



Politecnico di Torino

Corso di Laurea in Ingegneria Gestionale L8
A.A. 2023/2024

Sessione di Laurea Marzo 2024

**Software per la gestione e la
massimizzazione dei profitti nel settore
della vendita di imbarcazioni da diporto**

Relatore:

Prof. Fulvio Corno

Candidato:

Maida Gianluca (273910)

Sommario

Capitolo 1: Proposta di progetto	3
1.1 Studente proponente.....	3
1.2 Titolo della proposta	3
1.3 Descrizione del problema proposto.....	3
1.4 Descrizione della rilevanza gestionale del problema	4
1.5 Descrizione del data-set per la valutazione	4
1.6 Descrizione preliminare degli algoritmi coinvolti.....	5
1.7 Descrizione preliminare delle funzionalità previste per l'applicazione	6
Capitolo 2: Descrizione dettagliata del problema	7
2.1 Obiettivo	7
Capitolo 3: Descrizione del data-set.....	10
3.1 Descrizione.....	10
Capitolo 4: Descrizione delle strutture dati e degli algoritmi utilizzati	12
4.1 Strutture dati.....	12
4.2 Principali algoritmi.....	14
Capitolo 5: Diagramma delle classi delle principali parti dell'applicazione.....	20
5.1 Classi dell'applicazione.....	20
Capitolo 6: Interfaccia dell'applicazione	22
6.1 Funzionamento dell'applicazione	22
6.2 Video dimostrativo.....	24
Capitolo 7: Risultati Sperimentali.....	25
7.1 Primo Esempio: No Filtri	25
7.2 Secondo Esempio: Confronto con Primo Esempio.....	26
7.3 Terzo Esempio: Conferma Soluzione	27
7.4 Quarto Esempio: Doppia Soluzione	29
7.5 Quinto Esempio: Inserimento ID.....	31
Capitolo 8: Conclusioni Finali	33
8.1 Limiti dell'applicazione.....	33
8.2 Valutazione dei risultati.....	34
Capitolo 9: Licenza.....	35

Capitolo 1: Proposta di progetto

1.1 Studente proponente

S273910 Maida Gianluca

1.2 Titolo della proposta

Software per la gestione e la massimizzazione dei profitti nel settore della vendita di imbarcazioni da diporto.

1.3 Descrizione del problema proposto

La Società in questione si occupa della compravendita di barche di varia grandezza e tipologia; in particolare, acquista sia nuovo che usato per poi rivenderlo alla clientela gestendo rimessaggio, montaggio e/o manutenzione e per procedere al successivo varo in acqua.

Il software che svilupperò si propone di aiutare la società nell'individuare il prezzo di rivendita e nel trovare, attraverso un algoritmo ricorsivo, una soluzione ottimale sulla scelta delle barche da acquistare con il budget prefissato al fine di massimizzare il guadagno atteso dalla rivendita, rispettando i vincoli selezionati.

Il prezzo per la rivendita viene stabilito dalla società che ipotizza di ottenere un determinato profitto a partire da una serie di analisi su ogni singola barca.

Partendo, innanzitutto, dalle condizioni (nuovo o usato) calcolerò una percentuale fissa sul costo di acquisto delle singole barche al quale si aggiunge il costo di eventuali montaggi e/o lavorazioni per procedere al varo; quest'ultimo avrà un costo variabile in base alla categoria di appartenenza, calcolato secondo le dimensioni della barca (esempio da 7 a 15 metri, da 15 a 22 metri, ecc.).

Inoltre, sarà possibile selezionare dei filtri per rendere le proposte di acquisto più adatte alle esigenze della società.

Sarà prevista anche la possibilità di includere nella soluzione proposta una o più barche che la società vorrebbe necessariamente acquistare.

1.4 Descrizione della rilevanza gestionale del problema

La rilevanza del problema, da un punto di vista gestionale, riguarda la possibilità di individuare, attraverso un algoritmo ricorsivo, una soluzione ottimale che massimizzi il profitto tenendo conto dei vincoli espressi e di fornirla alla società in modo da finalizzare un acquisto che rispetti gli standard richiesti.

Nel caso preso in esame, infatti, l'applicazione non si limita a fornire una soluzione di acquisto che sia robusta e risponda in maniera ottimale al problema matematico, ma trova la sua rilevanza gestionale proprio nell'elasticità dell'algoritmo che permette, agendo sui vincoli, di soddisfare le esigenze dell'azienda.

1.5 Descrizione del data-set per la valutazione

Il data-set che verrà utilizzato è stato preso dal sito Kaggle:

<https://www.kaggle.com/datasets/karthikbhandary2/boat-sales>.

In particolare, i dati riguardano una serie di informazioni su imbarcazioni da diporto in vendita in tutto il mondo.

Il data-set verrà modificato in modo da selezionare esclusivamente le barche in vendita in Italia e ripulito da colonne superflue, doppioni e righe errate o non del tutto complete.

Una volta ripulito conterrà informazioni su:

- Prezzo di vendita in euro;
- Tipologia (ad esempio Sport Boat, Mega Yacht...);
- Manifattura;
- Condizioni (nuovo o usato);
- Anno di produzione;
- Lunghezza;
- Larghezza;
- Luogo di provenienza.

1.6 Descrizione preliminare degli algoritmi coinvolti

Innanzitutto, il software sarà in grado di fornire una lista di barche in vendita secondo una serie di condizioni selezionate dall'utente con un algoritmo di ricerca. Sarà possibile esprimere delle preferenze per quanto riguarda la condizione (nuovo o usato), il range di lunghezza desiderato, la tipologia di imbarcazione, l'anno minimo di produzione e il budget disponibile.

Inoltre, l'algoritmo è in grado di completare la ricerca anche nel caso in cui non siano espresse particolari esigenze con l'obbligo di inserire il budget massimo a disposizione. Dopodiché, l'aspetto principale dell'applicazione sarà un algoritmo ricorsivo che fornisce una soluzione ottimale sull'acquisto che l'azienda può effettuare, tradotto come potenziale massimo profitto realizzabile dalla successiva vendita delle barche. L'algoritmo partirà dalla lista delle barche compatibili con le necessità che l'utente ha selezionato in input (come descritto sopra nell'algoritmo di ricerca) e proporrà un pacchetto di barche acquistabili, con il budget selezionato, in modo da massimizzare il profitto atteso. Il risultato ottenuto, ovvero l'elenco di barche che massimizzerebbe il guadagno, potrà essere confermato e, di conseguenza, le medesime imbarcazioni non saranno più disponibili nel caso si voglia continuare con la ricerca di un ulteriore acquisto. Sarà possibile inserire nella soluzione una o più barche considerate come acquisto certo rispettando sempre il vincolo del budget prefissato.

Inoltre, l'algoritmo sarà in grado di proporre, oltre alla soluzione sopra descritta, una proposta di acquisto che supera di poco il budget prefissato per consentire all'utente di avere un quadro più completo e valutare, in caso, un possibile aumento del budget al fine di procedere con quella soluzione.

1.7 Descrizione preliminare delle funzionalità previste per l'applicazione

L'utente avrà a disposizione sulla parte superiore della schermata una serie di vincoli da impostare riguardo le caratteristiche delle barche all'interno del data-set. Un primo bottone sarà in grado di effettuare una ricerca delle barche con le caratteristiche definite prima e stamparle in modo tale da permettere all'utente di visualizzare il quadro completo delle possibilità di acquisto. Qualora, durante la ricerca, si individuasse una determinata barca da acquistare necessariamente, vi sarà la possibilità di inserirla, attraverso il suo id, per poi procedere con il calcolo della soluzione ottimale.

Un secondo bottone permetterà di generare una soluzione che massimizza l'investimento con il budget prefissato dall'utente e mostrerà l'elenco delle barche proposte e il guadagno atteso di ogni singola barca.

Sarà poi possibile, tramite altri due bottoni appositi, confermare o rifiutare la soluzione proposta. Nel primo caso verrà mostrato l'elenco degli acquisti finora effettuati, il capitale investito per tutti gli acquisti e il guadagno atteso totale e per ogni singola barca. Nel secondo, la soluzione verrà semplicemente scartata e inviterà l'utente a procedere con una nuova ricerca.

In entrambi, il programma resetterà i filtri selezionati.

Qualora, alla pressione del bottone che genera la soluzione ottimale, l'algoritmo trovi una soluzione che supera di poco il budget prefissato, il programma la mostrerà all'utente; in caso quest'ultimo optasse per la seconda soluzione, sarà possibile modificare il budget manualmente dall'apposita sezione ed effettuare nuovamente la ricerca per poi procedere alla conferma dell'acquisto.

Vi sarà, infine, la possibilità di resettare tutti gli acquisti già effettuati per ricominciare con una nuova ricerca.

Capitolo 2: Descrizione dettagliata del problema

2.1 Obiettivo

Uno dei principali obiettivi di un Ingegnere Gestionale è quello di ottimizzare le scelte aziendali in modo da allocare le giuste risorse per massimizzare i profitti.

L'applicazione implementata si propone di offrire un sostegno prezioso in un contesto aziendale specializzato nella compravendita di barche. Il software è stato concepito considerando l'importanza di fare scelte oculate per ottenere la soluzione più vantaggiosa, rispettando un budget prefissato, al fine di massimizzare il guadagno atteso dalla successiva rivendita delle imbarcazioni.

Quest'ultima comporta un significativo rischio per le società, le quali investono considerevoli somme di denaro con l'aspettativa di trovare degli acquirenti. Per tale ragione, l'applicazione può sostenere l'azienda e l'ingegnere incaricato nella scelta di una soluzione ottimale.

Per fare ciò, il software sviluppato seleziona, innanzitutto, una lista assortita di imbarcazioni ricavate da quelle già presenti nell'intero data-set le quali, però, rispetteranno i vincoli imposti dall'ingegnere in input. Tali preferenze permettono di eliminare a priori le barche che, con molta probabilità, non hanno grande mercato.

A questo punto l'applicazione effettua un'analisi su ogni singola imbarcazione e stima un ipotetico profitto atteso, ricavato dalla rivendita e dal varo in acqua.

La stima si basa su alcuni parametri intrinsechi alle imbarcazioni stesse, come ad esempio le condizioni generali (nuovo o usato), la lunghezza e l'anno di produzione.

Nel dettaglio, per quanto riguarda le CONDIZIONI, è stato considerato una maggiorazione del 5% sul nuovo e del 10% sull'usato rispetto al prezzo di acquisto.

In riferimento alla LUNGHEZZA vi sono delle norme di legge precise che riguardano la navigazione da diporto, le quali presentano precise definizioni e caratteristiche.

Con il termine Unità da diporto si considera qualunque tipo di mezzo destinato alla navigazione, ma vengono differenziati in base alla lunghezza e per tale ragione i costi di gestione sono altamente variabili. Ad esempio, il varo in acqua o le operazioni necessarie per rendere il mezzo a norma di legge (targa, dotazioni di sicurezza ecc.) presuppongono

dei costi maggiori per la società e di conseguenza un proporzionale aumento del prezzo di vendita. Ho riassunto realisticamente le principali differenze ed effettuato una stima sul sovraprezzo in funzione delle precedenti considerazioni:

- Se l'unità è inferiore ai 10 mt prende il nome di Natante e ho attribuito un aumento di 350€ sul profitto;
- Se è compresa tra i 10 e i 24 mt vengono chiamate Imbarcazioni; in questo caso ho ulteriormente suddiviso in tre scaglioni, dai 10 ai 15mt vi è un aumento di 500€, dai 15 ai 20mt un aumento di 750€ e dai 20 ai 24mt un aumento di 1.000€;
- Se l'unità supera i 24 mt di lunghezza viene identificata come Nave e ho attribuito un aumento di 2.000€ e un ulteriore 0,1% sul prezzo di vendita.

Per facilitare la comprensione all'interno della relazione tutti i tipi di unità da diporto verranno chiamate imbarcazione o semplicemente barca.

Ultima condizione rientra nel vincolo inerente all'ANNO DI PRODUZIONE dell'unità e anche in questo caso ho suddiviso in cinque sottogruppi le imbarcazioni. Quelle prodotte dopo il 2020 avranno un aumento sul profitto di 200€, quelle dopo il 2010 un aumento di 150€, quelle dopo il 2000 un aumento di 100€ e quelle dopo il 1990 un aumento di 50€. Infine, tutte le barche prodotte da oltre 30 anni, ovvero prima del 1990, vengono considerate imbarcazioni d'epoca e questo inevitabilmente prevede un aumento di prezzo di 100€.

Tali parametri, fissati nell'algoritmo, dovrebbero essere costantemente aggiornati, riadattati e modificati per rimanere sempre in linea con l'evoluzione del mercato, per far si che, con il passare del tempo, l'applicazione si dimostri il più possibile accurata nell'elaborazione delle sue stime.

A questo punto, il software è a conoscenza del budget massimo spendibile e del guadagno atteso, dunque avvia l'algoritmo ricorsivo che permette di ricercare, tra le diverse possibili combinazioni, quella che massimizza il potenziale profitto.

L'applicazione presenterà la soluzione all'utente e darà la possibilità di procedere confermando l'acquisto. Inoltre, sarà in grado di memorizzare quest'ultimo per

proseguire con un'altra ricerca, eliminando quindi dalla lista le barche già acquistate in precedenza.

Nella ricerca della soluzione sono state implementate ulteriori integrazioni per rendere l'applicazione più ricca e funzionale. In particolare, l'utente ha la possibilità di visualizzare la lista filtrata e, nel caso in cui ci fosse una barca particolarmente interessante, potrà inserirla nella soluzione, utilizzando il suo ID, in modo da considerarla tra le opzioni di acquisto nella lista finale.

Infine, un'ulteriore integrazione consente di trovare una soluzione che possa permettere un guadagno maggiore nel caso in cui l'acquisto di un set di barche superi di poco il budget disponibile. In questo caso l'applicazione presenterà entrambe le soluzioni all'utente che avrà così la possibilità di confermare l'acquisto di una delle due soluzioni proposte o, eventualmente, di annullare l'operazione.

Naturalmente bisogna considerare che il software non fornisce una soluzione in grado di garantire un guadagno certo poiché banalmente possono esserci situazioni in cui l'imbarcazione viene sopravvalutata o sottostimata, oppure perché, come nel caso di imbarcazioni usate, è necessario effettuare dei lavori di manutenzione o di restauro più costosi. Tra le caratteristiche distintive dell'applicazione, emerge la capacità di offrire, con estrema celerità e precisione, un supporto pratico e concreto nell'ottimizzazione dell'allocazione delle risorse economiche. Grazie a questa funzionalità, gli utenti possono contare su un valido alleato nel processo decisionale, potendo così massimizzare il rendimento dei loro investimenti e ottenere risultati più soddisfacenti nel mercato delle barche.

Capitolo 3: Descrizione del data-set

3.1 Descrizione

Il data-set utilizzato per questa applicazione è stato preso dal sito Kaggle:
<https://www.kaggle.com/datasets/karthikbhandary2/boat-sales>

Il data-set originale comprende un totale di circa 3000 righe e 10 colonne con annunci di imbarcazioni da diporto in vendita in tutto il mondo e contiene dettagli sulle barche come l'anno di costruzione, il tipo di barca, il prezzo e molto altro.

Il data-set è stato modificato in modo da selezionare tutti gli annunci di vendita in Italia e successivamente è stato ripulito, eliminando tutte le informazioni superflue, doppiioni e righe errate o non del tutto complete per alleggerire il data-set e rendere più rapida la ricerca di una soluzione ottimale da parte dell'algoritmo.

Le informazioni contenute nel data-set finale sono di seguito descritte nel diagramma ER:

BOAT	TYPE
Id	INT
Prezzo	INT
Tipologia	VARCHAR
Manifattura	VARCHAR
Condizioni	VARCHAR
Anno	INT
Lunghezza	DOUBLE
Larghezza	DOUBLE
Luogo	VARCHAR

La tabella "boat" contiene tutte le informazioni sulle barche in vendita a partire dalle quali verrà proposta una soluzione:

- Id: identificativo univoco della barca, progressivo, int;
- Prezzo: costo di vendita in euro, int;
- Tipologia: il tipo di imbarcazione (ad esempio Sport Boat, Mega Yacht...), string;
- Manifattura: cantiere nautico dell'imbarcazione, string;
- Condizioni: può essere nuova o usata, string;
- Anno: quello di produzione, int;
- Lunghezza: dimensione in metri della barca, double;
- Larghezza: dimensione in metri della barca, double;
- Luogo: dove si trova l'imbarcazione, string.

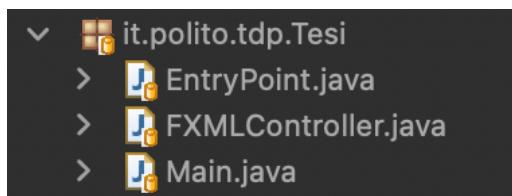
Capitolo 4: Descrizione delle strutture dati e degli algoritmi utilizzati

4.1 Strutture dati

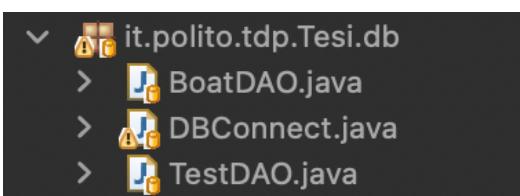
Il software è stato realizzato in linguaggio Java seguendo: il pattern MVC (Model View Controller) per quanto riguarda la parte logica/applicativa, il pattern DAO (Data Access Object) per l'interazione con la struttura dati e infine per realizzare l'interfaccia grafica in JavaFX è stato utilizzato il software SceneBuilder.

Il progetto è costituito dai seguenti tre package:

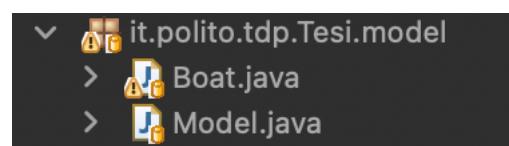
- Package `it.polito.tdp.Tesi`: qui sono presenti tre classi necessarie per garantire l'interazione con l'utente. Nella classe “Main” viene eseguita l'applicazione, mentre nell'Entry Point e nell'FXMLController sono definiti i metodi per garantire la corretta interazione tra l'utente e la logica applicativa del programma.



- Package `it.polito.tdp.Tesi.db`: con questo package il programma è in grado di accedere alla struttura dati ed effettuare le query SQL. Sono presenti la classe “DBConnect”, che implementa la connessione al database; la classe BoatDAO che contiene i metodi necessari per ottenere tutte le informazioni sulle imbarcazioni; infine, la classe “TestDAO” con cui è stato possibile testare i metodi durante l'implementazione del software.



- Package `it.polito.tdp.Tesi.model`: in questo package sono presenti due classi che permettono di effettuare le operazioni sugli oggetti all'interno del programma, ovvero le barche (classe chiamata "Boat"). All'interno della classe "Model" sono implementati tutti i metodi per estrarre i dati dal database e di implementare l'algoritmo ricorsivo che permette all'applicazione di generare una lista di barche che l'utente può eventualmente acquistare



4.2 Principali algoritmi

L'obiettivo principale dell'applicazione è la ricerca di una soluzione ottimale per massimizzare il profitto dato dalla compravendita di imbarcazioni. Per fare ciò il software inizia la ricerca partendo da una lista di imbarcazioni rispettando gli eventuali vincoli imposti.

L'algoritmo che ci permette di trovare la suddetta lista si trova all'interno della classe BoatDAO che, connettendosi al data-set "boat.sql", ci consente di caricare all'interno di una "List<Boat>" tutte le imbarcazioni ricercate.

```
    public List<Boat> listBoatSelected(int prezzo, List<String> condì, double Lmin, double Lmax, String tipologia, int Amin, int Amax){
        String sql = "SELECT * "
                    + "FROM boat b "
                    + "WHERE b.Lunghezza>=? AND b.Lunghezza<=? "
                    + "AND b.Prezzo<=? "
                    + "AND b.Anno>=? AND b.Anno<=? ";
        if(condì.size()>1) {
            sql=sql+"AND (b.Condizione=? OR b.Condizione=?)";
        }else {
            sql=sql+"AND b.Condizione=?";
        }
        if(!tipologia.equals("Qualsiasi")) {
            sql=sql+" AND b.Tipologia LIKE '%"+tipologia+"%'";
        }
        List<Boat> result = new ArrayList<Boat>();
        Connection conn = DBConnect.getConnection();
        try {
            PreparedStatement st = conn.prepareStatement(sql);
            st.setDouble(1, Lmin);
            st.setDouble(2, Lmax);
            st.setInt(3, prezzo);
            st.setInt(4, Amin);
            st.setInt(5, Amax);
            int j=6;
            for(int i=0;i<condì.size();i++) {
                st.setString(j,condì.get(i));
                j++;
            }
            ResultSet res = st.executeQuery();
            while (res.next()) {
                Boat b = new Boat(res.getInt("Id"), res.getInt("Prezzo"),res.getString("Tipologia"),res.getString("Manifattura"),
                                  res.getString("Condizione"), res.getInt("Anno"), res.getDouble("Lunghezza"),
                                  res.getDouble("Larghezza"), res.getString("Luogo"));
                result.add(b);
            }
            conn.close();
            return result;
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
```

Sempre all'interno della classe BoatDAO vi sono altri algoritmi che grazie all'interrogazione del database ci permettono di trovare tutte le differenti categorie presenti per riempire la ComboBox "TIPOLOGIA".

```
    public List<String> listAllTipologia() {
        String sql = "SELECT DISTINCT Tipologia "
                    + "FROM boat "
                    + "WHERE Tipologia NOT LIKE '%,%'";
        List<String> result = new ArrayList<String>();
        Connection conn = DBConnect.getConnection();
        try {
            PreparedStatement st = conn.prepareStatement(sql);
            ResultSet res = st.executeQuery();
            while (res.next()) {
                String b = res.getString("Tipologia");
                result.add(b);
            }
            conn.close();
            return result;
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
```

Inoltre, sempre all'interno della classe BoatDAO i seguenti algoritmi permettono di ricavare i valori minimi e massimi di Anno e Lunghezza per la ricerca nel caso in cui non siano stati selezionati dall'utente.

```

public double getLmin() {
    String sql = "SELECT MIN(Lunghezza) AS Lmin "
        + "FROM boat";
    Connection conn = DBConnect.getConnection();
    try {
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet res = st.executeQuery();
        double b=0;
        while (res.next()) {
            b = res.getDouble("Lmin");
        }
        conn.close();
        return b;
    } catch (SQLException e) {
        e.printStackTrace();
        return -1;
    }
}

public double getLmax() {
    String sql = "SELECT MAX(Lunghezza) AS Lmax "
        + "FROM boat";
    Connection conn = DBConnect.getConnection();
    try {
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet res = st.executeQuery();
        double b=0;
        while (res.next()) {
            b = res.getDouble("Lmax");
        }
        conn.close();
        return b;
    } catch (SQLException e) {
        e.printStackTrace();
        return -1;
    }
}

```

```

public double getLmin() {
    String sql = "SELECT MIN(Lunghezza) AS Lmin "
        + "FROM boat";
    Connection conn = DBConnect.getConnection();
    try {
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet res = st.executeQuery();
        double b=0;
        while (res.next()) {
            b = res.getDouble("Lmin");
        }
        conn.close();
        return b;
    } catch (SQLException e) {
        e.printStackTrace();
        return -1;
    }
}

public double getLmax() {
    String sql = "SELECT MAX(Lunghezza) AS Lmax "
        + "FROM boat";
    Connection conn = DBConnect.getConnection();
    try {
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet res = st.executeQuery();
        double b=0;
        while (res.next()) {
            b = res.getDouble("Lmax");
        }
        conn.close();
        return b;
    } catch (SQLException e) {
        e.printStackTrace();
        return -1;
    }
}

```

Una volta ricavata la lista su cui lavorerà l'algoritmo ricorsivo, ci spostiamo nella classe Model e possiamo così analizzare la ricorsione. Grazie al seguente metodo vengono inizializzate tutte le strutture dati necessarie per memorizzare i risultati e viene lanciato l'algoritmo per la ricerca della migliore proposta trovata.

```

    public List<Boat> barche(List<Boat> listaBarche, int budget) {
        best=new ArrayList<>();
        best2=new ArrayList<>();
        parziale=new ArrayList<>();
        prezzoP=0;
        barche_ricorsiva(parziale, budget, listaBarche);
        if(this.calcola_guadagno(best)>=this.calcola_guadagno(best2)) {
            best2=new ArrayList<>();
        }
        return best;
    }
}

```

Come possiamo notare nella schermata seguente l'algoritmo ricorsivo va alla ricerca di due soluzioni per massimizzare il profitto atteso. La prima soluzione di barche viene inserita nella lista “best” e rispetta a pieno il vincolo del budget imposto. Nella seconda soluzione, trovata dall'algoritmo e inserita nella lista “best2”, si è tenuto conto di un aumento del x% sul valore del

budget imposto in modo da presentare all'utente questa seconda possibilità per poter valutare un aumento del budget. Ovviamente, la seconda soluzione verrà presentata esclusivamente nel caso in cui il profitto atteso sia superiore rispetto alla prima soluzione.

```

private void barche_ricorsiva(List<Boat> parziale, int budget, List<Boat> listaBarche ) {
    if(parziale.size()>0 ) { //caso terminale
        int disponibile=budget-this.prezzoP;
        int disponibile2=budget+(budget/10)-this.prezzoP;
        if(this.calcola_guadagno(parziale)>this.calcola_guadagno(best)) {
            if(disponibile>=0) {
                if((disponibilita(disponibile,listaBarche) || disponibile==0 )) {
                    this.best=new ArrayList<>(parziale);
                    return;
                }
            }else if((disponibilita2(disponibile2,listaBarche) || disponibile2==0 )) {
                if(disponibilita2(disponibile2,listaBarche)) {
                    this.best2=new ArrayList<>(parziale);
                    return;
                }
            }
        }
    }

    for(Boat b:listaBarche) {
        if(budget>=(this.prezzoP+b.prezzo) && !parziale.contains(b)) {
            parziale.add(b);
            this.prezzoP+=b.prezzo;
            this.barche_ricorsiva(parziale, budget, listaBarche );
            parziale.remove(b);
            this.prezzoP-=b.prezzo;
        }else if((budget+(budget/10))>=(this.prezzoP+b.prezzo) && !parziale.contains(b)) {
            parziale.add(b);
            this.prezzoP+=b.prezzo;
            this.barche_ricorsiva(parziale, budget, listaBarche );
            parziale.remove(b);
            this.prezzoP-=b.prezzo;
        }
    }
}

```

Per facilitare la ricorsione sono stati implementati due metodi che forniscono all'algoritmo una lista aggiornata sui cui effettuare la ricerca, partendo da quella iniziale ed eliminando via via quelle inserite nella soluzione.

```

public boolean disponibilita(int disponibili,List<Boat> listaBarche) {
    ArrayList<Boat> lb=new ArrayList<>(listaBarche);
    Collections.sort(lb);
    for(Boat b:listaBarche){
        if(b.prezzo<=disponibili && !parziale.contains(b)){
            return false;
        }
    }
    return true;
}

public boolean disponibilita2(int disponib,List<Boat> listaBarche) {
    ArrayList<Boat> lb=new ArrayList<>(listaBarche);
    Collections.sort(lb);
    for(Boat b:listaBarche){
        if(b.prezzo<=disponib && !parziale.contains(b)){
            return false;
        }
    }
    return true;
}

```

Adesso analizziamo i principali metodi utilizzati nel FXMLController dove, appunto, ci si occupa della gestione dell'output da presentare nella TextArea. In particolare, un primo metodo, chiamato

“ActionLista”, si occupa di gestire il bottone “LISTA” che, in maniera dinamica, modifica il suo nome e le sue funzioni.

Grazie ad uno “switch case” possiamo valutare tutti i casi in cui il bottone si trasforma. In un primo caso, “LISTA”, il metodo presenta all’utente la lista completa di imbarcazioni che rispettano i vincoli selezionati. Il caso “ANNULLA”, si presenta in conseguenza alla generazione della soluzione, il bottone consente così di non confermare la soluzione generata dall’algoritmo ricorsivo. Sempre successivamente alla generazione della soluzione, ci si può ritrovare nel caso in cui l’algoritmo abbia trovato una seconda soluzione da valutare e modifica il bottone nel caso “CONFERMA 2° SOL”, consentendo all’utente di confermare l’acquisto di quest’ultima.

Infine, una volta confermato l’acquisto, il bottone consente di resettare il programma per ricominciare (caso “RESET”).

```
    @FXML
    void ActionLista(ActionEvent event) {
        initialize();
        switch(Lista.getText()) {
            case "LISTA":
                if(setFiltri()) {
                    if (!lBoatSel.size() != 0) {
                        TxtResult.appendText("LISTA\nLe barche che rispettano i criteri sono:\n");
                        for (Boat b : lBoatSel) {
                            TxtResult.appendText(b.toString() + "\n");
                        }
                    } else {
                        TxtResult.appendText("LISTA\nNon ci sono barche che rispettano i criteri.\nModificare i parametri inseriti"
                                + " o effettuare una nuova ricerca premendo sul bottone reset filtri");
                        Soluzione.setDisable(true);
                    }
                }
                break;
            case "ANNULLA":
                TxtResult.appendText("Acquisto annullato\nProcedere con una nuova soluzione");
                Soluzione.setText("GERA SOLUZIONE");
                Lista.setText("LISTA");
                BtnId.setDisable(false);
                break;
            case "CONFERMA 2° SOL":
                TxtResult.appendText("Acquisto confermato con successo!\n\nLista barche acquistate fin'ora:\n");
                Allbest.addAll(best2);
                for (Boat b : Allbest) {
                    TxtResult.appendText(b.toString() + "\n");
                }
                TxtResult.appendText("\n\nProcedere con una nuova ricerca con il budget rimanente, modificare i filtri/budget o ricominciare\n");
                budg = model.calcola_prezzo(best2);
                Budget.setText(Integer.toString(budg));
                Soluzione.setText("NUOVA RICERCA");
                Lista.setText("RESET");
                ResetFiltre.setText("RESET FILTRI");
                break;
            default:
                ActionResetFiltre(event);
                initialize();
                Allbest= new ArrayList<Boat>();
                Soluzione.setText("GERA SOLUZIONE");
                Lista.setText("LISTA");
                ResetFiltre.setText("RESET FILTRI");
                BtnId.setDisable(false);
        }
    }
```

In maniera analoga, il metodo “ActionSoluzione” gestisce, sempre grazie ad uno switch case, i vari casi in cui il bottone “GERA SOLUZIONE” può variare.

Come si può immaginare il primo caso, “GERA SOLUZIONE”, consente di ricercare una soluzione per massimizzare il profitto, il software è in grado di presentare all’utente la lista delle imbarcazioni

trovate, il budget speso e il guadagno atteso. Inoltre, propone, nel caso in cui esista, una seconda soluzione con un aumento di budget.

Una volta generata e proposta la soluzione il bottone cambia nome in “CONFERMA” (secondo case), così l’utente è in grado di procedere all’acquisto e salva le informazioni in modo da poter continuare la ricerca di altre soluzioni. Nel momento in cui si conferma è possibile visualizzare lo storico degli acquisti e il bottone ti consente di effettuare una nuova ricerca salvando gli acquisti effettuati fino a quel momento (case “NUOVA RICERCA”).

```
    @FXML
    void ActionSoluzione(ActionEvent event) {
        initialize();
        switch(Soluzione.getText()) {
            case "GENERA SOLUZIONE":
                best= new ArrayList<Boat>();
                best2= new ArrayList<Boat>();
                if(setFiltri()){
                    if(insertId!=0) {
                        budg=insertBoat.getPrezzo();
                        lBoatSel.remove(insertBoat);
                        best.add(insertBoat);
                        best2.add(insertBoat);
                    }
                    best.addAll(this.model.barche(this.lBoatSel, budg)); // RICERCA SOLUZIONE
                    insertId=0;
                    insertBoat=null;
                    if(best.size() != 0) {
                        TxtResult.appendText("SOLUZIONE\nLa soluzione ottimale è:\n");
                        for (Boat b : best) {
                            TxtResult.appendText(b.toString() + "\n");
                        }
                        TxtResult.appendText("\nBudget speso: "+model.calcola_prezzo(best)+"\nGuadagno totale atteso: "+model.calcola_guadagno(best));
                    } else {
                        TxtResult.appendText("SOLUZIONE\nNon ci sono barche che rispettano i criteri per una soluzione ottima.\n"
                                + "\nModificare i parametri inseriti o effettuare una nuova ricerca ");
                        Soluzione.setEditable(true);
                    }
                    Soluzione.setText("CONFERMA");
                    Lista.setText("ANNULLA");
                    if(!model.secondasol().isEmpty()) { // NEL CASO IN CUI CI SIA UNA SECONDA SOLUZIONE
                        best2.addAll(model.secondasol());
                        TxtResult.appendText("\nIn alternativa la soluzione ottima potrebbe essere:\n");
                        for (Boat b : best2) {
                            TxtResult.appendText(b.toString() + "\n");
                        }
                        int diff = model.calcola_prezzo(model.secondasol()) - budg;
                        TxtResult.appendText("\n\nSpendo " + diff + " in più rispetto al budget.");
                        TxtResult.appendText("\nGuadagno totale atteso dalla 2° soluzione: " + this.model.calcola_guadagno(model.secondasol()));
                        Lista.setText("CONFERMA 2° SOL");
                        ResetFiltri.setText("ANNULLA");
                    }
                }
                break;
            case "CONFERMA":
                TxtResult.appendText("Acquisto confermato con successo!\n\nLista barche acquistate fin'ora:\n");
                Allbest.addAll(best);
                for (Boat b : Allbest) {
                    TxtResult.appendText(b.toString() + "\n");
                }
                TxtResult.appendText("\n\nProcedere con una nuova ricerca con il budget rimanente, modificare i filtri/budget o ricominciare\n");
                budg-=model.calcola_prezzo(best);
                Budget.setText(Integer.toString(budg));
                Soluzione.setText("NUOVA RICERCA");
                Lista.setText("RESET");
                ResetFiltri.setText("RESET FILTRI");
                break;
            default:
                Soluzione.setText("GENERA SOLUZIONE");
                Lista.setText("LISTA");
                ResetFiltri.setText("RESET FILTRI");
                BtnId.setEditable(false);
                ActionLista(event);
        }
    }
```

Proseguendo all'interno del FXMLController, vi è il seguente metodo che si occupa di controllare se l'ID inserito dall'utente per essere compreso nella soluzione è valido. In caso affermativo, aggiorna la lista della soluzione e il relativo budget e, comunica la corretta esecuzione. In caso contrario, segnala tramite un messaggio l'errore e invita a inserire nuovamente un ID valido.

```
@FXML
void ActionBtnId(ActionEvent event) {
    try {
        insertId = Integer.parseInt(Id.getText());
        for(Boat b: this.lBoatSal) {
            if(b.getId()==insertId) {
                TxtResult.appendText("\nBarca inserita correttamente nella lista delle soluzioni\n");
                insertBoat=b;
                Id.clear();
                BtnId.setDisable(true);
                return;
            }
        }
        TxtResult.appendText("Inserire un ID corretto \n");
    } catch (NumberFormatException e) {
        TxtResult.appendText("Inserire un ID corretto \n");
    }
}
```

Per concludere questo capitolo vi è un ultimo metodo da considerare fondamentale per la corretta risoluzione del problema. Il metodo setGuadagno che consente di ipotizzare, secondo alcune considerazioni frutto di ricerca ed esperienza, il possibile valore di mercato di ogni singola imbarcazione. Il seguente metodo potrebbe e dovrebbe essere in continua evoluzione in modo da adattare e correggere le considerazioni effettuate per rendere il valore “guadagno atteso” il più realistico possibile.

```
public void setGuadagno() {
    //condizione
    if(condizione.equals("Nuovo")) {
        this.guadagno=(prezzo*5/100);
    }else {
        this.guadagno=(prezzo*10/100);
    }
    //lunghezza
    if(this.lunghezza<10) { // natanti
        this.guadagno+=200;
    }else if(this.lunghezza<15){ // barche tigraate
        this.guadagno+=350;
    }else if(this.lunghezza<20){
        this.guadagno+=500;
    }else if(this.lunghezza<24){
        this.guadagno+=600;
    }else{ // navi
        this.guadagno+=800;
        this.guadagno+=(prezzo*1/1000);
    }
    //anno
    if(this.anno>2020) {
        this.guadagno+=200;
    }else if(this.anno>2010){
        this.guadagno+=150;
    }else if(this.anno>2000){
        this.guadagno+=100;
    }else if(this.anno>1990){
        this.guadagno+=50;
    }else{
        this.guadagno+=100;
    }
}
```

Capitolo 5: Diagramma delle classi delle principali parti dell'applicazione

5.1 Classi dell'applicazione

Gli algoritmi principali di questa applicazione sono contenuti nelle classi *Model*, *Boat*, *BoatDAO* e *FXMLController* delle quali vengono riportati i seguenti diagrammi:

Diagramma del Model:

```
Model.java
Model
    • best
    • best2
    • parziale
    • prezzob
    • prezzob2
    • prezzop
    • prezzop2
    • barche(List<Boat>, int) : List<Boat>
    • barche_ricorsiva(List<Boat>, int, List<Boat>) : void
    • calcola_guadagno(List<Boat>) : int
    • calcola_prezzo(List<Boat>) : int
    • disponibilita(int, List<Boat>) : boolean
    • disponibilita2(int, List<Boat>) : boolean
    • secondasol() : List<Boat>
```

Diagramma classe Boat:

```
Boat.java
Boat
    ▲ anno
    ▲ condizione
    ▲ guadagno
    ▲ id
    ▲ larghezza
    ▲ lunghezza
    ▲ Luogo
    ▲ manifattura
    ▲ prezzo
    ▲ tipologia
    • Boat(Integer, Integer, String, String, String, Integer, Double, Double, String)
    • compareTo(Boat) : int
    • equals(Object) : boolean
    • getAnno() : Integer
    • getCondizione() : String
    • getGuadagno() : Integer
    • getId() : Integer
    • getLarghezza() : Double
    • getLunghezza() : Double
    • getLuogo() : String
    • getManifattura() : String
    • getPrezzo() : Integer
    • getTipologia() : String
    • hashCode() : int
    • setAnno(Integer) : void
    • setCondizione(String) : void
    • setGuadagno() : void
    • setId(Integer) : void
    • setLarghezza(Double) : void
    • setLunghezza(Double) : void
    • setLuogo(String) : void
    • setManifattura(String) : void
    • setPrezzo(Integer) : void
    • setTipologia(String) : void
    • toString() : String
```

Diagramma classe BoatDAO:

```
BoatDAO.java
  +-- BoatDAO
    +-- getAmax() : int
    +-- getAmin() : int
    +-- getLmax() : double
    +-- getLmin() : double
    +-- listAllBoat() : List<Boat>
    +-- listAllTipologia() : List<String>
    +-- listBoatSelected(int, List<String>, double, double, String, int, int) : List<Boat>
```

Diagramma classeFXMLController:

```
FXMLController.java
  +-- FXMLController
    +-- Allbest
    +-- AnnoMax
    +-- AnnoMin
    +-- best
    +-- best2
    +-- BtnId
    +-- budg
    +-- Budget
    +-- Condizione
    +-- dao
    +-- Id
    +-- insertBoat
    +-- insertId
    +-- IBoatSel
    +-- Lista
    +-- location
    +-- LunghezzaMax
    +-- LunghezzaMin
    +-- model
    +-- ResetFiltri
    +-- resources
    +-- Soluzione
    +-- Tipologia
    +-- TxtResult
    +-- ActionBtnId(ActionEvent) : void
    +-- ActionLista(ActionEvent) : void
    +-- ActionResetFiltri(ActionEvent) : void
    +-- ActionSoluzione(ActionEvent) : void
    +-- initialize() : void
    +-- setComboItems() : void
    +-- setFiltri() : boolean
    +-- setModel(Model) : void
```

Capitolo 6: Interfaccia dell'applicazione

6.1 Funzionamento dell'applicazione

L'applicazione si presenta in un'unica finestra come segue:

The screenshot shows the user interface of the BOAT SITE application. At the top center, the title "BOAT SITE" is displayed in blue. Below it, the word "FILTRI" is written in blue, indicating the section for filtering search criteria. The filters are organized into five groups: "CONDIZIONE" (with a dropdown menu), "LUNGHEZZA" (with "min" and "max" input fields), "TIPOLOGIA" (with a dropdown menu), "ANNO" (with "min" and "max" input fields), and "BUDGET" (with an empty input field). A "RESET FILTRI" button is located at the bottom right of the filter section. At the bottom of the window, there are several buttons: "ID" (input field), "INSERT ID" (button), "LISTA" (button), and "GENERA SOLUZIONE" (button).

L'interfaccia è stata realizzata in JavaFX grazie all'aiuto del software SceneBuilder, ed è stata suddivisa in tre sezioni:

La sezione superiore è dedicata alla selezione dei filtri da impostare per la ricerca.

Il primo valore che deve essere inserito è la ComboBox "CONDIZIONE" nella quale l'utente deve obbligatoriamente selezionare lo stato dell'imbarcazione scegliendo tra "nuovo", "usato" e, nel caso in cui non abbia una preferenza, la voce "qualsiasi".

Il secondo vincolo che può essere inserito nelle due TextField riguarda un range (min e max) di "LUNGHEZZA" espressa in metri.

Nel terzo filtro vi è la ComboBox "TIPOLOGIA", nella quale sono presenti ben 21 diverse categorie di imbarcazioni, come ad esempio Sport Boat, sempre con la possibilità di non esprimere alcuna preferenza, selezionando dal menù a tendina l'opzione "qualsiasi".

Il quarto parametro ti permette di indicare l'ANNO di produzione della barca, inserendo eventualmente nelle TextField un valore minimo e massimo.

Infine, l'utente deve impostare un BUDGET massimo per la ricerca, inoltre sulla destra si trova un pulsante "RESET FILTRI" che ti permette di azzerare i valori finora inseriti.

Il software è in grado di gestire eventuali errori di compilazione da parte dell'utente, nello specifico farà apparire un messaggio di errore sia nel caso in cui non venga selezionato alcun valore nei menù a tendina, sia se venga inserito un valore numerico non conforme alle sezioni lunghezza, anno e budget.

Nel caso in cui non si voglia selezionare alcun range di lunghezza e anno, il software effettuerà invece la ricerca includendo tutti i valori disponibili.

Nella sezione centrale vi è una TextArea all'interno della quale il programma presenta all'utente possibili messaggi di errore, la lista delle imbarcazioni selezionate e i risultati della ricorsione ricavati dall'esecuzione dell'algoritmo.

Nella sezione inferiore possiamo trovare sulla sinistra una TextField nella quale è possibile immettere l'ID di una imbarcazione da inserire obbligatoriamente nella soluzione ricercata, con un apposito bottone che ci permette di confermare

l'inserimento. Anche in questo caso l'algoritmo è in grado di gestire eventuali errori di digitazione, presentando un messaggio di conferma o meno nella TextArea.

Sempre nella sezione inferiore sulla destra possiamo trovare due bottoni dedicati alla ricerca della soluzione, entrambi i bottoni sono dinamici e variano il loro significato in base alla situazione in cui l'utente si trova. Ad esempio, una volta che il software ha presentato la soluzione ottimale, il bottone "GENERA SOLUZIONE" si trasforma nel bottone "CONFERMA" soluzione e il bottone "LISTA" diventa "ANNULLA" soluzione.

6.2 Video dimostrativo

È possibile visionare un video dimostrativo del funzionamento dell'applicazione al seguente indirizzo: <https://youtu.be/DV8gbuIhYhI>

Capitolo 7: Risultati Sperimentali

7.1 Primo Esempio: No Filtri

In un primo esempio, immaginiamo che l'utente non abbia preferenze particolari, di conseguenza nella ricerca non viene immesso alcun tipo di filtro ad eccezione del Budget, che, come da specifica, dev'essere inserito obbligatoriamente per generare la soluzione.

The screenshot shows a window titled "BOAT SITE". In the top right corner, there is a "LOGOUT" button. Below it, the word "SOLUZIONE" is displayed in blue. The main area is divided into two sections: "FILTRI" (Filters) and "SOLUZIONE" (Solution).

FILTRI:

- CONDIZIONE:** A dropdown menu set to "Qualsiasi".
- LUNGHEZZA:** Two input fields labeled "min" and "max".
- TIPOLOGIA:** A dropdown menu set to "Qualsiasi".
- ANNO:** Two input fields labeled "min" and "max".
- BUDGET:** An input field containing the value "12000".

SOLUZIONE:

La soluzione ottimale è:

Boat: id=1, prezzo=3300, tipologia=Pontoon Boat, condizione>New boat, anno=2018, lunghezza=4.35
Guadagno atteso:830

Boat: id=2, prezzo=3500, tipologia=Sport Boat, condizione>Used boat, anno=2004, lunghezza=4.7
Guadagno atteso:800

Boat: id=3, prezzo=4300, tipologia=Pontoon Boat, condizione>New boat, anno=2018, lunghezza=4.37
Guadagno atteso:930

Budget speso: 11100
Guadagno totale atteso: 2560

At the bottom of the window, there are several buttons: "ID" (input field), "INSERT ID" (button), "ANNULLA" (button), and "CONFERMA" (button). The "CONFERMA" button is highlighted with a blue border.

La soluzione ricavata consiste nella possibilità di acquistare tre imbarcazioni (id=1,2,3), spendendo 11.100 € e generando una previsione nel ricavo totale di 2.110 €.

7.2 Secondo Esempio: Confronto con Primo Esempio

In questo secondo esempio proviamo invece, oltre al Budget, ad inserire due specifiche nelle ComboBox “CONDIZIONE” e “TIPOLOGIA”, scegliendo così di ricercare rispettivamente un’imbarcazione nuova e il modello Sport Boat.

The screenshot shows the 'BOAT SITE' application window. At the top, there are three colored window control buttons (red, yellow, green). Below them, the title 'BOAT SITE' is centered. Underneath the title, the word 'FILTRI' is written in blue capital letters. The filter section contains five rows, each with a label in blue and a corresponding input field or dropdown menu:

- CONDIZIONE: A dropdown menu set to "New boat".
- LUNGHEZZA: Two input fields labeled "min" and "max".
- TIPOLOGIA: A dropdown menu set to "Sport Boat".
- ANNO: Two input fields labeled "min" and "max".
- BUDGET: An input field containing the value "12000".

Below the filters, a "RESET FILTRI" button is located. The next section is titled "SOLUZIONE" in blue. It displays the optimal solution found by the algorithm:

La soluzione ottimale è:
Boat: id=32, prezzo=10000, tipologia=Sport Boat, condizione>New boat, anno=2018, lunghezza=5.9
Guadagno atteso:1500

Budget speso: 10000
Guadagno totale atteso: 1500

At the bottom of the window, there are several buttons: an 'ID' input field, an 'INSERT ID' button, an 'ANNULLA' button, a 'CONFERMA' button (which is highlighted with a blue border), and a 'CONFERMA' button at the very bottom right.

L’algoritmo con questi filtri consiglia l’acquisto dell’imbarcazione id=32; con un Budget speso di 10.000 € e un guadagno totale atteso di 1.350 €. È utile notare come con lo stesso Budget, ma inserendo alcuni vincoli, la soluzione generata è composta da una sola imbarcazione, a fronte delle tre proposte precedentemente.

7.3 Terzo Esempio: Conferma Soluzione

Adesso immaginiamo di andare alla ricerca di imbarcazioni più specifiche. Inseriamo nelle ComboBox CONDIZIONE e TIPOLOGIA, rispettivamente un'imbarcazione usata e di tipo cabinata (Cabin Boat). Inoltre, inseriamo anche dei range di lunghezza tra gli 8 e i 20 metri e di anno compreso tra il 2000 e il 2015. Questa tipologia di imbarcazione richiede un costo di partenza molto più elevato; quindi, l'utente imposta un Budget massimo di 100.000€.

The screenshot shows the 'BOAT SITE' application window. At the top, there are three colored window control buttons (red, yellow, green). Below them is the title 'BOAT SITE'. Underneath the title is a section labeled 'FILTRI' (Filters) containing five input fields:

- CONDIZIONE:** A dropdown menu set to 'Used boat'.
- LUNGHEZZA:** Two input fields for range selection, showing '8' in the first and '20' in the second.
- TIPOLOGIA:** A dropdown menu set to 'Cabin Boat'.
- ANNO:** Two input fields for year selection, showing '2000' in the first and '2015' in the second.
- BUDGET:** An input field containing the value '100000'.

Below the filters is a 'RESET FILTRI' button. The next section is labeled 'SOLUZIONE' (Solution), which displays the optimal solution found by the application:

La soluzione ottimale è:

Boat: id=233, prezzo=30000, tipologia=Cabin Boat, condizione=Used boat, anno=2002, lunghezza=9.0
Guadagno atteso:3450

Boat: id=592, prezzo=69000, tipologia=Cabin Boat, condizione=Used boat, anno=2011, lunghezza=10.15
Guadagno atteso:7550

Budget speso: 99000
Guadagno totale atteso: 11000

At the bottom of the window are several buttons: 'ID' (disabled), 'INSERT ID', 'ANNULLA' (Cancel), 'CONFERMA' (Confirm), and another 'CONFERMA' button which is highlighted with a blue border.

La soluzione che è stata proposta dall'applicazione prevede l'acquisto di una prima barca (id=233) del 2002, di lunghezza 9mt ad un costo di soli 30.000€ e di una seconda (id=592) più recente, di lunghezza 10.15 mt ad un prezzo di 69.000€.

È utile notare la presenza di una notevole differenza tra le due nel guadagno atteso, scaturito: in primis dalla differenza di prezzo, e in secondo luogo dal fatto che la seconda imbarcazione va oltre i 10mt. Questo significa che rientra, per legge, tra quelle barche che vanno targate e ciò comporta un aumento del costo e di conseguenza del guadagno.

Nel caso in cui la soluzione sia di gradimento per l'utente, il passo successivo consiste nel confermare l'acquisto tramite l'apposito bottone. Comparirà una schermata, come da esempio, nella quale saranno riassunti tutti gli ordini effettuati con il totale del Budget speso e del Guadagno atteso.

BOAT SITE

FILTRI

CONDIZIONE	<input type="button" value="Used boat"/>
LUNGHEZZA	<input type="text" value="8"/> <input type="text" value="20"/>
TIPOLOGIA	<input type="button" value="Cabin Boat"/>
ANNO	<input type="text" value="2000"/> <input type="text" value="2015"/>
BUDGET	<input type="text" value="1000"/>

RESET FILTRI

Acquisto confermato con successo!

Lista barche acquistate fin'ora:

Boat: id=233, prezzo=30000, tipologia=Cabin Boat, condizione=Used boat, anno=2002, lunghezza=9.0
Guadagno atteso:3450

Boat: id=592, prezzo=69000, tipologia=Cabin Boat, condizione=Used boat, anno=2011, lunghezza=10.15
Guadagno atteso:7550

Budget speso: 99000
Guadagno totale atteso: 11000

Procedere con una nuova ricerca con il budget rimanente, modificare i filtri/budget o ricominciare

ID **INSERT ID**

RESET **NUOVA RICERCA**

7.4 Quarto Esempio: Doppia Soluzione

Dopo che l'utente ha confermato l'acquisto può procedere in due modi diversi. In un primo caso premendo il tasto “RESET” termina la simulazione e può ricominciare una nuova ricerca. In un secondo caso invece l'utente può continuare la ricerca attraverso il tasto “NUOVA RICERCA”. Ad esempio, mettiamo caso che voglia continuare ad investire il proprio denaro, questa volta nell'acquisto di uno Yacht usato con particolari specifiche. Per cui l'algoritmo, come in tutti gli altri casi, genera una soluzione permettendo all'utente di rientrare perfettamente nel Budget inserito, spendendo appunto un massimo di 600.000 € con un guadagno atteso di 62.500 €.

In alcuni casi, come quello seguente, l'algoritmo è in grado di generare una doppia soluzione, proponendo un'alternativa con un Budget superiore dell'1% rispetto a quello imposto inizialmente. Infatti, con una spesa di 5.000 € in più, ti permetterà di avere un Guadagno totale atteso di 63.000€.

The screenshot shows the 'BOAT SITE' application window. At the top, there's a toolbar with three colored circles (red, yellow, green). Below it is a header bar with the title 'BOAT SITE'. The main area is divided into sections:

- FILTRI**: A section for filtering search results.
 - CONDIZIONE**: A dropdown menu set to "Used boat".
 - LUNGHEZZA**: Two input fields for length, showing 15 and 18.
 - TIPOLOGIA**: A dropdown menu set to "Motor Yacht".
 - ANNO**: Two input fields for year, showing 2000 and "max".
 - BUDGET**: An input field showing 600000.
- SOLUZIONE**: A section displaying the results of the search.
 - Text: "La soluzione ottimale è:"
 - Text: "Boat: id=1103, prezzo=170000, tipologia=Motor Yacht, condizione=Used boat, anno=2003, lunghezza=15.2
Guadagno atteso:17850"
 - Text: "Boat: id=1139, prezzo=180000, tipologia=Motor Yacht, condizione=Used boat, anno=2006, lunghezza=17.0
Guadagno atteso:18850"
 - Text: "Boat: id=1279, prezzo=250000, tipologia=Motor Yacht, condizione=Used boat, anno=2003, lunghezza=17.05
Guadagno atteso:25850"
 - Text: "Budget speso: 600000
Guadagno totale atteso: 62550"
 - Text: "In alternativa la soluzione ottima potrebbe essere:"
 - Text: "Boat: id=1325, prezzo=270000, tipologia=Motor Yacht, condizione=Used boat, anno=2007, lunghezza=15.8
Guadagno atteso:27850"
 - Text: "Boat: id=1103, prezzo=170000, tipologia=Motor Yacht, condizione=Used boat, anno=2003, lunghezza=15.2
Guadagno atteso:17850"
 - Text: "Boat: id=1084, prezzo=165000, tipologia=Motor Yacht, condizione=Used boat, anno=2000, lunghezza=17.04
Guadagno atteso:17300"
 - Text: "Spendendo 5000 in più rispetto al budget.
Guadagno totale atteso dalla 2° soluzione: 63000"
- Buttons at the bottom:
 - An input field labeled "ID" and a button labeled "INSERTD ID".
 - A button labeled "CONFERMA 2° SOL".
 - A large blue button labeled "CONFERMA".

Adesso, come si può osservare, è possibile confermare l'una o l'altra opzione di acquisto. Facendo l'ipotesi in cui volessimo confermare la seconda soluzione, premendo l'apposito bottone "CONFERMA 2° SOL", possiamo inserire tra gli acquisti il set di imbarcazioni che sfiora il Budget. L'applicazione tiene in memoria le operazioni effettuate fino a quel momento. Infatti, come si evince dall'immagine la lista delle barche acquistate comprende: la seconda soluzione appena acquistata e quella confermata precedentemente nel terzo esempio.

The screenshot shows the 'BOAT SITE' application window. At the top, there are three colored window control buttons (red, yellow, green). Below them is the title 'BOAT SITE'. Underneath the title is a section labeled 'FILTRI' (Filters) in blue. This section contains five filter categories with their respective input fields:

- CONDIZIONE:** A dropdown menu set to 'Used boat'.
- LUNGHEZZA:** Two input fields for minimum and maximum length, both set to '15'.
- TIPOLOGIA:** A dropdown menu set to 'Motor Yacht'.
- ANNO:** Two input fields for minimum and maximum year, both set to '2000'.
- BUDGET:** An input field containing '-5000'.

Below the filters is a 'RESET FILTRI' button. The main content area displays the following messages and data:

- 'Acquisto confermato con successo!'
- 'Lista barche acquistate fin'ora:'
- Boat: id=233, prezzo=30000, tipologia=Cabin Boat, condizione=Used boat, anno=2002, lunghezza=9.0
Guadagno atteso:3450
- Boat: id=592, prezzo=69000, tipologia=Cabin Boat, condizione=Used boat, anno=2011, lunghezza=10.15
Guadagno atteso:7550
- Boat: id=1325, prezzo=270000, tipologia=Motor Yacht, condizione=Used boat, anno=2007, lunghezza=15.8
Guadagno atteso:27850
- Boat: id=1103, prezzo=170000, tipologia=Motor Yacht, condizione=Used boat, anno=2003, lunghezza=15.2
Guadagno atteso:17850
- Boat: id=1084, prezzo=165000, tipologia=Motor Yacht, condizione=Used boat, anno=2000, lunghezza=17.04
Guadagno atteso:17300
- Budget speso: 704000
Guadagno totale atteso: 74000
- 'Procedere con una nuova ricerca con il budget rimanente, modificare i filtri/budget o ricominciare'

At the bottom of the window are several buttons:

- An 'ID' input field.
- A 'INSERTD ID' button.
- A 'RESET' button.
- A 'NUOVA RICERCA' button.

7.5 Quinto Esempio: Inserimento ID

Nel quinto e ultimo esempio mi piacerebbe mostrare come il software permette di aggiungere nella ricerca di una soluzione il codice identificativo di una specifica imbarcazione. Poniamo il caso che la barca con id=362 sia di particolare interesse per l'utente, basta inserire il codice nella barra in fondo a sinistra e premere il bottone “INSERT ID”. A questo punto apparirà un messaggio di conferma del corretto inserimento e da qui è possibile continuare con la ricerca di una soluzione.

The screenshot shows the 'BOAT SITE' application window. At the top, there are three colored window control buttons (red, yellow, green). Below them is the title 'BOAT SITE'. Underneath the title is a section labeled 'FILTRI' (Filters) in blue. This section contains five filter categories: 'CONDIZIONE' (Condition), 'LUNGHEZZA' (Length), 'TIPOLOGIA' (Type), 'ANNO' (Year), and 'BUDGET' (Budget). Each category has associated input fields: 'Used boat' dropdown for Condition, 'min' and 'max' input fields for Length, 'Classic' dropdown for Type, 'min' and 'max' input fields for Year, and a single input field for Budget set to '70000'. To the right of these filters is a 'RESET FILTRI' button. Below the filters, a scrollable list displays search results:

- Guadagno atteso:4150
- Boat: id=299, prezzo=38000, tipologia=Classic,Flybridge, condizione=Used boat, anno=1970, lunghezza=15.0
Guadagno atteso:4650
- Boat: id=362, prezzo=45000, tipologia=Classic,Runabout, condizione=Used boat, anno=1962, lunghezza=6.1
Guadagno atteso:4950
- Boat: id=523, prezzo=60000, tipologia=Classic,Flybridge, condizione=Used boat, anno=1971, lunghezza=14.82
Guadagno atteso:6600
- Boat: id=545, prezzo=65000, tipologia=Classic,Motor Yacht, condizione=Used boat, anno=1991, lunghezza=10.8
Guadagno atteso:7050
- Boat: id=550, prezzo=65000, tipologia=Classic,Flybridge, condizione=Used boat, anno=1979, lunghezza=15.0
Guadagno atteso:7350

Below the results, a message states: 'Barca inserita correttamente nella lista delle soluzioni' (Boat inserted correctly into the list of solutions).

At the bottom of the window, there is a text input field containing '362' and a button labeled 'INSERT ID'. To the right of these are two more buttons: 'LISTA' and 'GENERA SOLUZIONE'.

Come possiamo vedere, l'algoritmo ha ricercato una combinazione che permette di massimizzare il profitto includendo nella lista finale l'imbarcazione con id=362. Anche qui, l'operazione si conclude con la conferma dell'acquisto o con l'annullamento della ricerca stessa.

BOAT SITE

FILTRI

CONDIZIONE	<input type="button" value="Used boat"/>
LUNGHEZZA	<input type="button" value="min"/> <input type="button" value="max"/>
TIPOLOGIA	<input type="button" value="Classic"/>
ANNO	<input type="button" value="min"/> <input type="button" value="max"/>
BUDGET	<input type="button" value="70000"/>

SOLUZIONE

La soluzione ottimale è:

Boat: id=362, prezzo=45000, tipologia=Classic,Runabout, condizione=Used boat, anno=1962, lunghezza=6.1
Guadagno atteso:4950

Boat: id=145, prezzo=24000, tipologia=Classic,Pontoon Boat, condizione=Used boat, anno=1968, lunghezza=5.65
Guadagno atteso:2850

Budget speso: 69000
Guadagno totale atteso: 7800

Capitolo 8: Conclusioni Finali

8.1 Limiti dell'applicazione

Il principale limite dell'applicazione riguarda il campo "guadagno atteso" nella classe "Boat", quest'ultimo può risultare in certi casi poco preciso e oggettivo. Questo perché sono state fatte delle assunzioni, grazie alle quali si è ipotizzato un valore di mercato per ogni imbarcazione ma, ovviamente, è fortemente influenzato dai singoli casi specifici; ad esempio, può capitare che una determinata barca venga sopravvalutata o sottovalutata dall'algoritmo.

Questo limite è scaturito dalle considerazioni effettuate, frutto della mia esperienza personale e di alcune ricerche, che tendono a generalizzare e creare dei sottogruppi troppo uniformi. L'algoritmo può essere sicuramente migliorato aggiungendo altri range/vincoli su altre caratteristiche delle imbarcazioni per rendere il valore del guadagno atteso maggiormente preciso.

Va però considerato che mancherà sempre l'aspetto umano e il programma è stato pensato appunto per supportare l'Ingegnere Gestionale nella scelta aziendale e anche per questo potrebbe essere implementata una funzione che permette di correggere manualmente il valore "guadagno atteso" per quelle barche che lo richiedono e rendere l'algoritmo molto più preciso.

8.2 Valutazione dei risultati

L'applicazione sviluppata raggiunge tutti gli obiettivi posti in fase di progettazione, permettendo di ottenere una stima di profitto come preventivato all'inizio della ricerca.

L'interfaccia risulta essere estremamente intuitiva, questo permette il facile utilizzo a coloro i quali vanno alla ricerca di nuove opportunità per la compravendita di imbarcazioni.

L'applicativo è in grado di fornire uno spunto importante affinché si possa indirizzare la società, interessata ad uno o più acquisti, in una direzione ottimale per la massimizzazione del profitto atteso.

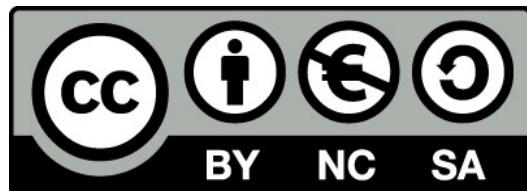
La rilevanza, dal punto di vista gestionale, riguarda l'aiuto che un software di questo genere può dare all'ingegnere che cerca di fare una scelta mirata e calibrata in base alle necessità che l'azienda impone.

Uno dei punti di forza riguarda sicuramente la flessibilità che quest'applicazione può avere, infatti, con il passare del tempo e l'evolversi del mercato, l'ingegnere è in grado di adattare e modificare la ricerca di una soluzione che sia il più possibile realistica.

In conclusione, ritengo che questo software possa essere migliorato e rafforzato implementando nuove funzionalità e nuovi vincoli per la ricerca di una soluzione ottima, adattandolo ad altre esigenze, come la capienza disponibile in magazzino/showroom.

Al giorno d'oggi vi è un continuo aumento d'interesse verso il settore nautico e le imbarcazioni da diporto, e così, grazie nuove tecnologie, sempre più imprese potranno affacciarsi in modo semplice e lineare a questo mondo.

Capitolo 9: Licenza



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

Copia della licenza consultabile al sito web: <http://creativecommons.org/licenses/by-nc-sa/4.0/>