UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Informatica

Documento dell'applicativo software

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

Anno Accademico 2022/2023

Autori
Tortora Salvatore, Terrecuso Francesco
Relative matricole: N86004033, N86004191

Indice

1 Modello Del Dominio

1.1 Analisi dei Requisiti

In questa sezione si analizza la richiesta del cliente, con lo scopo di definire le funzionalità che la base di dati deve soddisfare. Individueremo le entità e le associazioni del mini-word, per rappresentare in modo esaustivo e chiaro il dominio del problema.

> *Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione del personale di un'azienda. L'azienda possiede un certo numero di impiegati, raggruppabili in 4 categorie:*
>
> - *Dipendente Junior : colui che lavora da meno di tre anni nell'Azienda.*
> - *Dipendente Middle : colui che lavora da più di tre anni nell'Azienda. ma meno di sette.*
> - *Dipendente Senior : colui che lavora da più di sette anni nell'Azienda.*
> - *Dirigenti: 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.*

L'incipit della richiesta è chiara : fornire una base di Dati per la Gestione del personale di un Azienda. Le Entità principali da rappresentare nel nostro mini-word sono le seguenti : L'Impiegato e le sue specializzazioni, in particolare, l'impiegato Junior, Middle e Senior raggruppabili in un'unica classe Impiegato con attributo 'tipoImpiegato' La prima nota che risalta dalla richiesta del cliente è quella di dividere la tipologia di Impiegato per anni di appartenenza all'azienda specificando che gli scatti possibili di carriera sono svincolati dalle competenze effettive dell'impiegato e sono sanciti solamente dagli anni di lavoro. L'entità Dirigente fa eccezione da tali specializzazioni in quanto il suo ruolo non dipende dal tempo di appartenenza all'Azienda bensì dalle capacità ben specifiche.

> *È necessario tracciare tutti gli scatti di carriera per ogni dipendente.*

La richiesta è modellabile con una classe Storico in cui salvare ogni scatto di carriera per ogni impiegato, la data e il passaggio di ruolo; Oltre alla rappresentazione degli scatti di carriera per le specializzazioni junior, middle e Senior, bisogna considerare anche gli scatti Dirigenziali e, siccome la richiesta non è specifica a riguardo, consideriamo che un Impiegato può avere più scatti Dirigenziali.

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.

Abbiamo la presentazione di due nuove Entità del mini-word : Laboratorio e Progetto. L'attenzione è focalizzata sull'Entità Laboratorio, ad esso afferiscono gli Impiegati, ciò suggerisce l'associazione fra le due entità e si suppone che un impiegato può afferire a più laboratori; E' specificato anche che oltre gli afferenti, un laboratorio ha un responsabile scientifico il quale è Senior. Siccome la richiesta non è specifica a riguardo, si considera il responsabile scientifico univoco per ogni laboratorio, in quanto è specializzato nel topic di quest'utlimo.

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.

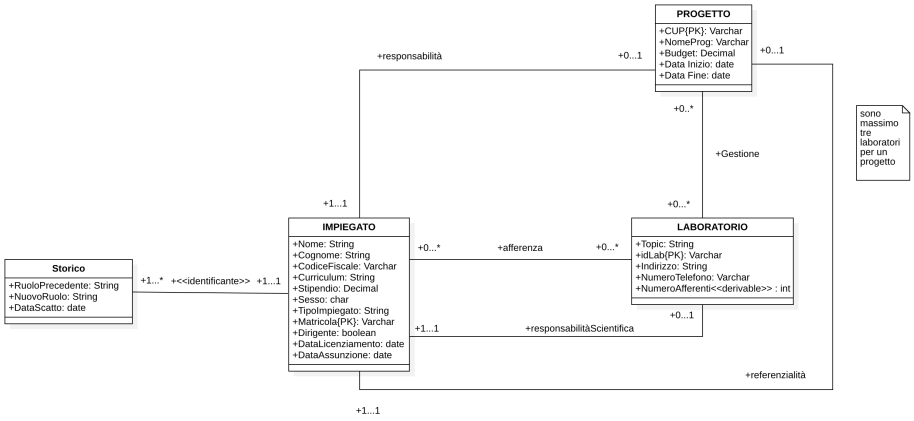
Il focus è spostato sulla nuova entità Progetto univocamente determinata da un attributo CUP e da un nome unico nel sistema. Da questa sezione riusciamo a capire che ogni progetto ha referente scientifico (Senior), e un responsabile (dirigente). Siccome la richiesta è vaga e non specifica, si suppone che un referente scientifico può essere associato ad un unico progetto, in equal modo un Dirigente può essere responsabile di un unico Progetto. Ciò suggerisce ovviamente le associazioni fra queste entità.

Al massimo 3 laboratori possono lavorare ad un progetto.

Questo punto è essenziale nel comprendere la relazione che esiste fra le entità Progetto e Laboratorio, specificandone i vincoli di gestione. Tuttavia la richiesta rimane vaga, motivo per il quale si consideri che un laboratorio può essere gestito da più progetti, mentre un progetto può gestire al più tre laboratori.

1.2 Dominio del Problema Generale

In seguito alle considerazioni espresse nella sezione precedente, si è prodotto il seguente schema concettuale espresso mediante diagramma UML:



2 Dizionari

2.1 Dizionario delle Entità

Entità	Descrizione	Attributi
Impiegato	Entità che fa parte del personale dell'Azienda	**Matricola**(Stringa): Identificante di un impiegato. **Nome**(Stringa): Nome di un impiegato. **Cognome**(Stringa): Cognome di un impiegato. **Codice Fiscale**(Stringa): codice fiscale di un impiegato. **Curriculum**(Stringa): Descrive le competenze di un Impiegato. **Stipendio**(Stringa): Descrive la somma complessiva dello stipendio. **Sesso**(Carattere): Specifica il sesso di un Impiegato. **tipo_impiegato**(stringa):rappresenta il ruolo attuale dell'impiegato. **Dirigente**(booleano):indica se attualmente l'impiegato è dirigente oppure no. **Data assunzione**(data): Descrive la data di assunzione di un Impiegato. **Data licenziamento**(Data): data di licenziamento dell'impiegato.
Storico	Entità Debole in cui si salvano tutti gli scatti di carriera di un impiegato	**Ruolo precedente**(Stringa): ruolo precedente allo scatto. **Nuovo ruolo**(Stringa): il ruolo di scatto. **Data scatto**(data): Data dello scatto di ruolo.

Entità	Descrizione	Attributi
Afferenza	Entità ad associazione tra Laboratorio e Impiegato.	
Laboratorio	Luogo in cui lavorano gli Impiegati	**Topic**(Stringa): Indica la materia trattata nel laboratorio. **Indirizzo**(Stringa): Indica la locazione del laboratorio. **Numero Telefononico**(intero): Descrive il riferimento telefonico. **Numero Afferenti**(Intero): Attributo derivato che descrive quanti afferenti ha un laboratorio.
Progetto	Ideazione e progettazione di un'Idea lavorativa	**CUP**(stringa): Codice Univoco del Progetto. **Nome Progetto**(stringa): Il nome del progetto unico nel sistema. **Budget**(Decimale):Capitale investito nel progetto. **Data Inizio**(data): data di avvio del progetto. **Data Fine**(data): data di fine progetto.

2.2 Dizionario delle Associazioni

Associazione	Tipologia	Descrizione
Afferenza	Molti-a-Molti	Associa ad un impiegato all'insieme di laboratori a cui lavora.
Responsabilità	Uno-a-Uno	Associa un Dirigente al suo Progetto.
Responsabilità scientifica	Uno-a-Uno	Associa un laboratorio al proprio responsabile scientifico.
Referenzialità	Uno-a-Uno	Associa un referente al suo progetto.
Gestione	Molti-a-Molti	Un laboratorio può essere gestito da più progetti, un progetto gestisce al massimo tre laboratori.
Identificante	Uno-a-Molti	Associa un Impiegato ai suoi scatti di carriera.

3 Model

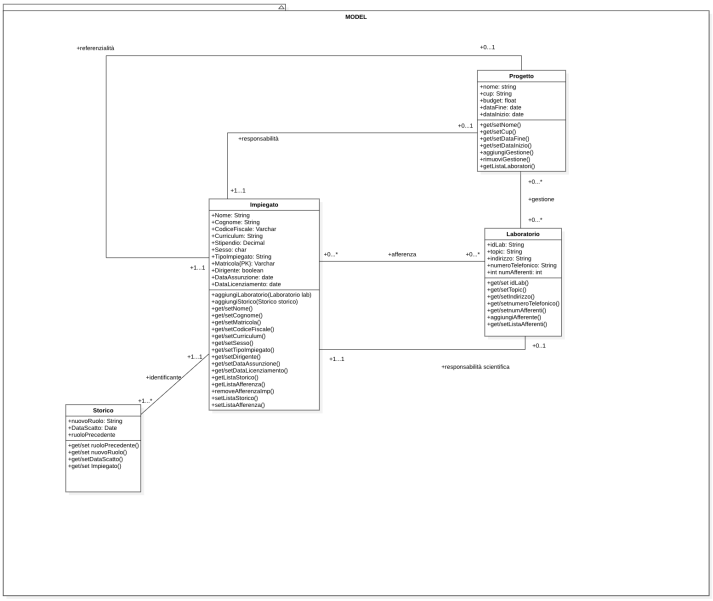
3.1 Traduzione delle classi e delle relazioni

Finito il processo in cui si analizza il dominio del problema, possiamo procedere col *mapping* di classi e associazioni passando alla creazione del Model del nostro problema, descrivendo il modo in cui gestisco le varie associazioni.

- **Identificante**: è un tipo di associazione *uno-a-molti totale*, motivo per il quale si è deciso di mappare tale associazione usando nella classe Storico il riferimento ad un Impiegato mentre per ogni Impiegato possiede un lista di riferimenti agli oggetti della classe Storico.
- **Afferenza**: per l'afferenza fra le entità Laboratorio e Impiegato,siccome la relazione è *molti-a-molti* si è deciso di implementare tale relazione attraverso due liste all'interno delle due corrispettive classi; Per ogni oggetto della classe Impiegato contiene una lista di rifermenti a laboratori a cui riferisce, mentre la classe Laboratorio ha una lista di riferimenti agli impiegati che afferiscono.
- **Gestione**: per l'afferenza fra le entità Laboratorio e Progetto,siccome la relazione è *molti-a-molti* si è deciso di implementare tale relazione attraverso due liste all'interno delle due corrispettive classi; Per ogni oggetto della classe laboratorio contiene una lista di rifermenti a progetti a cui riferisce, mentre la classe Progetto ha una lista di riferimenti ai Laboratori che gestisce.
- **Laboratorio**: La classe laboratorio contiene data l'associazione di responsabile scientifico(uno-a-uno), un riferimento ad un oggetto della classe Impiegato che definisce il responsabile scientifico; Oltreciò vengono rappresentati anche tutti gli attributi definiti all'interno del dizionario nella sezione 2.
- **Impiegato**: La classe impiegato oltre a contenere gli attributi descritti nel dizionario contiene dunque la lista degli storici ad esso riferito e la lista di laboratori a cui afferisce.
- **Progetto** : La classe progetto contiene oltre agli attributi descritti in precedenza nel dizionario, anche la lista di laboratori che gestisce, il riferimento all'oggetto impiegato che ne è referente scientifico e infine il riferimento all'oggetto impiegato che ne è responsabile.
- **Storico** : lo storico contiene oltre ai propri attributi anche il riferimento all'oggetto impiegato a cui si riferisce.

3.2 Package Model Diagram:

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



4 Comunicazione con il Database

Per la comunicazione con il Database dell'Azienda, si è scelto di usare il pattern architetturale **DAO** (Data Access Object) .
Il pattern serve alla gestione della persistenza dei dati e di gestire gli accessi a quest'ultimi; viene quindi utilizzato per fornire un'interfaccia unificata per l'accesso ai dati, in modo che l'applicazione non debba preoccuparsi di come i dati vengono archiviati o recuperati.

4.1 Connessione al Database

La connessione al database aziendale avviene all'interno del package Connessione **DBconnection** il quale contiene la classe Connessione che si occupa a creare un' istanza di connessione al database Postgres che si è scelto di tenere Online, in modo tale da rendere l'applicativo eseguibile su più calcolatori, senza neccessità di costruire in locale la propria base di dati; (per maggiori informazioni sul database guardare nella cartella del progetto, vi è una sottocartella contenenti l'implementazione della base di dati utilizzata.)

5 Package DAO:

5.0.1 Interfaccia AziendaDAO

Si è deciso di discernere le funzioni che fanno il load dei dati iniziali dalle subroutine usate durante l'esecuzione del programma; In particolare l'interfaccia Azienda si occupa di recuperare dal database i dati iniziali. Essa mette quindi a disposizione le funzioni che recuperano dalla base di dati i relativi Impiegati, Storici, Laboratori e Progetti.
Si vuole precisare che le afferenze e le gestioni non vengono caricate al momento di apertura del Software ma solamente quando richieste dall'utente, in questo modo si allegerisce il carico iniziale dei dati.

5.0.2 Interfaccia ImpiegatoDAO

L'interfaccia ImpiegatoDAO mette a disposizione le routine di aggiunta, eliminazione, modifica e lettura per un Impiegato.

5.0.3 Interfaccia LaboratorioDAO

L'interfaccia LaboratorioDAO mette a disposizione le funzioni di eliminazione, modifica, lettura e aggiunta di un laboratorio nel database.

5.0.4 Interfaccia ProgettoDAO

La seguente interfaccia, a differenza di quella del laboratorio, mette a disposizione oltre alle funzioni di modifica, anche quelle per aggiungere delle gestioni rispetto a qualche Laboratorio; In altre parole, è il progetto a scegliere quali sono i laboratori da gestire.

5.0.5 Interfaccia StoricoDAO

Nel software, il dump dei dati iniziali è fatto dall'Interfaccia DAO e, siccome la creazione e le eventuali modifiche degli storici sono automatizzate nel Database, l'interfaccia StoricoDAO non implementa alcun metodo, esso è stato aggiunto per completezza al package DAO.

6 Implementazioni del DAO

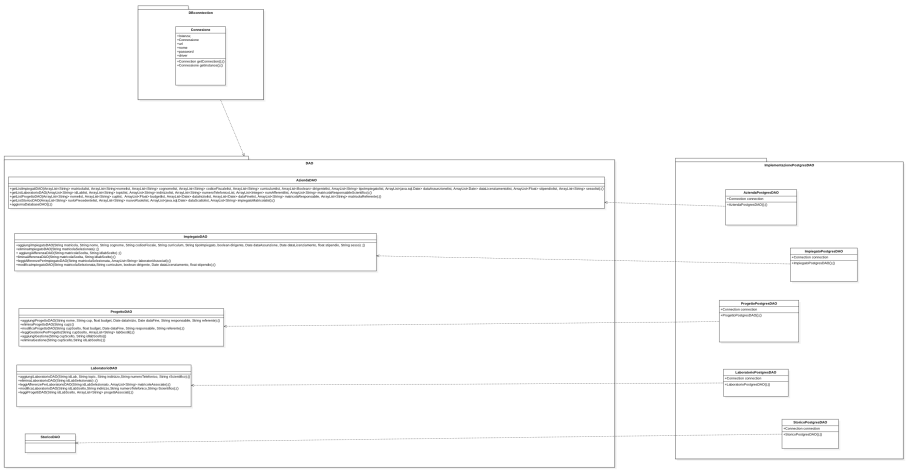
Le interfacce nel package DAO sono poi implementante per un database Postgres nel package **ImplementazioniPostgresDAO**.
In particolare le classi ImpiegatoPostgresDAO, AziendaPostgresDAO, LaboratorioPostgresDAO, ProgettoPostgresDAO e StoricoPostgresDAO definisco in che modo le varie query che vengono fatte al database e descrivono dunque la comunicazione fra il nostro software e il database utlizzato.

6.1 Exception handling

Si è pensato di gestire le eccezioni provenienti dai vari vincoli imposti sui dati nel database, in modo da inviare gli errori alla GUI e dunque all'utente. Un esempio è quello di inserire un impiegato con una data Licenziamento precedente alla data di assunzione; Siccome il Database ha il compito di tenere consistenti i dati, sarà esso a generare l'eccezione con corrispettivo messaggio di errore che viene catturato e poi visualizzato direttamente dall'utente, al quale è affidato il compito di gestire l'errore che lui stesso ha generato (in questo caso aggiustando la data di assunzione).

7 Package Diagram DAO

IL seguente Diagramma illustra l'architettura Dao utilizzata all'interno del software e designa quali sono i package, le classi e i metodi in uso e le loro relazioni:



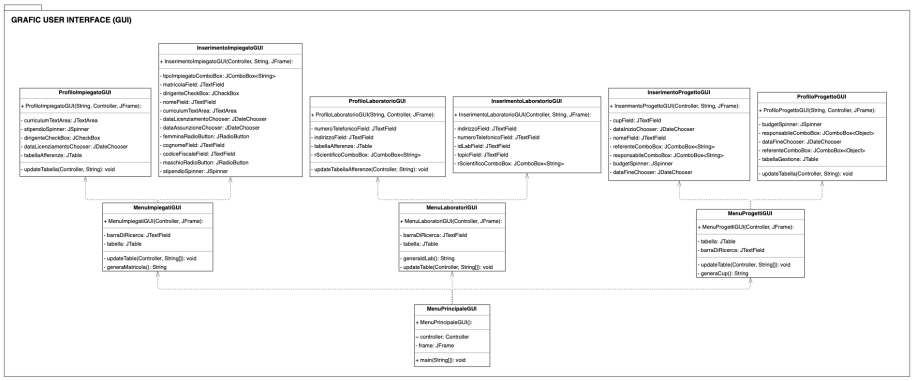
8 Graphical User Interface

In questa sezione viene descritto il modo in cui l'interfaccia grafica è stata implementata.
Inazitutto è stato scelto come framework Swing, appartenente alle Java Foundation Classes (JFC). L'interfaccia utente è implementata nel package GUI e i file vengono gestiti attraverso questo pattern :

- **Menu Principale** : è il menu dove scegliere quali entità fra Impiegato, Laboratorio e Progetto, si vuole visualizzare.
- **MenuImpiegati, MenuProgetti, MenuLaboratori**: rappresentano la lista di Laboratori o Impiegati o Progetti dell'Azienda; In questa parte della gui è possibile visualizzare tali liste. OltreModo è possibile accedere in modo dettagliato ai dati per ogni entità cliccando sul bottone ' Proflio ... ' o anche eliminare un entità cliccando su ' elimina '.
- **InserimentoImpiegato, InserimentoProgetto, InserimentoLaboratorio** è la scheramta in cui inserire i dati per creare una nuova entità all'interno dell'Azienda; Questa schermata è accessibile attraverso il bottone 'Inserisci' all'interno dei menu specificati sopra.
- **ProfiloImpiegato, ProfiloLaboratorio, ProfiloProgetto** è la schermata in cui vengono analizzati tutti i dati in modo spefico per un entità scelta. In questa sezione della GUI è possibile modificare gli attributi, aggiungere afferenze (tra laboratorio e impiegato), visualizzare lo storico di impiegato, aggiungere una gestione (tra un laboratorio e un progetto) [...]
  Per salvare le modifiche attuate basta cliccare sul bottone 'salva modifiche', altrimenti se si vuole ritornare al menu di partenza basta cliccare 'annulla'.

8.1 Package GUI diagram

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



9 Design pattern BCED

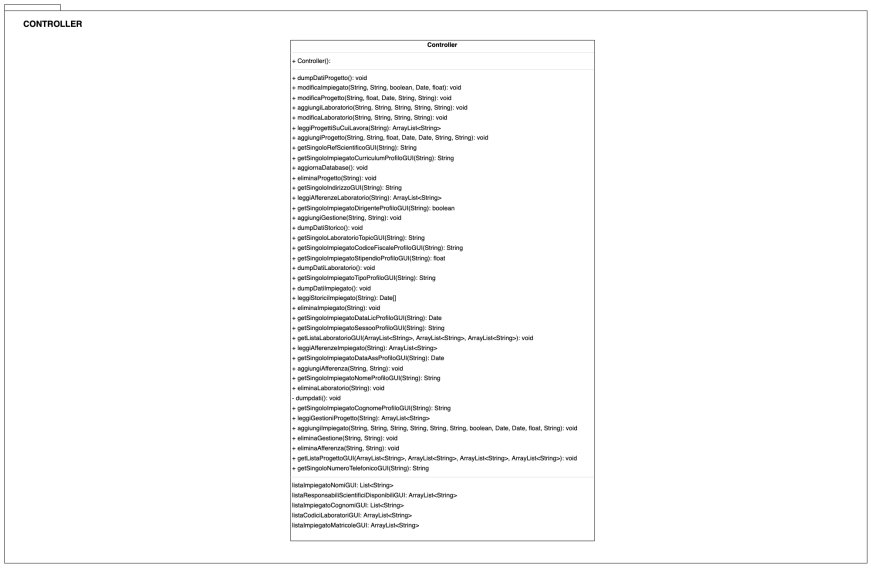
Prima di proseguire con il dominio della soluzione bisogna identificare il pattern utlizzato per creare l'appplicativo Java; In questo caso è stato scelto di gestire tale problema attraverso un approccio BCED;
Il design pattern citato sta ad indicare *Entity-controller-boundary-database*.

- **Entity**: l'insieme di classi che riproduce il diagramma del dominio delle informazioni da rappresentare, in questo caso è il nostro package MODEL.
- **controller** : l'insieme di classi che realizzano le operazioni algoritmiche, su richiesta diretta dalla GUI e gestendo i dati rappresentati nel Model e nel database.
- **Boundary** : Sta ad indicare il package GUI, in altre parole l'applicativo grafico sul quale interagisce l'utente.
- **Database** : è l'insieme del nostro package DAO che serve a gestire i dati del Database

L'esigenza dell'utilizzo di questo pattern è collegata al fatto di rendere l'intera progettazione più chiara. In particolare l'utilizzo di un design pattern favorisce il riuso del codice e la sua manutenzione.

9.1 Package Controller:

Il seguente diagramma illustra il package Controller con la sua relativa classe che rappresenta il cuore del nostro applicativo; In particolare qualsiasi mediazione fra Database, model e GUI è gestita attraverso i metodi implementati nel controller.

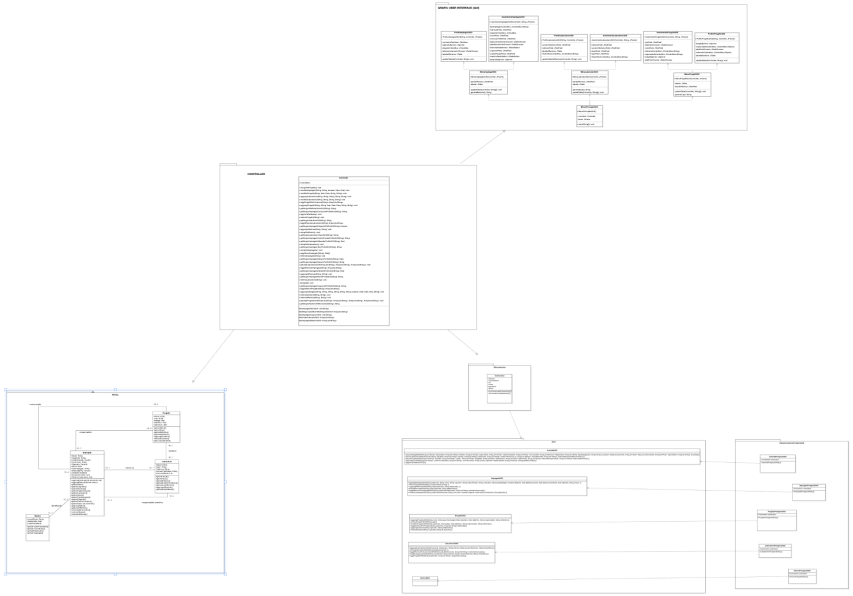


10 Dominio della Soluzione

In seguito a tutta l'implementazione descritta nelle sezioni precedenti e i vari design pattern usati, è allora possibile ora descrivere il dominio della soluzione.

10.1 Solution Diagram

Di seguito è riportato l'UML del dominio della soluzione.

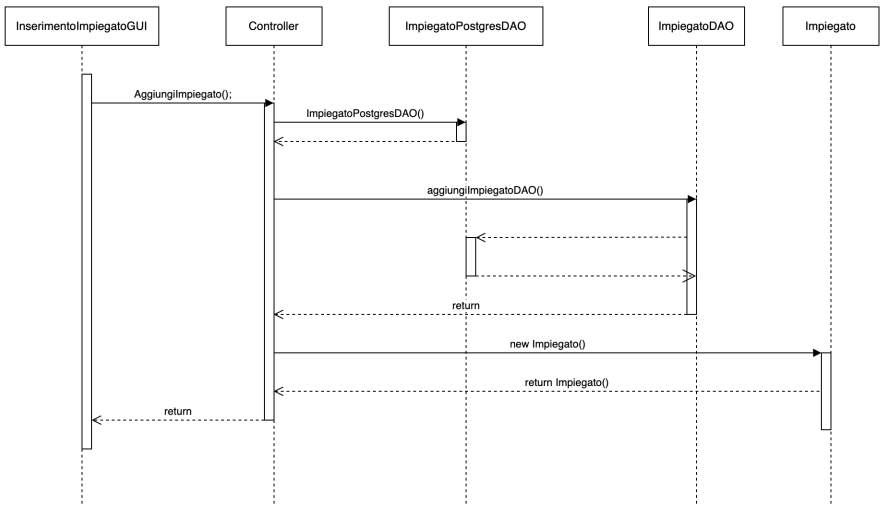


11 Sequence Diagram

IN quest'ultima sezione vengono analizzati due diagrammi di sequenza per descrivere il flusso di computazione quando due operazini in particolare vengono svolte;

11.1 Sequence Diagram Inserimento Impiegato :

Il seguente diagramma descrive in modo dettagliato il flusso del programma alla richiesta dell'utente di aggiungere un impiegato nell'Azienda.



11.2 Sequence Diagram Aggiungi Afferenza

Il seguente diagramma descrive invece, quando richiesto dall'utente, in che modo viene gestito l'aggiunta di una nuova afferenza fra un Impiegato ed un Laboratorio.