



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea in Ingegneria Gestionale

Classe L-8

Sessione di Laurea Luglio 2024

Laurea Triennale in Ingegneria Gestionale

**Software per l'analisi statistica di sistemi
energetici rinnovabili e valutazione degli
investimenti per i clienti**

Relatore:

Giuseppe Bruno Averta

Candidato:

Alexandra Elena Holota

ANNO ACCADEMICO 2023/2024

Indice

Sitografia	4
1 Proposta di progetto	5
1.1 Studente proponente	5
1.2 Titolo della proposta	5
1.3 Descrizione del problema proposto	5
1.4 Descrizione della rilevanza gestionale del problema	6
1.5 Descrizione dei data-set per la valutazione	7
1.6 Descrizione preliminare degli algoritmi coinvolti	8
1.7 Descrizione preliminare delle funzionalità previste per l'applicazione software	8
2 Descrizione dettagliata del problema affrontato	9
2.1 Analisi statistica	9
2.2 Soluzione per cliente	9
3 Descrizione del data-set utilizzato per l'analisi	10
3.1 Tabella: " <i>RenewableEnergySystems</i> "	11
3.2 Tabella: " <i>RenewableEnergy</i> "	12
3.3 Tabella: " <i>Options</i> "	13
4 Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati	14
4.1 UI - User Interface	14
4.1.1 Controller	15
4.1.2 View	20
4.2 Model	20
4.2.1 Classe Model	20
4.2.2 Metodi della classe Model	21
4.3 Database	24
4.3.1 Metodi classe DAO	24
5 Alcune videate dell'applicazione e link al video	27
5.1 Controlli sui dati inseriti	28
5.2 Link al video	28
6 Tabelle con risultati sperimentali ottenuti	31
6.1 Analisi generale	31
6.2 Analisi specifica	33
6.3 Soluzione per il cliente	34

7	Valutazioni sui risultati ottenuti e conclusioni	37
7.1	Analisi preliminare	37
7.2	Ricerca di soluzioni per il cliente	37
7.3	Conclusione	37
	Licenza	38

Elenco delle figure

3.1	Diagramma ER della struttura del data-set <i>"energy"</i>	10
3.2	Diagramma ER della tabella <i>"RenewableEnergySystems"</i>	12
3.3	Diagramma ER della tabella <i>"RenewableEnergy"</i>	12
3.4	Diagramma ER della tabella <i>"Options"</i>	13
4.1	Struttura package <i>UI</i>	14
4.2	Funzioni per la popolazione dei dropdowns	16
4.3	Funzioni per la lettura dei dati selezionati nei dropdowns	16
4.4	Metodo <code>get_Analisi(self, e)</code>	18
4.5	Metodo <code>get_SoluzioneCliente(self, e)</code>	19
4.6	Struttura package <i>Model</i>	20
4.7	Strutture delle classi <i>FonteRinnovabile</i> , <i>Option</i> e <i>SistemaEnergeticoRinnovabile</i>	20
4.8	Metodi per l'analisi preliminare	22
4.9	Metodo per costruire il grafo	22
4.10	Metodi per cercare le tre soluzioni ottimali	23
4.11	Struttura package <i>database</i>	24
4.12	Metodi della classe <i>DAO</i>	25
4.13	Metodi della classe <i>DAO</i>	26
5.1	Videata iniziale	27
5.2	Videata successiva all'analisi preliminare	28
5.3	Controllo sull'inserimento del sistema energetico	29
5.4	Controllo sull'inserimento delle preferenze dell'utente	29
5.5	Controllo sulla validità dell'inserimento dell'investimento iniziale	30
5.6	Controllo sulla validità dell'inserimento della capacita minima	30
6.1	Risultati analisi generale	32
6.2	Risultati analisi generica	32
6.3	Risultati analisi specifica	33
6.4	Risultati analisi specifica	34
6.5	Risultati della ricerca di soluzione per il cliente, caso default con tre soluzioni	35
6.6	Risultati della ricerca di soluzione per il cliente, caso particolare in cui si riportano tutte le soluzioni disponibili	35
6.7	Risultati della ricerca di soluzione per il cliente, caso particolare in cui non ci sono soluzioni da proporre	36

Sitografia

- Kaggle - “Dataset for renewable energy systems”, <https://www.kaggle.com/datasets/girumwondemagegn/dataset-for-renewable-energy-systems>

Capitolo 1

Proposta di progetto

1.1 Studente proponente

s297022 Holota Alexandra Elena

1.2 Titolo della proposta

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

1.3 Descrizione del problema proposto

L'idea progettuale consiste nello sviluppo di un software capace di fornire un'analisi preliminare, specifica o generica, riguardante l'installazione di sistemi energetici rinnovabili e una soluzione per il cliente in termini di investimenti iniziali. L'analisi preliminare specifica avviene su una singola fonte rinnovabile, scelta dall'utente, rispetto alla quale verranno riportati i valori minimi, massimi e medi scelti tra le seguenti categorie:

- capacità installata in megawatt (MW);
- produzione annuale di energia in megawattora (MWh);
- consumo energetico annuale in megawattora (MWh);
- capacità di stoccaggio di energia in megawattora (MWh);
- efficienza dei sistemi di accumulo di energia in percentuale;
- costi di investimento iniziali in USD;
- riduzione delle emissioni di gas a effetto serra in tonnellate di CO₂ equivalente;
- indice di riduzione dell'inquinamento atmosferico;
- numero di lavori creati.

Inoltre, l'utente potrà richiedere un'analisi generica su una singola fonte rinnovabile la quale verrà effettuata rispetto al valore minimo, massimo e medio relativo alle seguenti categorie di default:

- capacità installata in megawatt (MW);
- produzione annuale di energia in megawattora (MWh);
- consumo energetico annuale in megawattora (MWh);
- capacità di stoccaggio di energia in megawattora (MWh);
- costi di investimento iniziali in USD;
- indice di riduzione dell'inquinamento atmosferico.

Il cliente, successivamente, potrà richiedere un preventivo scegliendo

- la tipologia di fonte rinnovabile;
- l'investimento iniziale;
- il valore di capacità minimo necessario.

Inifine, il software proporrà le tre soluzioni ottimali al fine di rispettare le necessità del cliente, indicando: l'investimento iniziale, il valore di capacità installata e l'indice di riduzione dell'inquinamento atmosferico.

1.4 Descrizione della rilevanza gestionale del problema

Negli ultimi anni, la sostenibilità ambientale è diventata una priorità per molte aziende, non solo per motivi etici, ma anche per migliorare la propria reputazione ecologica, la cosiddetta Green Reputation. Inoltre, l'impegno verso la sostenibilità ambientale può aiutare le imprese a rafforzare la propria immagine, accrescere la fiducia dei consumatori e migliorare la loro posizione competitiva sul mercato. Una delle principali modalità in cui le aziende possono migliorare il loro impatto ambientale è l'investimento in fonti di energia rinnovabili. Sulla base di tali premesse, potrebbe essere utile poter visionare statistiche sulle produzioni, sul valore degli investimenti e sulle conseguenze ambientali relative all'installazione di differenti sistemi di energia rinnovabile.

1.5 Descrizione dei data-set per la valutazione

La risoluzione di tale problema avviene attraverso l'utilizzo di un data-set pubblicato sul sito Kaggle, in particolare si fa riferimento a “*Dataset for renewable energy systems*”, accessibile dal link:

<https://www.kaggle.com/datasets/girumwondemagegn/dataset-for-renewable-energy-systems>

Il set di dati contiene informazioni dettagliate su vari sistemi di energia rinnovabile:

- **id**: codice numerico che indentifica in modo univoco ogni sistema energetico rinnovabile;
- **Type_of_Renewable_Energy**: codice numerico che rappresenta il tipo di fonte di energia rinnovabile (1: Solare, 2: Eolico, 3: Idroelettrico, 4: Geotermico, 5: Biomassa, 6: Maremotrice, 7: Moto ondoso);
- **Installed_Capacity_MW**: capacità installata in megawatt (MW);
- **Energy_Production_MWh**: produzione annuale di energia in megawattora (MWh);
- **Energy_Consumption_MWh**: consumo energetico annuale in megawattora (MWh);
- **Energy_Storage_Capacity_MWh**: capacità di stoccaggio di energia in megawattora (MWh);
- **Storage_Efficiency_Percentage**: efficienza dei sistemi di accumulo di energia in percentuale;
- **Grid_Integration_Level**: codice numerico che rappresenta il livello di integrazione della griglia (1: completamente integrato, 2: parzialmente integrato, 3: integrazione minima, 4: microgriglia isolata);
- **Initial_Investment_USD**: costi di investimento iniziali in USD;
- **Funding_Sources**: codice numerico che rappresenta la fonte di finanziamento (1: Government, 2: Private, 3: Public-Private Partnership);
- **Financial_Incentives_USD**: incentivi finanziari in USD;
- **GHG_Emission_Reduction_tCO2e**: riduzione delle emissioni di gas a effetto serra in tonnellate di CO₂ equivalente (tCO₂e);
- **Air_Pollution_Reduction_Index**: Indice di riduzione dell'inquinamento atmosferico;
- **Jobs_Created**: numero di lavori creati;

L'obiettivo di tale data-set è quello di fornire una visione completa delle fonti di energie rinnovabili, utile per la ricerca e l'analisi nel campo dell'energia sostenibile.

1.6 Descrizione preliminare degli algoritmi coinvolti

Il programma sarà sviluppato usando il linguaggio Python, utilizzando i pattern MVC (Model View Controller) e DAO (Data Access Object) attraverso i quali vengono separati l'interfaccia utente, la logica applicativa e l'accesso ai dati.

Per lo sviluppo dell'analisi preliminare specifica e generale, verrà richiesto all'utente di scegliere la fonte rinnovabile da analizzare e una categoria. Successivamente, verranno raccolti i dati di minimo, massimo e medio tramite query SQL relativi alle preferenze espresse in precedenza.

Per lo sviluppo della soluzione per il cliente, sarà richiesta la selezione della tipologia di fonte rinnovabile, dell'investimento iniziale massimo e del valore minimo di capacità necessario. I dati utili per la ricerca delle tre soluzioni saranno estratti tramite query SQL, filtrando i sistemi che soddisfano le specifiche richieste dal cliente. Utilizzando un grafo e implementando un metodo basato sulla ricerca ricorsiva, si proporranno al cliente le tre soluzioni ottimali.

1.7 Descrizione preliminare delle funzionalità previste per l'applicazione software

L'applicazione si suddividerà in due sezioni: la prima dedicata all'analisi preliminare e la seconda alla ricerca della soluzione da proporre al cliente.

Il menù a tendina per la scelta della fonte rinnovabile sarà condiviso tra entrambe le sezioni.

Nella prima sezione, una volta scelta la fonte rinnovabile, il cliente potrà selezionare singolarmente le categorie da analizzare oppure optare per un'analisi generica, semplicemente lasciando vuoto il dropdown o scegliendo l'opzione "*Analisi generale*".

I risultati dell'analisi saranno visualizzati a schermo dopo che l'utente avrà premuto il pulsante "*Analisi*".

La seconda sezione sarà composta da due campi di testo in cui l'utente potrà inserire il valore massimo dell'investimento iniziale e il valore minimo di capacità desiderato. Dopo aver premuto il pulsante "*Trova*", le tre migliori soluzioni verranno mostrate a schermo.

Capitolo 2

Descrizione dettagliata del problema affrontato

Il software proposto rappresenta uno strumento utile per le scelte delle aziende che intendono adottare strategie energetiche sostenibili.

2.1 Analisi statistica

Le aziende possono valutare con precisione i benefici e i requisiti associati a diverse fonti rinnovabili, attraverso l'analisi fornita dal software. Questa comprende parametri tecnici come la capacità installata, la produzione annuale di energia, il consumo energetico, la capacità di stoccaggio e l'efficienza degli accumuli energetici. Questi dati tecnici sono importanti per capire quale tipo di sistema energetico sia più adatto alle specifiche esigenze operative dell'azienda, garantendo così un utilizzo ottimale delle risorse energetiche rinnovabili. Oltre agli aspetti tecnici, il software include un'analisi economica che copre i costi iniziali di investimento e il numero di posti di lavoro che può creare l'installazione di un sistema energetico rinnovabile.

L'implementazione di soluzioni energetiche innovative supporta le aziende nella transizione verso pratiche più sostenibili, con una conseguente riduzione dell'impatto ambientale e un contributo sia alla lotta al cambiamento climatico sia al miglioramento della reputazione aziendale. Questa trasformazione può portare ad un aumento della fiducia dei clienti, ad una maggiore lealtà al marchio e ad una crescita delle opportunità di investimento.

2.2 Soluzione per cliente

Per il cliente, il software offre la possibilità di definire criteri specifici come la tipologia di fonte rinnovabile, l'investimento iniziale disposto a finanziare e il valore minimo di capacità necessario, con il fine di ottenere indicazioni su misura che rispondano esattamente alle esigenze operative e finanziarie specificate. Infatti, il software, analizzando queste preferenze, propone i tre migliori sistemi energetici rinnovabili disponibili, evidenziando non solo i costi iniziali, ma anche i dettagli sulla capacità installata e l'impatto positivo sull'ambiente, come l'indice di riduzione dell'inquinamento atmosferico.

Capitolo 3

Descrizione del data-set utilizzato per l'analisi

Come indicato nella proposta, il dataset utilizzato è pubblicato sul sito Kaggle, in particolare si fa riferimento a “*Dataset for renewable energy systems*”, accessibile dal link :

<https://www.kaggle.com/datasets/girumwondemagegn/dataset-for-renewable-energy-systems>

Il data-set è stato creato con il nome “*energy*” ed era in origine costituito da una sola tabella “*RenewableEnergySystems*”. In seguito, sono state aggiunte manualmente due tabelle, “*RenewableSource*” e “*Options*”. In Figura 3.1 si riporta il diagramma ER della struttura del data-set.

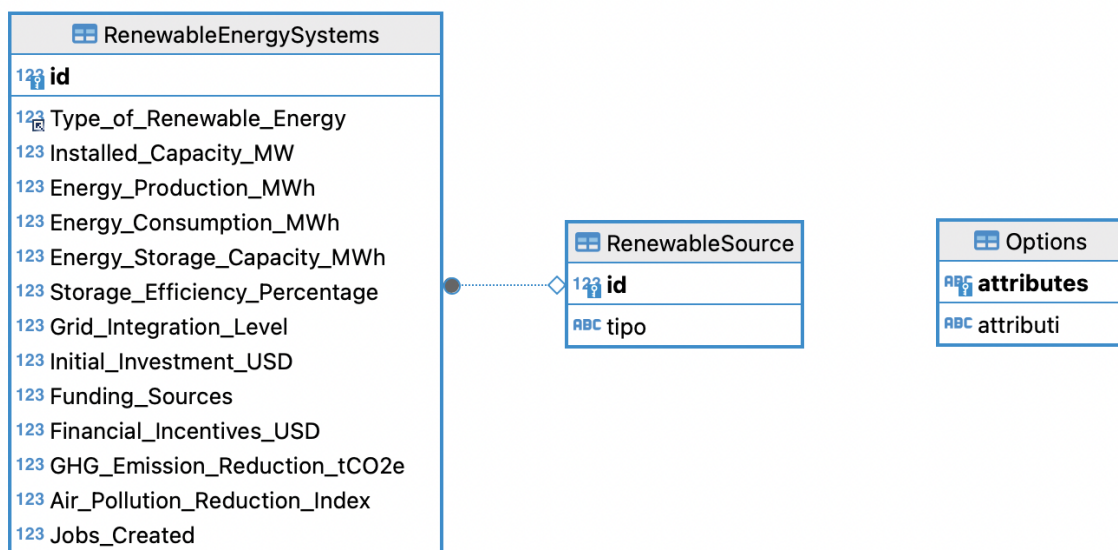


Figura 3.1: Diagramma ER della struttura del data-set “*energy*”

3.1 Tabella: "*RenewableEnergySystems*"

La tabella "*RenewableEnergySystems*" è un insieme di dati che ha lo scopo di fornire una visione completa dei sistemi di energia rinnovabile e delle loro caratteristiche. Questa tabella è molto utile per la ricerca e l'analisi nel campo dell'energia sostenibile grazie alla sua ampia quantità di dati.

La tabella è costituita dai seguenti attributi:

- **id**: codice numerico che identifica in modo univoco ogni sistema energetico rinnovabile;
- **Type_of_Renewable_Energy**: codice numerico che rappresenta il tipo di fonte di energia rinnovabile:
 - 1: Solare
 - 2: Eolico
 - 3: Idroelettrico
 - 4: Geotermico
 - 5: Biomassa
 - 6: Mareomotrice
 - 7: Moto ondoso
- **Installed_Capacity_MW**: capacità installata in megawatt (MW);
- **Energy_Production_MWh**: produzione annuale di energia in megawattora (MWh);
- **Energy_Consumption_MWh**: consumo energetico annuale in megawattora (MWh);
- **Energy_Storage_Capacity_MWh**: capacità di stoccaggio di energia in megawattora (MWh);
- **Storage_Efficiency_Percentage**: efficienza dei sistemi di accumulo di energia in percentuale;
- **Grid_Integration_Level**: codice numerico che rappresenta il livello di integrazione della griglia (1: completamente integrato, 2: parzialmente integrato, 3: integrazione minima, 4: microgriglia isolata);
- **Initial_Investment_USD**: costi di investimento iniziali in USD;
- **Funding_Sources**: codice numerico che rappresenta la fonte di finanziamento (1: Government, 2: Private, 3: Public-Private Partnership);
- **Financial_Incentives_USD**: incentivi finanziari in USD;
- **GHG_Emission_Reduction_tCO2e**: riduzione delle emissioni di gas a effetto serra in tonnellate di CO₂ equivalente (tCO₂e);
- **Air_Pollution_Reduction_Index**: indice di riduzione dell'inquinamento atmosferico;
- **Jobs_Created**: numero di lavori creati.

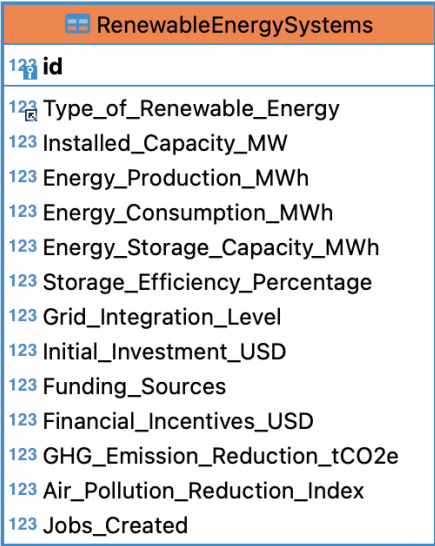


Figura 3.2: Diagramma ER della tabella "RenewableEnergySystems"

3.2 Tabella: "RenewableEnergy"

La tabella "RenewableEnergy" cataloga i vari tipi di fonti di energia rinnovabile, ciascuna identificata univocamente tramite un attributo "id" che funge da chiave esterna collegata all'attributo "Type_of_Renewable_Energy" nella tabella "RenewableEnergySystems". Ogni codice è associato al nome corrispondente della fonte, riportata nell'attributo "tipo".

Elenco dei codici, "id", e delle rispettive fonti di energia rinnovabile, "tipo":

- 1: Solare
- 2: Eolico
- 3: Idroelettrico
- 4: Geotermico
- 5: Biomassa
- 6: Maremotrice
- 7: Moto ondoso

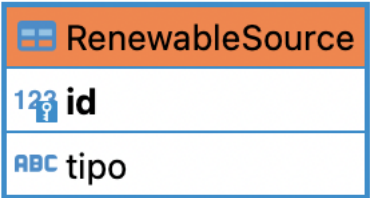


Figura 3.3: Diagramma ER della tabella "RenewableEnergy"

3.3 Tabella: "Options"

La tabella "Options" contiene gli attributi utilizzati nell'analisi specifica e generica delle fonti di energia rinnovabile. Gli attributi di questa tabella sono:

- **attributes:** Nome identificativo degli attributi presenti nella tabella "RenewableEnergySystems";
- **attributi:** Nome dell'attributo in italiano da mostrare nell'interfaccia.



Figura 3.4: Diagramma ER della tabella "Options"

Capitolo 4

Descrizione ad alto livello delle strutture dati e degli algoritmi utilizzati

Il software è stato sviluppato utilizzando il linguaggio di programmazione Python adottando i pattern MVC, Model View Controller, e DAO, Data Access Object. Sono stati realizzati tre Python packages, ciascuno con funzioni specifiche per una netta suddivisione delle funzioni all'interno dello sviluppo dell'applicazione.

I tre pacchetti Python sono l' *UI*, il *model* ed il *database* i quali implementano rispettivamente l'interfaccia utente, la logica applicativa e l'accesso ai dati.

4.1 UI - User Interface

Il package *UI* costituisce l'interfaccia visiva e interattiva tra l'utente e l'applicazione software. È progettato per rendere l'interazione con l'applicazione intuitiva, utilizzando elementi grafici quali: pulsanti, menu a tendina (dropdown) e campi di input (textfield). Nella specifica di questa applicazione, l'interfaccia è stata implementata con la libreria Flet di Python.

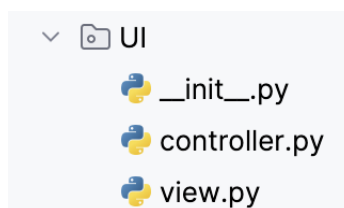


Figura 4.1: Struttura package *UI*

4.1.1 Controller

La classe "*Controller*" si occupa di gestire tutte le interazioni dell'utente nell'applicazione. Nel caso specifico, le interazioni includono la selezione della fonte da analizzare, la scelta dell'opzione per l'analisi e l'inserimento dei dati necessari per la ricerca delle tre soluzioni ottimali.

Popolamento dei menù a tendina e lettura dati

Per popolare i menù a tendina, dropdowns, sono state implementate le funzioni (Figura 4.2):

- **populated_ddFonteRinnovabile(self)**: acquisisce le fonti attraverso il metodo *getAllRenewableSource()* del *model* e le salva nella lista *fonti*. Per ogni fonte di energia rinnovabile nella lista, viene creata un'opzione dropdown e viene inserito:
 - "*data=f.id*": l'*id* della fonte sarà associato all'opzione selezionata.
 - ; "*on_click=self.readDDFonteRinnovabile*": la funzione sarà chiamata quando l'utente seleziona la specifica opzione;
 - "*text=f.tipo*": indica il testo visualizzato per l'opzione dropdown, ovvero il tipo di fonte di energia rinnovabile.

Ogni opzione creata viene aggiunta all'attributo *options* del dropdown *_ddFonteRinnovabile* nell'oggetto *_view*. Alla fine, la funzione *update_page()* viene chiamata su *_view* per aggiornare le modifiche apportate.

- **populated_ddAttributes(self)**: l'algoritmo segue lo stesso principio del metodo *populated_ddFonteRinnovabile(self)*, con l'eccezione dell'aggiunta manuale dell'opzione "*Analisi generale*" e la funzione di lettura, impostata su "*on_click=self.readDDAnalisi*"

Come anticipato, sono state implementate due funzione per la lettura dei dati selezionati dall'utente (Figura 4.3):

- **readDDFonteRinnovabile(self, e)**
- **readDDAnalisi(self, e)**

Entrambe le funzioni leggono i dati da un evento e assegnano alle variabili *self._analisi* e *self._fonteRinnovabile* i valori acquisiti.


```

1 usage (1 dynamic)  alexandrahlotota
def populated_ddFonteRinnovabile(self):
    fonti = self._model.getAllRenewableSource()
    for f in fonti:
        self._view._ddFonteRinnovabile.options.append(
            ft.dropdown.Option(
                data=f.id, on_click=self.readDDFonteRinnovabile,
                text=f.tipo))
    self._view.update_page()

1 usage (1 dynamic)  alexandrahlotota *
def populated_ddAttributes(self):
    self._view._ddAttributes.options.append(
        ft.dropdown.Option(
            data="Analisi Generale",
            on_click=self.readDDAnalisi,
            text="Analisi Generale"))
    opzioni = self._model.getAllOptions()
    for o in opzioni:
        self._view._ddAttributes.options.append(
            ft.dropdown.Option(
                data=o.attribute,
                text=o.attributo,
                on_click=self.readDDAnalisi
            ))
    self._view.update_page()

```

Figura 4.2: Funzioni per la popolazione dei dropdowns

```

1 usage  new *
def readDDFonteRinnovabile(self, e):
    if e.control.data is None:
        self._fonteRinnovabile = None
    else:
        self._fonteRinnovabile = e.control.data

2 usages  alexandrahlotota
def readDDAnalisi(self, e):
    if e.control.data is None:
        self._analisi = None
    else:
        self._analisi = e.control.data

```

Figura 4.3: Funzioni per la lettura dei dati selezionati nei dropdowns

Analisi preliminare

Il metodo `get_Analisi(self, e)` si occupa di recuperare e aggiungere i risultati delle analisi relative a fonti energetiche rinnovabili, in base alle preferenze espresse dall'utente. Il metodo (Figura 4.4):

- Controlla se è stata selezionata una fonte energetica rinnovabile dall'utente per l'analisi. Se non è stata selezionata, mostra un messaggio di errore e interrompe l'esecuzione.
- Determina se l'utente ha scelto di eseguire un'analisi generale o specifica in base all'opzione fornita. A seconda di questa scelta, utilizzano i metodi della classe *Model* appropriati per calcolare i valori minimi, massimi e medi.
- Aggiunge i risultati calcolati all'elemento `self._view._txt_resultAnalisi` per mostrare all'utente i dettagli dell'analisi effettuata.
- Abilita i controlli dell'interfaccia necessari per consentire all'utente di inserire i dati per effettuare la ricerca dei tre sistemi energetici ottimali.
- Infine, aggiorna l'interfaccia utente.

Soluzione per il cliente

Il metodo `get_SoluzioneCliente(self, e)` è progettato per gestire la ricerca e presentazione delle migliori soluzioni basate su input forniti dall'utente tramite un'interfaccia grafica.

Il metodo (Figura 4.5):

- Verifica se sono stati inseriti i valori nell'input `_txtImporto` e nell'input `_txtCapacita`. Nel caso in cui i dati non fossero stati inseriti, viene aggiunto un messaggio di errore e l'esecuzione del metodo termina; altrimenti, tenta di convertire il valore dell'investimento iniziale, e successivamente della capacità, in un numero float. Se la conversione fallisce, viene aggiunto un avviso indicando l'errore e l'esecuzione del metodo termina. Inoltre, controlla se è stata selezionata una fonte energetica rinnovabile dall'utente per l'analisi.
- Utilizza il metodo della classe *Model_model.buildGrafo* per costruire un grafo basato sulla fonte rinnovabile selezionata, l'investimento iniziale e la capacità inserita e, successivamente, chiama, sempre dalla classe *Model_model.getBestNodes()* per ottenere i migliori nodi (sistemi energetici) basati sul grafo costruito.
- Se vengono trovate delle soluzioni, i dettagli delle soluzioni vengono aggiunti all'elemento `self._view._txt_resultSoluzione`. Se non vengono trovate soluzioni, viene aggiunto un messaggio indicando che nessuna soluzione è stata trovata.
- Infine, aggiorna l'interfaccia utente.

```

1 usage (1 dynamic)  ± alexandraholota *
def get_Analisi(self, e):
    self._view._txt_resultAnalisi.controls.clear()
    if self._fonteRinnovabile is not None:
        if self._analisi == "Analisi Generale" or self._analisi==None:
            minimo = self._model.analisiGeneraleMin(self._fonteRinnovabile)
            massimo = self._model.analisiGeneraleMax(self._fonteRinnovabile)
            media = self._model.analisiGeneraleAVG(self._fonteRinnovabile)
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("Analisi Generale: "))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("I valori minimi sono: "))
            self.stampaValoriAnalisi(minimo)
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("-----"))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("I valori massimi sono: "))
            self.stampaValoriAnalisi(massimo)
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("-----"))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text("I valori medi sono: "))
            self.stampaValoriAnalisi(media)
        else:
            minimo = self._model.analisiSpecificaMin(self._fonteRinnovabile, self._analisi)
            massimo = self._model.analisiSpecificaMax(self._fonteRinnovabile, self._analisi)
            media = self._model.analisiSpecificaAVG(self._fonteRinnovabile, self._analisi)
            self._view._txt_resultAnalisi.controls.append(
                ft.Text(f"Analisi specifica per la categoria {self._view._ddAttributes.value}: "))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text(f"Il valore minimo trovato è: {minimo}"))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text(f"Il valore massimo trovato è: {massimo}"))
            self._view._txt_resultAnalisi.controls.append(
                ft.Text(f"Il valore medio calcolato è: {media} "))

            self._view._txtImporto.disabled = False
            self._view._txtCapacita.disabled = False
            self._view._btnSoluzione.disabled = False
            self._view.update_page()
    else:
        self._view._txt_resultAnalisi.controls.append(
            ft.Text(f"Non è stata scelto un sistema energetico rinnovabile da analizzare"))
        self._view.update_page()
    return

```

Figura 4.4: Metodo `get_Analisi(self, e)`

```

1 usage (1 dynamic)  ± alexandraholota *
def get_SoluzioneCliente(self, e):
    importo = -1
    capacita = -1
    self._view._txt_resultSoluzione.controls.clear()
    if self._view._txtImporto.value != "" and self._view._txtCapacita.value != "":
        try:
            importo = float(self._view._txtImporto.value)
        except ValueError:
            self._view._txt_resultSoluzione.controls.append(
                ft.Text("Investimento iniziale non valido, inserire un importo"))
            self._view.update_page()
            return
        try:
            capacita = float(self._view._txtCapacita.value)
        except ValueError:
            self._view._txt_resultSoluzione.controls.append(
                ft.Text("Capacità non valida"))
            self._view.update_page()
            return
    elif self._view._txtImporto.value == "" or self._view._txtCapacita.value == "":
        self._view._txt_resultSoluzione.controls.append(
            ft.Text("Inserire i dati richiesti"))
        self._view.update_page()
        return

    if self._fonteRinnovabile == None:
        self._view._txt_resultSoluzione.controls.append(
            ft.Text("Inserire la fonte rinnovabile"))
        self._view.update_page()
        return

    self.grafo = self._model.buildGrafo(self._fonteRinnovabile, importo, capacita)

    self._model.getBestNodes()

    if len(self._model.bestNodes) != 0:
        self._view._txt_resultSoluzione.controls.append(
            ft.Text(f"Abbiamo trovato {len(self._model.bestNodes)} soluzioni ottime: "))
        for n in self._model.bestNodes:
            self._view._txt_resultSoluzione.controls.append(ft.Text(
                f"{self._model._idMap[n].id} - "
                f"Investimento iniziale (USD): {self._model._idMap[n].Initial_Investment_USD} \n "
                f"Capacità installata (MW) : {self._model._idMap[n].Installed_Capacity_MW} \n "
                f"Indice di riduzione dell'inquinamento atmosferico: "
                f"{self._model._idMap[n].Air_Pollution_Reduction_Index} "))
    else:
        self._view._txt_resultSoluzione.controls.append(ft.Text(f"Non abbiamo trovato nessuna soluzione "))

    self._view.update_page()

```

Figura 4.5: Metodo `get_SoluzioneCliente(self, e)`

4.1.2 View

La classe *View* gestisce l'interfaccia grafica dell'applicazione.

Il metodo principale è `load_interface()` ed è responsabile di costruire e popolare l'interfaccia grafica con diversi elementi come titoli, dropdown (`_ddFonteRinnovabile` e `_ddAttributes`), pulsanti (`_btnAnalisi` e `_btnSoluzione`), e aree di testo (`_txt_resultAnalisi` e `_txt_resultSoluzione`).

4.2 Model

Il package *model* si assume il ruolo di gestore della logica applicativa, garantendo il corretto funzionamento dell'applicazione (Figura 4.6). Comprende quattro classi distinte:

1. **FonteRinnovabile**: include un identificatore univoco per la fonte, *id*, e una il nome, *tipo* della fonte stessa.
2. **Option**: è composta dal nome dell'attributo presente nella tabella "*RenewableEnergySystems*", utilizzato nell'analisi preliminare, e dalla sua versione italiana (attributo).
3. **SistemaEnergeticoRinnovabile**: gli attributi sono i campi della tabella "*RenewableEnergySystems*".
4. **Model**: si occupa della gestione di tutti i processi applicativi, verrà approfondita in seguito.

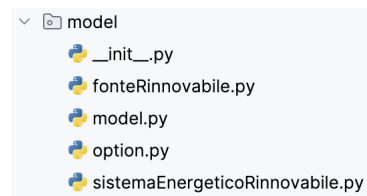


Figura 4.6: Struttura package *Model*

```
@dataclass()
class SistemaEnergeticoRinnovabile:
    id: int
    Type_of_Renewable_Energy: int
    Installed_Capacity_MW: float
    Energy_Production_MWh: float
    Energy_Consumption_MWh: float
    Energy_Storage_Capacity_MWh: float
    Storage_Efficiency_Percentage: float
    Grid_Integration_Level: int
    Initial_Investment_USD: float
    Funding_Sources: int
    Financial_Incentives_USD: float
    GHG_Emission_Reduction_tCO2e: float
    Air_Pollution_Reduction_Index: float
    Jobs_Created: int

@dataclass
class FonteRinnovabile:
    id: int
    tipo: str

@dataclass
class Option:
    attribute: str
    attributo: str
```

Figura 4.7: Strutture delle classi *FonteRinnovabile*, *Option* e *SistemaEnergeticoRinnovabile*

4.2.1 Classe Model

La classe **Model** rappresenta la logica delle operazioni. Acquisisce e prepara i dati dell'analisi da mandare alla classe *Controller*, si occupa della creazione di un grafo utilizzando la libreria NetworkX e implementa la funzione ricorsiva per la ricerca delle soluzioni ottimali da proporre al cliente.

4.2.2 Metodi della classe Model

Analisi preliminare

I metodi per la gestione delle operazioni di analisi per diverse fonti di energia rinnovabile sono riportate in Figura 4.8. I dati vengono acquisiti dal DAO attraverso i metodi sviluppati all'interno della classe DAO.

Soluzione per il cliente

I metodi per ricercare i tre sistemi energetici ottimali per il cliente sono riportate in Figura 4.9 ed in Figura 4.10 .

- **buildGrafo(self, fonte, prezzo, capacita)** (Figura 4.9): costruisce un grafo utilizzando la libreria NetworkX, basato su una specifica fonte di energia rinnovabile, un prezzo e una capacità forniti dall'utente. I nodi vengono recuperati dal DAO. In seguito, viene popolato il dizionario `_idMap`, che mappa l'*id* di ciascun nodo all'oggetto nodo corrispondente, e il dizionario `_idMapNodi`, che mappa l'*id* di ciascun nodo al rapporto tra l'investimento iniziale e la capacità installata; questo rapporto servirà per trovare le soluzioni ottimali. Utilizzando `combinations(self._idMapNodi.keys(), 2)`, si generano tutte le possibili coppie di nodi per creare gli archi.
- **getBestNodes(self), ricorsione(self, parziale, visitati), viciniAmmessi(self, ultimo, visitati)**: questi metodi sono utilizzati per trovare i tre sistemi energetici da proporre al cliente.
 - **getBestNodes(self)**: inizializza due variabili: *bestNodes* come lista vuota e *bestScore* con il valore positivo infinito. Quindi, per ogni nodo nel dizionario `_idMapNodi`, crea una lista parziale contenente solo quel nodo e un insieme di nodi visitati che include solo il nodo corrente. Chiama quindi il metodo `ricorsione` passando queste due strutture dati. Se il numero di nodi nel grafo è minore o uguale a tre, `getBestNodes(self)` assegna tutti i nodi come le migliori soluzioni e calcola il loro punteggio totale.
 - **ricorsione(self, parziale, visitati)**: esegue una ricerca ricorsiva per trovare combinazioni di nodi. Parte dall'ultimo nodo nella lista parziale e trova i suoi vicini ammessi usando il metodo `viciniAmmessi(self, ultimo, visitati)`. Se la lista parziale contiene già tre nodi, calcola il punteggio totale dei nodi nella lista. Se questo punteggio è inferiore al *bestScore* attuale, lo aggiorna e copia la lista parziale in *bestNodes*. Se invece la lista parziale ha meno di tre nodi, la funzione ricorre su ciascun vicino ammesso, aggiornando dinamicamente le strutture dati parziale e visitati.
 - **viciniAmmessi(self, ultimo, visitati)**: trova i nodi vicini all'ultimo nodo dato, escludendo quelli già visitati e quelli con un valore nel dizionario `_idMapNodi` superiore a quello dell'ultimo nodo. Ritorna una lista di nodi che soddisfano queste condizioni.

```

1 usage (1 dynamic)  ± alexandrahlotia
def analisiGeneraleMax(self, fonte):
    self._analisiMax = DAO.getMaxValueForType(fonte)
    massimo = {"Capacità Installata (MW)": self._analisiMax["Installed_Capacity_MW"],
               "Produzione di Energia (MWh)": self._analisiMax["Energy_Production_MWh"],
               "Consumo di Energia (MWh)": self._analisiMax["Energy_Consumption_MWh"],
               "Capacità di Stoccaggio di Energia (MWh)": self._analisiMax["Energy_Storage_Capacity_MWh"],
               "Investimento Iniziale (USD)": self._analisiMax["Initial_Investment_USD"],
               "Indice di Riduzione dell'Inquinamento Atmosferico": self._analisiMax["Air_Pollution_Reduction_Index"]}
    return massimo

1 usage (1 dynamic)  ± alexandrahlotia
def analisiGeneraleMin(self, fonte):
    self._analisiMin = DAO.getMinValueForType(fonte)
    minimo = {"Capacità Installata (MW)": self._analisiMin["Installed_Capacity_MW"],
              "Produzione di Energia (MWh)": self._analisiMin["Energy_Production_MWh"],
              "Consumo di Energia (MWh)": self._analisiMin["Energy_Consumption_MWh"],
              "Capacità di Stoccaggio di Energia (MWh)": self._analisiMin["Energy_Storage_Capacity_MWh"],
              "Investimento Iniziale (USD)": self._analisiMin["Initial_Investment_USD"],
              "Indice di Riduzione dell'Inquinamento Atmosferico": self._analisiMin["Air_Pollution_Reduction_Index"]}
    return minimo

1 usage (1 dynamic)  ± alexandrahlotia
def analisiGeneraleAVG(self, fonte):
    self._analisiAVG = DAO.getAVGValueForType(fonte)
    media = {"Capacità Installata (MW)": self._analisiAVG["Installed_Capacity_MW"],
             "Produzione di Energia (MWh)": self._analisiAVG["Energy_Production_MWh"],
             "Consumo di Energia (MWh)": self._analisiAVG["Energy_Consumption_MWh"],
             "Capacità di Stoccaggio di Energia (MWh)": self._analisiAVG["Energy_Storage_Capacity_MWh"],
             "Investimento Iniziale (USD)": self._analisiAVG["Initial_Investment_USD"],
             "Indice di Riduzione dell'Inquinamento Atmosferico": self._analisiAVG["Air_Pollution_Reduction_Index"]}
    return media

1 usage (1 dynamic)  ± alexandrahlotia
def analisiSpecificMax(self, fonte, categoria):
    self._analisiMax = DAO.getMaxValueForType(fonte)
    return self._analisiMax[categoria]

1 usage (1 dynamic)  ± alexandrahlotia
def analisiSpecificMin(self, fonte, categoria):
    self._analisiMin = DAO.getMinValueForType(fonte)
    return self._analisiMin[categoria]

1 usage (1 dynamic)  ± alexandrahlotia
def analisiSpecificAVG(self, fonte, categoria):
    self._analisiAVG = DAO.getAVGValueForType(fonte)
    return self._analisiAVG[categoria]

```

Figura 4.8: Metodi per l'analisi preliminare

```

2 usages (1 dynamic)  ± alexandrahlotia *
def buildGrafo(self, fonte, prezzo, capacita):
    self._grafo.clear()
    #aggiungi nodi
    self.nodes = DAO.getNodes(fonte, prezzo, capacita)
    self._idMap = {x.id: x for x in self.nodes}
    self._idMapNodi = {x.id: (x.Initial_Investment_USD/x.Installed_Capacity_MW) for x in self.nodes}
    self._grafo.add_nodes_from(self._idMapNodi.keys())
    #aggiungi archi
    self.edges = combinations(self._idMapNodi.keys(), r=2)
    self._grafo.add_edges_from(self.edges)

```

Figura 4.9: Metodo per costruire il grafo

```

2 usages (1 dynamic)  ⚡ alexandraholota *
def getBestNodes(self):
    self.bestNodes = []
    self.bestScore = float('inf')
    for node in self._idMapNodi.keys():
        parziale = [node]
        visited = set(parziale)
        self.ricorsione(parziale, visited)
    if len(self._idMapNodi.keys()) <= 3:
        self.bestNodes = list(self._idMapNodi.keys())
        self.bestScore = sum(self._idMapNodi[nodo] for nodo in self._idMapNodi.keys())

2 usages  ⚡ alexandraholota *
def ricorsione(self, parziale, visitati):
    ultimo = parziale[-1]
    vicini = self.viciniAmmessi(ultimo, visitati)
    # Se abbiamo già tre nodi, confrontiamo il loro punteggio totale con il bestScore
    if len(parziale) == 3:
        peso = sum(self._idMapNodi[nodo] for nodo in parziale)
        if peso < self.bestScore:
            self.bestScore = peso
            self.bestNodes = parziale[:]
        return
    # Ricorsione sui vicini
    for v in vicini:
        visitati.add(v)
        parziale.append(v)
        self.ricorsione(parziale, visitati)
        parziale.pop()
        visitati.remove(v)

1 usage  ⚡ alexandraholota *
def viciniAmmessi(self, ultimo, visitati):
    if ultimo not in self._grafo:
        return []
    vicini = self._grafo.neighbors(ultimo)
    risultati = []
    for v in vicini:
        if v not in visitati and self._idMapNodi[v] <= self._idMapNodi[ultimo]:
            risultati.append(v)
    return risultati

```

Figura 4.10: Metodi per cercare le tre soluzioni ottimali

4.3 Database

Il package *database* serve a stabilire, gestire la connessione con il database e a raccogliere i dati (Figura 4.11). I risultati sono ottenuti tramite le query SQL e, successivamente, vengono trasmessi alla classe *Model*. È stato sviluppato seguendo il pattern DAO, garantendo un accesso limitato e controllato per la lettura e la modifica dei dati.

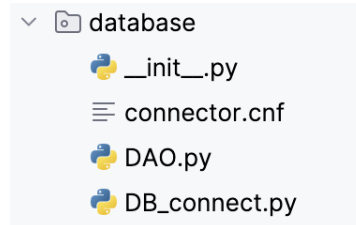


Figura 4.11: Struttura package *database*

- **DB_connect:** questa classe stabilisce la connessione con il database.
- **DAO:** sono sviluppati diversi metodi atti ad estrarre i dati necessari dal database mediante l'utilizzo di query SQL. In Figura 4.12 ed in Figura 4.13 vengono riportati alcuni dei metodi più significativi.

4.3.1 Metodi classe DAO

La classe DAO è costituita da sei metodi. I primi due, ovvero **getAllRenewableSource()** e **getAllOptions()**, si occupano dell'estrazione dei dati relativi alle fonti rinnovabile, dalla tabella "*RenewableSource*", e alle categorie che l'utente potrà selezionare per l'analisi preliminare dalla tabella "*Options*".

Più rilevanti sono i metodi:

- **getMinValueForType(fonte)**
- **getMaxValueForType(fonte)**
- **getAVGValueForType(fonte)**

Questi metodi sono usati per acquisire i valori minimi, massimi e medi degli elementi della tabella aventi come attributo *Type_of_Renewable_Energy* la fonte passata come parametro. In Figura 4.12 ed in Figura 4.13 sono evidenziate le query SQL.

Infine, per estrarre i nodi, che costituiscono il grafo, è stata implementato il metodo **getNodes(fonte, prezzo, capacita)**. Si selezionano i sistemi energetici aventi *Type_of_Renewable_Energy* uguali alla fonte passata come parametro, l' *Initial_Investment_USD* minore o uguale al prezzo e *Installed_Capacity_MW* maggiore o uguale alla capacità. In Figura 4.13 si evidenzia la query SQL utilizzata.

```

2 usages  ± alexandraholota *
@staticmethod
def getMinValueForType(fonte):
    cnx = DBConnect.get_connection()
    if cnx is not None:
        cursor = cnx.cursor(dictionary=True)
        query = '''SELECT MIN(Installed_Capacity_MW) as Installed_Capacity_MW,
                        MIN(Energy_Production_MWh) as Energy_Production_MWh,
                        MIN(Energy_Consumption_MWh) as Energy_Consumption_MWh,
                        MIN(Energy_Storage_Capacity_MWh) as Energy_Storage_Capacity_MWh,
                        MIN(Storage_Efficiency_Percentage) as Storage_Efficiency_Percentage,
                        MIN(Initial_Investment_USD) as Initial_Investment_USD,
                        MIN(GHG_Emission_Reduction_tCO2e) as GHG_Emission_Reduction_tCO2e,
                        MIN(Air_Pollution_Reduction_Index) as Air_Pollution_Reduction_Index,
                        MIN(Jobs_Created) as Jobs_Created
                    FROM RenewableEnergySystems res
                    WHERE Type_of_Renewable_Energy = %s'''
        cursor.execute(query, (fonte,))
        result = cursor.fetchone()
        cursor.close()
        cnx.close()
        return result
    else:
        print("Errore nella connessione")
        return None

2 usages  ± alexandraholota *
@staticmethod
def getMaxValueForType(fonte):
    cnx = DBConnect.get_connection()
    if cnx is not None:
        cursor = cnx.cursor(dictionary=True)
        query = '''SELECT MAX(Installed_Capacity_MW) as Installed_Capacity_MW,
                        MAX(Energy_Production_MWh) as Energy_Production_MWh,
                        MAX(Energy_Consumption_MWh) as Energy_Consumption_MWh,
                        MAX(Energy_Storage_Capacity_MWh) as Energy_Storage_Capacity_MWh,
                        MAX(Storage_Efficiency_Percentage) as Storage_Efficiency_Percentage,
                        MAX(Initial_Investment_USD) as Initial_Investment_USD,
                        MAX(GHG_Emission_Reduction_tCO2e) as GHG_Emission_Reduction_tCO2e,
                        MAX(Air_Pollution_Reduction_Index) as Air_Pollution_Reduction_Index,
                        MAX(Jobs_Created) as Jobs_Created
                    FROM RenewableEnergySystems res
                    WHERE Type_of_Renewable_Energy = %s'''
        cursor.execute(query, (fonte,))
        result = cursor.fetchone()
        cursor.close()
        cnx.close()
        return result
    else:
        print("Errore nella connessione")
        return None

```

Figura 4.12: Metodi della classe *DAO*

```

2 usages  ± alexandraholota *
@staticmethod
def getAVGValueForType(fonte):
    cnx = DBConnect.get_connection()
    if cnx is not None:
        cursor = cnx.cursor(dictionary=True)
        query = ''' SELECT AVG(Installed_Capacity_MW) as Installed_Capacity_MW,
                        AVG(Energy_Production_MWh) as Energy_Production_MWh,
                        AVG(Energy_Consumption_MWh) as Energy_Consumption_MWh,
                        AVG(Energy_Storage_Capacity_MWh) as Energy_Storage_Capacity_MWh,
                        AVG(Storage_Efficiency_Percentage) as Storage_Efficiency_Percentage,
                        AVG(Initial_Investment_USD) as Initial_Investment_USD,
                        AVG(GHG_Emission_Reduction_tCO2e) as GHG_Emission_Reduction_tCO2e,
                        AVG(Air_Pollution_Reduction_Index) as Air_Pollution_Reduction_Index,
                        AVG(Jobs_Created) as Jobs_Created
                    FROM RenewableEnergySystems res
                    WHERE Type_of_Renewable_Energy = %s'''
        cursor.execute(query, (fonte,))
        result = cursor.fetchone()
        cursor.close()
        cnx.close()
        return result
    else:
        print("Errore nella connessione")
        return None

1 usage  ± alexandraholota *
@staticmethod
def getNodes(fonte, prezzo, capacita):
    cnx = DBConnect.get_connection()
    if cnx is not None:
        result = []
        cursor = cnx.cursor(dictionary=True)
        query = ''' SELECT *
                    FROM RenewableEnergySystems res
                    WHERE res.Type_of_Renewable_Energy = %s
                        and res.Initial_Investment_USD <= %s
                        and res.Installed_Capacity_MW >= %s'''
        cursor.execute(query, (fonte, prezzo, capacita))
        for row in cursor:
            result.append(SistemaEnergeticoRinnovabile(**row))
        cursor.close()
        cnx.close()
        return result
    else:
        print("Errore nella connessione")
        return None

```

Figura 4.13: Metodi della classe *DAO*

Capitolo 5

Alcune videate dell'applicazione e link al video

Una volta avviata, l'applicazione si presenterà come illustrato nella videata in Figura 5.1.

Si potrà selezionare il sistema energetico rinnovabile, l'opzione dal menù a tendina "*Opzioni*" e effettuare l'analisi, attraverso il click sul bottone "*Analisi*".

Successivamente, verranno abilitate anche le caselle di testo per l'inserimento delle preferenze dell'utente, riguardanti l'investimento iniziale disposto ad effettuare e la capacità minima, ed il bottone "*Trova*" per effettuare la ricerca. Come mostrato in Figura 5.2

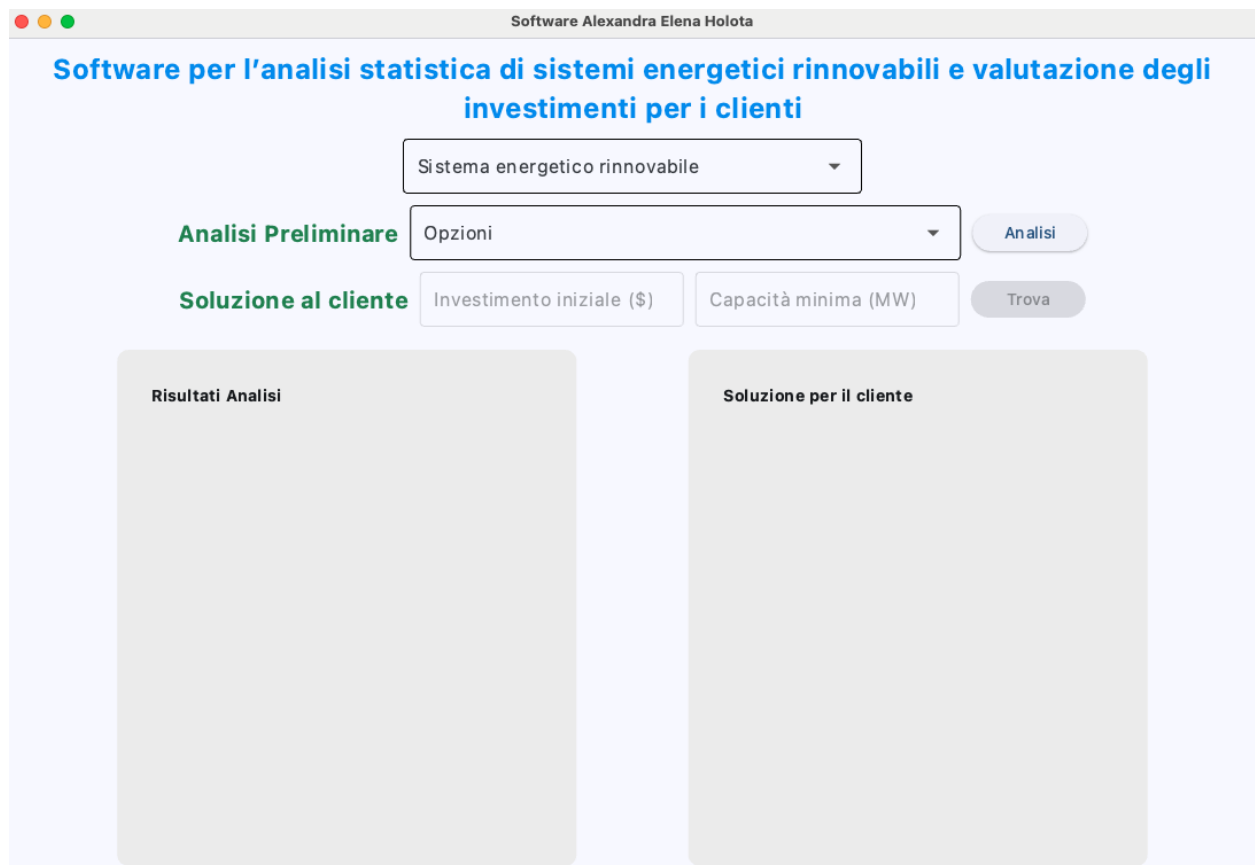


Figura 5.1: Videata iniziale

The screenshot shows a web application window titled "Software Alexandra Elena Holota". The main heading is "Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti". Below this, there are two dropdown menus: "Sistema energetico rinnovabile" with "Solare" selected, and "Opzioni" with "Indice di Riduzione dell'Inquinamento Atmosferico" selected. To the left of these dropdowns are the labels "Analisi Preliminare" and "Soluzione al cliente". To the right of the "Opzioni" dropdown is a button labeled "Analisi". Below the dropdowns are two input fields: "Investimento iniziale (\$)" and "Capacità minima (MW)". To the right of these fields is a button labeled "Trova". Below the input fields, there are two large grey rectangular boxes. The left box contains the text: "Analisi specifica per la categoria Indice di Riduzione dell'Inquinamento Atmosferico:", "Il valore minimo trovato è: 1.08425", "Il valore massimo trovato è: 99.9805", and "Il valore medio calcolato è: 51.61286589933354". The right box is titled "Soluzione per il cliente" and is currently empty.

Figura 5.2: Videata successiva all'analisi preliminare

5.1 Controlli sui dati inseriti

Sono stati implementati controlli sull'inserimento dei dati da parte dell'utente. In Figura 5.3, Figura 5.4, Figura 5.5 e Figura 5.6 vengono riportati degli esempi.

5.2 Link al video

Di seguito viene riportato il link relativo al video esplicativo del funzionamento dell'applicazione, pubblicato sulla piattaforma YouTube: https://youtu.be/4xp-AM_Ai8Y

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile

Opzioni

Analisi Preliminare

Soluzione al cliente

Non è stata scelto un sistema energetico rinnovabile da analizzare

Soluzione per il cliente

Figura 5.3: Controllo sull'inserimento del sistema energetico

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile

Opzioni

Analisi Preliminare

Soluzione al cliente

Capacità di Stoccaggio di Energia (MWh):
9995.97

Investimento Iniziale (USD): 499584000.0

Indice di Riduzione dell'Inquinamento
Atmosferico: 99.9738

I valori medi sono:

Capacità Installata (MW): 494.4036936945097

Produzione di Energia (MWh):
250384.79751406747

Consumo di Energia (MWh):
228664.10890684207

Capacità di Stoccaggio di Energia (MWh):
5063.074327466923

Investimento Iniziale (USD):
252402156.73211852

Inserire i dati richiesti

Figura 5.4: Controllo sull'inserimento delle preferenze dell'utente

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: **Eolico**

Opzioni: **Analisi Generale** **Analisi**

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$): **ciao** Capacità minima (MW): **242** **Trova**

Investimento Iniziale (USD): 499584000.0
Indice di Riduzione dell'Inquinamento Atmosferico: 99.9738

I valori medi sono:
Capacità Installata (MW): 494.4036936945097
Produzione di Energia (MWh): 250384.79751406747
Consumo di Energia (MWh): 228664.10890684207
Capacità di Stoccaggio di Energia (MWh): 5063.074327466923
Investimento Iniziale (USD): 252402156.73211852
Indice di Riduzione dell'Inquinamento Atmosferico: 49.85110642945734

Investimento iniziale non valido, inserire un importo

Figura 5.5: Controllo sulla validità dell'inserimento dell'investimento iniziale

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: **Eolico**

Opzioni: **Analisi Generale** **Analisi**

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$): **2800000** Capacità minima (MW): **ciao** **Trova**

Investimento Iniziale (USD): 499584000.0
Indice di Riduzione dell'Inquinamento Atmosferico: 99.9738

I valori medi sono:
Capacità Installata (MW): 494.4036936945097
Produzione di Energia (MWh): 250384.79751406747
Consumo di Energia (MWh): 228664.10890684207
Capacità di Stoccaggio di Energia (MWh): 5063.074327466923
Investimento Iniziale (USD): 252402156.73211852
Indice di Riduzione dell'Inquinamento Atmosferico: 49.85110642945734

Capacità non valida

Figura 5.6: Controllo sulla validità dell'inserimento della capacita minima

Capitolo 6

Tabelle con risultati sperimentali ottenuti

Si riportano alcuni risultati ottenuti.

6.1 Analisi generale

Nell'esempio in Figura 6.1 e Figura 6.2, la fonte selezionata corrisponde a quella *"Solare"* e la tipologia di analisi che si desidera è di tipo generale. È da sottolineare che per questa tipologia di analisi si può anche non scegliere nessuna opzione dal menù a tendina. In seguito al click sul bottone *"Analisi"*, nella sezione *"Risultati Analisi"*, verranno mostrati i valori di minimo, massimo e medi per le categorie:

- capacità installata (MW);
- produzione di energia (MWh);
- consumo di energia (MWh);
- capacità di stoccaggio di energia (MWh);
- investimento iniziali (USD);
- indice di riduzione dell'inquinamento atmosferico.

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: Solare

Opzioni: Analisi Generale

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$) Capacità minima (MW) Trova

Analisi Generale:

I valori minimi sono:

Capacità Installata (MW): 1.82673

Produzione di Energia (MWh): 1210.05

Consumo di Energia (MWh): 754.279

Capacità di Stoccaggio di Energia (MWh): 2.81548

Investimento Iniziale (USD): 1111600.0

Indice di Riduzione dell'Inquinamento Atmosferico: 1.08425

I valori massimi sono:

Capacità Installata (MW): 999.734

Produzione di Energia (MWh): 499991.0

Consumo di Energia (MWh): 449864.0

Soluzione per il cliente

Figura 6.1: Risultati analisi generale

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: Solare

Opzioni: Analisi Generale

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$) Capacità minima (MW) Trova

Soluzione per il cliente

Consumo di Energia (MWh): 449864.0

Capacità di Stoccaggio di Energia (MWh): 9998.04

Investimento Iniziale (USD): 499753000.0

Indice di Riduzione dell'Inquinamento Atmosferico: 99.9805

I valori medi sono:

Capacità Installata (MW): 491.7334814589383

Produzione di Energia (MWh): 252813.67605178573

Consumo di Energia (MWh): 228301.9907259486

Capacità di Stoccaggio di Energia (MWh): 5012.8668981410765

Investimento Iniziale (USD): 256192796.95729598

Figura 6.2: Risultati analisi generica

6.2 Analisi specifica

In generale, il cliente può selezionare una qualsiasi categoria tra quelle proposte, ovvero:

- Indice di Riduzione dell’Inquinamento Atmosferico;
- Consumo di Energia (MWh);
- Produzione di Energia (MWh);
- Capacità di Stoccaggio di Energia (MWh);
- Riduzione delle Emissioni di GHG (tCO₂);
- Investimento Iniziali (USD);
- Capacità Installata (MW);
- Posti di Lavoro Creati;
- Efficienza di stoccaggio (Percentuale=;

Nell’esempio in Figura 6.3 la fonte selezionata è *"Solare"* e la categoria della quale si desidera l’analisi è *"Indice di Riduzione dell’Inquinamento Atmosferico"*. I valori di minimo, massimo e medio per l’opzione selezionata verranno mostrati nella sezione *"Risultati Analisi"*, in seguito al click sul bottone *"Analisi"*.

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile
Solare

Opzioni
Indice di Riduzione dell'Inquinamento Atmosferico

Analisi

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$)

Capacità minima (MW)

Trova

Analisi specifica per la categoria Indice di Riduzione dell'Inquinamento Atmosferico:
Il valore minimo trovato è: 1.08425
Il valore massimo trovato è: 99.9805
Il valore medio calcolato è: 51.61286589933354

Soluzione per il cliente

Figura 6.3: Risultati analisi specifica

Nell'esempio in Figura 6.4 è stata selezionata la fonte *"Biomassa"* e la categoria *"Capacità Installata (MW)"*

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile
Biomassa

Opzioni
Capacità Installata (MW)

Analisi

Soluzione al cliente

Investimento iniziale (\$)

Capacità minima (MW)

Trova

Analisi specifica per la categoria Capacità Installata (MW):

- Il valore minimo trovato è: 1.26325
- Il valore massimo trovato è: 999.909
- Il valore medio calcolato è: 495.4310828614235

Soluzione per il cliente

Figura 6.4: Risultati analisi specifica

6.3 Soluzione per il cliente

In Figura 6.5 si seleziona il sistema energetico rinnovabile *"Solare"* e si inseriscono i valori 2800000 nella casella *"Investimento iniziale (\$)"* e 242 nella casella *"Capacità minima (MW)"*. Nella sezione *"Soluzione per il cliente"* verranno mostrati i tre sistemi energetici rinnovabili ottimali trovati, in seguito al click sul bottone *"Trova"*.

In Figura 6.6 ed in Figura 6.7 vengono mostrati rispettivamente tutte le soluzioni trovate, nel caso in cui esse siano inferiori a tre, o un messaggio di avviso per indicare che non sono stati trovati sistemi energetici rinnovabili con le caratteristiche indicate nelle due caselle di input

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: **Solare**

Opzioni: **Capacità Installata (MW)**

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$): **2800000**

Capacità minima (MW): **242**

Analisi

Trova

Analisi specifica per la categoria Capacità Installata (MW):

- Il valore minimo trovato è: 1.26325
- Il valore massimo trovato è: 999.909
- Il valore medio calcolato è: 495.4310828614235

Abbiamo trovato 3 soluzioni ottime:

- 10993 - Investimento iniziale (USD): 1586460.0
Capacità installata (MW) : 468.576
Indice di riduzione dell'inquinamento atmosferico: 96.5011
- 5284 - Investimento iniziale (USD): 1805490.0
Capacità installata (MW) : 607.613
Indice di riduzione dell'inquinamento atmosferico: 14.5767
- 13336 - Investimento iniziale (USD): 1128670.0
Capacità installata (MW) : 956.17
Indice di riduzione dell'inquinamento atmosferico: 98.4395

Figura 6.5: Risultati della ricerca di soluzione per il cliente, caso default con tre soluzioni

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile: **Geotermico**

Opzioni: **Capacità Installata (MW)**

Analisi Preliminare

Soluzione al cliente

Investimento iniziale (\$): **3800000**

Capacità minima (MW): **863**

Analisi

Trova

Analisi specifica per la categoria Capacità Installata (MW):

- Il valore minimo trovato è: 1.26325
- Il valore massimo trovato è: 999.909
- Il valore medio calcolato è: 495.4310828614235

Abbiamo trovato 2 soluzioni ottime:

- 5059 - Investimento iniziale (USD): 3347490.0
Capacità installata (MW) : 960.265
Indice di riduzione dell'inquinamento atmosferico: 76.4615
- 13432 - Investimento iniziale (USD): 3470200.0
Capacità installata (MW) : 863.568
Indice di riduzione dell'inquinamento atmosferico: 40.3596

Figura 6.6: Risultati della ricerca di soluzione per il cliente, caso particolare in cui si riportano tutte le soluzioni disponibili

Software Alexandra Elena Holota

Software per l'analisi statistica di sistemi energetici rinnovabili e valutazione degli investimenti per i clienti

Sistema energetico rinnovabile
Mareomotrice

Opzioni
Capacità Installata (MW)

Analisi

Soluzione al cliente

Investimento iniziale (\$)
280145

Capacità minima (MW)
863

Trova

Analisi specifica per la categoria Capacità Installata (MW):

- Il valore minimo trovato è: 1.26325
- Il valore massimo trovato è: 999.909
- Il valore medio calcolato è: 495.4310828614235

Non abbiamo trovato nessuna soluzione

Figura 6.7: Risultati della ricerca di soluzione per il cliente, caso particolare in cui non ci sono soluzioni da proporre

Capitolo 7

Valutazioni sui risultati ottenuti e conclusioni

Il software sviluppato offre due funzioni principali: l'analisi preliminare e la ricerca di soluzioni per il cliente, ciascuna con caratteristiche e tempi di elaborazione distinti.

7.1 Analisi preliminare

L'analisi preliminare è progettata per fornire risultati in tempo reale. Appena l'utente inserisce i dati richiesti, il software elabora le informazioni per offrire una panoramica indicativa delle caratteristiche di un determinato sistema di energia rinnovabile. Grazie all'analisi preliminare, gli utenti possono ottenere una chiara idea delle opzioni disponibili in modo immediato.

Il tempo di elaborazione è praticamente immediato, questo perchè i valori vengono acquisiti direttamente dal database e non vengono elaborati in maniera complessa dai metodi della classe *"Model"*.

7.2 Ricerca di soluzioni per il cliente

La ricerca di soluzioni per il cliente, invece, è una funzione più complessa e dettagliata, il cui tempo di elaborazione può variare in base ai dati inseriti dall'utente. Questa funzione permette di considerare molteplici scenari possibili per trovare la soluzione ottimale per le esigenze specifiche del cliente. Se i dati inseriti, come l'investimento iniziale e la capacità minima richiesta, sono congruenti e bilanciati, i tempi di elaborazione rimangono ragionevoli. Tuttavia, in presenza di dati inconsueti o squilibrati, il tempo di elaborazione può aumentare, poiché il software deve eseguire più ricorsioni per trovare le migliori soluzioni possibili.

7.3 Conclusione

Questo strumento rappresenta un supporto per le aziende che intendono approcciare la transizione verso una maggiore sostenibilità. L'analisi preliminare offre una rapida comprensione delle potenzialità delle diverse soluzioni di energia rinnovabile, mentre la ricerca di soluzioni per il cliente consente di esplorare scenari specifici. In questo modo, il software diventa un alleato per guidare le decisioni strategiche verso un futuro più sostenibile, facilitando il primo passo verso la trasformazione ecologica dell'azienda.

Licenza

Questa relazione tecnica è distribuita con Licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0)

Questo lavoro è distribuito con Licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

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.

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Alle seguenti condizioni:

- **Attribuzione** — Devi attribuire adeguatamente l'opera, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi farlo in qualsiasi modo ragionevole, ma non in alcun modo che suggerisca che il licenziante avalli te o il tuo utilizzo dell'opera.
- **Non commerciale** — Non puoi utilizzare il materiale per scopi commerciali.
- **Condividi allo stesso modo** — Se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi sotto la stessa licenza dell'originale.

Nota:

Non ci sono ulteriori restrizioni — Non puoi applicare termini legali o misure tecnologiche che limitino legalmente altri dal fare qualsiasi cosa la licenza permetta.

Per una copia della licenza completa, visita <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.