



Università degli studi di Napoli Federico II

Corso di Laurea Triennale in Informatica

Sistema di gestione del personale all'interno di un'azienda

Biagio Morgillo

N86004821

ANNO ACCADEMICO 2022/23

Indice

1	Requisiti identificati	3
1.1	Analisi dei requisiti	3
1.2	Diagramma del dominio generale	4
2	Dizionari	5
2.1	Dizionario delle Classi	5
2.2	Dizionario delle Associazioni	7
3	Soluzione nel dominio del problema	8
3.1	Modellare l'UML	8
3.2	Connettersi al database	9
3.2.1	Connessione al database	9
3.3	DAO	9
3.3.1	GestionaleDAO	9
3.3.2	ImpiegatoDAO	9
3.3.3	LaboratorioDAO	10
3.3.4	ProgettoDAO	10
3.4	ImplementazionePGresDAO	10
3.5	Graphical User Interface (GUI)	11
3.6	Controller	12
3.7	Dominio della soluzione del problema	13
4	Sequence diagram	14
4.1	Rimuovi Laboratorio	14
4.2	Aggiungi Laboratorio	15

Capitolo 1

Requisiti identificati

1.1 Analisi dei requisiti

Questa sezione serve a stabilire quali requisiti deve avere la base di dati a partire dalla richiesta fatta dal cliente. È importante trovare le entità e le relazioni che rappresenteranno in modo completo il dominio del problema. La richiesta riguarda la gestione degli impiegati in un'azienda un occhio di riguardo è tenuto soprattutto sulla loro anzianità, infatti sono state individuate 4 tipologie di impiegati:

- Dipendente junior: impiegato che lavora nell'azienda da meno di 3 anni;
- Dipendente middle: impiegato che lavora nell'azienda da più di 3 anni ma meno di 7 anni;
- Dipendente senior: impiegato che lavora nell'azienda da più di 7 anni;
- Dirigente: impiegato che ha ottenuto un certo merito. Non importa da quanto è stato assunto, l'importante è mostrare di avere una certa capacità.

I passaggi di ruolo avvengono in base all'anzianità.

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

Qui viene stabilita una delle entità chiave della base di dati, il Laboratorio, il quale si occupa di un argomento in particolare. Ha un certo numero di afferenti, quindi impiegati che lavorano in laboratorio tra cui si distingue un responsabile scientifico che è un impiegato 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. Al massimo 3 laboratori possono lavorare ad un progetto.

Un'altra delle entità chiave è stata stabilita qui, infatti si parlerà dell'entità Progetto, identificato da un CUP e da un nome, il progetto ha bisogno di un

referente scientifico, che è un dipendente senior ed da un responsabile, che è un dirigente. Gli impiegati, per lavorare a un progetto, devono utilizzare un laboratorio, però, massimo 3 laboratori possono lavorare a un progetto.

1.2 Diagramma del dominio generale

A seguire un'immagine che ritrae il diagramma generico per il dominio del problema:

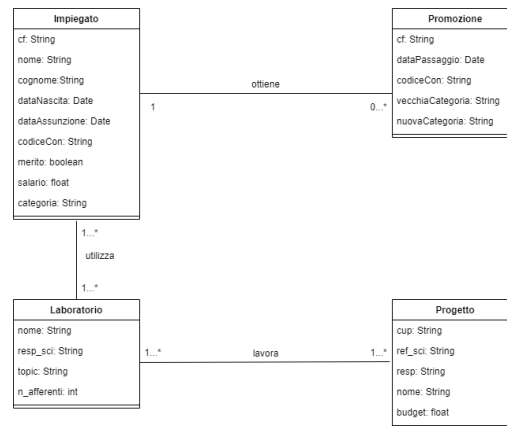


Figura 1: Diagramma UML generale

Capitolo 2

Dizionari

2.1 Dizionario delle Classi

Classi	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. CodiceCon (Stringa): codice contratto di un impiegato. Merito (Booleano): merito di un impiegato per essere dirigente. Salario (Numerico): salario di un impiegato. Categoria (Stringa): categoria di un impiegato nell'azienda. Età (Int): età di un impiegato.

Promozione	Entità che tiene traccia delle variazioni di carriera di un impiegato in azienda.	CF (Stringa): codice fiscale di un impiegato. DataPassaggio (Data): data in cui avviene la promozione dell'impiegato. CodiceCon (Stringa): codice del contratto che sta ricevendo la promozione. NuovaCategoria (Stringa): categoria a cui l'impiegato viene promosso.
Laboratorio	Entità che tiene traccia dei laboratori presenti in azienda.	Nome (Stringa): nome che identifica il laboratorio. Resp. Sci. (Stringa): codice fiscale di un impiegato senior. Topic (Stringa): argomento di cui si occupa il laboratorio. N.Afferenti (Int): numero di impiegati che lavorano in un laboratorio.
Progetto	Entità che tiene traccia dei progetti di cui si occupa l'azienda.	CUP (Stringa): codice unico che identifica il progetto. Ref. Sci. (Stringa): codice fiscale di un impiegato senior. Resp (Stringa): codice fiscale di un dirigente. Nome (Stringa): nome del progetto. Budget (Numerico): budget stanziato dall'azienda per il progetto.

2.2 Dizionario delle Associazioni

Associazione	Tipologia	Descrizione
Ottiene	Uno-a-Molti	Associazione che descrive come un impiegato può ottenere una o più promozioni.
Utilizza	Molti-a-Molti	Associazione che stabilisce come più impiegati possono utilizzare un laboratorio e come un impiegato può lavorare a un laboratorio.
Lavora	Uno-a-Molti	Associazione che stabilisce come al massimo tre laboratori possono lavorare a un progetto e come un laboratorio lavora al massimo a un progetto.

Capitolo 3

Soluzione nel dominio del problema

3.1 Modellare l'UML

Il punto d'inizio per trovare una soluzione al problema è stato quello di definire meglio le classi date dall'analisi dei requisiti posti dal problema. Il risultato finale è delineato dal pacchetto Model:

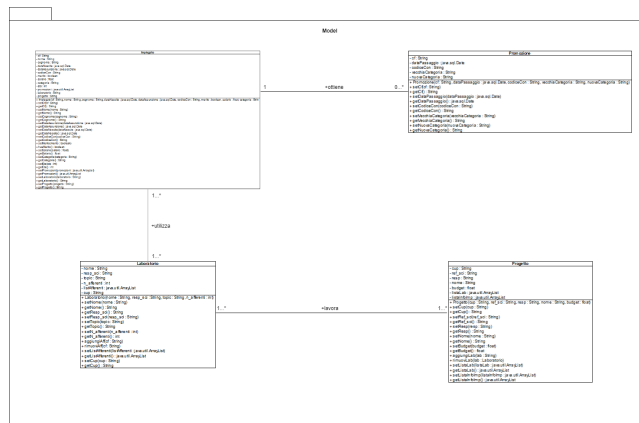


Figura 2: Package Model

3.2 Connettersi al database

Dovendo avere un collegamento a una base di dati che gestisce l'azienda, si deve, innanzitutto, stabilire una connessione e in questo caso si userà un pattern **DAO**: *Data-Access-Object*. Questo pattern renderà il codice più modulare e manutenibile, facilitando l'accesso e la manipolazione dei dati.

3.2.1 Connessione al database

La connessione al database dell'azienda è fornito dal pacchetto ConnectDB, contenente la classe DbConn, che effettua la connessione al database *Postgres*.

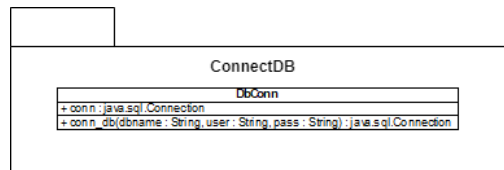


Figura 3: Package ConnectDB

3.3 DAO

3.3.1 GestionaleDAO

Interfaccia per facilitare il trasferimento di dati dalla base al Model, quindi ogni metodo associa i dati a: Impiegato, Promozione, Laboratorio e Progetto.

3.3.2 ImpiegatoDAO

Fornisce le operazioni per:

- aggiungere un impiegato;
- rimuovere un impiegato;
- promuovere un impiegato.

Fornisce anche i metodi per:

- prelevare i dati relativi alle promozioni;
- prelevare i dati sul laboratorio a cui lavora l'impiegato;
- prelevare i dati sul progetto a cui lavora l'impiegato.

3.3.3 LaboratorioDAO

Fornisce le operazioni per:

- aggiungere un laboratorio;
- rimuovere un laboratorio;
- aggiungere un nuovo afferente.

Fornisce anche i metodi per:

- prelevare le informazioni relative al responsabile scientifico (nome, cognome e codice fiscale);
- prelevare i progetti a cui sta lavorando lo specifico laboratorio.

3.3.4 ProgettoDAO

Fornisce le operazioni per:

- aggiungere un progetto;
- rimuovere un laboratorio;
- aggiungere un nuovo laboratorio.

Fornisce anche i metodi per:

- prelevare le informazioni relative a responsabile e referente scientifico;
- prelevare i nomi dei laboratori che lavorano al progetto.

3.4 ImplementazionePGresDAO

Fornisce l'implementazione di tutte le interfacce proposte dal pacchetto DAO, quindi sviluppa una logica per GestionaleDAO, ImpiegatoDAO, LaboratorioDAO e ProgettoDAO, rispettivamente, in ImplementazioneGestionaleDAO, ImplementazioneImpiegatoDAO, ImplementazioneLaboratorioDAO, ImplementazioneProgettoDAO.

A seguire la visualizzazione di come comunicano i pacchetti: ConnectDB, DAO e ImplementazionePGresDAO:

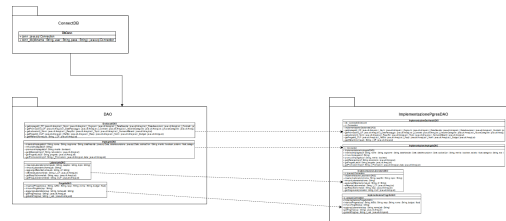


Figura 4: Packages
ConnectDB-DAO-ImplementazionePGresDAO

3.5 Graphical User Interface (GUI)

Per implementare le operazioni che svolge il DAO si è fatto uso di un'interfaccia grafica sviluppata tramite il framework Swing. L'interfaccia è contenuta nel pacchetto View e implementa:

- Home: schermata principale da cui si può raggiungere la schermata per Impiegato, Laboratorio e Progetto.
- ImpiegatoGUI, LaboratorioGUI e ProgettoGUI: sono schermate che mostrano le tabelle contenenti le informazioni riguardo le rispettive tabelle e permettono di svolgere le operazioni sulle tabelle.
- AddImpiegatoGUI, AddLaboratorioGUI, AddProgettoGUI: sono micro-schermate che permettono di fare l'inserimento nelle rispettive tabelle.

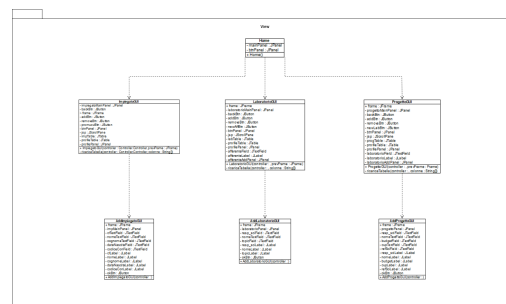


Figura 5: Package View

3.6 Controller

Per rendere possibile l'immagazzinamento dei dati in modo permanente è stato usato il modello BCED, che estende il pattern BCE. È composto da:

- **Boundary:** è l'interfaccia (GUI) che permette di far operare l'utente.
- **Controller:** è l'insieme di metodi che permettono di svolgere le operazioni nella GUI tramite i dati presenti nel Model.
- **Entity:** l'insieme di classi presenti nel pacchetto Model.
- **Database:** si occupa dell'interazione col database.

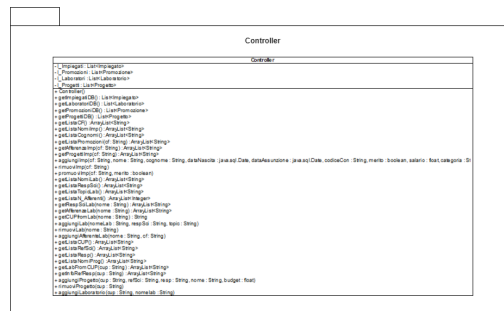


Figura 6 Package Controller

3.7 Dominio della soluzione del problema

Dopo l'analisi dei requisiti e dopo aver definito i pacchetti utili ai requisiti richiesti si può mostrare il dominio della soluzione:

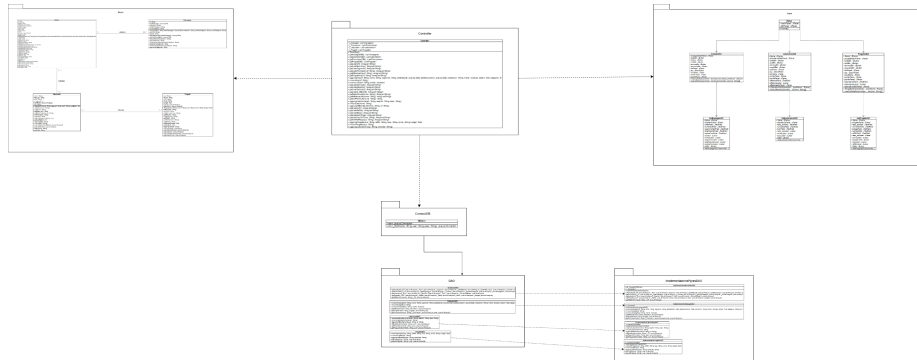


Figura 7 Dominio della soluzione

Capitolo 4

Sequence diagram

4.1 Rimuovi Laboratorio

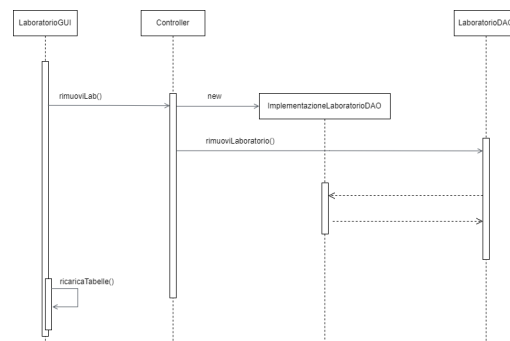


Figura 8 Sequence Diagram per la rimozione di un Laboratorio

4.2 Aggiungi Laboratorio

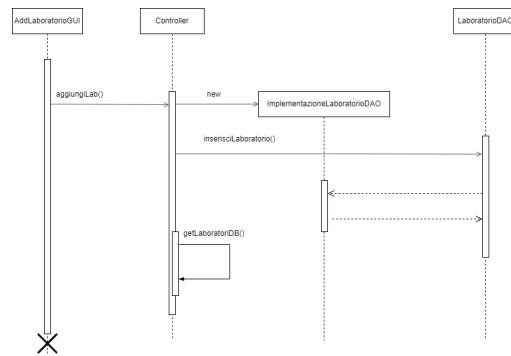


Figura 9 Sequence Diagram per aggiungere un Laboratorio