



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Corso di Laurea Triennale in Informatica

**Sistema di gestione
di personale e progetti all'interno
di un'azienda**

Mattia Scotellaro N86004846
Marta Pagliuso N86004844

ANNO ACCADEMICO 2022/2023

Indice

1	Modello di Dominio	2
1.1	Analisi dei requisiti	2
1.2	Dominio del Problema Generale	3
2	Dizionari	4
2.1	Dizionario delle Classi	4
2.2	Dizionario delle Associazioni	5
3	Model	6
3.1	Traduzione delle Classi e delle Relazioni	6
3.2	Package Model Diagram	7
4	Comunicazione con il Database	8
4.1	Connessione al Database	8
5	Package DAO	9
5.1	Interfaccia GestionaleDAO	9
5.2	Interfaccia ImpiegatoDAO	9
5.3	Interfaccia LaboratorioDAO	9
5.4	Interfaccia ProgettoDAO	9
5.5	Interfaccia AttrezzaturaDAO	9
6	Implementazione del DAO	9
7	Package Diagram DAO	10
8	Graphical User Interface	10
8.1	Package Diagram GUI	11
9	Design Pattern BCED	12
9.1	Package Controller	12
10	Dominio della Soluzione	13
10.1	Solution Diagram	13
11	Sequence Diagram	14
11.1	Aggiungi Impiegato	14
11.2	Elimina Impiegato	14

1 Modello di Dominio

1.1 Analisi dei requisiti

In questa sezione, esamineremo la richiesta del cliente con l'obiettivo di delineare le caratteristiche che il sistema deve possedere. Identifieremo attributi e metodi dell'applicativo al fine di rappresentare in modo completo e comprensibile il dominio del problema.

L'azienda possiede un certo numero di impiegati, raggruppabili in 4 categorie:

1. *Dipendente junior: colui che lavora da meno di 3 anni all'interno dell'azienda;*
2. *Dipendente middle: colui che lavora da meno di 7 ma più di 3 anni per l'azienda;*
3. *Dipendente senior: colui che lavora da almeno 7 anni all'interno dell'azienda;*
4. *Dirigente: la classe dirigente non ha obblighi temporali di servizio. Chiunque può diventare dirigente, se mostra di averne le capacità.*

I passaggi di ruolo avvengono per anzianità di servizio.

In questa parte viene messa in evidenza la divisione gerarchica dei dipendenti all'interno dell'azienda. Ci sono 3 categorie principali: Junior, Middle e Senior. Un dipendente può salire di livello in base agli anni di servizio conseguiti all'interno dell'azienda. Eccezione fatta per la quarta categoria, ovvero quella del Dirigente, in cui può partecipare un qualsiasi impiegato, a patto che ne abbia le capacità.

Nell'azienda vengono gestiti laboratori e progetti. Un laboratorio ha un particolare topic di cui si occupa, un certo numero di afferenti ed un responsabile scientifico che è un dipendente senior.

Qui vengono specificate due delle classi gestite all'interno dell'azienda. La prima è quella del Laboratorio, caratterizzato da un topic, ovvero in cosa è specializzato; il numero di afferenti, ovvero il numero di impiegati che ci lavorano e il responsabile scientifico che gestisce il laboratorio a lui assegnato con il vincolo che sia un dipendente Senior.

Un progetto è identificato da un CUP (codice unico progetto) e da un nome (unico nel sistema). Ogni progetto ha un referente scientifico, il quale deve essere un dipendente senior dell'ente, ed un responsabile che è uno dei dirigenti.

La seconda classe è quella del Progetto. Questo è identificato dal CUP, un codice alfanumerico di 15 caratteri, e da un nome, entrambi unici nel sistema. Di un Progetto si devono occupare un referente scientifico e un responsabile, rispettivamente un dipendente senior e un dirigente.

Al massimo 3 laboratori possono lavorare a un progetto.

Il vincolo in questione è molto chiaro: a ogni Progetto possono essere affidati da un minimo di 1 a un massimo di 3 Laboratori.

Un laboratorio ha diverse attrezzature (computer, robot, dispositivi mobili, sensori, ...) acquistati con i fondi di un determinato progetto.

La caratterizzazione della classe Laboratorio si fa più precisa con l'aggiunta di una nuova classe: Attrezzatura. Questa non viene descritta in modo specifico come le precedenti. Le sue specifiche possono essere scelte arbitrariamente in base alle esigenze.

Tracciare quindi gli acquisti fatti sui fondi di un progetto, con l'ovvio vincolo che il costo totale delle attrezzature non può superare il 50% del budget di un progetto.

Qui viene presentato il primo vincolo inter-relazionale che coinvolge le classi Progetto e Attrezzatura. In particolare, l'acquisto di un'Attrezzatura può avvenire un numero limitato di volte, in base al 50% del budget stipulato per un Progetto.

Con il restante 50% è possibile assumere dipendenti per l'azienda con un "contratto a progetto". Questi contratti hanno una scadenza, a differenza degli altri che sono a tempo indeterminato.

Gli impiegati dell'azienda sono divisi in due categorie: coloro che possiedono un contratto a tempo indeterminato e coloro che hanno un contratto "a progetto". Ciò significa che il contratto di questi ultimi è valido solo per il periodo necessario alla realizzazione del Progetto a cui è associato. Non vengono fatte altre specifiche, dunque eventuali condizioni sono a discrezione dello sviluppatore.

1.2 Dominio del Problema Generale

Considerando l'analisi dei requisiti svolta nella sezione precedente, viene ora presentata la prima iterazione del Class Diagram.

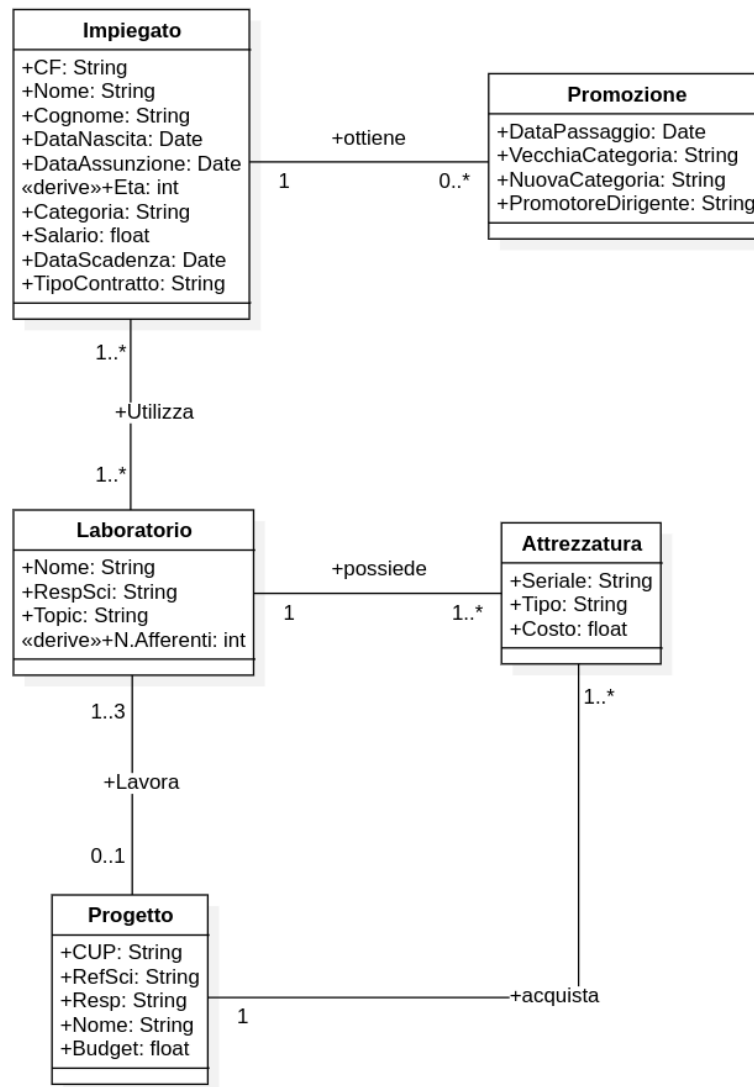


Figura 1: Diagramma UML

2 Dizionari

2.1 Dizionario delle Classi

Entità	Descrizione	Attributi
Impiegato	Entità che tiene traccia degli impiegati dell'azienda	CF (Stringa): codice univoco che identifica un impiegato. Nome (Stringa): nome di un impiegato. Cognome (Stringa): cognome di un impiegato. DataNascita (Data): data di nascita di un impiegato. DataAssunzione (Data): data di assunzione di un impiegato nell'azienda. Eta (Int): età di un impiegato. Categoria (Stringa): categoria di un impiegato nell'azienda. Salario (Numerico): salario di un impiegato. DataScadenza (Data): data di scadenza di un contratto per un impiegato a progetto. TipoContratto (Stringa): indica se il contratto è a tempo indeterminato o "a progetto".
Promozione	Entità che tiene traccia degli scatti di carriera di un impiegato.	CF (Stringa): codice fiscale di un impiegato. DataPassaggio (Data): data in cui avviene lo scatto di carriera di un impiegato. VecchiaCategoria (Stringa): categoria di un impiegato al momento dello scatto di carriera. NuovaCategoria (Stringa): categoria ottenuta dopo lo scatto di carriera. PromotoreDirigente (Stringa): dirigente che permette a un senior di passare a dirigente.
Laboratorio	Entità che tiene traccia dei laboratori dell'azienda.	Nome (Stringa): nome di un laboratorio. Resp.Sci (Stringa): responsabile scientifico di un laboratorio. Topic (Stringa): indica di cosa si occupa un laboratorio. N.Afferenti (Int): indica il numero di impiegati che lavorano a un laboratorio.
Attrezzatura	Entità che tiene traccia delle attrezzature acquistate	Seriale (Stringa): identificativo di un'attrezzatura. Tipo (Stringa): indica la tipologia di un'attrezzatura. Costo (Numeric): costo di un'attrezzatura. NomeLab (Stringa): nome del laboratorio a cui è associata un'attrezzatura. CUP (Stringa): identificativo del progetto per cui è stata acquistata un'attrezzatura.

Progetto	Entità che tiene traccia dei progetti gestiti dall'azienda.	CUP (Stringa): identificativo di un progetto. Ref.Sci (Stringa): referente scientifico di un progetto, necessariamente senior. Resp. (Stringa): responsabile di un progetto, necessariamente dirigente. Nome (Stringa): nome di un progetto. Budget (Numeric): fondi a disposizione per un progetto.
----------	---	---

2.2 Dizionario delle Associazioni

Associazione	Tipologia	Descrizione
Ottiene	Uno-a-Molti	Associazione che specifica che un impiegato può ottenere una o più promozioni nell'arco della carriera.
Utilizza	Molti-a-Molti	Un impiegato può lavorare in più laboratori e, allo stesso tempo, in un laboratorio possono lavorare più impiegati.
Possiede	Uno-a-Molti	Associa più attrezzature in un laboratorio e ogni attrezzatura in un solo laboratorio.
Lavora	Uno-a-Molti	A un progetto lavorano al più tre laboratori; un laboratorio lavora al massimo a un progetto.
Acquista	Uno-a-Molti	Per un progetto vengono acquistate più attrezzature; un'attrezzatura viene adoperata per un solo progetto.

3 Model

3.1 Traduzione delle Classi e delle Relazioni

Una volta finita l'analisi del modello concettuale, passiamo al *mapping* di classi e associazioni.

- **Ottiene:** si tratta di una relazione *Uno-a-Molti parziale*. Si basa sulla presenza di un riferimento, all'interno della classe Promozione, alla classe Impiegato.
- **Utilizza:** si tratta di una relazione *Molti-a-Molti totale*. È stata risolta introducendo una classe di associazione che contiene due riferimenti: uno per la classe Impiegato, l'altro per la classe Laboratorio.
- **Possiede:** si tratta di una relazione *Uno-a-Molti totale*. Si basa sulla presenza del riferimento NomeLab della classe Attrezzatura che si riferisce alla classe Laboratorio.
- **Lavora:** si tratta di una relazione *Molti-a-Molti totale*. È stata risolta introducendo una classe di associazione che contiene due riferimenti: uno per la classe Laboratorio, l'altro per la classe Progetto.
- **Acquista:** si tratta di una relazione *Uno-a-Molti totale*. Si basa sulla presenza del riferimento CUP, nella classe Attrezzatura, alla classe Progetto.
- **Impiegato:** oltre a contenere gli attributi di un impiegato già specificati nel Model, contiene anche riferimenti ai suoi scatti di carriera e ai laboratori a cui afferisce.
- **Promozione:** oltre a contenere gli attributi di una promozione già specificati nel Model, contiene anche un riferimento all'impiegato associato.
- **Laboratorio:** oltre a contenere gli attributi di un laboratorio già specificati nel Model, contiene anche riferimenti agli impiegati che ci lavorano, le attrezzature utilizzate nello stesso e il progetto associato.
- **Progetto:** oltre a contenere gli attributi di un progetto già specificati nel Model, contiene anche riferimenti ai laboratori associati e le attrezzature acquistate.
- **Attrezzatura:** oltre a contenere gli attributi di un'Attrezzatura già specificati nel Model, contiene anche riferimenti al laboratorio in cui viene utilizzato e al progetto che la acquista.

3.2 Package Model Diagram

Il seguente diagramma raffigura il package model implementato nel codice sorgente dell'applicativo:

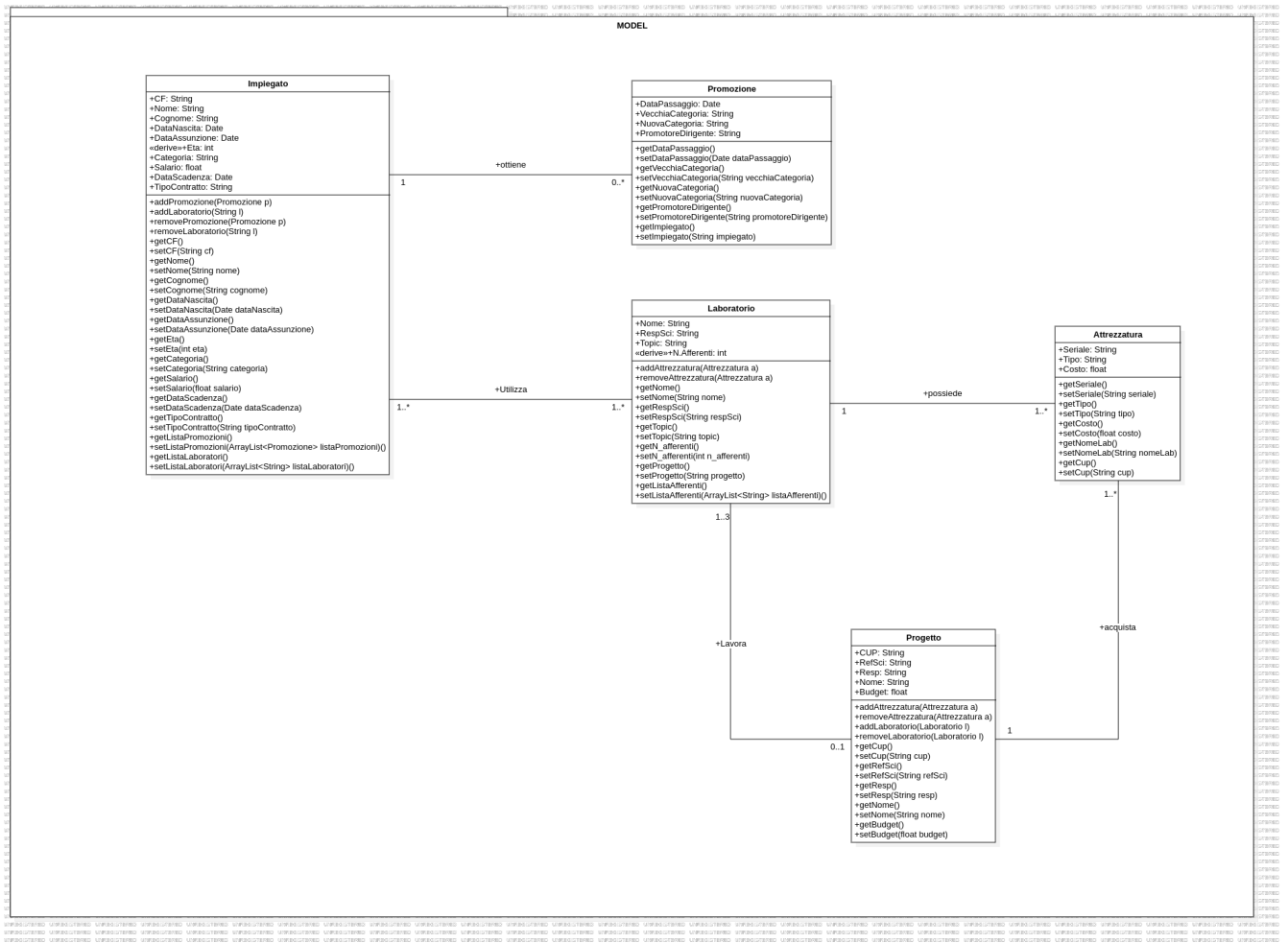


Figura 2: Package Model

4 Comunicazione con il Database

Per la comunicazione con il database utilizzeremo il pattern **DAO**, acronimo per Data Access Object. Questo pattern promuove la modularità e la manutenibilità del codice, rendendo più facile gestire le operazioni di accesso e manipolazione dei dati in un'applicazione.

4.1 Connessione al Database

La connessione al database aziendale si realizza all'interno del pacchetto *Database*. Questo pacchetto include la classe **ConnessioneDatabase**, la quale ha il compito di istanziare una connessione al database Postgres. La decisione è stata presa per mantenere il database online, consentendo così l'esecuzione dell'applicativo su diversi calcolatori senza la necessità di creare localmente la propria base di dati.

5 Package DAO

5.1 Interfaccia GestionaleDAO

La seguente interfaccia è stata implementata per facilitare il trasferimento di tutti i dati dal database al Model. Esiste quindi un metodo per ogni entità al fine di estrarre dalla base di dati i relativi Impiegati, Laboratori, Progetti, Promozioni e Attrezzature. Si specifica che il metodo per ottenere le afferenze non viene richiamato a prescindere, ma solo quando necessario.

5.2 Interfaccia ImpiegatoDAO

La seguente interfaccia si occupa di:

- Assumere impiegato a contratto indeterminato;
- Assumere impiegato con contratto a progetto;
- Licenziare un impiegato;
- Promuovere un impiegato;
- Ottenere le afferenze di un impiegato;
- Sostituire il responsabile scientifico/referente scientifico/responsabile.

5.3 Interfaccia LaboratorioDAO

La seguente interfaccia si occupa di:

- Aggiungere un laboratorio;
- Rimuovere un laboratorio;
- Aggiungere un afferente;
- Acquistare un'attrezzatura;

5.4 Interfaccia ProgettoDAO

La seguente interfaccia si occupa di:

- Aggiungere un progetto;
- Eliminare un progetto;
- Collegare un laboratorio;

5.5 Interfaccia AttrezzaturaDAO

La seguente interfaccia si occupa di:

- Aggiungere un'attrezzatura;
- Eliminare un'attrezzatura;

6 Implementazione del DAO

Le interfacce del package DAO vengono implementate tramite le classi presenti nel package **Implementazione-PostgresDAO**, che sono: ImplementazioneAttrezzaturaDAO, ImplementazioneGestionaleDAO, ImplementazioneImpiegatoDAO, ImplementazioneLaboratorioDAO, ImplementazioneProgettoDAO.

Nel momento in cui viene lanciata un'eccezione nel database, la GUI mostra un messaggio di errore comprensibile per l'utente e anche la sua versione completa per un utente più esperto.

7 Package Diagram DAO

Il diagramma seguente offre una panoramica dell'architettura DAO implementata nel software, identificando i pacchetti, le classi e i metodi utilizzati, insieme alle loro interconnessioni.

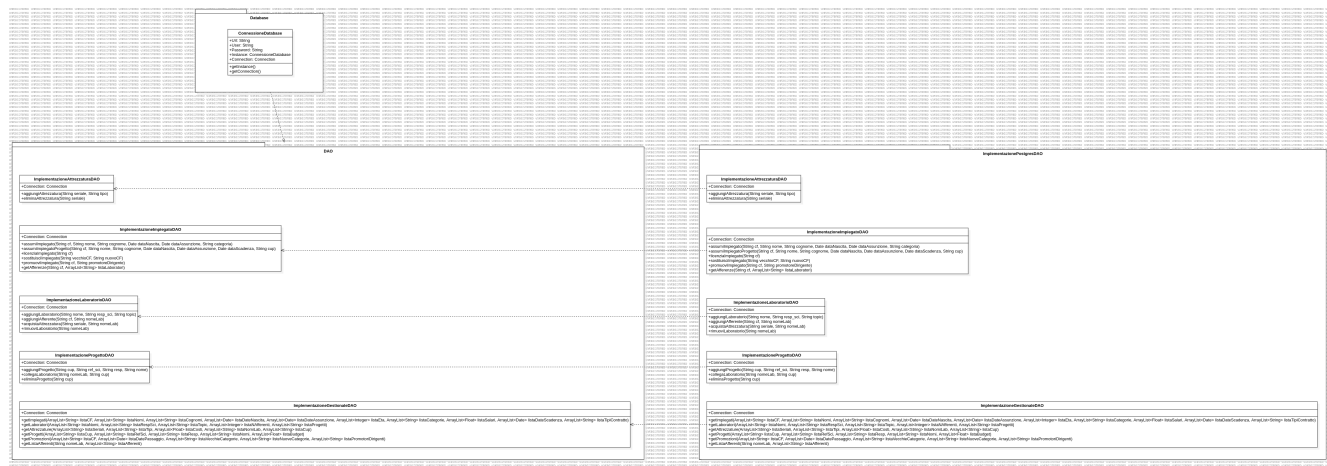


Figura 3: Package DAO

8 Graphical User Interface

In questa sezione, si fornirà una descrizione dell'implementazione dell'interfaccia grafica. È stata selezionata la libreria Swing come framework. L'interfaccia utente è stata realizzata all'interno del package GUI, utilizzando il seguente pattern di gestione dei file:

- **Home**: la pagina principale che reindirizza alle successive.
- **ImpiegatoGUI, AttrezzaturaGUI, LaboratorioGUI, ProgettoGUI**: macro pagine che contengono i dati più importanti delle classi principali. È possibile effettuare altre operazioni che si differenziano in base alla pagina in cui ci si trova.
- **AddAfferenteGUI, AddAttrezzaturaGUI, AddImpiegatoGUI, AddLaboratorioGUI, AddProgettoGUI**: è possibile effettuare l'inserito del dato omonimo.
- **AfferentiGUI**: mostra gli afferenti di un laboratorio.
- **ProfiloAttrezzaturaGUI, ProfiloImpiegatoGUI**: mostra tutti i dati di un'attrezzatura o di un impiegato rispettivamente.
- **PromuoviGUI**: permette di promuovere un impiegato da *senior* a *dirigente*, andando a selezionare il codice fiscale del *promotore dirigente*.
- **SelezionaCupGUI, SelezionaSerialeGUI**: il primo permette di selezionare un progetto da associare a un determinato laboratorio, mentre il secondo permette di acquistare un'attrezzatura per un laboratorio.
- **SostituisciGUI**: permette di selezionare un impiegato di categoria appropriata, per sostituirlo a un responsabile o un referente già presenti.

8.1 Package Diagram GUI

Il seguente diagramma descrive l'implementazione usata per il package GUI:

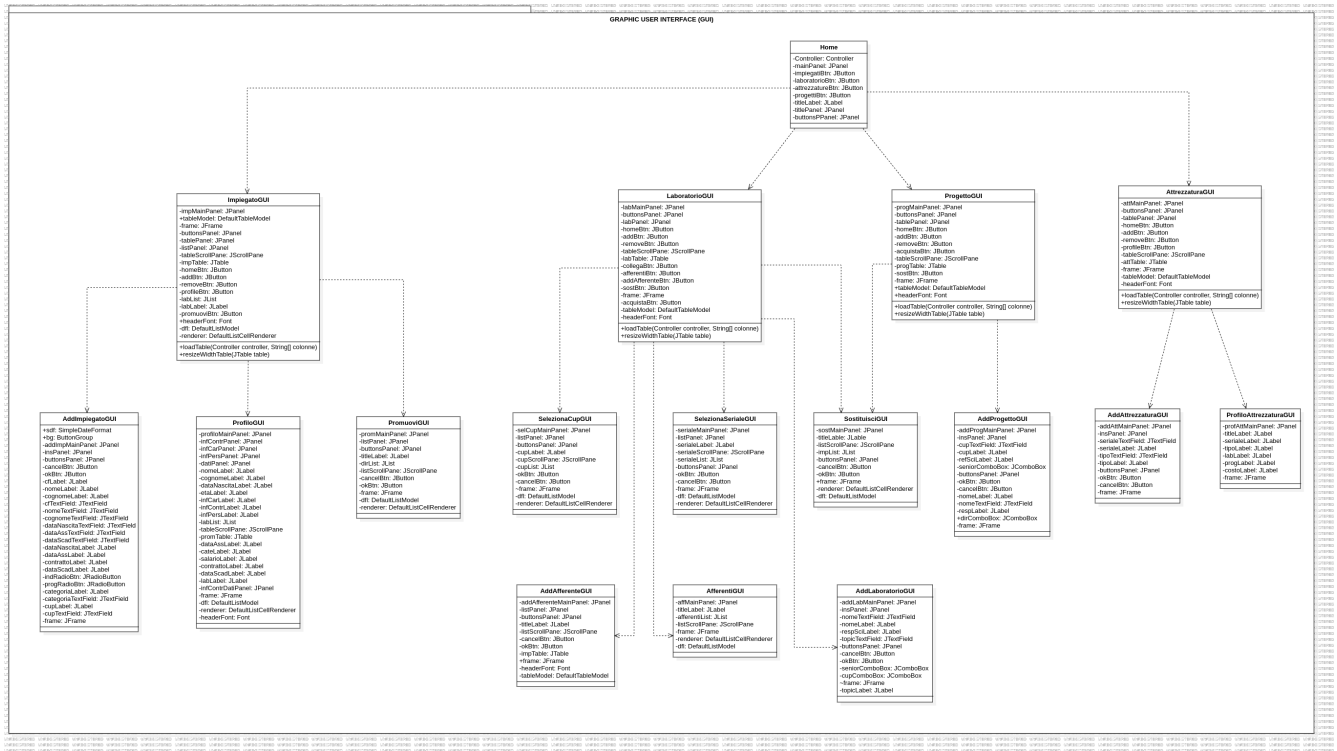


Figura 4: Package GUI

9 Design Pattern BCED

Prima di giungere alla spiegazione del dominio della soluzione, bisogna specificare il pattern utilizzato per creare l'applicativo in Java. In tale contesto è stato scelto il **modello BCED**, estensione del modello BCE. Questo perché permette di immagazzinare in modo permanente i dati all'interno del database. È composto da:

- **Boundary**: in questo caso è la GUI, ovvero l'interfaccia del software con l'esterno.
- **Controller**: insieme di classi che realizzano le operazioni, su richiesta diretta della GUI e gestendo i dati rappresentati nel Model.
- **Entity**: insieme di classi presenti nel package Model.
- **Database**: si occupa di interagire direttamente con la base di dati.

9.1 Package Controller

Il seguente diagramma raffigura il package Controller implementato nel codice sorgente dell'applicativo:

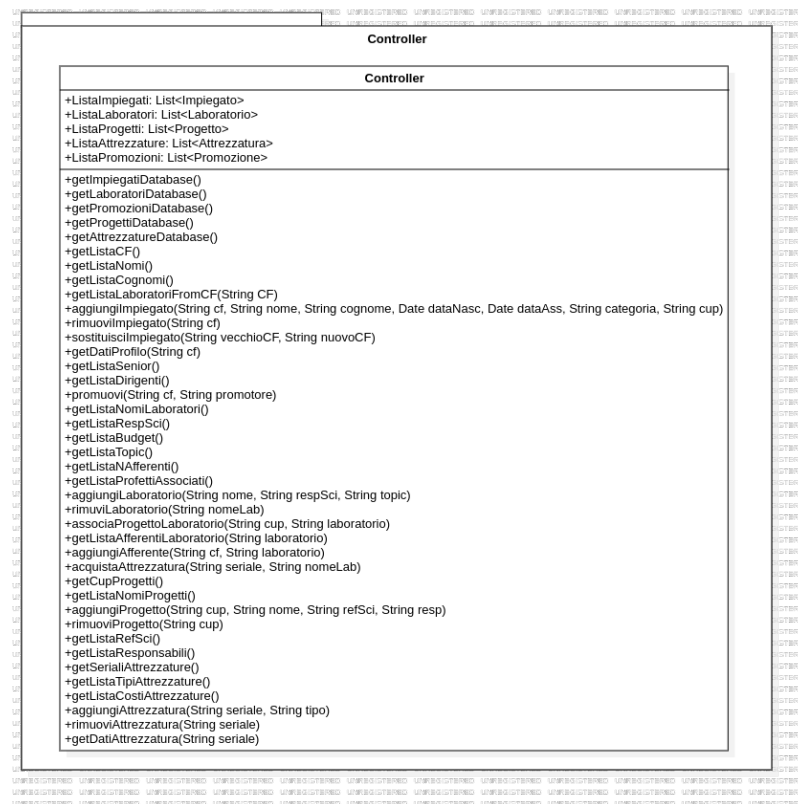


Figura 5: Package Controller

10 Dominio della Soluzione

In seguito all'analisi fatta precedentemente, è possibile ora presentare il dominio della soluzione.

10.1 Solution Diagram

Il seguente diagramma rappresenta il dominio della soluzione:

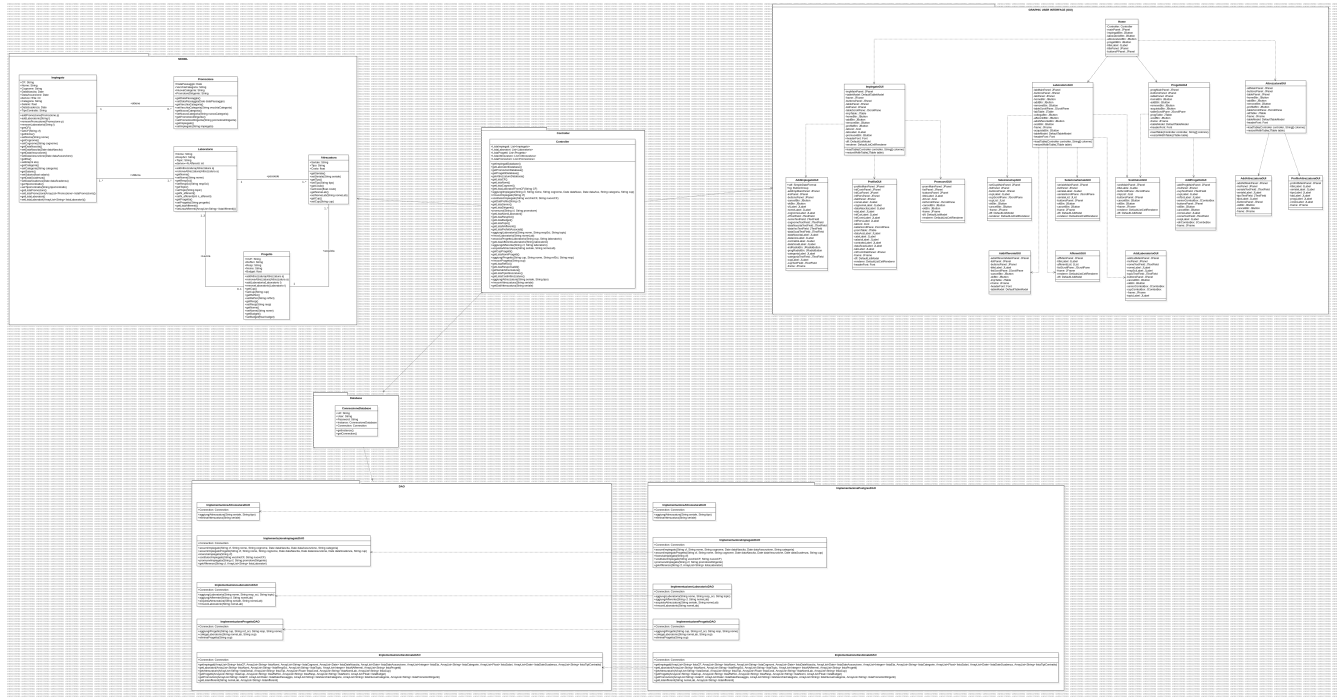


Figura 6: Solution Diagram

11 Sequence Diagram

In questa sezione vengono mostrati due sequence diagram che rappresentano l'esecuzione di due funzioni, in particolare l'inserimento e l'eliminazione di un impiegato.

11.1 Aggiungi Impiegato

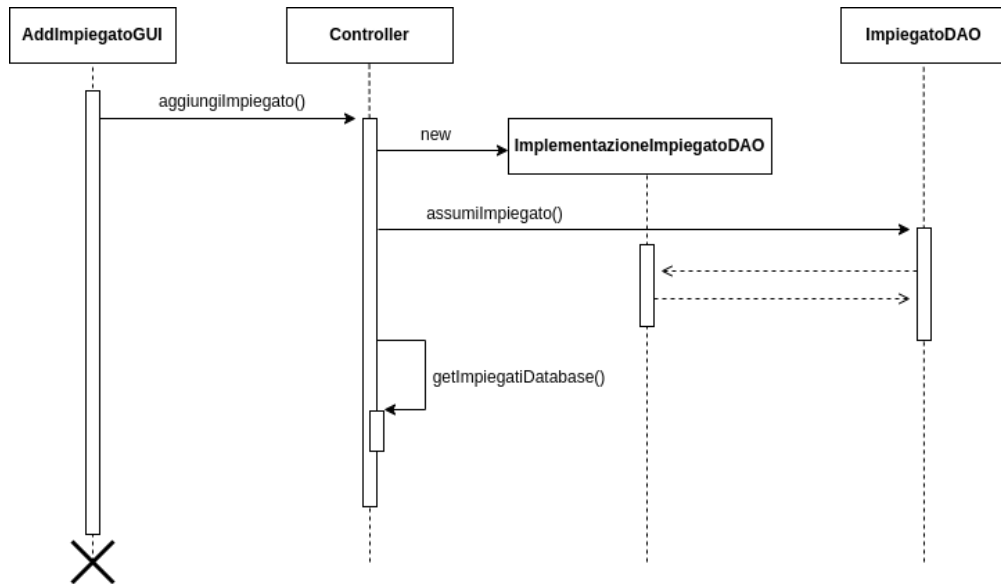


Figura 7: Sequence Diagram Inserimento Impiegato

11.2 Elimina Impiegato

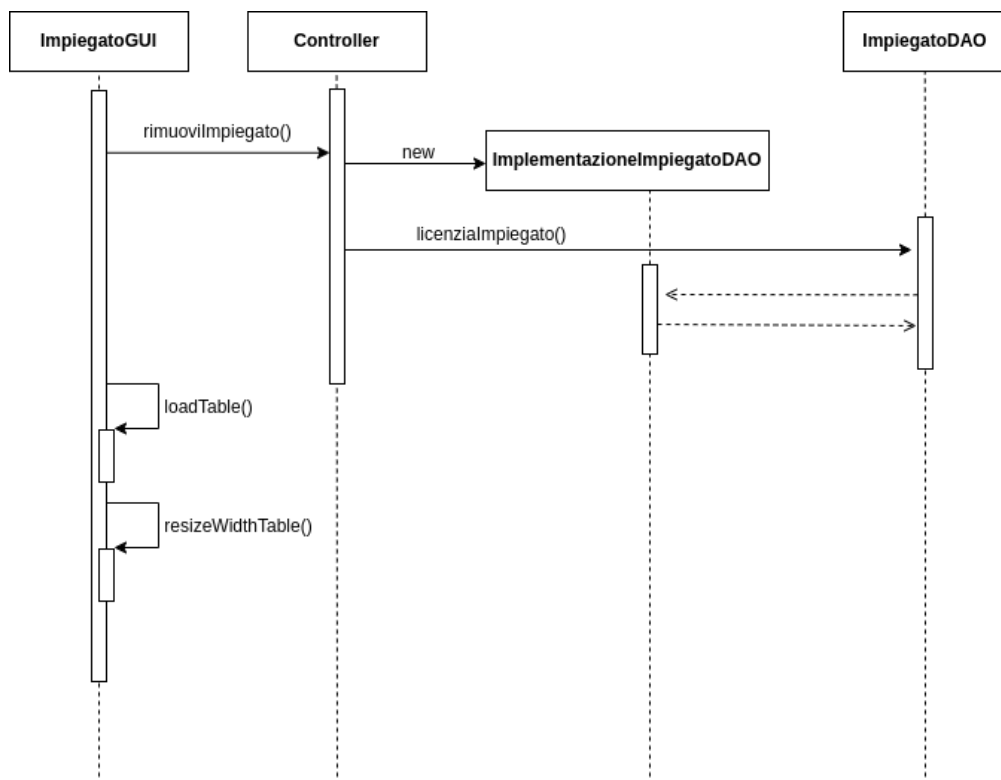


Figura 8: Sequence Diagram Eliminazione Impiegato