes descargarte el proyecto finalizado desde la web de alumnos, para comparar tu código.

Aquí concluimos la parte obligatoria del curso. Ahora, si deseas completar tu formación y desafiarte, te proponemos un ejercicio:

Añade a nuestra app, una imagen del autor que cambie en función de la frase seleccionada.