

Politecnico di Torino

Dipartimento di ingegneria gestionale e della produzione



Corso di Laurea Triennale in Ingegneria Gestionale
Classe L8 – Ingegneria dell'informazione A.A. 2022/2023

Applicazione per la valutazione della risorsa energetica idonea alla realizzazione di CER

Relatore

Prof. Fulvio Corno

Candidato

Roberta De Seriis 261469

Dicembre 2022

Indice

Capitolo 1: Proposta di progetto.....	1
1.1 Studente proponente	1
1.2 Titolo della proposta.....	1
1.3 Descrizione del problema proposto	1
1.4 Descrizione della rilevanza gestionale del problema	1
1.5 Descrizione dei data-set per la valutazione	1
1.6 Descrizione preliminare degli algoritmi coinvolti.....	2
1.7 Descrizione preliminare delle funzionalità previste per l'applicazione software	2
Capitolo 2: Descrizione dettagliata del problema affrontato.....	3
2.1 Introduzione.....	3
2.2 Contesto operativo/statale	4
Capitolo 3: Descrizione del data-set	5
Capitolo 4: Descrizione strutture dati e algoritmi coinvolti.....	7
4.1 Strutture dati	7
4.2 Algoritmi principali.....	8
Capitolo 5: Diagramma delle classi delle parti principali	16
Capitolo 6: Esempi di utilizzo dell'applicazione	17
6.1 Videata dell'applicazione	17
6.2 Video dimostrativo dell'applicazione software.....	18
Capitolo 7: Risultati sperimentali ottenuti	19
Capitolo 8: Considerazioni finali.....	20
8.1 Valutazione dei risultati ottenuti	20
8.2 Conclusioni.....	20
Riferimenti.....	22
Indice delle figure.....	23

Capitolo 1: Proposta di progetto

1.1 Studente proponente

S261469 Roberta De Seriis

1.2 Titolo della proposta

Applicazione per la valutazione della risorsa energetica idonea alla realizzazione di CER

1.3 Descrizione del problema proposto

L'idea progettuale consiste nella realizzazione di un'applicazione JavaFX che analizza dal punto di vista energetico il territorio italiano (con risoluzione regionale), al fine di redistribuire i finanziamenti Statali nell'ambito delle energie rinnovabili per sostenere lo sviluppo delle Comunità Energetiche Rinnovabili (CER).

Il programma riceve in input determinati parametri di analisi per il calcolo della produzione energetica comunale prima e della ricorsione poi. Utilizzando i dati inseriti in input e le informazioni presenti nel data-set, restituisce la combinazione (regioni + risorsa energetica + budget assegnato) che massimizza il numero di comuni ai quali destinare gli investimenti per l'incremento di nuova capacità di energia elettrica, seguendo i criteri riportati nel PNRR, Piano Nazionale di Ripresa e Resilienza (www.mise.gov.it).

L'output viene generato valutando determinati criteri inerenti agli attuali livelli di produzione di energia da fonti rinnovabili, in particolare energia prodotta da impianti eolici e solari fotovoltaici. Dati ripresi dal sito dell'ISTAT (www.istat.it).

1.4 Descrizione della rilevanza gestionale del problema

La nascita di nuove CER è un problema di rilevanza gestionale, affrontato con frequenza negli ultimi anni per favorire e accelerare il processo di transizione energetica. La CER si delinea come un insieme di persone fisiche, giuridiche ed enti (privati, PMI, PA ecc.) il cui scopo è produrre e condividere energia mediante impianti alimentati da fonti rinnovabili. Un'errata analisi economico-ambientale porta ad investimenti in risorse inadatte al territorio di riferimento. Grazie all'analisi di fattori energetici (es. produzione totale netta di energia elettrica delle fonti rinnovabili) ed economici (fondi di finanziamento stanziati dallo Stato italiano) tale applicativo, analizzando le risorse disponibili sul territorio, permette di individuare la fonte di energia elettrica (eolica e/o solare fotovoltaica) da incrementare e conseguentemente la tipologia di impianto da installare.

1.5 Descrizione dei data-set per la valutazione

La risoluzione di tale problema avviene attraverso l'utilizzo di data-set pubblicati sul sito Istat, in particolar modo quello relativo al tema "Energia" con risoluzione territoriale regionale.

Il data-set include i dati necessari allo scopo dell'applicazione. Essendo da esso esclusi quelli relativi al numero di comuni presenti in Italia è possibile per questi ultimi far riferimento al sito (www.tuttitalia.it, 2022) che riprende i dati aggiornati all' 01/01/2022 (Istat).

Tra le informazioni riportate, emergono:

- ♦ produzione lorda degli impianti da fonti rinnovabili;
- ♦ potenza efficiente lorda delle fonti rinnovabili;
- ♦ numero di comuni e numero di comuni con meno di cinquemila abitanti.

1.6 Descrizione preliminare degli algoritmi coinvolti

Nell'applicazione sono presenti algoritmi ricorsivi che si occupano di ricercare la soluzione più adatta, considerando stima per comune di produzione netta di energia elettrica generata da Solare Fotovoltaico ed Eolico, numero di comuni con meno di cinquemila abitanti a cui affidare il budget di finanziamento e tipologia di produzione da incrementare (eolico e/o solare fotovoltaico).

Mediante la ricorsione, sfruttando i parametri presenti nel data-set, viene individuata la combinazione (regioni + risorsa energetica + budget assegnato) che massimizza il numero di comuni con meno di cinquemila abitanti ai quali destinare gli investimenti per installare nuova capacità di energia elettrica, seguendo i parametri riportati nel PNRR (Piano Nazionale di Ripresa e Resilienza).

Successivamente il programma effettua le stampe nell'apposita sezione "Output".

1.7 Descrizione preliminare delle funzionalità previste per l'applicazione software

L'applicazione si suddivide in tre sezioni:

- ♦ **produzione per comune:** calcola la stima di produzione netta di energia elettrica per comune;
- ♦ **parametri ricorsione:** avvia le tre tipologie di ricorsione, la prima utilizza i parametri relativi alla produzione di energia elettrica da impianti eolici, la seconda i dati inerenti alla produzione di energia elettrica da impianti solari fotovoltaici e l'ultima, riguarda l'incremento di produzione di energia elettrica generata da entrambe le FER (Fonti Energetiche Rinnovabili). Le prime 2 ricorsioni sono pressoché simili, sebbene differiscano per i data-set a cui fanno riferimento;
- ♦ **output:** restituisce gli output delle precedenti sezioni. In particolare, nella textArea vengono riportati gli output delle ricorsioni o eventuali errori per il mancato inserimento dei parametri in input. Nella TableView vengono riportati i risultati attinenti alla prima sezione e specificatamente le produzioni nette per comune di energia elettrica generata dagli impianti solari fotovoltaici descritti nella prima colonna e quelli eolici descritti nella seconda.

I dati inseriti dall'utente (input) necessari per la produzione dei risultati (output) sono:

- ♦ area di riferimento: Italia, Italia PS (regioni più sviluppate), Italia IT (regioni in transizione), Italia MS (regioni meno sviluppate);
- ♦ numero di comuni (con meno di cinquemila abitanti);
- ♦ produzione di: eolico e/o solare fotovoltaico;
- ♦ budget di finanziamento (in euro);

Capitolo 2: Descrizione dettagliata del problema affrontato

2.1 Introduzione

La produzione di energia elettrica sta diventando sempre più green e sostenibile, grazie all'aumento delle FER. L'Italia è all'avanguardia in Europa e nel Mondo sul fronte delle rinnovabili, con una quota percentuale in costante crescita, incentivata dai molteplici finanziamenti messi a disposizione dallo Stato e dall'UE (Unione europea).

La generazione di energia elettrica da FER, presenta rilevanti differenze, dovute soprattutto alle caratteristiche del territorio e alla distribuzione delle risorse rinnovabili. In Italia, il fotovoltaico è più presente al Sud, grazie alla minore latitudine e al maggior irraggiamento solare. Accade diversamente per l'energia eolica che viene raccolta per lo più nelle grandi isole, Sicilia e Sardegna e in alcune delle regioni dell'Italia meridionale, tra le quali Puglia, Campania e Basilicata.

Secondo quanto riportato dal Rapporto Statistico 2020 del GSE (www.gse.it, s.d.) : *“la Puglia si conferma la regione italiana caratterizzata dalla maggiore produzione di energia elettrica da impianti fotovoltaici (3.839 GWh, pari al 15,4% del totale nazionale); seguono la Lombardia con il 9,8%, l'Emilia Romagna con il 9,6% e il Veneto con l'8,7%. Valle d'Aosta e Liguria sono invece le regioni con produzioni più modeste (rispettivamente 0,1% e 0,5% del totale nazionale). Anche per quanto riguarda la produzione da impianti eolici la Puglia detiene il primato, con 4.802 GWh di energia elettrica prodotta, seguita da Sicilia (2.765 GWh) e Campania (3.209 GWh); considerate insieme, queste tre regioni coprono il 57,4% del dato complessivo nazionale”.*

In seguito vengono riportate due mappe esemplificative pubblicate nel Rapporto Statistico 2020 del GSE, inerenti alla distribuzione regionale della produzione di energia elettrica da impianti solari fotovoltaici (Figura 1) e da impianti eolici (Figura 2) nel 2020.

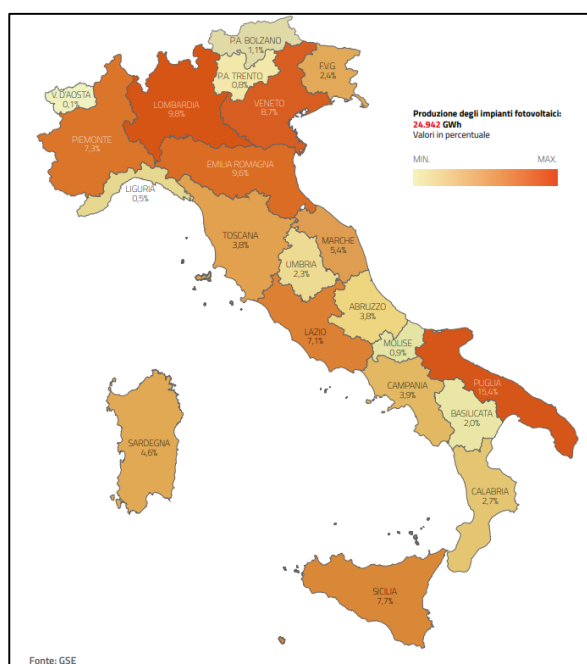


Figura 1: Produzione solare fotovoltaico

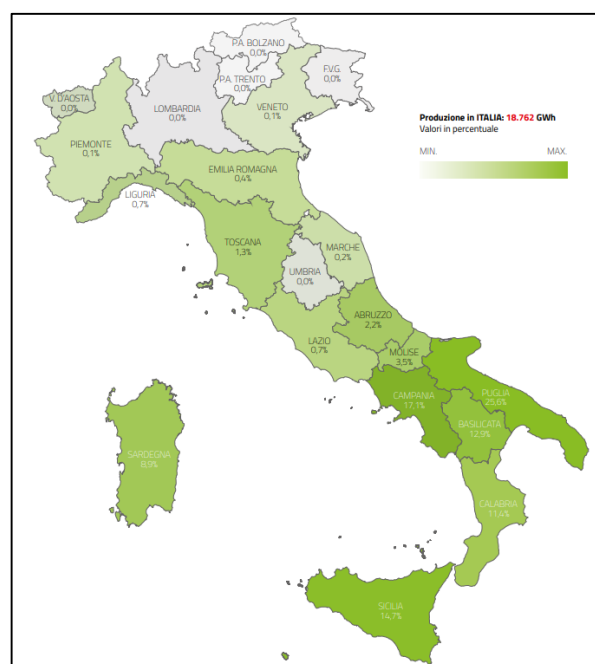


Figura 2: Produzione eolica

2.2 Contesto operativo/statale

Negli ultimi anni, il mercato dei “prosumer” (consumatori e produttori allo stesso tempo) in campo energetico è in continua crescita anche grazie agli incentivi statali previsti per i progetti che riguardano l'autoproduzione di energia. A seguito della pandemia mondiale di Covid-19, allineandosi al piano di ripresa europeo NGEU (Next Generation EU, programma che prevede investimenti e riforme per accelerare la transizione ecologica e digitale), lo Stato italiano ha presentato il PNRR (Piano Nazionale di Ripresa e Resilienza) che si articola in sei Missioni e sedici Componenti. Tra le sei missioni rientra la “rivoluzione verde e transizione ecologica”, alla quale vengono assegnati 59,46 miliardi di euro, dei quali 23,78 miliardi di euro a “M2C2- Energia rinnovabile, idrogeno, rete e mobilità sostenibile”. Di quest'ultimo ammontare 2,2 miliardi di euro riguardano investimenti in: “Promozione rinnovabili per le comunità energetiche e l'autoconsumo”.

Tra i compiti dello Stato italiano rientra quello di redistribuire i finanziamenti. Per tale scopo è prevista una procedura piuttosto lunga sia per ragioni burocratiche che di tempistica relativa all'effettiva valutazione dell'adequatezza e dell'efficacia delle proposte presentate. Nell'ambito delle risorse rinnovabili e seguendo le linee guida per gli investimenti in “Promozione rinnovabili per le comunità energetiche e l'autoconsumo”, l'applicazione software sviluppata si occuperà di distribuire in modo ottimale il budget di finanziamento alle diverse regioni italiane.

Le specifiche per gli investimenti, riportate nel PNRR, riguardano espressamente le Comunità Energetiche Rinnovabili (CER) e le strutture collettive di autoproduzione e prendono in considerazione le aree in cui si prevede il maggior impatto socio-territoriale. Per gli investimenti, si procede all'individuazione di PA (Pubblica Amministrazione), famiglie e microimprese dei comuni con meno di cinquemila abitanti, per sostenere l'economia dei piccoli Comuni, spesso a rischio di spopolamento, nonché di rafforzare la coesione sociale. L'obiettivo è garantire le risorse necessarie per installare 2.000 MW di nuova capacità di generazione elettrica, distribuendola tra CER e auto-consumatori di energie rinnovabili.

Come riportato dal PNRR *“La realizzazione di questi investimenti contribuirà a una riduzione delle emissioni di gas serra stimata in circa 1,5 milioni di tonnellate di CO2 l'anno”*.

Seguendo il criterio di coesione sociale, le regioni italiane sono state suddivise in tre classi:

- ♦ PS - più sviluppate
- ♦ MS - meno sviluppate
- ♦ IT - in transizione

Le regioni appartenenti a tali aree sono definite dal Ministero per il Sud e la coesione territoriale (www.ministroperilsud.gov.it, 2021-2027) come segue:

- ♦ regioni più sviluppate, con un PIL pro capite superiore al 90 % della media dell'UE;
- ♦ regioni meno sviluppate, con un PIL pro capite inferiore al 75 % della media dell'UE;
- ♦ regioni in transizione, con un PIL pro capite compreso tra il 75 % e il 90 % della media dell'UE.

Capitolo 3: Descrizione del data-set

Il data-set utilizzato per l'analisi è composto da una tabella "Regione", che riporta i dati energetici, quelli relativi al numero totale di comuni e al numero di comuni con meno di 5.000 abitanti raggruppandoli per regione. Il data-set è stato ottenuto dall'estrapolazione e dall'elaborazione dei dati dal database dell'Istat "Energia" e dai dati ripresi dal Rapporto Statistico del GSE sulle FER mentre la parte inerente al numero di comuni, è stata ricavata dal sito www.tuttitalia.it che contiene i dati aggiornati all' 01/01/2022 (Istat).

Regione
<pre> + nome :String + produzioneTotNEElettrica :double + produzioneTotNEElettrica_FOTOVOLTAICO :double + prodTotNEElettrica_EOLICO :double + potenzaEfficienteLorda%FER :double + potenzaEfficienteLordaEOLICO :double + potenzaEfficienteLorda_FOTOVOLTAICO :double + potenzaEfficienteLordaTotale :double + famiglieAllacciateAllaRete :int + superficieCentriAbitati :int + popolazioneResidenteMedia :int + consumiFinaliLordiEnergia :int + consumiCopertiDaiImpiantiEol :int + consumiCopertiDaiImpiantiFot :int + statoRegione :String + numeroComuniConMenodi5000Abitanti :int + numeroComuniTot :int </pre>

Figura 3: classe Regione, realizzata con Signavio Academic (www.signavio.com, s.d.)

La tabella Regione contiene le seguenti informazioni:

- nome della regione italiana;
- prodTotNEElettrica: produzione totale netta di Energia Elettrica, espressa in GWh;
- prodTotNEElettrica_FOTOVOLTAICO: produzione netta di energia elettrica da impianti solari fotovoltaici, espressa in GWh;
- prodTotNEElettrica_EOLICO: produzione netta di energia elettrica da impianti eolici, espressa in GWh. La Direttiva 2009/28/CE prevede che l'energia prodotta da fonte eolica venga contabilizzata applicando una procedura di normalizzazione, al fine di attenuare gli effetti delle variazioni climatiche tra i diversi anni. La produzione normalizzata è funzione della produzione osservata e della potenza installata negli ultimi 5 anni, secondo la seguente formula, riportata di seguito e presa dal Rapporto Statistico del GSE sulle FER 2020;

$$Q_{N(norm)} = \frac{C_N + C_{N-1}}{2} \cdot \frac{\sum_{i=N-n}^N Q_i}{\sum_{j=N-n}^N \left(\frac{C_j + C_{j-1}}{2} \right)}$$

N = anno di riferimento
Q_N (norm) = produzione normalizzata
Q_i = produzione reale anno i
C_j = potenza totale installata anno j
n = min (4; numero di anni precedenti l'anno N per i quali sono disponibili i dati su potenza e produzione).

Figura 4: Produzione normalizzata - Rapporto Statistico del GSE sulle FER 2020

- ♦ **potenzaEfficienteLorda%FER**: percentuale di potenza efficiente lorda delle fonti rinnovabili, sul totale della potenza efficiente lorda, espressa in MW;
- ♦ **potenzaEfficienteLorda_EOLICO**: potenza efficiente lorda degli impianti eolici, espressa in MW;
- ♦ **potenzaEfficienteLorda_FOTOVOLTAICO**: potenza efficiente lorda degli impianti solari fotovoltaici, espressa in MW;
- ♦ **potenzaEfficienteLordaTotale**: potenza efficiente lorda, relativa a fonti rinnovabili e non, espressa in MW;
- ♦ **famiglieAllacciate**: totale delle famiglie allacciate alla rete elettrica nazionale;
- ♦ **superficieCentriAbitati**: superficie dei centri abitati, espressa in Km²;
- ♦ **popolazioneResidenteMedia**;
- ♦ **consumiFinaliLordiEnergia**: consumi finali lordi di energia, delle famiglie allacciate alla rete elettrica nazionale, espressi in GWh;
- ♦ **consumiCopertiDaImpiantiEol**: consumi coperti con la produzione di energia elettrica da impianti eolici, espressi in GWh;
- ♦ **consumiCopertiDaImpiantiFot**: consumi coperti con la produzione di energia elettrica da impianti solari fotovoltaici, espressi in GWh;
- ♦ **statoRegione**: stato delle ragioni dovuto ad una classificazione basata sul PIL pro capite (quantità di prodotto interno lordo ipoteticamente prodotta, in un certo periodo di tempo, da una persona (www.wikipedia.org, s.d.)) delle regioni italiane, in relazione alla media europea. Le regioni italiane sono così suddivise (riprese dal ministero per il Sud e la coesione territoriale, e riportate in seguito in ordine decrescente di PIL pro capite):
 - i. più sviluppate (PS): Trentino-Alto Adige, Lombardia, Emilia-Romagna, Valle d'Aosta, Veneto, Lazio, Friuli-Venezia Giulia, Toscana, Piemonte, Liguria;
 - ii. in transizione (IT): Marche, Abruzzo, Umbria;
 - iii. meno sviluppate (MS): Sardegna, Basilicata, Molise, Puglia, Campania, Sicilia, Calabria;
- ♦ **numeroComuniConMenoDicinquemila**: numero di comuni italiani con meno di cinquemila Abitanti;
- ♦ **numeroComuniTot**: numero di comuni italiani per ogni regione.

Capitolo 4: Descrizione strutture dati e algoritmi coinvolti

4.1 Strutture dati

Il software è stato programmato in Java, usando il pattern **MVC** (Model View Controller) e il pattern **DAO** (Data Access Object). Mediante il software SceneBuilder è stato possibile realizzare l'interfaccia grafica in JavaFX. Come consigliato dai pattern MVC e DAO, il progetto è costituito da tre package per separare il modello, l'accesso al database e l'interfaccia utente.

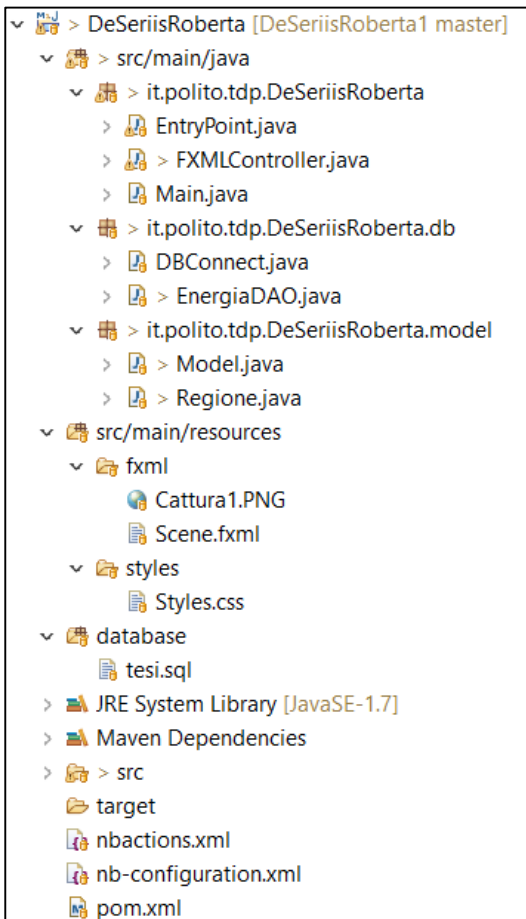


Figura 5: Struttura dati

Il package *it.polito.tdp.DeSeriisRoberta* contiene la classe principale del progetto, Main, la classe EntryPoint, utilizzata per l'avvio del programma e la classe FXMLController, che collega il Model all'interfaccia grafica.

Il package *it.polito.tdp.DeSeriisRoberta.db* contiene le classi necessarie per l'accesso ai dati presenti nel database.

Il package *it.polito.tdp.DeSeriisRoberta.model* contiene la classe logica principale dell'applicazione (Model) e la classe di oggetti (Regione) creata per il funzionamento dell'applicazione.

4.2 Algoritmi principali

Il database presenta informazioni relative alla produzione netta di energia elettrica sviluppata da impianti eolici e solari fotovoltaici ma tali dati sono a risoluzione regionale.

Per convertirli da regionali a comunali, nella classe Model, sono stati creati due metodi: *calcolaProduzioneFotovoltaicoComune()*, *calcolaProduzioneEolicoComune()*.

```
public List<Regione> calcolaProduzioneFotovoltaicoComune(List<Regione> regioni) {
    double produzioneFotComunale;
    for (Regione r: regioni) {
        if(r.getProdNEElett_Fot_GWh(>0) {
            produzioneFotComunale= r.getProdNEElett_Fot_GWh()/r.getComuniTot();
            r.setProdFotovoltaicoPerComune(produzioneFotComunale);
        }
        else
            produzioneFotComunale=0;
    }
    reg = new LinkedList<Regione>(regioni);
    return regioni;
}

public List<Regione> calcolaProduzioneEolicoComune(List<Regione> regioni) {
    double produzioneEolComunale;
    for (Regione r: regioni) {
        if(r.getProdNEElett_Eol_GWh(>0) {
            produzioneEolComunale= r.getProdNEElett_Eol_GWh()/r.getComuniTot();
            r.setProdEolicoPerComune(produzioneEolComunale);
        }
        else
            produzioneEolComunale=0;
    }
    reg = new LinkedList<Regione>(regioni);
    return regioni;
}
```

Figura 6: Metodi *calcolaProduzioneComune()* (eolico e solare fotovoltaico) - Model

Essi vengono richiamati nel metodo *statoRegioni(String tipo)* che calcola la produzione comunale del set di regioni selezionate dall'utente. L'utente, infatti, può selezionare: Italia (tutte le regioni italiane), PS (regioni più sviluppate), IT (regioni in transizione) e MS (regioni meno sviluppate). I valori così calcolati vengono settati negli attributi **ProdSolarePerComune** e **ProdEolicoPerComune** della classe Regione.

```
public List<Regione> statoRegioni(String tipo) {
    regioniItalia = dao.getAllRegioni();
    regioniSelezionate = new ArrayList<>();

    for (Regione r: regioniItalia) {
        if(r.getStatoRegione().compareTo(tipo)==0) {
            regioniSelezionate.add(r);
        }
    }
    this.calcolaProduzioneFotovoltaicoComune(regioniSelezionate);
    this.calcolaProduzioneEolicoComune(regioniSelezionate);

    return regioniSelezionate;
}
```

Figura 7: Metodo *statoRegioni()* – Model

La selezione dell'insieme di regioni da parte dell'utente è possibile grazie al metodo *doProduzioneTot()* dell'*FXMLController*.

```
@FXML
void doProduzioneTot(ActionEvent event) {
    txtResult.clear();

    produzione=true;
    areaScelta = this.cmbAreaDiRiferimento.getValue();

    this.btnProduzioneTot.setDisable(false);
    this.btnAvvia.setDisable(false);
    this.btnReset.setDisable(false);

    if(areaScelta==null) {
        txtResult.appendText("Seleziona un'area di riferimento");
        return;
    }

    if(areaScelta.compareTo("Italia")==0) {
        List<Regione> produzioneRegioni = model.getRegioni();
        model.calcolaProduzioneEolicoComune(produzioneRegioni);
        model.calcolaProduzioneFotovoltaicoComune(produzioneRegioni);
        tblOutput.setItems(FXCollections.observableArrayList(produzioneRegioni));
    }

    if(areaScelta.compareTo("Italia: PS")==0) {
        List<Regione> produzioneC = model.statoRegioni("PS");
        tblOutput.setItems(FXCollections.observableArrayList(produzioneC));
    }

    if(areaScelta.compareTo("Italia: MS")==0) {
        List<Regione> produzioneC = model.statoRegioni("MS");
        tblOutput.setItems(FXCollections.observableArrayList(produzioneC));
    }

    if(areaScelta.compareTo("Italia: IT")==0) {
        List<Regione> produzioneC = model.statoRegioni("IT");
        tblOutput.setItems(FXCollections.observableArrayList(produzioneC));
    }
}
```

Figura 8: Metodo *doProduzione()* – *FXMLController*

La stampa dei risultati riguardanti la produzione comunale eolica e fotovoltaica avviene nella tabella *tblOutput*, attraverso i *setCellValueFactory()*, riportati nell'*initialize()* dell'*FXMLController*.

Nel Model, i dati comunali di produzione di energia elettrica ottenuti vengono utilizzati per i metodi ricorsivi. Le ricorsioni sono tre: una relativa alla produzione eolica, un'altra relativa alla produzione solare fotovoltaica e l'ultima relativa alla produzione sia eolica che solare fotovoltaica.

I metodi ricorsivi riportati si dividono in due principali *calcolaMigliore()*, *cercaMigliore()* e due complementari *sommaComuni()* (comune a tutte le ricorsioni) e *calcolaParam()*.

Si possono analizzare i metodi ricorsivi uno per volta:

- ♦ il metodo *cercaMiglioreEol(List<Regione> parziale, int L, int n)* che, passando una lista di regioni (selezionate dall'utente), un livello di riferimento (L) e un intero (n), il metodo restituisce il massimo numero di comuni (n) con meno di cinquemila abitanti a cui assegnare il finanziamento per la realizzazione di nuovi impianti eolici, la cui produzione eolica comunale (in percentuale) è minore rispetto a quella degli altri comuni italiani;
- ♦ il metodo *calcolaMiglioreEol(int n)* che avvia la ricorsione e, richiamando il metodo precedente, modifica di volta in volta la lista di regioni (parziale) e il numero di comuni (n) a cui destinare il finanziamento restituendo così la soluzione migliore.

```
public List<Regione> calcolaMiglioreEol (int n) {
    miglioreE = new LinkedList<Regione>();
    List<Regione> parziale = new LinkedList<Regione>();
    aM = Double.MAX_VALUE;

    cercaMiglioreEol(parziale,0,n);
    return miglioreE;
}

private void cercaMiglioreEol(List<Regione> parziale, int L, int n) {
    int sommaComuni = sommaComuni(parziale);

    if(sommaComuni>n)
        return;

    if(sommaComuni != 0 && sommaComuni <= n) {
        double param = calcolaParametroEol(parziale);

        if(param < aM && sommaComuni > sommaComuni(miglioreE)) {
            miglioreE = new LinkedList<Regione>(parziale);
            aM = param;
        }
    }

    if(L == reg.size()) {
        return;
    }

    if(reg.get(L).getProdEolicoPerComune()>0) {
        parziale.add(reg.get(L));
        this.cercaMiglioreEol(parziale, L+1, n);

        parziale.remove(reg.get(L));
        this.cercaMiglioreEol(parziale, L+1, n);
    }
}
```

Figura 9: Metodi ricorsivi: *calcolaMiglioreEol()* e *CercaMiglioreEol()* – Model

I metodi supplementari richiamati nei due precedenti sono:

- ♦ Il metodo *calcolaParametroEol(List<Regione> parziale)* che calcola la produzione eolica per comune in percentuale (rispetto al numero di comuni italiani).

```

private double calcolaParametroEol(List<Regione> parziale) {
    double s = 0;
    double nC = 0;

    for (Regione r:parziale) {
        s += r.getProdEolicoPerComune();
        nC += r.getComuniTot();
    }
    return (s/nC);
}

```

Figura 10: Metodo calcolaParametroEol() – Model

- ♦ Il metodo *sommaComuni(List<Regione> parziale)* che calcola il numero di comuni italiani con meno di cinquemila abitanti.

```

private int sommaComuni(List<Regione> parziale) {
    int nCom = 0;

    for(Regione r:parziale) {
        nCom += r.getComuniMeno5000();
    }
    return nCom;
}

```

Figura 11: Metodo sommaComuni() –Model

Il metodo *sommaComuni*, essendo comune a tutte le ricorsioni, viene riportato una sola volta.

I metodi ricorsivi relativi alla produzione di energia elettrica da impianti solari fotovoltaici sono:

- ♦ il metodo *calcolaMiglioreFot (int n)* che avvia la ricorsione e richiamando il metodo precedente modifica di volta in volta la lista di regioni (parziale) e il numero di comuni (n) a cui destinare il finanziamento restituendo così la soluzione migliore.

```

public List<Regione> calcolaMiglioreFot (int n) {
    miglioreS = new LinkedList<Regione>();
    List<Regione> parziale = new LinkedList<Regione>();
    aM = Double.MAX_VALUE;

    cercaMiglioreFot(parziale,0,n);
    return miglioreS;
}

```

Figura 12: Metodo cercaMiglioreFot() – Model

- ♦ il metodo *cercaMiglioreFot(List<Regione> parziale, int L, int n)* che, passando una lista di regioni (selezionate dall'utente), un livello di riferimento (L) e un intero (n), il metodo restituisce il massimo numero di comuni (n) con meno di cinquemila abitanti a cui assegnare il finanziamento per la realizzazione di nuovi impianti solari fotovoltaici, la cui produzione fotovoltaica comunale (in percentuale) è minore rispetto a quella degli altri comuni italiani;

```

private void cercaMiglioreFot(List<Regione> parziale, int L, int n) {
    int sommaComuni = sommaComuni(parziale);

    if(sommaComuni>n)
        return;

    if(sommaComuni != 0 && sommaComuni <= n) {
        double param = calcolaParametroFot(parziale);
        if(param < aM && sommaComuni > sommaComuni(miglioreS)) {
            miglioreS = new LinkedList<Regione>(parziale);
            aM = param;
        }
    }

    if(L == reg.size()) {
        return;
    }

    if(reg.get(L).getProdFotovoltaicoPerComune()>0) {
        parziale.add(reg.get(L));
        this.cercaMiglioreFot(parziale, L+1, n);

        parziale.remove(reg.get(L));
        this.cercaMiglioreFot(parziale, L+1, n);
    }
}

```

Figura 13: Metodo *cercaMiglioreFot()* – Model

- ♦ Il metodo *calcolaParametroFot(List<Regione> parziale)* che calcola la produzione eolica per comune in percentuale (rispetto al numero di comuni italiani).

```

private double calcolaParametroFot(List<Regione> parziale) {
    double s = 0;
    double nC = 0;

    for(Regione r:parziale) {
        s += r.getProdFotovoltaicoPerComune();
        nC += r.getComuniTot();
    }

    return s/nC;
}

```

Figura 14: Metodo *calcolaParametroFot()* – Model

L'ultima ricorsione, relativa alla scelta tra solare fotovoltaico ed eolico, è più complessa ed è articolata in più metodi. Quelli principali sono:

- ♦ il metodo *calcolaMigliore (int n)* che avvia la ricorsione e richiamando il metodo precedente modifica di volta in volta la lista di regioni (parziale) e il numero di comuni (n) a cui destinare il finanziamento restituendo così la soluzione migliore.

```

public void calcolaMigliore (int n) {
    miglioreS = new LinkedList<Regione>();
    miglioreE = new LinkedList<Regione>();
    List<Regione> parzialeE = new LinkedList<Regione>();
    List<Regione> parzialeS = new LinkedList<Regione>();
    aME = Double.MAX_VALUE;
    aMS = Double.MAX_VALUE;

    cercaMiglioreFotEol(parzialeE, parzialeS ,0,n);
}

```

Figura 15: Metodo ricorsivo calcolaMigliore() _solare ed eolico – Model

- ♦ il metodo *cercaMiglioreFotEol(List<Regione> parziale, int L, int n)* che, passando una lista di regioni (selezionate dall'utente), un livello di riferimento (L) e un intero (n), il metodo restituisce il massimo numero di comuni (n) con meno di cinquemila abitanti a cui assegnare il finanziamento per la realizzazione di nuovi impianti solari fotovoltaici e eolici, la cui produzione comunale eolica o solare, in percentuale, è minore rispetto a quella degli altri comuni italiani. Ne deriva che i finanziamenti per l'eolico sono assegnati alle regioni la cui produzione comunale è minore per l'eolico rispetto al fotovoltaico e quelli per il fotovoltaico alle regioni la cui produzione comunale è minore per il fotovoltaico rispetto all'eolico.

```

private void cercaMiglioreFotEol(List<Regione> parzialeE, List<Regione> parzialeS, int L, int n) {
    int sommaComuniE = sommaComuni(parzialeE);
    int sommaComuniS = sommaComuni(parzialeS);

    if(sommaComuniE + sommaComuniS > n)
        return;

    if(sommaComuniE + sommaComuniS <= n) {
        double paramE = calcolaParametroEol(parzialeE);
        double paramS = calcolaParametroFot(parzialeS);

        if(paramE < aME ) {
            miglioreE = new LinkedList<Regione>(parzialeE);
            aME = paramE;
        }

        if(paramS < aMS ) {
            miglioreS = new LinkedList<Regione>(parzialeS);
            aMS = paramS;
        }
    }

    if(L == reg.size()) {
        return;
    }

    if(reg.get(L).getProdEolicoPerComune() < reg.get(L).getProdFotovoltaicoPerComune()
        && reg.get(L).getProdEolicoPerComune() != 0) {
        parzialeE.add(reg.get(L));
        this.cercaMiglioreFotEol(parzialeE, parzialeS, L+1, n);
        parzialeE.remove(reg.get(L));
        this.cercaMiglioreFotEol(parzialeE, parzialeS, L+1, n);
    }
    if (reg.get(L).getProdEolicoPerComune() > reg.get(L).getProdFotovoltaicoPerComune()
        && reg.get(L).getProdFotovoltaicoPerComune() != 0 ){
        parzialeS.add(reg.get(L));
        this.cercaMiglioreFotEol(parzialeE, parzialeS, L+1, n);
        parzialeS.remove(reg.get(L));
        this.cercaMiglioreFotEol(parzialeE, parzialeS, L+1, n);
    }
}

```

Figura 16: Metodo ricorsivo cercaMiglioreFotEol() –Model

- ♦ il metodo *calcolaBudget(float budget, List<Regione>migliore)* che riceve in input un budget totale di finanziamento e la lista di regioni ottenuta dalle ricorsioni (“migliore”) e calcola l’importo da assegnare ad ogni comune, dividendo il budget totale per il numero di comuni con meno di cinquemila abitanti, ai quali verrà assegnato il finanziamento, come indicato nel PNRR.

```
public float calcolaBudget (float budget, List<Regione> migliore) {
    int nComuni = 0;
    float budgetAssegnato = 0;

    for(Regione r:migliore) {
        nComuni += r.getComuniMeno5000();
    }

    budgetAssegnato = (budget/nComuni);
    return budgetAssegnato;
}
```

Figura 17: Metodo *calcolaBudget()* – Model

I metodi ricorsivi e il metodo *calcolaBudget()* vengono poi richiamati nell’FXMLController attraverso il metodo *doRicorsione()*, permettendo all’utente di interagire con il programma.

In particolare la lista ottenuta dal metodo *calcolaMigliore()* viene inserita all’interno del ciclo for che restituisce le regioni a cui assegnare il finanziamento. Nel ciclo for viene moltiplicato il risultato ottenuto dal metodo *calcolaBudget()* per il numero di comuni delle regioni a cui destinare il finanziamento. Infine viene restituito il budget assegnato ad ogni regione.

```
void doRicorsione(ActionEvent event) {
    txtResult.clear();
    areaScelta = this.cmbAreaDiRiferimento.getValue();
    produzioneDi = this.boxParamRicorsione.getValue();

    int numeroComuni;
    float budgetFinanziamento;

    try {
        numeroComuni=Integer.parseInt(this.txtNumeroComuni.getText());
        budgetFinanziamento=Float.parseFloat(this.txtBudget.getText());
        if (this.boxParamRicorsione.getValue() == null) {
            txtResult.appendText("Selezionare la fonte di energia elettrica da incrementare\n");
            return;
        }
    } catch (NumberFormatException e) {
        txtResult.clear();
        txtResult.appendText("Inserire i parametri per la ricorsione");
        return;
    }
    if(areaScelta==null) {
        txtResult.appendText("Selezionare un'area di riferimento");
        return;
    }
    if(produzione==false) {
        txtResult.appendText("Calcolare prima la produzione per comune");
    }

    if(produzioneDi.compareTo("Eolico")==0) {
        float budgetAssegnato=0;
        txtResult.appendText("Regioni a cui assegnare i finanziamenti per l'Eolico:\n");

        for(Regione r: model.calcolaMiglioreEol(numeroComuni)) {
            budgetAssegnato=model.calcolaBudget(budgetFinanziamento, model.calcolaMiglioreEol(numeroComuni))*r.getComuniMeno5000();
            txtResult.appendText(r+ " , " + r.getComuniMeno5000() + " comuni, ");
            txtResult.appendText("budget assegnato: " + String.format("%.2f",BigDecimal.valueOf(budgetAssegnato)) + " €\n");
        }
    }
}
```

Figura 18: Metodo *doRicorsione()* _ Pt.1 - FXMLController

```

if(produzioneDi.compareTo("Fotovoltaico")==0) {
    float budgetAssegnato=0;
    txtResult.appendText("Regioni a cui assegnare i finanziamenti per il Solare Fotovoltaico:\n");

    for(Regione r: model.calcolaMiglioreFot(numeroComuni)) {
        txtResult.appendText(r+ ", " + r.getComuniMeno5000() + " comuni, " );
        budgetAssegnato=model.calcolaBudget(budgetFinanziamento, model.calcolaMiglioreFot(numeroComuni))*r.getComuniMeno5000();
        txtResult.appendText("budget assegnato: "+ String.format("%.2f", BigDecimal.valueOf(budgetAssegnato))+ " €\n");
    }
}

if(produzioneDi.compareTo("Eolico e Fotovoltaico")==0) {
    float budgetAssegnato=0;
    numeroComuni=Integer.parseInt(this.txtNumeroComuni.getText());
    model.calcolaMigliore(numeroComuni);

    List<Regione> temporanea = new LinkedList<Regione>(model.getMiglioreE());
    temporanea.addAll(model.getMiglioreS());

    txtResult.appendText("Regioni a cui assegnare i finanziamenti per l'Eolico:\n");
    for(Regione r: model.getMiglioreE()) {
        txtResult.appendText(r+ ", " + r.getComuniMeno5000() + " comuni, " );
        budgetAssegnato=model.calcolaBudget(budgetFinanziamento,temporanea)*r.getComuniMeno5000();
        txtResult.appendText("budget assegnato: "+ String.format("%.2f", BigDecimal.valueOf(budgetAssegnato))+ " €\n");
    }

    txtResult.appendText("\nRegioni a cui assegnare i finanziamenti per il Solare Fotovoltaico:\n");
    for(Regione r: model.getMiglioreS()) {
        txtResult.appendText(r+ ", " + r.getComuniMeno5000() + " comuni, " );
        budgetAssegnato=model.calcolaBudget(budgetFinanziamento, temporanea)*r.getComuniMeno5000();
        txtResult.appendText("budget assegnato: "+ String.format("%.2f", BigDecimal.valueOf(budgetAssegnato))+ " €\n");
    }
}
}

```

Figura 19: Metodo doRicorsione()_ Pt.2 - FXMLController

Le stampe degli output della ricorsione vengono riportate nella textArea TxtResult.

Capitolo 5: Diagramma delle classi delle parti principali

Seguendo i patterns MVC e DAO, il collegamento al database viene effettuato mediante la classe DBConnect. I dati sono ottenuti dal database attraverso la classe EnergiaDAO del package: *it.polito.tdp.DeSeriisRoberta.db*.

La classe **Regione** a cui fa riferimento il metodo *getAllRegioni()* (della classe EnergiaDAO) è stata creata per raggruppare tutte le informazioni ottenute dal database. Agli attributi ripresi dal database ne sono stati aggiunti altri due (*prodFotovoltaicaPerComune*, *prodEolicaPerComune*) nei quali vengono inseriti i valori calcolati dai metodi:

- ♦ *calcolaProduzioneFotovoltaicoComune(List<Regione> regioni);*
- ♦ *calcolaProduzioneEolicoComune(List<Regione> regioni).*

Nella medesima classe sono stati inoltre creati get e set degli attributi, per renderli fruibili anche alle altre classi.

La classe **Model** è quella in cui sono riportati i principali algoritmi.

Essa si compone come segue:

Model

```
Map<String, Regione> idMap;  
List<Regione> regioniItalia;  
List<Regione> reg;  
List<Regione> regioniSelezionate;  
List<Regione> miglioreE;  
List<Regione> miglioreS;  
double aM;  
double aME;  
double aMS;
```

```
public Model()  
public List<Regione> calcolaProduzioneFotovoltaicoComune (List<Regione> regioni)  
public List<Regione> calcolaProduzioneEolicoComune (List<Regione> regioni)  
public List<Regione> statoRegioni (String tipo)  
public List<Regione> calcolaMiglioreEol (int n)  
private void cercaMiglioreEol (List<Regione> parziale, int L, int n)  
private double calcolaParametroEol (List<Regione> parziale)  
private int sommaComuni (List<Regione> parziale)  
public List<Regione> calcolaMiglioreFot (int n)  
private void cercaMiglioreFot (List<Regione> parziale, int L, int n)  
private double calcolaParametroFot (List<Regione> parziale)  
public void calcolaMigliore (int n)  
private void cercaMiglioreFotEol (List<Regione> parzialeE, List<Regione> parzialeS, int L, int n)  
public List<Regione> getMiglioreE()  
public List<Regione> getMiglioreS()  
public float calcolaBudget (float budget, List<Regione> migliore)
```

Capitolo 6: Esempi di utilizzo dell'applicazione

6.1 Videata dell'applicazione

La schermata principale dell'applicativo si compone di tre sezioni:

- ♦ produzione per comune;
- ♦ parametri ricorsione;
- ♦ output.

The screenshot shows a JavaFX application window titled "JavaFX and Maven". The main content area has a header with the logo "ECER" and the subtitle "Energia e Comunità Energetiche Rinnovabili". Below the header is a central illustration depicting a sustainable energy landscape with a factory, a wind turbine, solar panels, a house, and trees. The interface is divided into three main sections:

- Produzione per Comune**: Contains a dropdown menu for "Area di riferimento" and a green button labeled "Calcola Produzione".
- Parametri Ricorsione**: Contains two input fields: "Numero di comuni" and "Budget di finanziamento (€)". Below these is a dropdown for "Fonte di energia elettrica" and a green button labeled "Avvia ricorsione".
- Output**: Features a large empty text area for results. Below it is a table with the following structure:

Regione	Produzione Fotovoltaico (GWh)	Produzione Eolico (GWh)
Nessun contenuto nella tabella		

A green "Reset" button is located at the bottom right of the output section.

Figura 20: Schermata applicazione – immagine CER (danilodiflorio, s.d.)

Nella prima sezione (Produzione per Comune) è possibile selezionare un'area di riferimento tra:

- ♦ Italia;
- ♦ Italia: PS;
- ♦ Italia: MS;
- ♦ Italia: IT.

Dopo la selezione, cliccando su **Calcola Produzione**, nella *TableView* della sezione Output, verranno stampate le produzioni fotovoltaiche ed eoliche per comune, espresse in GWh e raggruppate per regione.

Nella seconda sezione (Parametri Ricorsione), dopo aver selezionato il numero massimo di comuni a cui conferire il finanziamento, il budget totale di finanziamento e la fonte di produzione di energia elettrica da incrementare, cliccando su **Avvia Ricorsione**, si otterranno gli output desiderati riportati nella *TextArea* **txtResult**.

Cliccando su **Reset** verranno ripuliti *TextField*, *TextArea* e *TableView*.

6.2 Video dimostrativo dell'applicazione software

È possibile guardare un video dimostrativo sul funzionamento del programma al seguente link:

(<https://youtu.be/kYK9aQIIITU>)

Capitolo 7: Risultati sperimentali ottenuti

I risultati ottenuti dalle ricorsioni variano in base ai parametri in input. Si riportano di seguito alcuni dei risultati ottenuti.

Selezionando come area di riferimento “Italia”, come numero di comuni massimo “1000”, come budget di finanziamento (2'200'000'000 euro) e come produzione da massimizzare “Eolico”, i risultati ottenuti sono:

Output
Regioni a cui assegnare i finanziamenti per l'Eolico: Abruzzo, 253 comuni, budget assegnato: 1.434.536.064,00 € Emilia-Romagna, 135 comuni, budget assegnato: 765.463.936,00 €

Figura 21: Output Eolico (Italia)

Selezionando come area di riferimento Italia: MS e cambiando la produzione da massimizzare in “Fotovoltaico”, i risultati ottenuti sono:

Output
Regioni a cui assegnare i finanziamenti per il Solare Fotovoltaico: Calabria, 326 comuni, budget assegnato: 898.746.880,00 € Campania, 344 comuni, budget assegnato: 948.370.944,00 € Molise, 128 comuni, budget assegnato: 352.882.208,00 €

Figura 22: Output Fotovoltaico (Italia: MS)

Selezionando come area di riferimento Italia e cambiando la produzione da massimizzare in “Eolico e Fotovoltaico”, i risultati ottenuti sono:

Output
Regioni a cui assegnare i finanziamenti per l'Eolico: Emilia-Romagna, 135 comuni, budget assegnato: 620.041.792,00 € Regioni a cui assegnare i finanziamenti per il Solare Fotovoltaico: Campania, 344 comuni, budget assegnato: 1.579.958.272,00 €

Figura 23: Output Eolico e Fotovoltaico (Italia)

Il tempo di elaborazione delle singole ricorsioni, inserendo i dati riportati precedentemente, è pari a 0,0007 millisecondi.

Capitolo 8: Considerazioni finali

8.1 Valutazione dei risultati ottenuti

Nel complesso i risultati ottenuti dall'esecuzione del programma sono coerenti con il contesto attuale. I finanziamenti vengono destinati alle regioni che al momento presentano una produzione eolica o solare inferiore a quella delle altre regioni, considerando il numero di comuni. Ad esempio, nel caso dell'output "Eolico e Fotovoltaico" la Campania, nonostante abbia una produzione solare più alta di altre regioni, è preferita per l'elevato numero di comuni.

Le ricorsioni che puntano a massimizzare il numero di comuni con meno di cinquemila abitanti a cui conferire i finanziamenti per l'incremento di nuova capacità eolica e/o solare risultano dunque corrette e costituiscono un supporto per l'adozione delle decisioni relative alla ripartizione dei fondi.

Si può ritenere che, la scarsa disponibilità dei dati aggiornati unita all'assenza di informazioni con risoluzione inferiore a quella regionale, abbiano reso carente la componente algoritmica non consentendo una maggiore precisione relativa all'analisi e alla valutazione della soluzione migliore.

8.2 Conclusioni

Gli obiettivi principali del programma si possono ritenere raggiunti: il software funziona e potrebbe essere utilizzato per gli scopi individuati. Soluzioni più dettagliate potrebbero essere implementate sviluppando ulteriormente il programma. Questa versione base del software potrebbe essere di supporto a coloro i quali verranno chiamati a decidere relativamente alle procedure di ripartizione dei fondi di finanziamento.

Questa relazione tecnica è rilasciata con licenza Creative Commons BY-NC-SA.

Tu sei libero di:

- Condividere - riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato
- Modificare - remixare, trasformare il materiale e basarti su di esso per le tue opere

Alle seguenti condizioni:

- Attribuzione - Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.
- Non Commerciale - Non puoi utilizzare il materiale per scopi commerciali.
- StessaLicenza - Se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

Per visualizzare una copia di questa licenza, visitare : <https://creativecommons.org/licenses/by-nc-sa/4.0/>



Riferimenti

www.mise.gov.it. (2021). Tratto da

https://www.mise.gov.it/images/stories/documenti/PNRR_Aggiornato

www.istat.it. (2020). Tratto da <https://www.istat.it/it/archivio/16777>

www.gse.it. (s.d.). Tratto da

https://www.anita.it/public/files/pagine/D621_Rapporto%20statistico%20del%20GSE%20Energia%20da%20Fonti%20Rinnovabili%20in%20Italia.pdf

www.ministroperilsud.gov.it. (2021-2027). Tratto da

<https://www.ministroperilsud.gov.it/it/approfondimenti/fondi-strutturali-europei-2021-2027/il-ciclo-di-programmazione-2021-2027-in-italia>

www.tuttitalia.it. (2022). Tratto da <https://www.tuttitalia.it/regioni/numero-comuni/>

www.signavio.com. (s.d.). Tratto da <https://academic.signavio.com/>

www.wikipedia.org. (s.d.). Tratto da https://it.wikipedia.org/wiki/Reddito_pro_capite

www.danilodiflorio.it. Tratto da <http://danilodiflorio.it/le-comunita-energetiche-cosa-sono-e-perche-sceglierle/>. Danilodiflorio. (s.d.).

www.youtube.com. Tratto da <https://youtu.be/kYK9aQIITU>. Link al video dimostrativo.

creativecommons.org. Tratto da <https://creativecommons.org/licenses/by-nc-sa/4.0/>. Licenza.

Indice delle figure

Figura 1: Produzione solare fotovoltaico	3
Figura 2: Produzione eolico	3
Figura 3: classe Regione, realizzata con Signavio Academic [6]	5
Figura 4: Produzione normalizzata - Rapporto Statistico del GSE sulle FER 2020.....	5
Figura 5: Struttura dati	7
Figura 6: Metodi calcolaProduzioneComune() (eolico e solare fotovoltaico) - Model.....	8
Figura 7: Metodo statoRegioni() – Model	8
Figura 8: Metodo doProduzione() – FXMMLController	9
Figura 9: Metodi ricorsivi: calcolaMiglioreEol() e CercaMiglioreEol() – Model.....	10
Figura 10: Metodo calcolaParametroEol() - Model.....	11
Figura 11: Metodo sommaComuni() –Model	11
Figura 12: Metodo cercaMiglioreFot() – Model.....	11
Figura 13: Metodo cercaMiglioreFot() – Model.....	12
Figura 14: Metodo calcolaParametroFot() – Model	12
Figura 15: Metodo ricorsivo calcolaMigliore() _solare ed eolico – Model.....	13
Figura 16: Metodo ricorsivo cercaMiglioreFotEol() –Model	13
Figura 17: Metodo calcolaBudget() – Model.....	14
Figura 18: Metodo doRicorsione() _ Pt.1 - FXMMLController	14
Figura 19: Metodo doRicorsione()_ Pt.2 - FXMMLController	15
Figura 20: Schermata applicazione – immagine CER [8].....	17
Figura 21: Output Eolico (Italia).....	19
Figura 22: Output Fotovoltaico (Italia: MS)	19
Figura 23: Output Eolico e Fotovoltaico (Italia).....	19