

POLITECNICO DI TORINO

Dipartimento di Ingegneria Gestionale e della
Produzione

Corso di Laurea in Ingegneria Gestionale
Classe L-8 – Ingegneria dell'Informazione

Tesi di Laurea

TasteTrip: applicazione per il turismo della regione Puglia



Relatore:

Prof. Fulvio Corno

Candidato:

Palmisano Vito

248020

A.A. 2019/2020

Indice



Indice	I
Indice figure.....	III
Indice tabelle.....	IV
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.....	2
2 Introduzione	3
3 Descrizione del problema affrontato	4
4 Descrizione del data-set	5
4.1 B&B	5
4.2 Attività turistiche	6
4.3 Luoghi d'interesse.....	6
4.4 Stabilimenti balneari	7
4.5 Tabella comuni.....	7
5 Strutture dati ed algoritmi	8
5.1 Strutture dati utilizzate	8
5.1.1 Package TasteTrip	8
5.1.2 Package db.....	8
5.1.3 Package model.....	8
5.2 Algoritmi utilizzati.....	9
5.2.1 Lettura dati dal database	9
5.2.2 Ricorsione.....	13
6 Diagramma delle classi principali	19
7 Interfaccia dell'applicazione	20
8 Risultati sperimentali.....	22
8.1 Ricerca sul territorio di un'intera Provincia.....	22

8.1.1	Esempio 1	22
8.1.2	Esempio 2	23
8.2	Ricerca con settaggio di un comune specifico	24
8.2.1	Esempio 3	24
8.3	Osservazioni dovute ai vari test effettuati.....	25
9	Valutazioni e conclusioni	26
	Riferimenti.....	27

Indice figure

Figura 1: metodo del Model per l'aggiunta dei comuni di una provincia specifica.....	10
Figura 2: metodo del TasteTripDAO per la selezione dei comuni dal DB	10
Figura 3: metodo del Model per l'aggiunta dei comuni che hanno una distanza dal comune selezionato, minore della distanza massima	11
Figura 4: metodo del Model per l'aggiunta dei B&B	11
Figura 5: metodo del TasteTripDAO per la selezione dei B&B dal DB	12
Figura 6: metodo del Model per l'aggiunta delle attività turistiche	12
Figura 7: metodo del TasteTripDAO per la selezione delle attività dal DB	13
Figura 8: metodo del Model per effettuare la ricorsione su tutti i B&B necessari.....	14
Figura 9: metodo cerca del Model, parte 3, condizione di terminazione, condizione per l'aggiunta di un percorso ed aggiornamento del percorso migliore	15
Figura 10: metodo cerca del Model, parte 2, caso in cui la durata del viaggio sia maggiore di un giorno.....	17
Figura 11: metodo cerca del Model, parte 2, caso in cui la durata del viaggio sia pari ad un giorno.....	18
Figura 12: schermata iniziale dell'applicazione	20
Figura 13: esempio inserimento dati, con conseguente risultato.....	21

Indice tabelle

Tabella 1: struttura dati tabella beb	5
Tabella 2: struttura dati tabella attivita_turistiche	6
Tabella 3: struttura dati tabella luoghi_interesse	6
Tabella 4: struttura dati tabella stabilimenti_balneari	7
Tabella 5: struttura dati tabella comuni_puglia	7
Tabella 6: diagramma delle principali classi del package model	19
Tabella 7: diagramma delle principali classi del package db	19

1 Proposta di progetto

1.1 Studente proponente

s248020 Vito Palmisano

1.2 Titolo della proposta

TasteTrip: applicazione per il turismo della regione Puglia

1.3 Descrizione del problema proposto

Organizzare una vacanza senza conoscere a pieno il territorio da visitare, rende sicuramente più difficoltoso scegliere, tra le varie esperienze, quelle più in linea con i propri gusti.

Obiettivo di TasteTrip è garantire un'esperienza di viaggio sul territorio Pugliese, che soddisfi a pieno le esigenze dell'utente e che rispecchi la tipologia di vacanza desiderata. A tal proposito, si vogliono proporre all'utente degli itinerari sviluppati a partire dalle sue preferenze: le zone che principalmente vorrebbe visitare, il budget che sarebbe disposto a spendere, la distanza massima che vorrebbe percorrere, il periodo dell'anno ed il numero di giorni a disposizione.

1.4 Descrizione della rilevanza gestionale del problema

Dal punto di vista gestionale l'applicazione avrebbe rilevanza per il turismo Pugliese, in quanto consentirebbe all'utente di cercare, con una singola applicazione e attraverso pochi e semplici passaggi, sia le attività da svolgere che i luoghi in cui pernottare.

1.5 Descrizione dei data-set per la valutazione

Si utilizzeranno più data-set, i quali saranno principalmente di 2 tipologie. Una prima tipologia di data-set conterrà informazioni riguardanti B&B e strutture simili presenti sull'intero territorio Pugliese. Una seconda tipologia di data-set conterrà le informazioni riguardanti:

- attività turistiche (attività sportive, attività di formazione, ecc.); [1]
- luoghi di interesse (castelli, riserve, chiese, grotte, masserie, trulli, ecc.); [2]
- stabilimenti balneari. [3]

Per quanto riguarda la prima tipologia è stato trovato un data-set delle strutture alberghiere e dei B&B che si affidano al portale Airbnb, aggiornato ad Aprile 2020. Questo data-set è fornito gratuitamente [4] e non contiene dati sensibili riguardanti gli host.

La seconda tipologia di data-set comprende 3 data-set, uno per ognuna delle tre descrizioni date sopra. Anche questi data-set sono stati trovati su siti che ne forniscono l'utilizzo gratuito ed in tutti i casi l'autore viene identificato come 'Regione Puglia'.

È importante specificare che tutti i data-set contengono le informazioni riguardanti il comune, la latitudine e la longitudine, in cui è situato l'elemento specifico del data-set. Vi è inoltre un ulteriore data-set contenente esclusivamente i dati sui singoli comuni, così da poter collegare facilmente fra di loro i data-set descritti precedentemente.

1.6 Descrizione preliminare degli algoritmi coinvolti

- Primo passo è raccogliere dai vari data-set i dati necessari su comuni, B&B ed attività, scelti in base alle selezioni dell'utente.
- Secondo passo è quello di eseguire una ricorsione per trovare le migliori combinazioni di attività e B&B rispettanti i dati di input.

Riguardo ai vincoli specificati dall'utente, egli imposta un budget massimo che è disposto a spendere, la distanza massima che è disposto a percorrere per raggiungere una determinata meta, il giorno della partenza ed il giorno del ritorno. Il budget massimo è un vincolo che riguarda la somma dei costi dei pernottamenti e di tutte le attività svolte, mentre il numero di giorni pone un vincolo sulla quantità di attività che è possibile svolgere durante la vacanza.

Nella ricorsione si massimizza la spesa della vacanza, restando al di sotto del budget inserito dall'utente, questo perché si vuole creare l'itinerario migliore per il turista e per far ciò si ipotizza che ad una maggiore spesa corrisponda una maggiore qualità della vacanza.

1.7 Descrizione preliminare delle funzionalità previste per l'applicazione

L'Utente ha la possibilità di selezionare, da un apposito menu a tendina, la provincia in cui vorrebbe svolgere la vacanza. Se lo desidera, può scendere più nel dettaglio, selezionando il comune specifico in cui vorrebbe soggiornare.

Seleziona successivamente le tipologie di attività che gradirebbe svolgere, che siano esse di tipo culturale, ricreativo, sportivo, ecc.

Una volta impostati l'andata, il ritorno, il budget che è disposto a spendere, la distanza massima che è disposto a percorrere per raggiungere un qualsiasi punto d'interesse ed il numero di viaggiatori, l'utente riceve in output il miglior percorso possibile, cioè quello che ha il costo massimo minore o uguale al budget inserito,

2 Introduzione

“È stato durante queste vacanze da emigrante alla rovescia che è avvenuta in me la spaccatura tra due diversi modi di vivere. Così oggi mi ritrovo con due anime: quella nordica (ordinata, efficiente, futuribile, perfezionista, esigente verso sé e verso gli altri) e quella meridionale (disordinata, brada, sensuale, onirica, mistica)”. [5]

Così Lucio Dalla descrive, in un'intervista rilasciata a L'Europeo, il suo rapporto con la Puglia.

Non saprei trovare parole migliori per descrivere il rapporto con le mie due Terre.

L'unica differenza, nel mio caso, è quella di non essere un “emigrante alla rovescia”, ma un “emigrante per necessità”, sempre voglioso e bisognoso di nuovi stimoli ed a caccia della possibilità di poter avere un futuro stabile e soddisfacente.

Vorrei tanto non esistesse questa divisione così netta fra i due mondi, Nord e Sud, ma al momento la realtà non è così semplice.

Io sogno un futuro in cui il divario fra Nord e Sud venga azzerato, ma per realizzare una realtà del genere serve tempo e specialmente serve il duro lavoro di tutti coloro che credono nelle potenzialità del proprio territorio e che non si arrendono alla realtà dei fatti attuale.

Esistono migliaia di giovani, che giorno dopo giorno, lottano per il loro Sud ed io vorrei far parte di loro. Per il momento continuo ad apprendere e a migliorare le mie abilità, per poi magari, un giorno, provare a fare la differenza nel mio piccolo.

È con questa mentalità che ho deciso di realizzare questa applicazione, non appena ho visto la possibilità di poter rendere la mia Regione protagonista del mio elaborato finale. Ovviamente sono ben consapevole che essa non porterà a nessun reale beneficio per la Regione, ma voglio far sì che questo elaborato funga da monito per chiunque legga queste parole, per me in primis.

3 Descrizione del problema affrontato

TasteTrip vorrebbe avere il compito di incentivare il turismo nella Regione Puglia e per fare ciò si pone l'obiettivo di facilitare al massimo l'organizzazione di una vacanza sul territorio regionale, risolvendo un problema con cui spesso ci si imbatte nel momento in cui si prova ad organizzare in autonomia l'itinerario della propria vacanza.

Il problema in questione è dato dall'impossibilità di conoscere tutte le attività disponibili su un vasto territorio, com'è quello della Regione Puglia, e dalla conseguente impossibilità di poter creare il viaggio perfetto e veramente su misura per i propri gusti.

A tutto ciò si unisce la difficoltà dovuta dall'organizzazione di un viaggio che non sfori il budget disponibile e che non lasci scontento nessun viaggiatore riguardo alle attività da svolgere.

Con l'utilizzo di opportuni database e di un'adeguata applicazione sviluppata in linguaggio Java, TasteTrip permette all'utente di ricercare, grazie ad un'interfaccia semplice ed intuitiva, le attività da svolgere ed i luoghi in cui pernottare, che massimizzano la qualità della propria esperienza di viaggio, il tutto nel rispetto delle proprie esigenze e dei propri gusti.

4 Descrizione del data-set

Tutti i dati necessari alla realizzazione dell'applicazione sono presi da cinque differenti tabelle: una tabella contenente le informazioni sui B&B presenti sul territorio della regione Puglia, tre differenti tabelle contenenti le informazioni sulle varie attività svolgibili sul territorio, divise secondo tre tipologie (attività turistiche, luoghi d'interesse e stabilimenti balneari), ed un'ultima tabella contenente le informazioni sui comuni della Regione.

Tutte le tabelle sono state modificate col fine di renderne più semplice l'utilizzo ai fini pratici dell'applicazione. Ad alcuni attributi sono stati dati dei valori casuali, lì dove il valore era null, mentre in alcune tabelle si è deciso di eliminare delle tuple.

4.1 B&B

Le informazioni riguardanti i B&B sono prese dalla tabella 'beb' [4].

Questa è composta da più di 3600 elementi, ognuno dei quali rappresenta un B&B caratterizzato in ordine da: il proprio identificativo numerico, il proprio nome, nome del proprietario, provincia del comune, comune, latitudine, longitudine, tipo di stanza (singola, appartamento, ecc.), prezzo per una notte e numero minimo di notti per cui bisogna soggiornare.

In precedenza il database era composto da circa 35000 elementi, che rendevano

quasi inutilizzabile l'applicazione, a causa dell'elevata quantità di dati che venivano considerati all'interno dell'algoritmo di ricorsione.

A causa di ciò si è deciso di ridurre, in modo random (utilizzando le opportune query), ad un numero più esiguo, la quantità degli elementi.

#	Nome	Tipo di dati
1	id	INT
2	name	VARCHAR
3	host_name	VARCHAR
4	neighbourhood_group	VARCHAR
5	neighbourhood	VARCHAR
6	latitude	DECIMAL
7	longitude	DECIMAL
8	room_type	VARCHAR
9	price	INT
10	minimum_nights	INT

Tabella 1: struttura dati tabella beb

4.2 Attività turistiche

La tabella 'attività_turistiche' [1] è composta da 570 elementi, ognuno rappresentante una singola attività, tipica del luogo in cui ha sede.

In questa tabella è possibile trovare le seguenti tipologie di attività: "Luoghi d'interesse naturale", "Alla scoperta della città", "Attività di formazione", "Attività sportiva".

L'attributo nome contiene il nome dell'attività specifica, mentre l'attributo tipologia contiene appunto una delle quattro tipologie appena citate.

L'attributo prezzo rappresenta il prezzo da pagare per svolgere l'attività. Essendo che non tutti gli elementi presentavano un valore per questo campo, si è deciso di inserire dei valori casuali, compresi fra 5 e 20, lì dove il valore era settato a null.

#	Nome	Tipo di dati
1	id	INT
2	nome	VARCHAR
3	tipologia	VARCHAR
4	cap	INT
5	comune	VARCHAR
6	indirizzo	VARCHAR
7	numeroCivico	VARCHAR
8	latitudine	DECIMAL
9	longitudine	DECIMAL
10	prezzo	DECIMAL

Tabella 2: struttura dati tabella attività_turistiche

4.3 Luoghi d'interesse

La tabella 'luoghi_interesse' [2] è composta da 648 elementi, ognuno rappresentante un luogo rilevante sotto vari possibili punti di vista.

In questa tabella è possibile trovare le seguenti tipologie di attività: "Luoghi di interesse architettonico e storico", "Luoghi d'interesse naturale", "Musei", "Luoghi di culto", "Località marittime", "Alla scoperta della città", "Attività sportiva".

L'attributo nomeAttrattore contiene il nome dell'attività specifica, mentre l'attributo tipologia contiene appunto una delle sette tipologie appena citate.

Una caratteristica importante è l'assenza di un attributo 'prezzo', data la natura dei luoghi; di conseguenza, nell'applicazione, ogni attività presa da questa tabella avrà un costo pari a zero.

#	Nome	Tipo di dati
1	id	INT
2	nomeAttrattore	VARCHAR
3	tipologia	VARCHAR
4	risorsaTerritoriale	VARCHAR
5	cap	INT
6	comune	VARCHAR
7	indirizzo	VARCHAR
8	numeroCivico	VARCHAR
9	latitudine	DECIMAL
10	longitudine	DECIMAL

Tabella 3: struttura dati tabella luoghi_interesse

4.4 Stabilimenti balneari

La tabella 'stabilimenti_balneari' [3] non ha un campo 'tipologia' come nei due precedenti casi, dato che all'interno di questa tabella è presente un'unica tipologia, data appunto dal nome stesso della tabella.

Anche in questa tabella è presente una particolarità nel prezzo, il quale non è impostato come valore decimale, ma come stringa all'interno della voce 'Prezzi_alta_o_unica_stagione'.

Per semplicità riporto un esempio di questo attributo:

'lettini (2-3)#lettini-stagione (150-200)#ombrellone (6-9)#ombrellone-stagione (400-500)#sedie-sdraio (2-3)'

Le informazioni che in genere ho estrapolato da una stringa del genere, sono quelle riguardanti il costo di un lettino ed il costo di un ombrellone, cioè il primo valore che si trova all'interno della parentesi (in questo caso costo lettino pari a 2,00€ e costo ombrellone pari a 6,00€), ignorando volutamente, per semplicità, ogni altro tipo di informazione.

Durante lo sviluppo si è ipotizzato che uno stabilimento balneare possa essere aperto solo dal 01/05 fino al 30/09, e che 2 persone possano condividere un singolo ombrellone, mentre una terza persona ne dovrà prendere uno a parte; ogni lettino è invece personale.

#	Nome	Tipo di dati
1	Comune	VARCHAR
2	Provincia	VARCHAR
3	Nome_operatore	VARCHAR
4	Indirizzo	VARCHAR
5	Numero_civico	VARCHAR
6	Prezzi_alta_o_unica_stagione	VARCHAR
7	ID_operatore	INT
8	Latitudine	DECIMAL
9	Longitudine	DECIMAL

Tabella 4: struttura dati tabella stabilimenti_balneari

4.5 Tabella comuni

La tabella 'comuni_puglia' è stata ottenuta incrociando i comuni delle varie tabelle descritte fin ora, dato che ottenere una tabella del genere non è possibile gratuitamente e comporta dei costi.

Per fare ciò sono state utilizzate le opportune query per la creazione di tabelle e per l'aggiunta di tuple.

#	Nome	Tipo di dati
1	id	INT
2	cap	INT
3	nomeProvincia	VARCHAR
4	nome	VARCHAR
5	provincia	VARCHAR
6	lat	DECIMAL
7	lng	DECIMAL

Tabella 5: struttura dati tabella comuni_puglia

5 Strutture dati ed algoritmi

5.1 Strutture dati utilizzate

Per lo sviluppo dell'applicazione sono stati utilizzati il pattern MVC (Model-View-Controller) ed il pattern DAO (Data-Access-Object), con lo scopo di tenere ben separate fra loro l'interfaccia utente, l'accesso ai dati del database e la logica del programma.

A questo scopo sono stati creati tre differenti packages, uno per ognuno dei componenti appena citati.

5.1.1 Package TasteTrip

All'interno del package TasteTrip vi sono tre classi: Main, EntryPoint e FXMLController. Il Main è adibito all'avvio dell'applicazione, mentre l'EntryPoint gestisce l'avvio della schermata che l'utente visualizzerà (Figura 12) ed attraverso la quale inserirà i vari dati di input. Questa schermata è stata creata utilizzando il software SceneBuilder.

La classe FXMLController ha invece il compito di gestire l'interazione fra la View dell'utente ed il Model, convertendo adeguatamente gli input dell'utente in dati gestibili dal Model, e gli output generati dal Model in informazioni facilmente comprensibili dall'utente.

5.1.2 Package db

Vista la necessità di avere una banca di dati dalla quale leggere informazioni utili allo scopo dell'applicazione, si è resa necessaria la creazione di un package il cui compito fosse la sola interazione fra Model e database, seguendo le regole del pattern DAO.

Anche all'interno del package db vi sono tre classi: DBConnect, TasteTripDAO e TestDAO.

La classe DBConnect si occupa di creare il collegamento al database.

È la classe TasteTripDAO quella che ha tutti i metodi necessari alla lettura dei dati dal database (Tabella 7). A questi metodi si accede tramite la classe Model, contenuta nel package model.

Infine la classe TestDAO è una classe utile, in fase di programmazione, ad eseguire dei test sul corretto funzionamento della classe TasteTripDAO.

5.1.3 Package model

In questo package è contenuta tutta la parte logica dell'applicazione, la quale viene implementata attraverso l'utilizzo di sei classi: Model, TestModel, Comune, BeB, Attivita e Percorso (Tabella 6).

Con i dati ricevuti in input attraverso la classe FXMLController, utilizzando i metodi della classe Model, vengono eseguite tutte le operazioni necessarie alla generazione dell'output richiesto.

La classe TestModel ha lo scopo di verificare, in fase di programmazione, il corretto funzionamento dei metodi della classe Model.

Le tre classi Comune, BeB ed Attivita hanno il compito di rappresentare le varie tuple delle tabelle presenti nel database.

Un oggetto Comune ha gli attributi mostrati nella Tabella 6, i quali corrispondono ai dati della tabella comuni_puglia del DB. Essendo che nella tabella, un comune si può ripetere

più volte, con differenti CAP e valori ad esso associati di latitudine e longitudine, ogni comune ha un attributo `HashMap<Integer, LatLng>`, la cui chiave è il CAP ed il cui valore sono le coordinate del comune, espresse da un oggetto della classe `LatLng`.

Inoltre, ogni Comune contiene una lista degli oggetti `BeB` ed una lista degli oggetti `Attivita` che hanno come attributo il comune specifico.

Un oggetto `BeB` corrisponde ad una singola tupla della tabella `beb` del DB.

Un oggetto `Attivita` può invece corrispondere ad una delle tuple di più tabelle. Può far parte della tabella `attivita_turistiche`, `luoghi_interesse`, o `stabilimenti_balneari`, le quali hanno tutte la caratteristica di contenere elementi rappresentabili, ai fini dell'applicazione, con la stessa classe, nonostante l'essere tre tabelle differenti.

Ognuna di queste tre classi è dotata di un costruttore e degli opportuni metodi `get`, `set`, `hashCode`, `equals` e `toString`.

La classe `Percorso` è una classe utilizzata come supporto per la generazione dell'output.

Un oggetto `Percorso` è caratterizzato da un Comune, un `BeB`, una lista di `Attivita`, un valore numerico decimale, rappresentativo del costo del Percorso e dalla durata, espressa in numero di giorni, del viaggio.

5.2 Algoritmi utilizzati

Per parlare degli algoritmi utilizzati, bisogna prima descrivere l'importanza dei vari vincoli inseriti dall'utente.

Egli seleziona una provincia dal primo elenco, successivamente, se lo desidera, imposta anche un comune specifico, seleziona poi almeno una delle 9 tipologie di attività disponibili ed imposta: il giorno della partenza ed il giorno del ritorno, un budget massimo che sarebbe disposto a spendere, la distanza massima che sarebbe disposto a percorrere per raggiungere una determinata meta ed il numero di viaggiatori (Figura 12).

Nelle due date citate, sono implicite la durata della vacanza ed il periodo dell'anno durante il quale la si vorrebbe svolgere. Per quanto riguarda il periodo, esso è utile per escludere la possibilità di andare in un lido nei mesi non estivi (4.4Stabilimenti balneari).

Il budget massimo è un vincolo che riguarda la somma dei costi dei pernottamenti e di tutte le attività svolte, di conseguenza, al variare delle attività svolte, varierà la spesa complessiva.

Il numero di giorni pone invece un vincolo sulla quantità di attività che è possibile svolgere durante la vacanza.

All'interno dell'applicazione si ipotizzerà che non sia possibile svolgere più di due differenti attività nell'arco della stessa giornata (una al mattino ed una al pomeriggio).

5.2.1 Lettura dati dal database

Il primo passo consiste nella lettura dal DB dei dati, scelti in base alle selezioni dell'utente.

Nel caso l'utente non abbia selezionato un comune specifico, per la ricerca dei B&B e delle attività vengono presi in considerazione tutti i comuni della provincia selezionata.

```

public void addComuniBySelezioneProvincia(String sigla){
    comuni = new ArrayList<Comune>();
    comuni.addAll(dao.getCommuniByProvincia(sigla));
}

```

Figura 1: metodo del Model per l'aggiunta dei comuni di una provincia specifica

```

public List<Comune> getCommuniByProvincia(String sigla) {

    String sql = "SELECT * FROM comuni_puglia WHERE provincia = ? ORDER BY nome";
    List<Comune> comuni = new ArrayList<Comune>();

    try {
        Connection conn = DBConnect.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, sigla);
        ResultSet res = st.executeQuery();

        while (res.next()) {
            boolean esistente = false;
            for(Comune c : comuni) {
                if( c.getNome().equals(res.getString("nome"))) {
                    c.addCapCoordinate(res.getInt("cap"), res.getDouble("lat"), res.getDouble("lng"));
                    esistente = true;
                }
            }
            if(!esistente) {
                comuni.add(new Comune(res.getString("nome"), res.getString("provincia"),
                    res.getString("nomeProvincia"), new LatLng(res.getDouble("lat"),
                    res.getDouble("lng")), res.getInt("cap")));
            }
        }

        conn.close();

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }

    return comuni;
}

```

Figura 2: metodo del TasteTripDAO per la selezione dei comuni dal DB

Nel caso in cui invece sia stato selezionato un comune specifico, la ricerca dei B&B viene effettuata solo sul territorio di quest'ultimo, mentre la ricerca delle attività viene svolta, oltre che sul territorio del comune, anche sui territori dei comuni vicini, che hanno una distanza massima da quello scelto pari alla distanza inserita dall'utente. Tutte le distanze sono calcolate attraverso l'utilizzo dei dati relativi a latitudine e longitudine presenti in ogni data-set.

```

public void addComuniBySelezioneSpecificaComune(Comune comune){
    comuni = new ArrayList<Comune>();
    comuni.add(comune);
    List<Comune> templList = dao.getAllCommuni();

    for(Comune c : templList) {
        boolean minore = false;
        if( !c.equals(comune) && !comuni.contains(c) ) {
            for(LatLng coor1 : c.getMapCapCoordinate().values()) {
                for(LatLng coor2 : comune.getMapCapCoordinate().values()) {
                    Double distanza = LatLngTool.distance(coor1, coor2, LengthUnit.KILOMETER);
                    if(distanza < distanzaMax) {
                        minore = true;
                    }
                }
            }
            if(minore) {
                comuni.add(c);
            }
        }
    }
}

```

Figura 3: metodo del Model per l'aggiunta dei comuni che hanno una distanza dal comune selezionato, minore della distanza massima

I B&B sono scelti anche in base alla durata del viaggio, a seconda del numero minimo di notti di pernottamento richiesto dalle varie strutture.

```

public void addBeBComune(Comune comune){
    if(comune == null) {
        if(numGiorni!=1) {
            for(Comune c : comuni) {
                c.addListaBeB(dao.getBeBComune(c, numGiorni-1, numPersone));
            }
        }
    }
    else {
        comune.addListaBeB(dao.getBeBComune(comune, numGiorni-1, numPersone));
    }
}

```

Figura 4: metodo del Model per l'aggiunta dei B&B


```

public List<BeB> getBeBComune(Comune comune, long numNotti, int numPersone) {

    String sql = "SELECT `name`, host_name, neighbourhood, room_type, latitude,"
        + " longitude, price FROM beb WHERE minimum_nights<=? AND neighbourhood=?";
    List<BeB> listaBeb = new ArrayList<>();

    try {
        Connection conn = DBConnect.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        st.setLong(1, numNotti);
        st.setString(2, comune.getNome());
        ResultSet res = st.executeQuery();

        while (res.next()) {
            listaBeb.add(new BeB(res.getString("name"), res.getString("host_name"),
                comune, res.getString("room_type"), new Latlng( res.getDouble("latitude"),
                    res.getDouble("longitude") ), res.getDouble("price")*numPersone));
        }

        conn.close();

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }

    return listaBeb;
}

```

Figura 5: metodo del TasteTripDAO per la selezione dei B&B dal DB

Le attività vengono aggiunte ad ogni comune della lista ‘comuni’ in base alle tipologie selezionate dall’utente.

Inoltre si tiene traccia delle tipologie che sono state selezionate, il che sarà utile durante la ricorsione, per non inserire nel Percorso, Attività che siano tutte della stessa tipologia. Di seguito, nella Figura 6 e nella Figura 7, vi è mostrato come vengono aggiunte le attività turistiche; l’aggiunta dei luoghi d’interesse e degli stabilimenti balneari viene effettuata nello stesso modo.

```

public void addAttivitaTuristicheComuni(List<String> tipologie) {
    for(String t : tipologie) {
        if(!tipoAttivita.contains(t)) {
            tipoAttivita.add(t);
        }
    }
    for(Comune c : comuni) {
        for(String t : tipologie) {
            c.addListaAttivita(dao.getAttivitaTuristicheComuni(c, t, numPersone));
        }
    }
}

```

Figura 6: metodo del Model per l’aggiunta delle attività turistiche

```

public List<Attivita> getAttivitaTuristicheComuni(Comune comune, String tipologia, int numPersone){
    List<Attivita> attivita = new ArrayList<Attivita>();

    String sql = "SELECT nome, tipologia, comune, indirizzo, cap, latitudine, longitudine, "
        + "prezzo FROM attivita_turistiche WHERE comune = ? AND tipologia = ?";

    try {
        Connection conn = DBConnect.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, comune.getNomeProvincia());
        st.setString(2, tipologia);
        ResultSet res = st.executeQuery();
        while (res.next()) {
            attivita.add(new Attivita(res.getString("nome"), tipologia, comune,
                res.getString("indirizzo"), res.getInt("cap"),
                new LatLng(res.getDouble("latitudine"), res.getDouble("longitudine")),
                res.getDouble("prezzo")*numPersone));
        }

        conn.close();

    } catch (SQLException e) {
        throw new RuntimeException(e);
    }

    return attivita;
}

```

Figura 7: metodo del TasteTripDAO per la selezione delle attività dal DB

5.2.2 Ricorsione

Il secondo passo è quello di eseguire una ricorsione per trovare le migliori combinazioni di attività e B&B rispettanti i dati di input.

In base alle tipologie di attività selezionate e alla durata del viaggio, si viene a creare una situazione nella quale si hanno due possibili casi:

- numero di tipologie di attività selezionate ≥ 2 * numero giorni vacanza
- numero di tipologie di attività selezionate < 2 * numero giorni vacanza.

In entrambi i casi, la ricorsione restituisce un oggetto Percorso contenente il B&B e la combinazione di attività che massimizzano la spesa.

Le attività sono contenute in una lista di dimensione $2 * (\text{numero giorni vacanza})$.

Nel caso (1) le varie liste contengono le diverse combinazioni di tipologie possibili, senza mai contenere tutte le tipologie di attività selezionate dall'utente.

Nel caso (2) nelle liste vengono inserite più di una volta alcune delle tipologie di attività selezionate.

La ricorsione viene effettuata con lo scopo di massimizzare la spesa della vacanza, restando al di sotto del budget inserito dall'utente, questo perché si vuole creare l'itinerario migliore per il turista e per far ciò si ipotizza che ad una maggiore spesa corrisponda una maggiore qualità della vacanza.

In definitiva, il Percorso che massimizza la spesa è il percorso più consigliato.

Inoltre all'utente viene fornita, per dargli una scelta più vasta, una lista dei 10 percorsi migliori corrispondenti alle tipologie scelte, cioè i 10 percorsi che hanno costo totale massimo fra tutti ed inferiore o uguale al costo del percorso migliore (attenzione, la

dimensione della lista dei percorsi migliori potrebbe anche essere inferiore a 10, così come potrebbe non essere trovato nessun percorso che rispetti i vincoli inseriti). Ovviamente tutti i percorsi forniti saranno tali da rispettare i limiti imposti dall'utente. Nella ricorsione vengono considerati i vari casi:

- Viaggio di più giorni e selezione del comune effettuata:
 - Itero la ricorsione su tutti i B&B del comune selezionato.
- Viaggio di più giorni e selezione del comune non effettuata:
 - Itero la ricorsione su tutti i B&B di tutti i comuni della provincia.
- Viaggio di un solo giorno:
 - Nessun comune e nessun B&B da selezionare.

```
public Percorso ricorsione(Comune comune){

    altriPercorsi = new ArrayList<Percorso>();
    bestPercorso = new Percorso(comune, 0, numGiorni);

    if(comune != null && numGiorni>1 && comune.getListaBeB().isEmpty()) {
        return bestPercorso;
    }

    Percorso parziale = new Percorso(comune, 0, numGiorni);
    List<Attivita> attivita = new ArrayList<Attivita>();

    for(Comune c : comuni) {
        attivita.addAll(c.getListaAttivita());
    }
    int i = 0;
    for(Attivita a : attivita) {
        a.setOrdine(i);
        i++;
    }

    if(numGiorni>1) {
        if(comune != null) {
            for(BeB b : comune.getListaBeB()) {
                if( b.getPrezzo() <= spesaMax ) {
                    parziale.setBeb(b);
                    parziale.addCosto(parziale.getBeb().getPrezzo());
                    cerca(b, parziale, attivita, comune, tipoAttivita);
                    parziale.removeCosto(parziale.getBeb().getPrezzo());
                    parziale.unsetBeb();
                }
            }
        }else {
            for(Comune c : comuni) {
                for(BeB b : c.getListaBeB()) {
                    if( b.getPrezzo() <= spesaMax ) {
                        parziale.setBeb(b);
                        parziale.setComune(b.getComune());
                        parziale.addCosto(parziale.getBeb().getPrezzo());
                        cerca(b, parziale, attivita, comune, tipoAttivita);
                        parziale.removeCosto(parziale.getBeb().getPrezzo());
                        parziale.unsetBeb();
                    }
                }
            }
        }
    }
    else { // Se numGiorni==1, non ci sono notti di pernottamento da considerare.
        cerca(null, parziale, attivita, comune, tipoAttivita);
    }

    return bestPercorso;
}
```

Figura 8: metodo del Model per effettuare la ricorsione su tutti i B&B necessari

All'interno del metodo cerca, richiamato durante la ricorsione, la condizione di terminazione si ha nel momento in cui la lista di Attività ha raggiunto dimensione pari a "2*durata del viaggio".

In questo caso, se il percorso parziale rispetta tutti i vincoli, viene aggiunto alla lista dei potenziali migliori percorsi.

Per aggiornare il percorso migliore, si seguono i seguenti 2 vincoli:

- Il costo del Percorso parziale dev'essere strettamente maggiore del costo dell'attuale Percorso migliore
- Il costo del Percorso parziale dev'essere uguale al costo dell'attuale Percorso migliore e le tipologie delle sue Attività devono essere più variegata rispetto a quelle delle Attività del Percorso migliore attuale; questo per cercare di non rendere la vacanza monotona.

```
private void cerca(BeB b, Percorso parziale, List<Attività> attività, Comune comune, List<String> tipiRestanti) {
    if(parziale.getAttività().size()==2*numGiorni) {
        if( parziale.getCosto() <= spesaMax && !altriPercorsi.contains(parziale) ) {
            altriPercorsi.add(new Percorso(parziale));
            if(parziale.getCosto() >= bestPercorso.getCosto()) {
                if(parziale.getCosto() > bestPercorso.getCosto() ) {
                    bestPercorso = new Percorso(parziale);
                }
            }
            else {
                List<String> tipologiePresentiParziale = new ArrayList<String>();
                List<String> tipologiePresentiBest = new ArrayList<String>();
                for(Attività a : parziale.getAttività()) {
                    if(!tipologiePresentiParziale.contains(a.getTipologia())) {
                        tipologiePresentiParziale.add(a.getTipologia());
                    }
                }
                for(Attività a : bestPercorso.getAttività()) {
                    if(!tipologiePresentiBest.contains(a.getTipologia())) {
                        tipologiePresentiBest.add(a.getTipologia());
                    }
                }
                if(tipologiePresentiParziale.size() > tipologiePresentiBest.size()) {
                    bestPercorso = new Percorso(parziale);
                }
            }
        }
    }
    return;
}
```

Figura 9: metodo cerca del Model, parte 1, condizione di terminazione, condizione per l'aggiunta di un percorso ed aggiornamento del percorso migliore

Nel metodo cerca, si tengono ancora in considerazione molti aspetti:

- Viaggio di più giorni:
 - Nel calcolo della distanza, tengo in considerazione la distanza fra il BeB passato come parametro e l'Attività attuale.
- Viaggio di un solo giorno:
 - Nel calcolo della distanza, tengo in considerazione la distanza fra l'Attività attuale e tutte le Attività già inserite nella lista del Percorso.
- La lista tipiRestanti contiene la tipologia dell'Attività attuale:
 - Procedo con la ricorsione.
- La lista tipiRestanti risulta vuota:

- Riempio nuovamente la lista con tutte le tipologie iniziali per continuare la ricerca; sono nel caso in cui:
numero di tipologie di attività selezionate $< 2 * \text{numero giorni vacanza}$.
- La lista tipiRestanti non è vuota, ma non contiene la tipologia dell'Attività attuale:
 - Verifico che esistano ancora Attività, non utilizzate, aventi una delle tipologie presenti nella lista.
Nel caso la risposta sia affermativa, ignoro l'Attività attuale fino a quando non trovo un'Attività di una delle tipologie presenti nella lista.
Nel caso invece in cui la risposta sia negativa, procedo con il riempire nuovamente la lista con tutte le tipologie iniziali per continuare la ricerca; sono nel caso:
numero di tipologie di attività selezionate $< 2 * \text{numero giorni vacanza}$.

Per diminuire i tempi di ricerca sono stati aggiunti dei vincoli ulteriori, non essenziali, quali:

- Verifica che la somma fra il costo parziale ed il costo dell'Attività attuale sia minore del budget.
- Ogni Attività è stata dotata di un attributo numerico intero 'ordine' il quale serve a far sì che non si ripetano le stesse combinazioni di Attività, posizionate in ordine diverso.
Per far ciò, la ricorsione viene effettuata solo nel caso in cui l'ordine dell'Attività attuale sia maggiore dell'ordine dell'ultima Attività inserita nella lista.

6 Diagramma delle classi principali



Tabella 6: diagramma delle principali classi del package model

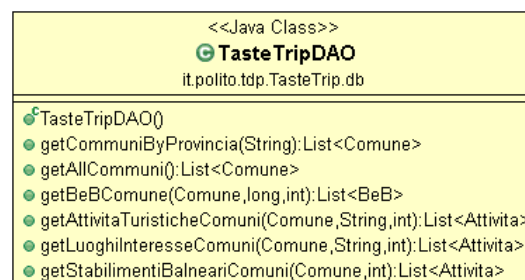


Tabella 7: diagramma delle principali classi del package db

7 Interfaccia dell'applicazione

Figura 12: schermata iniziale dell'applicazione

TasteTrip
Applicazione per il turismo della regione Puglia

Selezionare la provincia:
 Barletta-Andria-Trani

Selezionare, se lo si desidera, un comune:

Selezionare le tipologie di attività che si vorrebbero svolgere:

<input type="checkbox"/> Alla scoperta della città'	<input checked="" type="checkbox"/> Attività di formazione
<input checked="" type="checkbox"/> Luoghi d'interesse naturale	<input type="checkbox"/> Luoghi di interesse architettonico e storico
<input checked="" type="checkbox"/> Musei	<input checked="" type="checkbox"/> Luoghi di culto
<input type="checkbox"/> Attività sportiva	<input type="checkbox"/> Località marittime
<input checked="" type="checkbox"/> Stabilimenti balneari	

Selezionare il periodo in cui si vorrebbe effettuare la propria vacanza:

Andata: 01/09/2020 Ritorno: 03/09/2020

Spesa massima che si è disposti a sostenere: 500 €

Distanza massima che si è disposti a percorrere: 10 Km

Numero di viaggiatori: 2

Reset **Calcola itinerario**

Il viaggio più confortevole e lussuoso che rispetta le caratteristiche volute è:

Soggiorna nella magnifica cittadina di Trani.
 Fatti ospitare da Loic, Estelle & Co nel B&B "Amazing apt with swimming-pool" situato in Trani.
 Tipologia di camera: Entire home/apt
 Prezzo: 364,00 €

Figura 13: esempio inserimento dati, con conseguente risultato

Video su Youtube che ne mostra l'utilizzo: <https://youtu.be/qZpRBM-stcg>.

8 Risultati sperimentali

Nei seguenti esempi si riporta l'input inserito e l'output ricevuto in risposta a seguito della ricorsione.

L'output non viene riportato in forma completa, a causa della sua lunghezza; verrà riportato ogni volta il percorso migliore e, se necessario ai fini dell'esempio, alcuni degli altri percorsi trovati.

8.1 Ricerca sul territorio di un'intera Provincia

Nel caso non venga selezionato nessun comune specifico, la ricerca del B&B viene effettuata su tutti i comuni dell'intero territorio della Provincia selezionata.

8.1.1 Esempio 1

Input:

Provincia:	Barletta-Andria-Trani
Comune:	null
Tipologie di attività selezionate:	<ul style="list-style-type: none">- Alla scoperta della città- Attività di formazione- Luoghi d'interesse naturale- Stabilimenti balneari
Andata:	01/09/2020
Ritorno:	02/09/2020
Budget:	450,00 €
Distanza massima:	10 km
Numero viaggiatori:	2 persone

Output:

Soggiorna nella magnifica cittadina di Trani.

Fatti ospitare da Loic, Estelle & Co nel B&B "Amazing apt with swimming-pool" situato in Trani.

Tipologia di camera: Entire home/apt

Prezzo: 364,00 €

Sul territorio potrai eseguire le seguenti attività:

Alla scoperta dei sapori del frantoio, Alla scoperta della città, Bisceglie, Via Vecchia Corato, costo: 40,00 €

Archimede incontra..., Attività di formazione, Trani, Piazza Re Manfredi, costo: 10,00 €

Suoni e ritmi del MedioEco, Attività di formazione, Trani, Piazza Re Manfredi, costo: 16,00 €

Trani tra il Doge e i D'Asburgo, Attività di formazione, Trani, Piazza Manfredi Re, costo: 20,00 €

Costo totale: 450,00 €

Di seguito vengono elencati i 10 percorsi migliori subito dopo quello già riportato:

1) Soggiorna nella magnifica cittadina di Trani.

Fatti ospitare da Loic, Estelle & Co nel B&B "Amazing apt with swimming-pool" situato in Trani.

Tipologia di camera: Entire home/apt

Prezzo: 364,00 €

Sul territorio potrai eseguire le seguenti attività:

Alla scoperta dei sapori del frantoio, Alla scoperta della città', Bisceglie, Via Vecchia Corato, costo: 40,00 €
 Nella terra dell'olio e della poesia, Attività' di formazione, Bisceglie, Via Ruvo, costo: 0,00 €
 LIDO TERME, Stabilimenti Balneari, Margherita di Savoia, lungomare cristoforo Colombo, costo: 25,00 €
 Federico II e gli Ebrei, Alla scoperta della città', Trani, Piazza Manfredi Re, costo: 20,00 €
 Costo totale: 449,00 €
 [...]

8.1.2 Esempio 2

Input:

Provincia:	Foggia
Comune:	null
Tipologie di attività selezionate:	<ul style="list-style-type: none"> - Alla scoperta della città - Attività di formazione - Luoghi d'interesse naturale - Luoghi di culto
Andata:	01/09/2020
Ritorno:	02/09/2020
Budget:	300,00 €
Distanza massima:	20 km
Numero viaggiatori:	2 persone

Output:

Il viaggio più confortevole e lussuoso che rispetta le caratteristiche volute è:

Soggiorna nella magnifica cittadina di Foggia.
 Fatti ospitare da Daniela nel B&B "Suite Familiare in Casa Fuiani" situato in Foggia.
 Tipologia di camera: Private room
 Prezzo: 200,00 €
 Sul territorio potrai eseguire le seguenti attività:
 Il menù delle scampagnate di un tempo, Attività di formazione, Foggia, Basilica Santuario Incoronata, costo: 28,00 €
 Chiesa del Monte Calvario (detta Chiesa delle Croci), Luoghi di culto, Foggia, Piazza Sant'Eligio, costo: 0,00 €
 Scoprendo Lucera, Alla scoperta della città', Lucera, Piazza Nocelli, costo: 32,00 €
 Un percorso di arte cultura e tradizione, Alla scoperta della città', Lucera, Piazza Nocelli, costo: 38,00 €
 Costo totale: 298,00 €

Di seguito vengono elencati i 10 percorsi migliori subito dopo quello già riportato:
 [...]

7) Soggiorna nella magnifica cittadina di Foggia.
 Fatti ospitare da Daniela nel B&B "Suite Familiare in Casa Fuiani" situato in Foggia.
 Tipologia di camera: Private room
 Prezzo: 200,00 €
 Sul territorio potrai eseguire le seguenti attività:
 Herdonia a quiz al Museo Civico di Foggia, Attività di formazione, Foggia, Piazza Nigri, costo: 36,00 €

Chiesa del Monte Calvario (detta Chiesa delle Croci), Luoghi di culto, Foggia, Piazza Sant'Eligio, costo: 0,00 €

Un percorso di arte cultura e tradizione, Alla scoperta della città', Lucera, Piazza Nocelli, costo: 38,00 €

Un giorno a Herdonia, Alla scoperta della città', Ortona, Herdonia, costo: 24,00 €

Costo totale: 298,00 €

Osservazioni:

Gli ulteriori percorsi, che vanno dal primo fino al sesto, risultano molto simili al Percorso migliore, si tratta della variazione di una singola Attività in ogni lista, è dal settimo percorso che si nota una differenza maggiore.

8.2 Ricerca con settaggio di un comune specifico

Nel caso venga selezionato un comune specifico, la ricerca del B&B viene effettuata solo sul suo territorio. Per le attività vengono prese in considerazione tutte quelle alla distanza massima impostata.

8.2.1 Esempio 3

Input:

Provincia:	Brindisi
Comune:	Cisternino
Tipologie di attività selezionate:	<ul style="list-style-type: none">- Musei- Luoghi di culto- Attività sportiva- Stabilimenti balneari
Andata:	01/09/2020
Ritorno:	04/09/2020
Budget:	400,00 €
Distanza massima:	15 km
Numero viaggiatori:	2 persone

Output:

Soggiorna nella magnifica cittadina di Cisternino.

Fatti ospitare da NÃ¹evÃ¹ nel B&B "Eco friendly artists trullo" situato in Cisternino.

Tipologia di camera: Entire home/apt

Prezzo: 320,00 €

Sul territorio potrai eseguire le seguenti attività:

Conservatorio botanico della Valle d'Itria, Musei, Cisternino, Contrada Figazzano, costo: 0,00 €

Lido Tavernese, Stabilimenti Balneari, Fasano, Appia Antica - Posto di Tavernese - Torre Canne, costo: 18,00 €

Chiesa rupestre di San Giovanni Battista, Luoghi di culto, Fasano, Strada Provinciale Fasano-Savelletri, costo: 0,00 €

Selva di Fasano, Attività sportiva, Fasano, Piazza Ciaia, costo: 0,00 €

Lido il Faro, Stabilimenti Balneari, Ostuni, Via Consolato Veneziano - Villanova Zona Fontanelle, costo: 25,00 €

Parapendio tra volo e natura, Attività sportiva, Ostuni, Strada complanare alla superstrada SS379 tra Torre Canne e Rosamarina, costo: 36,00 €

Museo di Civiltà preclassiche della Murgia meridionale e Parco Archeologico di Santa Maria Agnano, Musei, Ostuni, Via Cattedrale, costo: 0,00 €
Convento e chiesa delle Monacelle, Luoghi di culto, Ostuni, Via Cattedrale, costo: 0,00 €
Costo totale: 399,00 €

8.3 Osservazioni dovute ai vari test effettuati

In tutti gli esempi riportati, i tempi necessari per eseguire l'algoritmo sono al di sotto della decina di secondi, ma non sempre è così.

Una ricerca effettuata sull'intera Provincia di Lecce tiene in considerazione 97 comuni, 1935 B&B ed un massimo di quasi 500 attività (quest'ultimo numero ovviamente varia in base alle selezioni dell'utente).

Una quantità di dati del genere, inserita all'interno di un algoritmo di ricorsione come quello di TasteTrip, porta ad avere dei tempi di calcolo esageratamente elevati.

Risulta fattibile la ricerca sull'intero territorio leccese solo nel caso in cui si effettui un viaggio di un solo giorno e di conseguenza non si abbia la necessità di un B&B.

Non con tutte le Province si ha lo stesso problema: Taranto, Foggia e Barletta-Andria-Trani, risultano Province a più bassa densità di B&B ed attività. Ciò facilita la ricerca sui loro territori.

Nonostante ciò risulta difficile eseguire una ricerca, su questi territori, per viaggi di durata maggiore ai 2 giorni.

Differente è il caso in cui si vada a selezionare un comune specifico in cui effettuare la ricerca dei B&B (8.2.1Esempio 3), facendo attenzione però a non inserire una distanza troppo elevata, che porterebbe la difficoltà di ricerca ad essere simile a quella che si ha in una ricerca su un'intera Provincia, se non addirittura maggiore.

9 Valutazioni e conclusioni

Trattandosi di un'applicazione che utilizza un algoritmo di ricorsione e conoscendo la dimensione non esigua dei database, è importante tenere sott'occhio i tempi necessari per la creazione dell'output desiderato, che potrebbero essere anche dell'ordine di minuti nel caso di ricerche effettuate su un vasto territorio o su territori con elevata concentrazione di B&B ed attività.

Eseguendo vari test si nota ciò che già dalla semplice lettura del codice di TasteTrip era deducibile e cioè come al crescere dei valori inseriti dall'utente, crescano anche i tempi necessari al programma per eseguire tutti i calcoli.

Una distanza maggiore considererà un numero maggiore di Comuni ed Attività sul territorio, ciò implica un maggior numero di dati in pasto alla ricorsione

Una durata del viaggio maggiore comporterà un aumento delle chiamate effettuate al metodo cerca del Model, con conseguente dilatazione dei tempi di ricerca.

Anche la spesa massima ha un effetto sui tempi di ricerca, anche se più esiguo rispetto agli attributi precedenti. Ciò a causa dei controlli effettuati all'interno della ricorsione per cercare di ottimizzare al massimo l'algoritmo.

Ciò che fa veramente la differenza sono però le singole zone scelte, che potrebbero avere un'elevata densità di B&B ed attività, così come invece potrebbero non averne.

I punti deboli di TasteTrip sono da ritrovare nella difficoltà computazionale della ricorsione, che porta a non poter effettuare le ricerche che tengono in considerazione un'elevata quantità di B&B ed attività.

In definitiva, anche se non in maniera impeccabile, TasteTrip riesce a portare a termine il compito che era stato prefissato: creare una vacanza su misura per l'utente, che risulti personalizzata secondo i suoi gusti e le sue necessità.

Riferimenti

- [1] ARET Pugliapromozione, «Attività da svolgere in Puglia - Dati Puglia.it,» [Online]. Available: <https://www.dataset.puglia.it/dataset/attivita-da-svolgere-in-puglia>.
- [2] ARET Pugliapromozione, «Luoghi di interesse turistico, culturale, naturalistico - Dati Puglia.it,» [Online]. Available: <https://www.dataset.puglia.it/dataset/luoghi-di-interesse-turistico-culturale-naturalistico>.
- [3] ARET Pugliapromozione, «Regione Puglia - Stabilimenti balneari - DatiOpen.it,» [Online]. Available: http://www.datiopen.it/it/opendata/Regione_Puglia_Stabilimenti_balneari.
- [4] M. Cox, «Get the Data - Inside Airbnb. Adding data to the debate,» [Online]. Available: <http://insideairbnb.com/get-the-data.html>.
- [5] Wikipedia, «Lucio Dalla - Wikipedia,» [Online]. Available: https://it.wikipedia.org/wiki/Lucio_Dalla.



Quest'opera è distribuita con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

Copia della licenza consultabile al sito web:

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