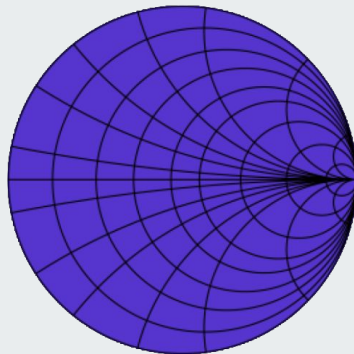




Line Calc

**Applicazione per il calcolo
delle Linee di Trasmissione**

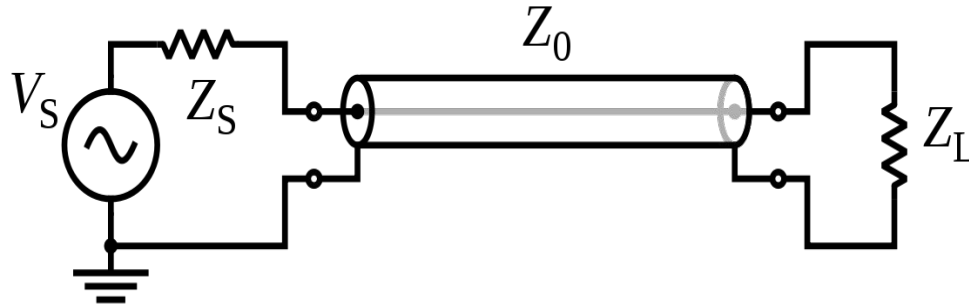
Andrea Argnani, Matteo Scardovi
19 Febbraio 2024



L'idea

L'idea per questa applicazione nasce dall'esame di Campi Elettromagnetici, in particolare dallo studio delle **linee di trasmissione**.

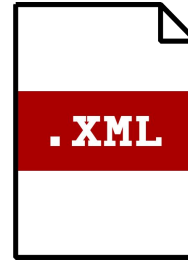
L'obiettivo è di fornire una serie di strumenti utili per calcolare parametri caratteristici e dimensionare adattatori di linea.



Tecnologie Utilizzate

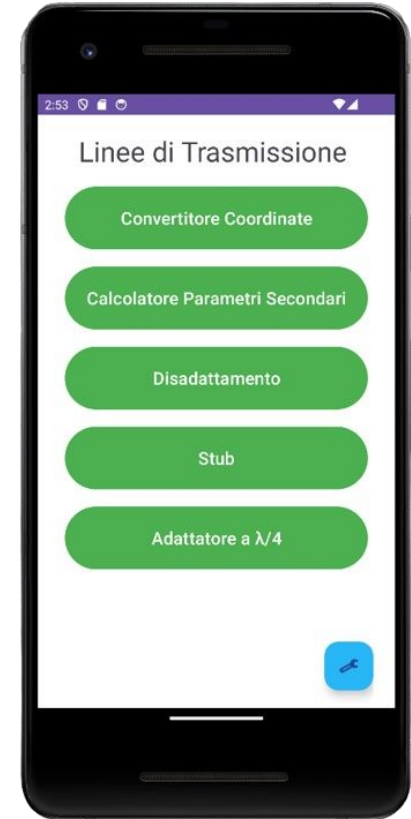
Per lo sviluppo dell'applicazione sono state utilizzate diverse tecnologie:

- Android Studio: IDE ufficiale di Android
- Java, per la parte di backend
- XML, per la parte grafica



Funzionalità

- Conversione di **numeri complessi** da forma lineare a polare e viceversa
- Calcolo dei **parametri secondari** delle linee di trasmissione
- Calcolo **disadattamento** del carico
- Dimensionamento **Adattatori a Stub**
- Dimensionamento **Adattatori a $\lambda/4$**



Codice

Le varie activity si basano su un backbone comune, modificato secondo necessità.

L'acquisizione dei valori avviene attraverso la funzione *getFromEditText*.

Per i calcoli che riguardano i numeri complessi viene impiegata la libreria Complex.

```
private double getFromEditText(EditText et, Spinner sp){  
    if(et.getText().toString().equals("")) {  
        et.setText("0");  
        return 0;  
    } else{  
        double val = Float.parseFloat(et.getText().toString());  
        val *= Math.pow(10, (sp.getSelectedItemPosition()-3)*3);  
        return val;  
    }  
}
```



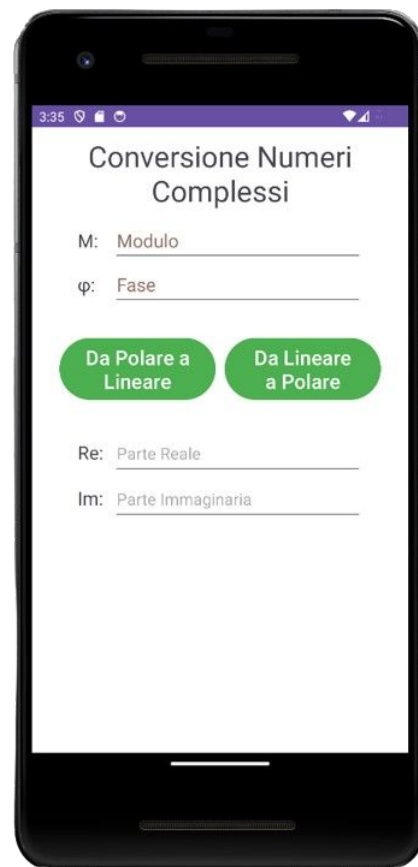
Activity

Numeri Complessi

Conversione di un numero complesso da forma algebrica a forma polare e viceversa.

```
double fRealValue = fAmplitude * Math.cos(fPhase);  
double fImaginalValue = fAmplitude * Math.sin(fPhase);
```

```
Complex c = new Complex(fRealValue, fImaginalValue);  
double fAmplitude = c.abs();  
double fPhase = c.arg();
```



Calcolatore Parametri Secondari

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

```
// Calcolo i parametri di trasmissione
Complex Zc, gamma;
Complex Z = new Complex(R, imaginary: omega * L);
Complex Y = new Complex(G, imaginary: omega * C);
Zc = Z.divide(Y);
Zc = Zc.sqrt();
```

The image shows a smartphone screen displaying a web application titled "Calcolatore Parametri Secondari". The app has a purple header bar with the time "2:25" and standard Android navigation icons. Below the title, there are five input fields, each with a label, a unit, and a multiplier: "R [Q/m]: Resistenza *10^0", "G [s/m]: Conduttanza *10^0", "C [C/m]: Capacità *10^0", "L [H/m]: Induttanza *10^0", and "ω [rad/s]: Pulsazione *10^0". Each field has a text input area and a dropdown arrow. Below these fields is a green button labeled "Calcola". At the bottom of the screen, there are labels for the calculated parameters: "Z_c", "Γ:", "α:", and "β".

Calcolatore Parametri Secondari

$$\alpha = \frac{1}{\sqrt{2}} \sqrt{\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)} - (\omega^2 LC - RG)}$$

$$\beta = \frac{1}{\sqrt{2}} \sqrt{\sqrt{(R^2 + \omega^2 L^2)(G^2 + \omega^2 C^2)} + (\omega^2 LC - RG)}$$

```
double alfa = (R * R + omega * omega * L * L) * (G * G + omega * omega * C * C);  
alfa = Math.sqrt(alfa);  
double beta = alfa;  
double ab = omega * omega * L * C - R * G;  
alfa = alfa - ab;  
beta = beta + ab;  
alfa = Math.sqrt(alfa / 2);  
beta = Math.sqrt(beta / 2);
```

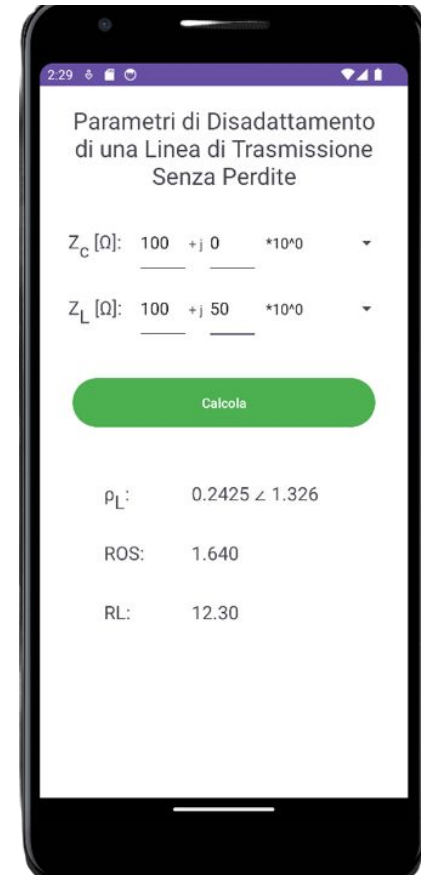
The image shows a smartphone screen displaying a web application titled "Calcolatore Parametri Secondari". The interface includes five input fields for parameters: R [Q/m] (Resistenza), G [s/m] (Conduttanza), C [C/m] (Capacità), L [H/m] (Induttanza), and ω [rad/s] (Pulsazione). Each field has a unit label, a description, a value input area, and a multiplier dropdown menu currently set to *10^0. A green "Calcola" button is positioned below the inputs. The results section at the bottom lists Z_c, Γ, α, and β.

Parametri di Disadattamento

$$\rho_L = \frac{Z_L - Z_C}{Z_L + Z_C}$$

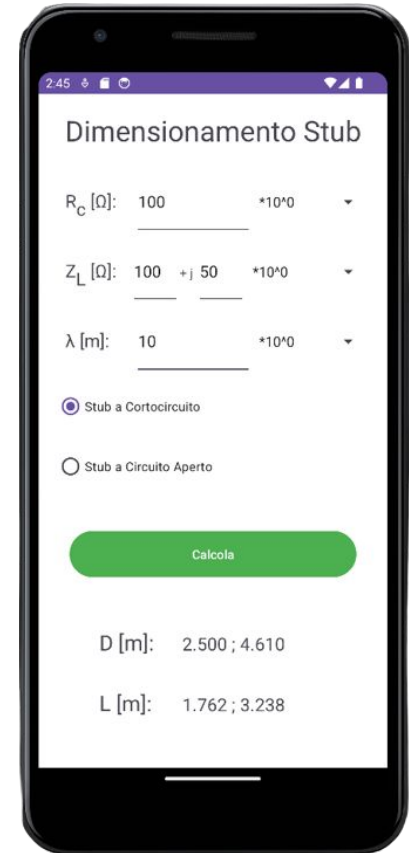
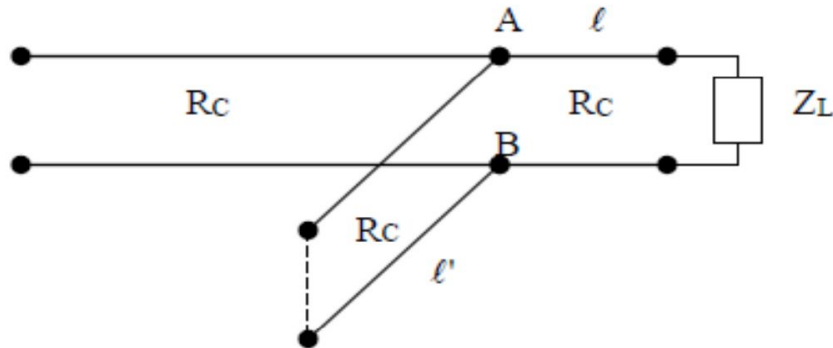
$$ROS = \frac{1 + |\rho_L|}{1 - |\rho_L|}$$

$$RL = -20 \log_{10} |\rho_L| = 20 \log_{10} \left(\frac{S + 1}{S - 1} \right)$$

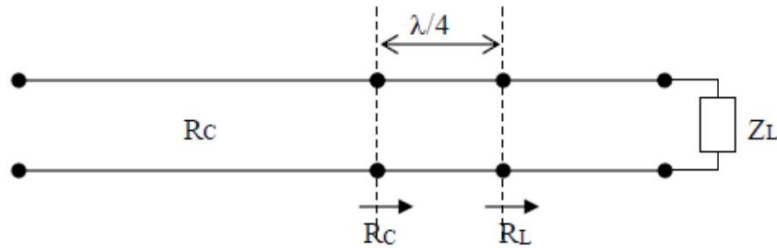


Stub

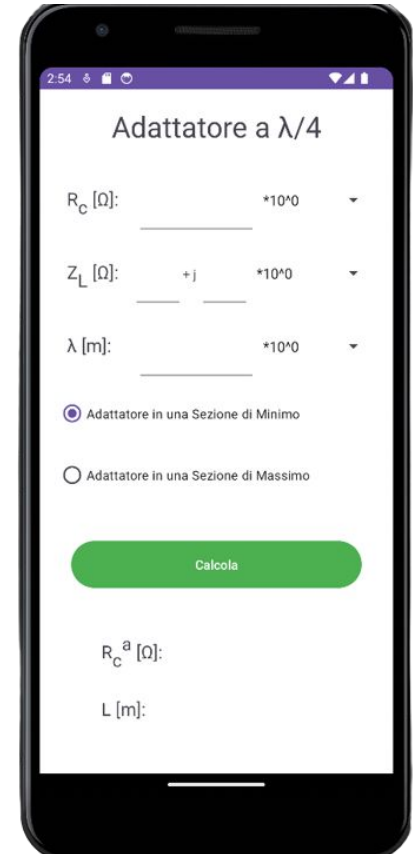
Activity che serve per dimensionare gli Stub, componenti utili per migliorare il trasferimento di potenza portando il valore del ROS il più possibile vicino a 0.



Adattatori a $\lambda/4$



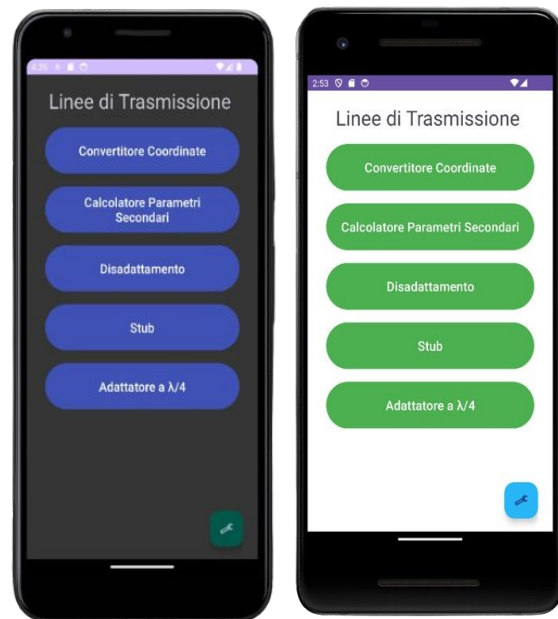
```
rho_L = ZL.subtract(Zc);  
rho_L = rho_L.divide(Zc.add(ZL));  
ROS = (1 + rho_L.abs()/(1 - rho_L.abs()));  
RadioButton r_min = findViewById(R.id.radioMin);  
if(r_min.isChecked()) {  
    Ra = Rc / Math.sqrt(ROS);  
    L = (lambda/4)*(1 - rho_L.arg()/Math.PI);  
}else{  
    Ra = Rc * Math.sqrt(ROS);  
    L = (-rho_L.arg()/(4*Math.PI))*lambda;  
}
```



Impostazioni

Permettono il cambio di lingua, del tema e del numero di cifre significative

```
public void updatePreferenceTheme(SharedPreferences prefs) {  
    String theme = prefs.getString( key: "theme", defValue: "light");  
    if (theme.equals("light"))  
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);  
    else if(theme.equals("dark"))  
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);  
    else  
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_FOLLOW_SYSTEM);  
    recreate();  
}
```



Tema Scuro vs Tema Chiaro



Grazie per l'attenzione

Codice Sorgente:

<https://github.com/argnaan/LineCalc>

APK:

https://github.com/argnaan/LineCalc/releases/download/tag/LineCalc0_2.apk