

E0602 – CitasCelebres

En este ejercicio realizaremos dos modificaciones a nuestra aplicación citas célebres. Crear una nueva versión **E0602_CitasCelebres** para no alterar las anteriores como copia de la última versión.

1. **Añadir un Menú de Ayuda**
2. **Usar un fichero de recursos para cargar los autores y citas por defecto.**

Menú de Ayuda

Debéis añadir un menú Ayuda a los menús ya existentes en la aplicación.

Cuando el usuario seleccione este menú se abrirá una nueva Activity llamada **ayuda** que mostrará un scroll view con el contenido de un fichero de recursos ayuda.txt que redactaréis y guardaréis en la carpeta apropiada.

La actividad ayuda se limitará a cargar el contenido del fichero línea a línea y a mostrarlo en un scroll view en su layout.

Citas por defecto

Crearemos un **fichero de recursos** para almacenar los autores y sus citas, que tenemos ya definidas en la versión anterior. El fichero se llamará **citas.txt** (Éstos serán los autores y citas por defecto de nuestra aplicación. En versiones posteriores añadiremos autores y citas dinámicamente)

El fichero será texto plano y tendrá el siguiente formato (línea a línea) y no habrá autores repetidos. Cada autor tendrá, su nombre, la cita, la url de su biografía y el nombre sin extensión de su drawable.

```
<autor>
<cita>
<url>
<drawable>
<autor>
<cita>
<url>
<drawable>
...
```

1. Crear una Clase Autor

Crearemos una nueva clase java que gestionará las citas de un autor.

Datos Miembro

La clase tendrá:

- Un ArrayList llamado **citas** para almacenar las citas de este autor.
- Un String **nombre** que será el nombre del autor.
- Un String **url_biografia** que será la url de su biografía
- Un String **drawabe** que será el nombre sin extensión de su fichero drawable.
- Un int **refered**; que mantendrá el índice de la cita referenciada.

Constructor

El constructor recibirá como parámetros:

- El nombre del autor y lo almacenará en la variable nombre.
- La url de su biografía
- El nombre de su drawable

Inicializará **refered** a cero.

Métodos

void Add(String cita) : Añadirá una cita que se pasa como parámetro al ArrayList **citas**.

String getCita() : Devolverá una cita indicada por la variable estática **refered** que incrementará tras cada llamada. Si se llega a la última se colocará cero. Utilizar la operación módulo.

String getNombre(): Devolverá el nombre del autor.

String getUrl(): Devolverá la url de la biografía del autor.

String getDrawable(): Devolverá el nombre del drawable del autor.

2. Crear una Clase CitasCelebres

Crearemos una nueva clase java que gestionará la carga de los autores y las citas de éstos desde el fichero de recursos.

En versiones posteriores utilizaremos otros orígenes de datos.

Datos Miembro

Un List<Autor> llamado **autores** para almacenar una lista de objetos **Autor**.

Constructor

Al constructor habrá que pasarle this como contexto, la Activity.

Métodos

void loadResourceFile(void) :

El método leerá el fichero de recurso citas.txt para lo que deberá acceder al contexto para llamar al método **getResources()**

Conforme va leyendo línea a línea el fichero:

 Crearé un objeto Autor pasándole el nombre del autor leído del fichero.

 Llamaré a su método Add pasándole la cita leída del fichero.

 Añadiré el nuevo Autor a la lista **autores**

String[] getNombreAutores() :

El método devolverá un array de strings con los nombres de los autores.

String[] getUrlAutores() :

El método devolverá un array de strings con las urls de las biografías de todos los autores.

String getNombreAutor(int autor) :

El método devolverá el nombre del autor en la posición autor

String getUrlAutor(int autor) :

El método devolverá la url de la biografía del autor en la posición autor

String getDrawableAutor(int autor) :

El método devolverá el nombre del drawable del autor en la posición autor

String getCitaAutor(int autor) :

El método devolverá una cita del autor en la posición autor

Modificaciones

En la actividad principal de la aplicación, en el método `onCreate()` deberemos realizar la carga del fichero de recursos, por lo que deberemos instanciar un objeto de tipo `CitasCelebres` y llamar a su método `loadResourceFile()`;

Obtendremos un array de nombres de autores llamando al método `getNombreAutores()` del objeto `CitasCelebres`

Obtendremos un array de url de autores llamando al método `getUrlAutores()` del objeto `CitasCelebres`

Revisar el código para modificarlo apropiadamente que las citas vengan de lo leído desde el fichero y no desde los recursos de tipo string como en versiones anteriores.

Consideraciones

Debéis tener en cuenta estos aspectos:

- Los nombres de los botones se aplicarán dinámicamente a partir de los datos cargados desde el fichero `citas.txt`
- Se deben utilizar las urls de cargadas desde el fichero `citas.txt`
- Si no se ha pulsado ningún autor, los botones Biografía y Voz deberán mostrar un Toast que alerte de que primero hay que elegir autor.
- Cuando se invalida la activity (por ejemplo por ir a la activity de preferencias o biografía o por un cambio de orientación), cuando se recupera se deben seguir mostrando correctamente la cita y la imagen del último autor seleccionado.
 - Para garantizar esto y que las preferencias también se mantengan, utilizar las funciones
 - `onSaveInstanceState(Bundle savedInstanceState)`
 - `onRestoreInstanceState(Bundle savedInstanceState)`
 - `onResume()`
- No se debe oír el saludo inicial dos veces
- Procurar reutilizar código tanto como sea posible utilizando métodos de la clase accesibles por tanto desde cualquier método de la clase.

Siguiendo esta consideración, el código de uno de los botones de autores podría ser como el siguiente:

```
btn_autor1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        ix_autor=0;
        setFoto(ix_autor);
        setCita(ix_autor);
        if(b_speak_now){
            speakNow(txt_cita.getText().toString());
        }
    }
});
```

En el click se establece el índice del autor pulsado, se asigna su foto y su cita en base a ese índice y se procede a su locución en caso de estar activa la voz automática.