



UNIVERSIDAD
PONTIFICIA
DE SALAMANCA



Aplicación Android

Programación Aplicaciones

12/06/2020

Nuria Pacheco Sánchez

Universidad Pontificia de Salamanca
Salamanca

Manual del Programador

En esta parte se recoge la parte técnica de la aplicación que he desarrollado en Android. Todos los ficheros java.

Los primero ficheros que vamos a ver, son los ficheros Java. En primer lugar veremos, el fichero MainActivity.class

En la clase MainActivity extiende de la AppCompatActivity , he implementa los métodos onClickListener (que lo veremos más abajo) y el TextView.OnEditorActionListener.

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener, TextView.OnEditorActionListener {
```

Variables

edNombre-> Es una variable de tipo editText, donde se recogerá el nombre que nos pase el usuario en la TextBox (en android EditText)

```
private EditText edNombre;
```

Método onCreate -> Este método se ejecuta para cargar la lógica de básica de la aplicación en Android. En este método llámanos al método findViewById , que lo veremos más abajo

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

Método findViewById -> En este método nos encargamos de cargar las Views por sus ides que hemos puesto para trabajar java. También llámanos a los distintos listener

```
private void findViewById(){
    Button empezar = findViewById(R.id.button);
    this.edNombre = findViewById(R.id.ed);

    edNombre.setOnEditorActionListener(this);
    empezar.setOnClickListener(this);
}
```

En onClick-> Creamos un intent para poder pasar a la segunda pantalla, y le pasamos el nombre del usuario, a través de putExtra , llamamos start para que inicie esa actividad.

```
@Override
public void onClick(View v) {
    Intent intent = new Intent(this, SegundaPantalla.class);
    intent.putExtra("nombre", edNombre.getText().toString());
    startActivity(intent);
}
```

En EditorAction-> En este método a través del actionId cogemos la acción que tiene el botón de abajo del teclado, cuando se activa el EditText, si damos a este botón . Hace lo mismo que el botón empezar/Start ó Debut

```
@Override
public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
    switch (actionId) {
        case EditorInfo.IME_ACTION_DONE:
            Intent intent = new Intent(this, SegundaPantalla.class);
            intent.putExtra("nombre", edNombre.getText().toString());
            //intent.putExtra("contador", contador);
            startActivity(intent);
    }
}
```

```

        break;
    }
    return false;
}
}

```

El método ValidarNombre()-> Crea una notificación cuando el nombre introducido es vacío

```

public void ValidarNombre(){

    AlertDialog di = new AlertDialog.Builder(this)

        .setTitle(R.string.Label_nombrevacio)

        .setNeutralButton(getResources().getText(R.string.Label_volver),new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
            which) {

                }
            }).show();

        }
    }
}

```

Ahora veremos el Archivo SegundaPantalla.class.

```

public class SegundaPantalla extends AppCompatActivity implements
View.OnClickListener {

```

Variables

Inten -> Esta variable representa el número máximo de intentos que deja la sacar la palabra

contador -> Esta variable refleja el tiempo máximo que se tiene para averiguar la palabra

tvOculto -> TextView que contendrá la palabra en la pantalla de la app

tvNombre -> TextView que contiene el nombre del usuario

tvtiempo -> TextView que sacará el tiempo por pantalla de la app

tvintentos -> TextView que sacará el número de intentos que tenemos

pais -> Variable que recogerá en qué idioma está la aplicación

btA...btZ -> Recogen los botones que simulan el teclado de la app, estos botones están

metidos en GridLayout

inte -> Recoge el Intent que nos llega de la Activity Main

countDownTimer -> variable que se usa para CountTimer, es decir para crear el

temporizador

palabraOculto -> Es un objeto de Palabra

```

private static int Inten=5;
private long contador = 180000;
private TextView tvOculto;
private TextView tvNombre;
//TextView tIdio;
private TextView tvtiempo;
private TextView tvintentos;

```

```

    private String pais;
    private Button btÑ, btA
    , btB, btC, btD, btE, btF, btG, btH, btI, btJ, btK, btL, btM, btN, btO, btP, btQ, btR, btT, btU,
    btS, btV, btX, btY, btW, btZ;
    Intent inte;
    private CountdownTimer countdownTimer;
    private Palabra palabraOculta=null;

```

Metodo onCreate -> Este método se ejecuta para cargar la lógica de básica de la aplicación en android

En este método llámanos al método findViews y showedito , que lo veremos más abajo

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_segunda_pantalla);
    findViews();
    showEdito();
}

```

Metodo findViews -> En este método nos encargamos de cargar las Views por sus ids que hemos puesto para trabajar java. También llámanos a los distintos listener de los botones

```

public void findViews() {
    this.inte = getIntent();
    this.tvNombre = findViewById(R.id.tv_nombre);
    this.tvtiempo = findViewById(R.id.tv_tiempo);
    this.tvOculta = findViewById(R.id.tv_oculta);
    this.tvintentos = findViewById(R.id.tv_puntuacion);
    btA = findViewById(R.id.bt_a);
    btB = findViewById(R.id.bt_b);
    btC = findViewById(R.id.bt_c);
    btD = findViewById(R.id.bt_d);
    btE = findViewById(R.id.bt_e);
    btF = findViewById(R.id.bt_f);
    btG = findViewById(R.id.bt_g);
    btH = findViewById(R.id.bt_h);
    btI = findViewById(R.id.bt_i);
    btJ = findViewById(R.id.bt_j);
    btK = findViewById(R.id.bt_k);
    btL = findViewById(R.id.bt_l);
    btM = findViewById(R.id.bt_m);
    btN = findViewById(R.id.bt_n);
    btÑ = findViewById(R.id.bt_ñ);
    btO = findViewById(R.id.bt_o);
    btP = findViewById(R.id.bt_p);
    btQ = findViewById(R.id.bt_q);
    btR = findViewById(R.id.bt_r);
    btS = findViewById(R.id.bt_s);
    btT = findViewById(R.id.bt_t);
    btU = findViewById(R.id.bt_u);
    btV = findViewById(R.id.bt_v);
    btW = findViewById(R.id.bt_w);
    btX = findViewById(R.id.bt_x);
    btY = findViewById(R.id.bt_y);
    btZ = findViewById(R.id.bt_z);

    btA.setOnClickListener(this);
    btB.setOnClickListener(this);
    btC.setOnClickListener(this);
    btD.setOnClickListener(this);

```

```

btE.setOnClickListener(this);
btF.setOnClickListener(this);
btG.setOnClickListener(this);
btH.setOnClickListener(this);
btI.setOnClickListener(this);
btJ.setOnClickListener(this);
btK.setOnClickListener(this);
btL.setOnClickListener(this);
btM.setOnClickListener(this);
btN.setOnClickListener(this);
btÑ.setOnClickListener(this);
btO.setOnClickListener(this);
btP.setOnClickListener(this);
btQ.setOnClickListener(this);
btR.setOnClickListener(this);
btS.setOnClickListener(this);
btT.setOnClickListener(this);
btU.setOnClickListener(this);
btV.setOnClickListener(this);
btW.setOnClickListener(this);
btX.setOnClickListener(this);
btY.setOnClickListener(this);
btZ.setOnClickListener(this);

```

```

}

```

El método EliminarÑ*-> Este método lo llamamos en showEdito, en el caso Francés y en caso Ingles, la letra ñ del teclado aparece como no deshabilitado.

```

private void EliminarÑ(Button btÑ, String idioma) {
    switch (idioma){
        case "FR":
            btÑ.setEnabled(false);
            break;
        case "US":
            btÑ.setEnabled(false);
            break;
    }
}

```

En onClick-> Con los id de las vistas botón, vamos viendo que botón a pausado el usuario, vamos cogiendo el valor de ella y comprobamos a través del método de comprobar letra de palabra, después la palabra que nos devuelve , comprobamos si no contiene ("_") , puesto esto significa que hemos acertado toda la palabra , entonces salta un notificación como que hemos acertado la palabra oculta y paramos el tiempo.

* Si no estamos en el anterior caso, comprobamos que si la palabra nueva es distinta a la anterior, si es así modificamos en pantalla la palabra oculta con la letra que ha pulsado el usuario. Si no es ninguno de los dos anteriores, entonces la letra pausada es incorrecta, por lo tanto restamos un intento, y lo modificamos por pantalla.

```

@Override
public void onClick(View v) {
    String val;
    String anterior = this.tvOculta.getText().toString();
    String[] temp = tvintentos.getText().toString().split(":");
    int C = Integer.parseInt(temp[1]);

    switch (v.getId()){

```

```

        case R.id.bt_a:
            val =
palabraOculta.comprobarletra(btA.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();

                this.tvOculta.setText(val);
                countDownTimer.cancel();
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{

                CambiarIntentos(C);
            }
            break;
        case R.id.bt_b:
            val =
palabraOculta.comprobarletra(btB.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_c:
            val =
palabraOculta.comprobarletra(btC.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_d:
            val =
palabraOculta.comprobarletra(btD.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();

                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }

```

```

        }
        break;
    case R.id.bt_e:
        val =
palabraOculta.comprobarletra(btE.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();

            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_f:
        val =
palabraOculta.comprobarletra(btF.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_g:
        val =
palabraOculta.comprobarletra(btG.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_h:
        val =
palabraOculta.comprobarletra(btH.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;

```

```

        case R.id.bt_i:
            val =
palabraOculta.comprobarletra(btI.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_j:
            val =
palabraOculta.comprobarletra(btJ.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_k:
            val =
palabraOculta.comprobarletra(btK.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_l:
            val =
palabraOculta.comprobarletra(btL.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculta.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculta.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_m:
            val =

```



```

palabraOculta.comprobarletra(btM.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_n:
        val =
palabraOculta.comprobarletra(btN.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_ñ:
        val =
palabraOculta.comprobarletra(btÑ.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_o:
        val =
palabraOculta.comprobarletra(btO.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_p:
        val =
palabraOculta.comprobarletra(btP.getText().toString(),anterior);

        if(!val.contains("_")) {

```

```

        this.countDownTimer.cancel();
        this.tvOculto.setText(val);
        AciertaPalabra();
    }else if (!val.equals(anterior)) {
        this.tvOculto.setText(val);
    }else{
        CambiarIntentos(C);
    }
    break;
case R.id.bt_q:
    val =
palabraOculto.comprobarletra(btQ.getText().toString(),anterior);

    if(!val.contains("_")) {
        this.countDownTimer.cancel();
        this.tvOculto.setText(val);
        AciertaPalabra();
    }else if (!val.equals(anterior)) {
        this.tvOculto.setText(val);
    }else{
        CambiarIntentos(C);
    }
    break;
case R.id.bt_r:
    val =
palabraOculto.comprobarletra(btR.getText().toString(),anterior);

    if(!val.contains("_")) {
        this.countDownTimer.cancel();
        this.tvOculto.setText(val);
        AciertaPalabra();
    }else if (!val.equals(anterior)) {
        this.tvOculto.setText(val);
    }else{
        CambiarIntentos(C);
    }
    break;
case R.id.bt_s:
    val =
palabraOculto.comprobarletra(btS.getText().toString(),anterior);

    if(!val.contains("_")) {
        this.countDownTimer.cancel();
        this.tvOculto.setText(val);
        AciertaPalabra();
    }else if (!val.equals(anterior)) {
        this.tvOculto.setText(val);
    }else{
        CambiarIntentos(C);
    }
    break;
case R.id.bt_t:
    val =
palabraOculto.comprobarletra(btT.getText().toString(),anterior);

    if(!val.contains("_")) {
        this.countDownTimer.cancel();
        this.tvOculto.setText(val);
        AciertaPalabra();
    }

```

```

        }else if (!val.equals(anterior)) {
            this.tvOculto.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
        case R.id.bt_u:
            val =
palabraOculto.comprobarletra(btU.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculto.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculto.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_v:
            val =
palabraOculto.comprobarletra(btV.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculto.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculto.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_w:
            val =
palabraOculto.comprobarletra(btW.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculto.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculto.setText(val);
            }else{
                CambiarIntentos(C);
            }
            break;
        case R.id.bt_x:
            val =
palabraOculto.comprobarletra(btX.getText().toString(),anterior);

            if(!val.contains("_")) {
                this.countDownTimer.cancel();
                this.tvOculto.setText(val);
                AciertaPalabra();
            }else if (!val.equals(anterior)) {
                this.tvOculto.setText(val);
            }else{

```

```

        CambiarIntentos(C);
    }
    break;
    case R.id.bt_y:
        val =
palabraOculta.comprobarletra(btY.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    case R.id.bt_z:
        val =
palabraOculta.comprobarletra(btZ.getText().toString(),anterior);

        if(!val.contains("_")) {
            this.countDownTimer.cancel();
            this.tvOculta.setText(val);
            AciertaPalabra();
        }else if (!val.equals(anterior)) {
            this.tvOculta.setText(val);
        }else{
            CambiarIntentos(C);
        }
        break;
    }
}

```

El método cambiar Intentos -> Este método restamos los intentos , cuando llega a cero , salta una notificación como que se ha acabado el juego

```

private void CambiarIntentos(int intentos) {
    if (intentos > 1) {
        intentos -= 1;

        this.tvintentos.setText(getResources().getText(R.string.Label_puntuacion) +
String.valueOf(intentos));
    } else {

        this.tvintentos.setText(getResources().getText(R.string.Label_puntuacion) +
"0");
        CrearAlerta();
    }
}

```

CrearAlerta->Este método crea un alert, cuando el juego se acabado por tiempo o por intentos tiene un botón neutral , que se llama volver, donde si es pulsado nos vuelve a inicio de la app

```

public void CrearAlerta(){
    ImageView image = new ImageView(this);
    image.setImageResource(R.drawable.triste);

    AlertDialog di = new AlertDialog.Builder(this)

        .setTitle(R.string.Laber_fin)

```

```

        .setView(image)

        .setNeutralButton(getResources().getText(R.string.Label_volver), new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                countDownTimer = null;
                finish();
            }
        }).show();

        Button Neutral = di.getButton(AlertDialog.BUTTON_NEUTRAL);

        LinearLayout parent = (LinearLayout) Neutral.getParent();
        parent.setGravity(Gravity.CENTER_HORIZONTAL);
        View leftSpacer = parent.getChildAt(1);
        leftSpacer.setVisibility(View.GONE);
    }

```

Este método AciertaPalabra-> Crea una notificación , diciendo has acertado la palabra. Tiene dos opciones a la pregunta ¿Quiere volver a jugar?
un botón No -> Donde este vuelve al inicio de la aplicación
un botón Si -> Donde este vuelve a generarte una nueva palabra

```

    public void AciertaPalabra(){
        new AlertDialog.Builder(this)
            .setTitle(getResources().getText(R.string.Label_acierto))
            .setMessage(getResources().getText(R.string.Laber_pregunta))
            .setPositiveButton(R.string.button_Si, new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Intent intent = getIntent();
                overridePendingTransition(0, 0);
                finish();
                overridePendingTransition(0, 0);
                startActivity(intent);
            }
        })
            .setNegativeButton(R.string.button_No, new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        }).show();
    }

```

El metodo ShowEditor-> Se encarga de los cambios de los diferentes view al inicio de la aplicación. Es decir, si está habilitado o no el btÑ , muestra los intentos iniciales, el tiempo inicial, el nombre del usuario, crea "_" tanta longitud tenga la palabra oculta

```

    public void showEdito() {

        String nombre = getResources().getString(R.string.Label_nombre) + " "
        + inte.getStringExtra("nombre");
        this.tvNombre.setText(nombre);
    }

```

```

        this.pais = Locale.getDefault().getCountry();
        EliminarÑ(btÑ,pais);
        this.palabraOculata = new Palabra(pais);
        // String palabraOcu= palabraOculata.cargarPalabra(pais);
        String longitud = palabraOculata.longitudPalabraOculata();
        //Log.i("Palabra",palabraOcu);
        tvOculata.setText(longitud);

        this.tvintentos.setText(getResources().getText(R.string.Label_puntuacion).toString()+String.valueOf(Inten));
        ContarTiempo();

    }

```

ContarTiempo->Este método es el encargado de crear el tiempo a través del CountdownTimer, ir modificando este a través del método update.

```

    public void ContarTiempo(){

        countdownTimer = new CountdownTimer(contador,1000){

            @Override
            public void onTick(long millisUntilFinished) {
                contador = millisUntilFinished;
                updateText();
            }

            @Override
            public void onFinish() {

                CrearAlerta();
            }

        }.start();

    }

```

El método Update-> Hace el formato del tiempo y lo saca en el TextView tvtiempo

```

    private void updateText() {
        int minutes = (int) (contador / 1000) / 60;
        int seconds = (int) (contador / 1000) % 60;
        String timeLeftFormatted =
        getResources().getString(R.string.Label_tiempo)+" "
            +String.format(Locale.getDefault(), "%02d:%02d", minutes,
seconds);
        tvtiempo.setText(timeLeftFormatted);

    }

}

```

Ahora continuaremos viendo el fichero Palabra.class

```

public class Palabra {
    Variables
    * español/ingles/frances-> Es una lista que recogerá las palabras en español/ingles/frances
del juego
    * idioma-> Recogerá el idioma que este la aplicación
    * rand-> Es una variable para Random
    * oculta -> String que recogerá la palabra oculta
    * forma-> String para el formato de salida por pantalla
    private List<String> español;
    private List <String> ingles;
    private List <String> frances;
    private String idioma;
    private Random rand;
    private String oculta=null;
    private String linea;
    private String forma="";

```

constructor de la clase

Donde inicializaremos las listas, rand y asociamos el idioma que nos pasen

```

public Palabra(String idiom){
    this.idioma=idiom;
    frances = new ArrayList<String>();
    ingles = new ArrayList<String>();
    español = new ArrayList<String>();
    rand = new Random();
}

```

cargarPalabra -> Este método se encarga de cargar la listas , dependiendo del idioma que nos hayan pasado y elegirá una palabra oculta totalmente aleatoria

```

public String cargarPalabra () {
    int item;
    String palabraOculta=null;
    switch (idioma) {
        case "FR":
            frances.add("ESPAGNE");
            frances.add("COLOMBIE");
            frances.add("ITALIE");
            frances.add("GRECE");
            frances.add("ALLEMAGNE");
            item = rand.nextInt(frances.size());
            palabraOculta = frances.get(item);
            break;
        case "US":
            ingles.add("SPAIN");
            ingles.add("COLOMBIA");
            ingles.add("ITALY");
            ingles.add("GREECE");
            ingles.add("GERMANY");
            item = rand.nextInt(ingles.size());
            palabraOculta = ingles.get(item);
            break;
        case "ES":
            español.add("ESPAÑA");
            español.add("COLOMBIA");
            español.add("ITALIA");
            español.add("GRECIA");
            español.add("ALEMANIA");
            item = rand.nextInt(español.size());

```

```

        palabraOculto = español.get(item);
        break;
    }

    return palabraOculto;
}

```

Método comprobarLetra -> Este método comprobará si la letra que nos pasan es correcta, y está en la palabra y llamamos a método formato, para que salga el formato " _ _ _ _ ", que será la string que devuelva

```

public String comprobarLetra(String le, String anterior){
    String [] quitarEspacios = anterior.split(" ");
    char [] ocul = oculto.toCharArray();

    for (int i =0 ; i<ocul.length;i++) {
        if (ocul[i] == le.charAt(0)) {
            quitarEspacios[i]=le;
        }
    }
    String formato = formato(quitarEspacios);
    Log.i("Comprobacion formato",formato);
    return formato ;
}

```

En este método Formato -> Creamos el formato "_ A A _" del juego del ahorcado, devolvemos la string con el formato

```

private String formato(String[] quitarEspacios ) {
    forma="";
    for(int i=0; i<quitarEspacios.length;i++){
        if(quitarEspacios[i].equals("_")){
            forma+=" _"+" ";
        }else{
            forma+=quitarEspacios[i]+" ";
        }
    }
    return forma;
}

```

Método longitud Palabra, crea la palabra oculta y pondrá el formato " _ _ _ _ " inicial

```

public String longitudPalabraOculto(){
    oculto=cargarPalabra();
    Log.i("Palabra Oculta",oculto);
    linea="";
    int i =0;
    Log.i("Longitud",String.valueOf(oculto.length()));
    while (i<oculto.length()){

        linea+=" _"+" ";
        i++;
    }
    return linea;
}

```

```

}
}

```


Manual del usuario.

En este enlace esta el manual de usuario .

https://docs.google.com/document/d/18yaNGs6fPZ4OvTQP1RYgeMDpkys_SKBK5tu3oOer-zQ/edit?usp=sharing

Este es un video que se muestra el funcionamiento de la app.

<https://www.youtube.com/watch?v=qfKFO97N8cs&t=55s>

Bibliografía Usada:

<https://code.tutsplus.com/tutorials/android-user-interface-design-creating-a-numeric-keypad-with-gridlayout--mobile-8677>

<https://codinginflow.com/tutorials/android/countdowntimer/part-1-countdown-timer>

<https://www.youtube.com/watch?v=IjNa5LyieaE>

<https://stackoverflow.com/questions/17488766/refresh-activity-without-re-opening-it>

<https://stackoverflow.com>