

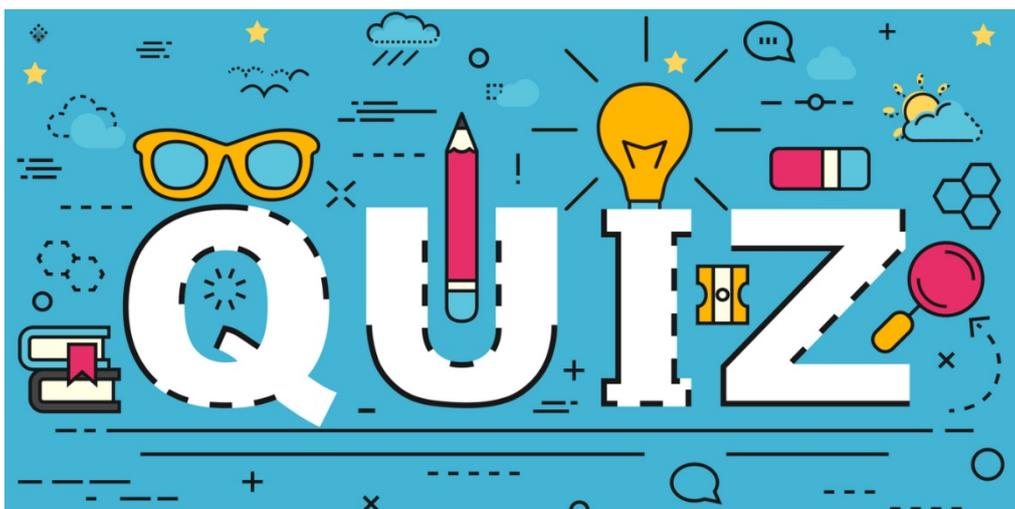
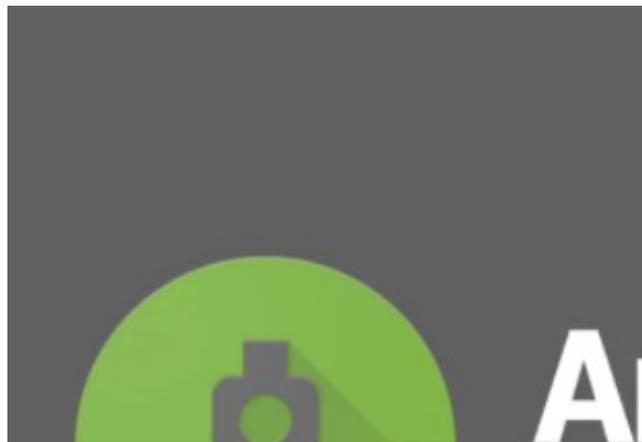


**ITIS-LS "Francesco Giordani" Caserta**

**Studente Angelo De Luca 5Binf**

**A.S. 2020-2021**

**Realizzazione Quiz In Android studio**



## Descrizione del progetto

Realizzare un quiz in ambiente Android Studio seguendo il "Format ideato dal prof. Ranucci" : ad ogni domanda vengono presentati 3 distrattori ed una risposta corretta. Se la risposta non è corretta viene riproposta la stessa domanda ed il punteggio di 4 punti viene diminuito di uno. Se la risposta è corretta viene proposto un approfondimento. Il gioco-quiz è finalizzato all'apprendimento di un argomento. Il gioco a quiz ha come finalità l'apprendimento: si passa alla domanda successiva solo se è stata scelta la risposta giusta ed è stato visualizzato il relativo approfondimento.

### Storia dell'azienda e della sua struttura organizzativa

L'Istituto iniziò la sua attività il 1° ottobre 1961 utilizzando i locali di palazzo Catemario, nei pressi di piazza Marconi e fu intitolato a Francesco Giordani, già titolare della cattedra di Chimica all'Università di Napoli. L'anno seguente furono utilizzati i locali di Villa Rosa, sull'Appia, ove fu istituito il primo laboratorio tecnologico, con attrezzature all'avanguardia, tant'è che i professori dell'Università di Napoli, come il prof. Cittadini, se ne servivano per eseguire prove sui materiali. L'Istituto intanto cresceva e furono utilizzati alcuni locali del Liceo Scientifico "A. Diaz" per collocare i reparti di Macchine utensili, Saldature ed Aggiustaggio. Inoltre veniva aperta una sede staccata a Carinola. L'Istituto industriale, sorto per formare Periti Meccanici, nel 1966 ampliò l'offerta formativa inaugurando la specializzazione per Elettrotecnici. A Villa Rosa fu allestita una Sala Macchine Elettriche e un laboratorio di Misure Elettriche. Furono utilizzati il Palazzo Vescovile, sul Corso Trieste, e il Palazzo Landolfi in Corso Giannone, per installare i Laboratori di Macchine Idrauliche e quelli di Aggiustaggio. Nel contempo si lasciava Villa Rosa, ormai fatiscente, e nuovi locali furono presi in locazione in via Ricciardelli e in via Tevere. Poi gli edifici dell'ex ACI divennero la sede principale dell'Istituto. In questi anni sedi staccate erano: Carinola, Marcianise, Capriati al Volturno, Alife, Piedimonte e Capua e fu istituito anche un corso serale. Dal 1973 al 1985, l'Istituto aprì la specializzazione di Elettronica. Si chiuse l'esperienza del corso serale e, finalmente, venne assegnata la nuova sede di via Laviano, progettata nel 1962.

### Presentazione degli stakeholder

prof Ennio Ranucci(informatica) e prof Margherita Puca(laboratorio informatica), prof. Tracanna Patrizia(Gestione Progetto, Organizzazione Impresa, Angelo De Luca (studente 5Binf 2020/2021).

### Il prospetto degli aspetti amministrativi e degli investimenti

Gli esercizi e le esercitazioni sono valutate anche dal punto di vista del costo prevedendo un costo orario di 23€ per il capo progetto, 4€ all'ora, per l'analista e 3€ all'ora per il programmatore.

### L'analisi del knowhow e prospetto delle risorse umane necessarie (team)

Android studio: i principali tag e XML

Java: strutture di controllo del flusso delle istruzioni e sottoprogrammi

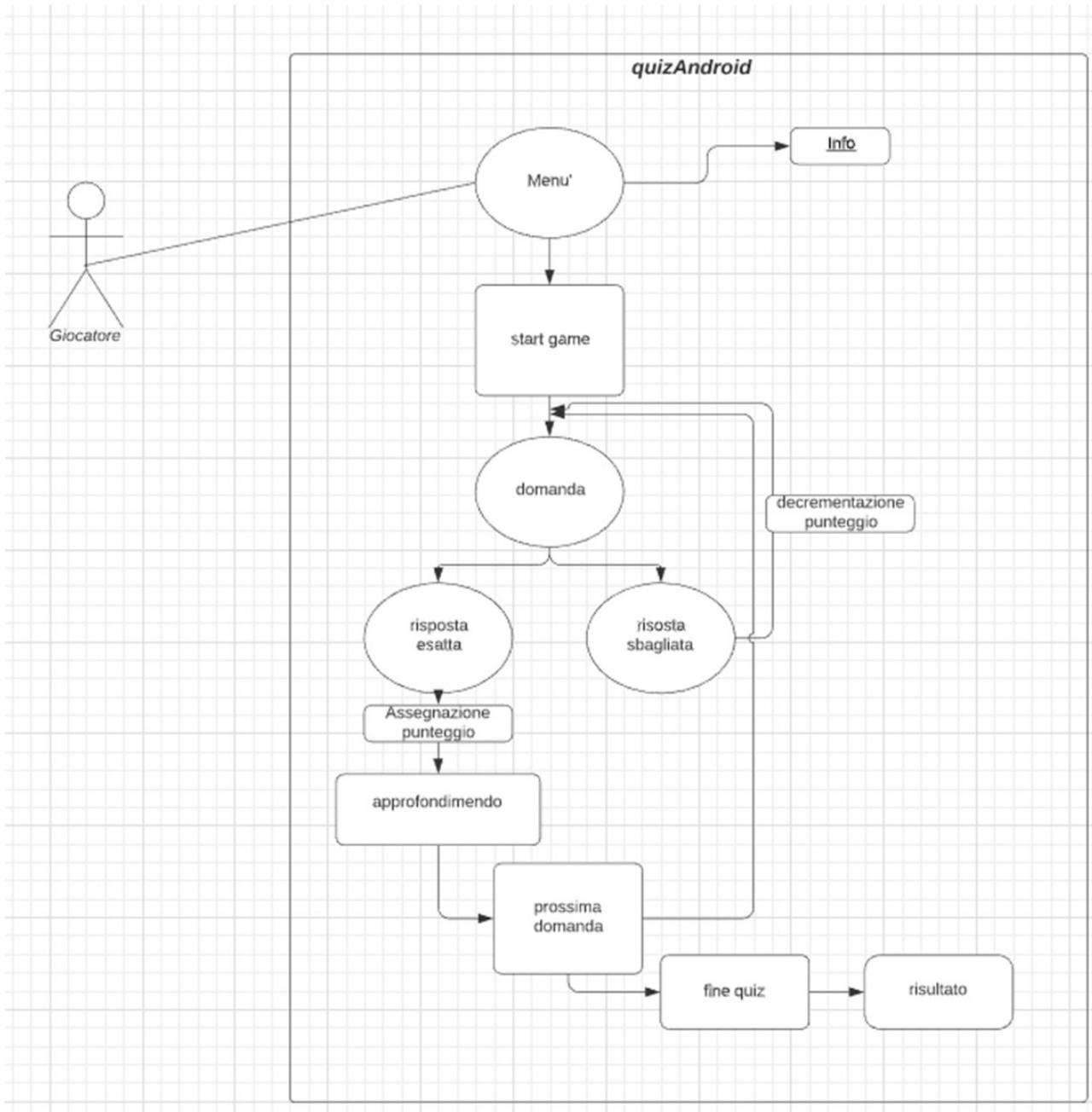
### Piano di progetto comprendente lo sviluppo temporale e le tappe principali di verifica

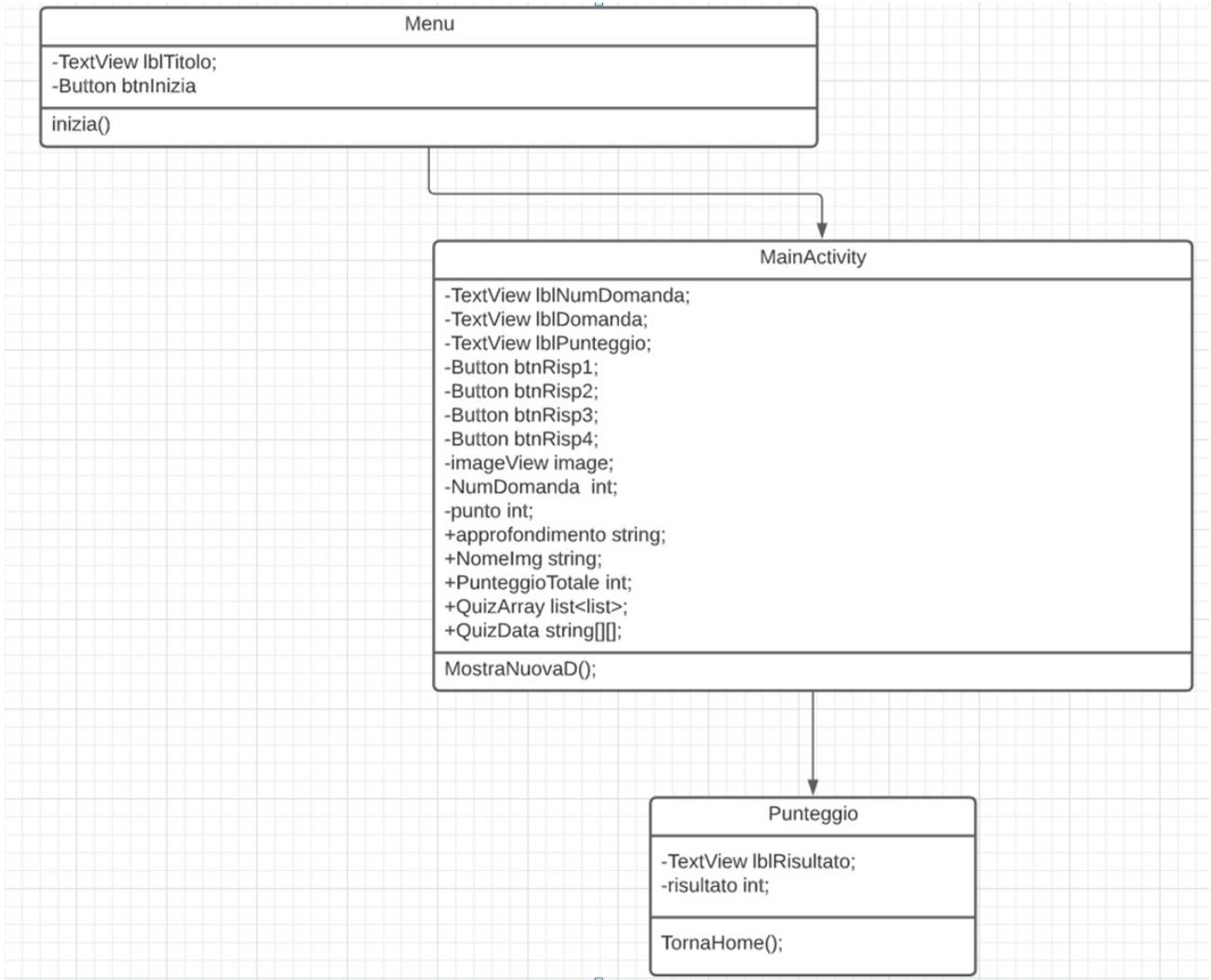
|    | Nome                     | Durata       | Previsto |      |
|----|--------------------------|--------------|----------|------|
| 1  | ☐ QUIZ ANDROID           | 10 giorni?   | ☑        | 02/1 |
| 2  | ☐ scelta linguaggio      | 0,3 giorni?  | ☑        | 02/1 |
| 3  | download android studio  | 0,1 giorni?  | ☑        | 02/1 |
| 4  | scelta emulatore         | 0,15 giorni? | ☑        | 02/1 |
| 5  | ☐ Interfaccia XML        | 1,5 giorni?  | ☑        | 03/1 |
| 6  | formattazione            | 0,25 giorni? | ☑        | 03/1 |
| 7  | codice                   | 0,25 giorni? | ☑        | 04/1 |
| 8  | ☐ Coding                 | 3,25 giorni? | ☑        | 05/1 |
| 9  | menu                     | 0,25 giorni? | ☑        | 05/1 |
| 10 | disposizioni domande     | 0,25 giorni? | ☑        | 06/1 |
| 11 | verifica risposte        | 0,25 giorni  | ☐        | 09/1 |
| 12 | punteggio + scheda final | 0,25 giorni? | ☑        | 10/1 |
| 13 | documentazione           | 3 giorni?    | ☑        | 11/1 |

The Gantt chart visualizes the project schedule from November 2nd to 9th, 2020. The tasks are represented by horizontal bars of varying lengths and colors (black, blue, red). The chart shows the duration of each task and its position relative to the project start and end dates. A red bar is visible at the end of the timeline, likely representing a final task or milestone.

# UML





## GIOCO A QUIZ PER L'APPRENDIMENTO

Scegliere "Attività vuota" android minimum 4.0

ProgettoAndroid: pixel 2 api 24 -

### Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical">

    <TextView
        android:id="@+id/Contlbl"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="D1"
        android:textSize="50sp"
        android:layout_marginTop="30dp" />

    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/image" />

    <TextView
        android:textAlignment="center"
        android:id="@+id/lblDomanda"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="domanda"
        android:textSize="30sp"
        android:layout_marginTop="20dp"/>

    <Button
        android:id="@+id/btnRisp1"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:text="risposta 1"
        android:layout_marginTop="10dp"
        android:onClick="VerificaRisposta" />

    <Button
        android:id="@+id/btnRisp2"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:text="risposta 2"
        android:layout_marginTop="10dp"
        android:onClick="VerificaRisposta" />

    <Button
        android:id="@+id/btnRisp3"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:text="risposta 3"
        android:layout_marginTop="10dp"
        android:onClick="VerificaRisposta"/>

    <Button
        android:id="@+id/btnRisp4"
        android:layout_width="200dp"
        android:layout_height="40dp"
        android:text="risposta 4"
        android:layout_marginTop="10dp"
        android:onClick="VerificaRisposta"/>
</LinearLayout>
```

## MainActivity.java

```
package com.e.progettoandroid3;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

public class MainActivity extends AppCompatActivity {

    private TextView lblNumDomanda; // numero domanda corrente
    private TextView lblDomanda; // domanda corrente
    private TextView lblpunteggio; // punteggio
    private Button btnRisp1; // bottone 1
    private Button btnRisp2; // bottone 2
    private Button btnRisp3; // bottone 3
    private Button btnRisp4; // bottone 4
    private ImageView Image; // immagine rispettiva alla domanda
    private String RispostaGiusta; // variabile dove viene salvata di volta in volta la
risposta giusta
    private int NumDomanda = 1;
    private int punto = 5;
    private int PunteggioTotale = 0;
    String Approfondimento; // variabile dove viene salvato l'approfondimento corrente.

    String QuizData[][] = { //matrice dove disporremo tutte le variabili necessarie
nel seguente modo
        //Domanda ; risposta giusta ; distrattore 1 ; distrattore 2 ;
distrattore 3; Nome imagine ;Approfondimento;
        {"domanda 1", "risposta giusta", "distrattore 1", "distrattore 2",
"distrattore 3", " Approfondimento"},
        {"domanda 2", "risposta giusta", "distrattore 1", "distrattore 2",
"distrattore 3", " Approfondimento"},
        {"domanda 3", "risposta giusta", "distrattore 1", "distrattore 2",
"distrattore 3", " Approfondimento"},
        {"domanda 4", "risposta giusta", "distrattore 1", "distrattore 2",
"distrattore 3", " Approfondimento"},
        {"domanda 5", "risposta giusta", "distrattore 1", "distrattore 2",
"distrattore 3", " Approfondimento"}
        //ATTENZIONE: nel caso in cui salvate le immagini in un'altra cartella che
non sia quella del
        //progetto originale dovete specificarne il percorso ad esempio la stringa
assumerà valore di "nomeCartella/Nome immagine"
    };
    ArrayList<ArrayList<String>> QuizArray = new ArrayList<>(); // creazione dell'array
di liste dove caricheremo la matrice QuizData per poter usufruire delle funzioni
dell'ArrayList

    public MainActivity() {
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //associamo gli oggetti java con quelli dell'xml
    }
}
```

```

    lblNumDomanda = (TextView) findViewById(R.id.lblNumDomanda);
    lblDomanda = (TextView) findViewById(R.id.lblDomanda);
    btnRisp1 = (Button) findViewById(R.id.btnRisp1);
    btnRisp2 = (Button) findViewById(R.id.btnRisp2);
    btnRisp3 = (Button) findViewById(R.id.btnRisp3);
    btnRisp4 = (Button) findViewById(R.id.btnRisp4);
    Image = (ImageView) findViewById(R.id.image);
    lblpunteggio = (TextView) findViewById(R.id.lblpunteggio);
    for (int i = 0; i < QuizData.length; i++) { //carichiamo nel'array
list QuizData
        ArrayList<String> tmpArray = new ArrayList<>();
        tmpArray.add(QuizData[i][0]); // domanda
        tmpArray.add(QuizData[i][1]); // risposta esatta
        tmpArray.add(QuizData[i][2]); // distrattore 1;
        tmpArray.add(QuizData[i][3]); // distrattore 2;
        tmpArray.add(QuizData[i][4]); // distrattore 3;
        tmpArray.add(QuizData[i][5]); // approfondimento

        QuizArray.add(tmpArray);
    }
    MostraNuovaD();
}

public void MostraNuovaD() { // funzione per mostrare una nuova domanda
    punto = 3;
    lblNumDomanda.setText("D" + NumDomanda); //imposta il numero della domanda
corrente
    lblpunteggio.setText(PunteggioTotale + "/15"); //punteggio corrente
    Random random = new Random(); //scelta di una domanda casuale
    int randomnum = random.nextInt(QuizArray.size());
    ArrayList<String> quiz = QuizArray.get(randomnum); //preleva tutta la lista di
posizione random
    lblDomanda.setText(quiz.get(0)); // imposta la domanda
    RispostaGiusta = quiz.get(1); //salva la risposta giusta
    Approfondimento = quiz.get(5);
    punto = 3;
    quiz.remove(0); // rimuove la domanda dalla lista
    quiz.remove(4); // rimuove l'approfondimento dalla lista
    Collections.shuffle(quiz);

    btnRisp1.setText(quiz.get(0));
    btnRisp2.setText(quiz.get(1));
    btnRisp3.setText(quiz.get(2));
    btnRisp4.setText(quiz.get(3));

    QuizArray.remove(randomnum); //rimuove l'intera lista di quella posizione
così la domanda nn verrà ripetuta
    //riabilita tutti i bottoni per essere cliccati
    btnRisp1.setEnabled(true);
    btnRisp2.setEnabled(true);
    btnRisp3.setEnabled(true);
    btnRisp4.setEnabled(true);
}

public void VerificaRisposta(View view) { // evento che verifica se la risposta è
corretta e che scatta nel momento in cui si clicca un bottone
    Button btnRisp = (Button) findViewById(view.getId());
    String btnText = btnRisp.getText().toString();

    String TitoloAvviso;
    lblpunteggio.setText(PunteggioTotale + "/15");

```

```

    if (btnText.equals(RispostaGiusta)) { //se il testo del bottone è uguale alla
risposta corretta allora mostra la scheda di approfondimento
        TitoloAvviso = "CORRETTO!";
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle(TitoloAvviso);
        builder.setMessage("Approfondimento: " + Approfondimento);
        builder.setPositiveButton("ok", new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                if (NumDomanda == 5) { // se il num delle domande è pari a 5 vuol
dire che sono state fatte tutte le domande
                    lblNumDomanda.setText("Finito");
                } else { //se non è l'ultima domanda allora incrementa il
punteggio e mostra la nuova domanda
                    PunteggioTotale = PunteggioTotale + punto;
                    NumDomanda++;
                    MostraNuovaD();
                }
            }
        });

        builder.setCancelable(false);
        builder.show();
    } else { // se si risponde in modo errato allora il punto viene decremtato e
il giocatore può riprovare e il bottone selezionato viene disabilitato
        TitoloAvviso = "errore";
        punto = punto - 1;
        btnRisp.setEnabled(false);
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle(TitoloAvviso);
        builder.setMessage("Hai sbagliato! dai su Riprova!");
        builder.setPositiveButton("riprova", new DialogInterface.OnClickListener()
{
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
        });
        builder.setCancelable(false);
        builder.show();
    }
}
}
}

```