

Corso di Ingegneria del Software Deliverable di progetto	2022-2023
---	-----------

# “Ingegneria del Software” 2022-2023

**Docente: Prof. Angelo Furfaro**

## <Organigramma Aziendale>

<b>Data</b>	<22/06/2023>
<b>Documento</b>	Documento Finale – D3

<b>Team Members</b>		
<b>Nome e Cognome</b>	<b>Matricola</b>	<b>E-mail address</b>
Simone Chiarelli	223473	chrsmn01s14f537b@studenti.unical.it

# Sommario

---

List of Challenging/Risky Requirements or Tasks .....	3
A. Stato dell'Arte .....	4
B. Raffinamento dei Requisiti.....	6
<i>A.3 Scenari d'uso dettagliati</i> .....	11
<i>A.4 Excluded Requirements</i> .....	12
<i>A.5 Assunzioni</i> .....	13
<i>A.6 Use Case Diagrams</i> .....	15
C. Architettura Software.....	17
E. Scelte Progettuali (Design Decisions).....	27
Struttura Gerarchica Organigramma: Composite.....	27
Interfaccia Grafica: Observer .....	28
Raccolta Informazioni Organigramma: Visitor .....	29
Punto Accesso Globale per Informazioni Azienda: Singleton .....	30
Gestione persistenza dei dati.....	31
F. Progettazione di Basso Livello .....	32
G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs) .....	38
Appendix. Prototype .....	41

***List of Challenging/Risky Requirements or Tasks***

<b>Challenging Task</b>	<b>Date the task is identified</b>	<b>Date the challenge is resolved</b>	<b>Explanation on how the challenge has been managed</b>
Creazione di un modello per la gestione della logica dell'organigramma.	2/06	8/06	Impiego del pattern Composite al fine di creare un modello per la gestione di Unità, Impiegati e Ruoli definiti nell'organigramma.
Creazione di un'interfaccia grafica apposita finalizzata all'interazione utente-applicazione.	3/06	13/06	Utilizzo di librerie apposite di proprietà del linguaggio al fine di creare una barra degli strumenti attraverso la quale l'utente sia in grado di modificare lo stato corrente dell'organigramma
Creazione di un modello di visualizzazione dinamico in grado di aggiornarsi al variare dello stato interno dell'organigramma	4/06	16/06	Impiego del pattern Observer al fine di realizzare viste in ascolto per eventuali cambiamenti dello stato e in grado di aggiornarsi in funzioni di esse.
Creazione di un modello finalizzato alla persistenza dei dati, in grado di consentire salvataggio e caricamento dei dati da memoria secondaria	5/06	18/06	Utilizzo del formato json tramite apposite librerie proprietarie del linguaggio, al fine di consentire la memorizzazione dello stato interno corrente dell'organigramma.
Creazione di un meccanismo di visualizzazione delle informazioni statistiche relative allo stato dell'organigramma.	5/06	17/06	Impiego del Pattern Visitor al fine di effettuare operazioni di visita alla struttura interna dell'organigramma prelevando informazioni circa lo stato dello stesso.
Creazione di una classe Azienda che racchiuda lo stato totale corrente dell'organigramma.	9/06	15/06	Impiego del Pattern Singleton al fine di definire un solo punto di accesso globale per Azienda.
Stesura deliverable di progetto	1/06	20/06	Realizzazione relazione inerente al progetto.

## A. Stato dell'Arte

---

L'espressione organigramma è usata per intendere un grafico con il quale si rappresenta visivamente la struttura organizzativa di un'impresa, esso permette di chiarire, quali siano gli organi dell'azienda ed il tipo di relazioni esistenti tra tali organi. Nell'organigramma si usano dei rettangoli al fine di indicare gli organi e delle linee per indicare le relazioni esistenti tra i vari organi.

Esistono svariate applicazioni software volte alla creazione di organigrammi aziendali.

Di seguito si analizzeranno le più comuni funzionalità delle applicazioni volte alla realizzazione di organigrammi aziendali, ponendo particolare enfasi sulla logica di visualizzazione della struttura e sulle operazioni basilari che un prototipo di applicazione dovrebbe garantire.

1. *Rappresentazione dell'organigramma*: Le moderne applicazioni software per organigrammi aziendali offrono una vasta gamma di opzioni per la rappresentazione visuale dell'organigramma. Queste includono diagrammi gerarchici, grafici a blocchi, alberi genealogici e organigrammi interattivi. Alcune applicazioni consentono anche la personalizzazione dell'aspetto dell'organigramma, come lo stile, i colori, le icone e i collegamenti ipertestuali.
2. *Interfaccia utente intuitiva*: Le applicazioni per organigrammi aziendali si concentrano sempre di più sull'usabilità e sull'esperienza utente. L'interfaccia utente dovrebbe essere intuitiva, facile da navigare e consentire l'interazione diretta con l'organigramma. Funzionalità come il trascinamento degli elementi, il ridimensionamento, l'aggiunta di collegamenti e l'organizzazione dinamica dei nodi dell'organigramma sono diventate comuni.
3. *Collaborazione e condivisione*: Le moderne applicazioni consentono la collaborazione in tempo reale su organigrammi aziendali. Piattaforme basate su cloud permettono a più utenti di lavorare contemporaneamente sullo stesso organigramma, consentendo la modifica, l'annotazione e i commenti in tempo reale. Inoltre, è possibile condividere l'organigramma con altre persone attraverso link, inviti o esportazione in diversi formati (ad esempio PDF, immagini, documenti).
4. *Integrazione con altri sistemi*: Le applicazioni per organigrammi aziendali possono integrarsi con altri sistemi utilizzati all'interno dell'azienda. Ad esempio, è possibile sincronizzare l'organigramma con il sistema di gestione delle risorse

umane (HRM) per ottenere automaticamente informazioni sui dipendenti, i ruoli e la struttura organizzativa. L'integrazione con strumenti di produttività come le suite per ufficio o le piattaforme di collaborazione può semplificare la creazione e l'aggiornamento dell'organigramma.

5. *Funzionalità avanzate:* Alcune applicazioni offrono funzionalità avanzate come la generazione automatica dell'organigramma a partire da dati strutturati, la possibilità di filtrare e visualizzare diverse viste dell'organigramma (ad esempio per dipartimenti o posizioni), l'analisi dei dati organizzativi (ad esempio attraverso grafici e statistiche) e l'integrazione con strumenti di analisi e-business intelligence.
  6. *Accessibilità e mobilità:* Un'applicazione moderna per organigrammi aziendali dovrebbe essere accessibile da diverse piattaforme, inclusi dispositivi mobili come smartphone e tablet. Le applicazioni basate su web o le app mobili consentono agli utenti di accedere e interagire con l'organigramma da qualsiasi luogo e in qualsiasi momento.
  7. *Sicurezza e privacy:* Le applicazioni per organigrammi aziendali dovrebbero garantire la sicurezza e la privacy dei dati.
-

## B. Raffinamento dei Requisiti

---

L'organigramma rappresenta la struttura gerarchica dell'azienda, con le diverse unità organizzative e i ruoli dei dipendenti all'interno di esse.

L'obiettivo principale è tradurre le esigenze e le funzionalità richieste dagli utenti in requisiti specifici che consentano di gestire e manipolare l'organigramma aziendale in modo efficiente.

L'organigramma dell'azienda può essere considerato come una struttura dati gerarchica, in cui le unità organizzative sono rappresentate come nodi e le relazioni gerarchiche tra di esse sono rappresentate come collegamenti tra i nodi.

È necessaria una chiara comprensione delle operazioni che devono essere supportate, come la modifica del nome delle unità, l'aggiunta e la rimozione di unità, l'aggiunta e la rimozione di impiegati, l'assegnazione di ruoli agli impiegati.

È necessario identificare e definire in modo preciso i requisiti funzionali, che riguardano le operazioni specifiche che l'applicazione deve supportare, come l'aggiunta di una sotto unità, la rimozione di un'unità o l'assegnazione di un impiegato a un'unità con un determinato ruolo.

Inoltre, è importante considerare anche i requisiti non funzionali che verranno scelti per l'applicazione tra tutti quelli possibili, quali la sicurezza dei dati, la scalabilità dell'applicazione, la facilità d'uso e l'efficienza delle operazioni.

---

### A.1 Servizi (con prioritizzazione)

---

L'applicazione offre una serie di servizi che permettono di interagire con l'applicazione al fine di creare un organigramma aziendale.

1. **Modifiche allo stato dell'organigramma** (alta priorità, media complessità): è necessario realizzare una struttura interconnessa tra le varie entità dell'organigramma in modo tale che la modifica di una si ripercuota sullo stato delle altre.
  - a. Modifica Nome Unità:
    - i. L'utente deve essere in grado di selezionare un'unità esistente.
    - ii. L'utente deve poter inserire un nuovo nome per l'unità selezionata.
    - iii. L'applicazione deve salvare il nuovo nome dell'unità.
  - b. Aggiungi Sotto Unità:
    - i. L'utente deve essere in grado di selezionare un'unità esistente a cui aggiungere una sotto unità.
    - ii. L'utente deve poter inserire i dettagli della nuova sotto unità, come il nome.
    - iii. L'applicazione deve creare una nuova sotto unità collegata all'unità selezionata.
  - c. Rimuovi Unità:
    - i. L'utente deve essere in grado di selezionare un'unità esistente da rimuovere.
    - ii. L'applicazione deve verificare se l'unità selezionata ha sotto unità.
    - iii. Se l'unità ha sotto unità, l'applicazione deve impedire la rimozione e fornire un messaggio di avviso.
    - iv. Se l'unità non ha sotto unità, l'applicazione deve eliminare l'unità selezionata.
  - d. Aggiungi Impiegato Azienda:
    - i. L'utente deve poter inserire i dettagli del nuovo impiegato, come il nome e le informazioni personali.
    - ii. L'applicazione deve creare un nuovo impiegato all'interno dell'azienda.
  - e. Rimuovi Impiegato Azienda:
    - i. L'utente deve essere in grado di selezionare un impiegato esistente da rimuovere.
    - ii. L'applicazione deve eliminare l'impiegato selezionato dall'azienda.

- f. Assegna Impiegato ad Unità:
  - i. L'utente deve poter selezionare un impiegato esistente.
  - ii. L'utente deve poter selezionare un'unità esistente a cui assegnare l'impiegato.
  - iii. L'utente deve poter selezionare un ruolo consentito per l'unità selezionata.
  - iv. L'applicazione deve assegnare l'impiegato all'unità con il ruolo selezionato.
- g. Rimuovi Impiegato da Unità:
  - i. L'utente deve poter selezionare un'unità esistente dalla quale rimuovere l'assegnamento di un impiegato.
  - ii. L'utente deve poter selezionare l'impiegato da rimuovere dall'unità.
  - iii. L'applicazione deve rimuovere l'assegnamento dell'impiegato dall'unità.
- h. Crea Ruolo:
  - i. L'utente deve poter inserire i dettagli del nuovo ruolo, come il nome e la descrizione.
  - ii. L'applicazione deve creare un nuovo ruolo all'interno dell'azienda.
- i. Rimuovi Ruolo:
  - i. L'utente deve poter selezionare un ruolo esistente da eliminare.
  - ii. L'applicazione deve verificare se ci sono impiegati assegnati a quell'unità con il ruolo selezionato.
  - iii. Se ci sono impiegati assegnati solo a quel ruolo nell'unità, l'applicazione deve rimuovere l'assegnamento di quegli impiegati dall'unità.
  - iv. L'applicazione deve eliminare il ruolo selezionato.
- j. Assegna Ruolo ad Unità:
  - i. L'utente deve poter selezionare un ruolo esistente.
  - ii. L'utente deve poter selezionare un'unità esistente a cui assegnare il ruolo.
  - iii. L'applicazione deve assegnare il ruolo all'unità selezionata.
- k. Rimuovi Ruolo da Unità:
  - i. L'utente deve poter selezionare un'unità esistente dalla quale rimuovere l'assegnamento di un ruolo.
  - ii. L'utente deve poter selezionare il ruolo da rimuovere dall'unità.
  - iii. L'applicazione deve rimuovere l'assegnamento del ruolo dall'unità.



## 2. Visualizzazione grafica dello stato corrente dell'organigramma

- a. Visualizzazione dell'organigramma con rettangoli (media priorità, alta complessità): *è necessario realizzare una struttura grafica in grado di adattarsi all'inserimento di entità nell'organigramma, in modo tale che continui a risultare chiara e leggibile da parte dell'utente, occorre evitare che vi siano sovrapposizioni.*
  - i. Le unità sono rappresentate come rettangoli
  - ii. Le unità sono collegate alle relative sotto unità mediante segmenti.
  - iii. In questa visualizzazione vengono mostrate solo le unità, non gli impiegati.
  - iv. La struttura grafica dell'organigramma si adatta dinamicamente all'inserimento e all'eliminazione delle unità.
- b. Visualizzazione dell'organigramma ad albero (alta priorità, bassa complessità): *essendo la visualizzazione incentrata esclusivamente sulle etichette associate alle varie entità, non è necessaria la gestione di complessi elementi grafici.*
  - i. È possibile visualizzare la struttura dell'organigramma come un albero.
  - ii. Selezionando una specifica unità, è possibile espandere il suo contenuto, se presente.
  - iii. vengono mostrate eventuali sotto unità ed impiegati dell'unità selezionata.
- c. Rappresentazione degli impiegati e dei loro ruoli (alta priorità, bassa complessità): *se le funzionalità associate alla modifica dell'organigramma risultano ben implementate, non diviene complicato realizzare un meccanismo di visualizzazione basilare per quanto riguarda la visualizzazione di impiegati e ruoli.*
  - i. Un impiegato all'interno di un'unità viene visualizzato come figlio dell'unità nella gerarchia.
  - ii. Nelle informazioni dell'impiegato sono indicati i ruoli che egli ricopre all'interno dell'unità.

## 3. Memorizzazione su memoria secondaria (media priorità, alta complessità): è necessario gestire le dipendenze che intercorrono tra le varie entità facente parti della struttura interna dell'organigramma, al fine di garantire una corretta memorizzazione di esse, e di conseguenza un corretto ripristino degli stati precedenti.

- a. Salvataggio dell'organigramma:

- i. Si consente di salvare lo stato corrente dell'organigramma in un file in formato valido persistente su memoria secondaria.
  - ii. Il file salvato può successivamente essere ricaricato nell'applicazione.
- b. Caricamento dell'organigramma:
  - i. Si consente di caricare un organigramma precedentemente salvato in un valido formato su memoria secondaria, precedentemente salvato dall'apposita funzione offerta dall'applicazione
- c. Ripristino dell'organigramma allo stato iniziale:
  - i. Consente di azzera l'organigramma, ripristinandolo allo stato iniziale.

## A.2 Requisiti non Funzionali

---

1. *Prestazioni*: L'applicazione deve essere in grado di gestire grandi quantità di dati e fornire risposte rapide agli utenti. L'interfaccia utente deve presentarsi reattiva e l'applicazione deve essere in grado di gestire l'aggiornamento in tempo reale dell'organigramma, anche con un numero elevato di dipendenti
2. *Scalabilità*: L'applicazione deve essere in grado di gestire un numero crescente di dipendenti e di adattarsi a future espansioni aziendali senza subire una significativa perdita di prestazioni. Deve essere possibile aggiungere o rimuovere dipendenti senza compromettere la stabilità o l'usabilità dell'applicazione.
3. *Affidabilità*: L'applicazione deve risultare stabile e non soggetta a malfunzionamenti frequenti. Deve essere in grado di gestire eventuali errori in modo elegante e ripristinarsi senza perdere dati importanti.
4. *Usabilità*: L'interfaccia utente dell'applicazione deve essere intuitiva e facile da usare, anche per utenti non esperti. Deve essere possibile navigare facilmente nell'organigramma aziendale, visualizzare informazioni dettagliate sui dipendenti e apportare modifiche in modo semplice.
5. *Portabilità*: L'applicazione deve essere in grado di funzionare su diverse piattaforme e ambienti, come ad esempio Windows, macOS o Linux. Deve essere indipendente dall'hardware e dalle configurazioni specifiche dell'ambiente di esecuzione.
6. *Manutenibilità*: Il codice dell'applicazione deve essere ben strutturato, modulare e facilmente comprensibile. Deve essere possibile apportare modifiche, effettuare

correzioni o estensioni senza causare effetti indesiderati su altre parti dell'applicazione

---

### *A.3 Scenari d'uso dettagliati*

---

Di seguito si riportano gli scenari d'uso più significativi per l'applicazione in esame, ulteriori casi d'uso relativi ad altre operazioni fornite dall'applicazione stessa possono essere ricondotti ai seguenti con piccole modifiche.

#### **Scenario 1: Creazione di una nuova unità**

*Descrizione:* Un utente desidera creare una nuova unità all'interno del sistema di gestione aziendale. L'utente accede alla piattaforma e seleziona l'opzione "Crea Sotto Unità". Vengono richiesti i dettagli dell'unità come il nome dell'unità che si desidera creare. Dopo aver inserito tutte le informazioni necessarie, l'utente conferma la creazione dell'unità. Il sistema verifica la validità dei dati e crea una nuova unità aziendale. L'utente riceve una conferma visiva dell'avvenuta creazione e potrà visualizzare la struttura dell'organigramma aggiornata.

#### **Scenario 2: Creazione di un nuovo ruolo**

*Descrizione:* Un utente vuole creare un nuovo ruolo per definire le responsabilità e le autorizzazioni dei dipendenti che possono ricoprire all'interno di un'unità. L'utente seleziona l'opzione "Crea nuovo ruolo". Vengono richiesti i dettagli del ruolo, come il nome e, dopo aver inserito tutte le informazioni necessarie, l'utente conferma la creazione del ruolo.

Il sistema crea un nuovo ruolo nel e l'utente riceve una conferma visiva dell'avvenuta creazione e l'elenco dei ruoli viene aggiornato al fine di includere quello appena creato

#### **Scenario 3: Creazione di un impiegato**

*Descrizione:* Un utente vuole creare un nuovo impiegato al fine di assegnarlo alle varie unità con i ruoli consentiti in esse. L'utente seleziona l'opzione "Crea nuovo impiegato". Vengono richiesti i dettagli dell'impiegato, come il nome e, dopo aver inserito tutte le informazioni necessarie, l'utente conferma la creazione dell'impiegato.

Il sistema crea un nuovo impiegato, l'utente riceve una conferma visiva dell'avvenuta creazione e l'elenco degli impiegati viene aggiornato al fine di includere quello appena creato.

#### **Scenario 4: Assegnazione di un dipendente ad un'unità**

*Descrizione:* Un utente desidera assegnare un dipendente a un'unità specifica all'interno dell'azienda. L'utente seleziona l'opzione "Assegna dipendente a unità".

Viene visualizzato un elenco di dipendenti disponibili e un elenco di unità all'interno dell'azienda e i relativi ruoli che in esse è possibile ricoprire. L'utente seleziona il dipendente desiderato, l'unità di destinazione e il ruolo che egli ricoprirà nell'unità.

Il sistema effettua l'assegnazione. L'utente riceve una conferma visiva dell'avvenuta assegnazione e l'elenco dei dipendenti e delle unità viene aggiornato di conseguenza.

#### **Scenario 5: Assegnazione di un ruolo ad un'unità**

*Descrizione:* Un utente desidera assegnare un ruolo a un'unità specifica all'interno dell'azienda. L'utente seleziona l'opzione "Assegna ruolo ad unità".

Viene visualizzato un elenco di ruoli disponibili e un elenco di unità all'interno dell'azienda a cui è possibile assegnare il ruolo. L'utente seleziona ruolo e l'unità di destinazione.

Il sistema effettua l'assegnazione. L'utente riceve una conferma visiva dell'avvenuta assegnazione e l'elenco dei ruoli consentiti per quell'unità viene aggiornato, in questo modo potrà essere, in un secondo momento, assegnato un impiegato a quell'unità con quel determinato ruolo.

---

### **A.4 Excluded Requirements**

Nella definizione dei requisiti per un'applicazione di organigramma, alcuni requisiti possono risultare escludibili in base alle specifiche dell'applicazione o alle esigenze degli utenti. Di seguito sono requisiti ritenuti escludibili per l'applicazione in esame:

1. *Funzionalità avanzate di formattazione:* l'applicazione è focalizzata principalmente sulla creazione e la visualizzazione di organigrammi, non è necessario fornire funzionalità di formattazione avanzate come stili di testo personalizzati, tabelle o colori personalizzati per le unità.
2. *Collaborazione in tempo reale:* l'applicazione è destinata a un utilizzo individuale e non richiede la collaborazione simultanea di più utenti; dunque, la funzionalità di collaborazione in tempo reale viene ritenuta escludibile.

3. *Integrazione con altri software aziendali*: l'applicazione non è destinata a essere integrata con altri sistemi aziendali o software; dunque, non risulta necessario considerare requisiti specifici di integrazione.
4. *Funzionalità di generazione automatica dell'organigramma*: l'applicazione si concentra sulla creazione manuale e la modifica dell'organigramma da parte degli utenti, non è necessario implementare funzionalità di generazione automatica dell'organigramma in base a criteri specifici.
5. *Monitoraggio delle modifiche e tracciabilità*: l'applicazione non richiede una tracciabilità completa delle modifiche apportate all'organigramma o un registro delle attività degli utenti; dunque, i requisiti relativi a questa funzionalità possono essere escludibili.
6. *Funzionalità di esportazione in formati specifici*: l'applicazione non richiede l'esportazione dell'organigramma in formati specifici come PowerPoint o Visio, i requisiti correlati a tali formati possono essere escludibili.
7. *Supporto per reti aziendali complesse*: l'applicazione è destinata a un utilizzo individuale o a piccoli gruppi di utenti; dunque, non è necessario considerare requisiti di supporto per reti aziendali complesse come autenticazione.

---

## A.5 Assunzioni

---

Durante lo sviluppo del progetto sull'applicazione dell'organigramma aziendale, sono state prese diverse decisioni e assunzioni riguardanti i requisiti più rilevanti.

Di seguito sono riportate le principali effettuate durante il progetto:

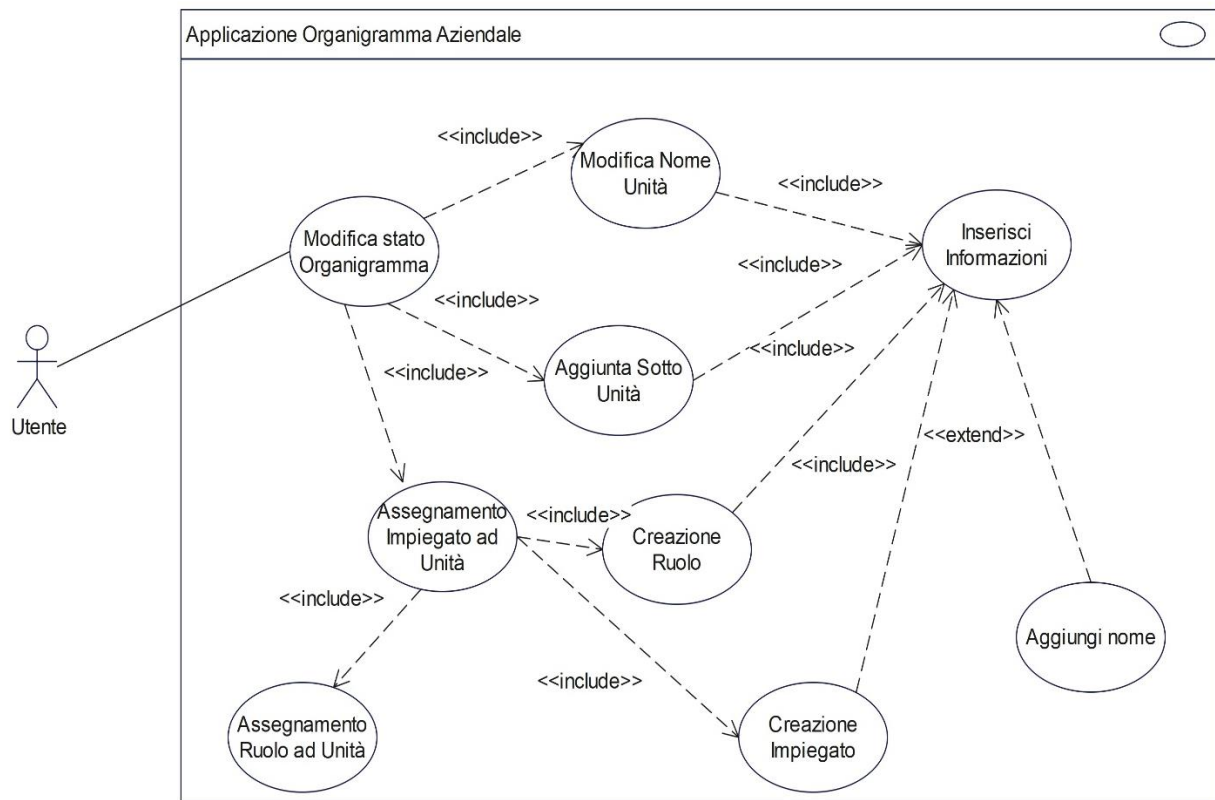
1. *Rappresentazione dell'organigramma*: Una delle decisioni cruciali riguardava la scelta della rappresentazione dell'organigramma all'interno dell'applicazione. Dopo un'attenta valutazione, è stato deciso di utilizzare una rappresentazione grafica basata su rettangoli e collegamenti tra unità. Questo approccio visivo offre una chiara visualizzazione della struttura gerarchica e delle relazioni tra le unità aziendali.
2. *Tipi di visualizzazione*: È stata presa la decisione di implementare due tipi di visualizzazione per l'organigramma: una basata su rettangoli e una visualizzazione

ad albero. Questa scelta è stata fatta per consentire agli utenti di scegliere la modalità di visualizzazione più adatta alle proprie esigenze e preferenze. La visualizzazione a rettangoli offre una rappresentazione più compatta e dinamica dell'organigramma, mentre la visualizzazione ad albero fornisce una panoramica più dettagliata della struttura gerarchica.

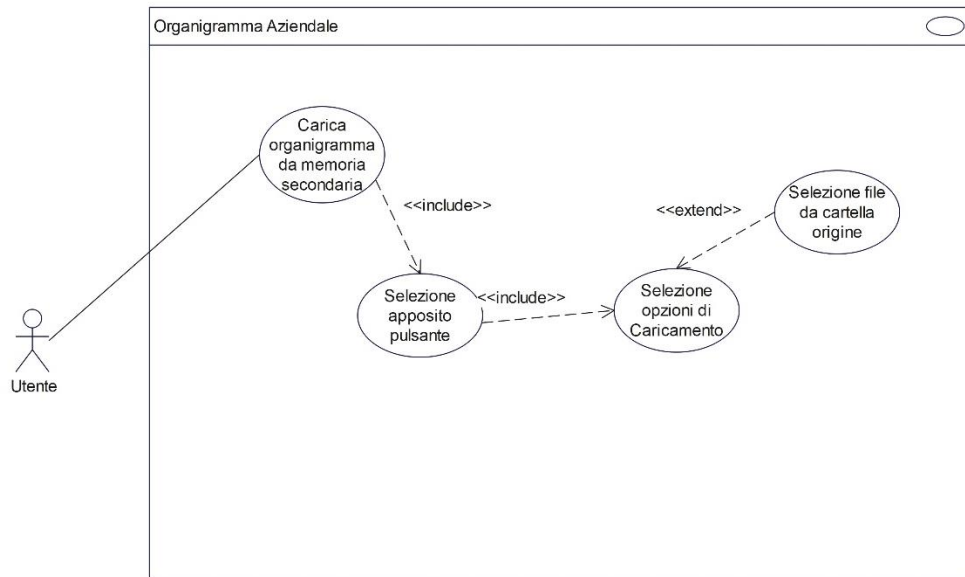
3. *Gestione delle unità*: È stata assunta la decisione di consentire agli utenti di aggiungere, modificare e rimuovere unità all'interno dell'organigramma. Inoltre, è stato deciso di consentire l'aggiunta di sotto unità per creare una gerarchia multilivello. Questa decisione è stata presa per fornire flessibilità e adattabilità all'applicazione, consentendo agli utenti di rappresentare in modo accurato la struttura organizzativa della propria azienda.
4. *Gestione degli impiegati*: È stata presa la decisione di consentire agli utenti di aggiungere, rimuovere e assegnare impiegati alle unità dell'organigramma. Inoltre, è stato deciso di associare ruoli specifici agli impiegati all'interno di ciascuna unità. Questa scelta è stata fatta per consentire una gestione più dettagliata dei ruoli e delle responsabilità all'interno dell'organigramma aziendale.
5. *Barra degli strumenti*: È stata presa la decisione di fornire una barra degli strumenti con una serie di comandi per facilitare l'interazione dell'utente con l'applicazione. I comandi includono la modifica dei nomi delle unità, l'aggiunta di sotto unità, la rimozione di unità, l'aggiunta e la rimozione di impiegati, l'assegnazione di impiegati alle unità, la creazione e la rimozione di ruoli, nonché l'assegnazione di ruoli alle unità. Questa decisione mirava a garantire un'interfaccia utente intuitiva e user-friendly, facilitando le operazioni di gestione dell'organigramma.
6. *Salvataggio e caricamento dell'organigramma*: È stata presa la decisione di consentire agli utenti di salvare lo stato corrente dell'organigramma in un file .json e di caricare organigrammi precedentemente salvati. Questo approccio consente agli utenti di conservare le proprie configurazioni personalizzate dell'organigramma e di ripristinare facilmente uno stato precedente dell'applicazione.
7. *Riguardo rimozione Unità*: è possibile effettuare la rimozione solo di unità che non abbiano sotto unità, in questo modo si previene l'eliminazione indesiderata di una intera porzione dell'organigramma. Al fine di eliminare un'unità in mezzo alla gerarchia, è necessario partire dai nodi foglia.

## A.6 Use Case Diagrams

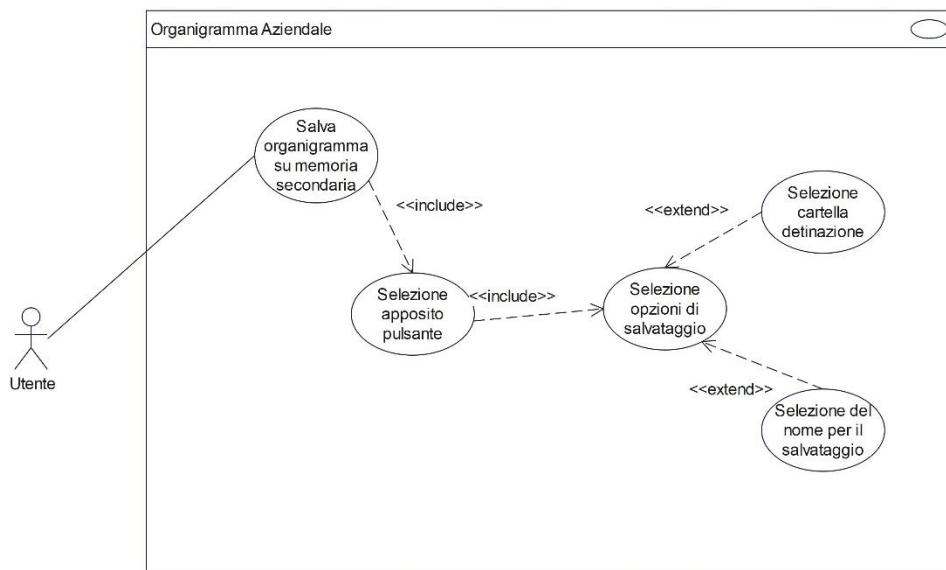
### Scenario di possibile modifica dell'organigramma



## Caricamento dati da memoria secondaria



## Salvataggio dati in memoria secondaria

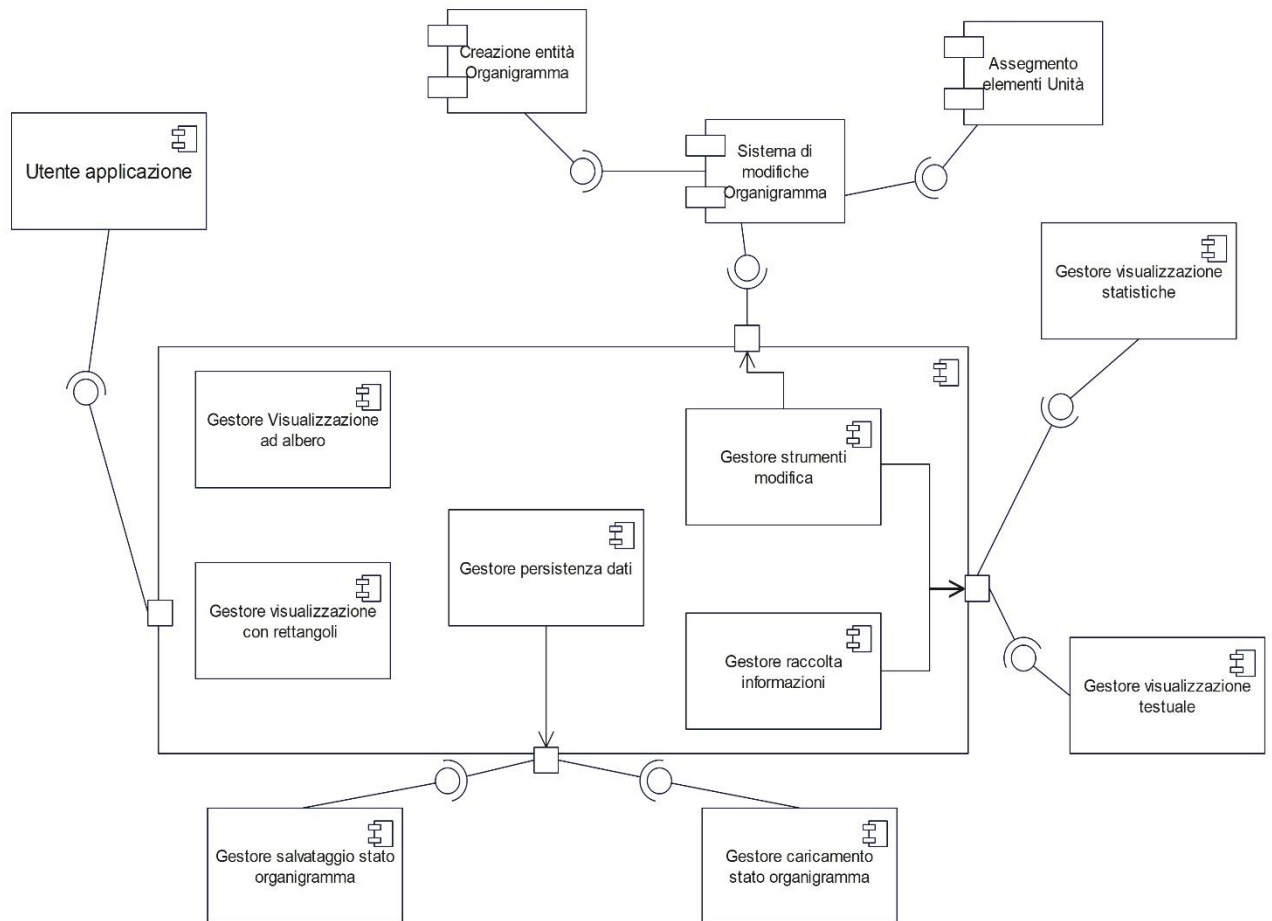




## C. Architettura Software

### Component Diagram

*Visione statica della struttura dell'applicazione*



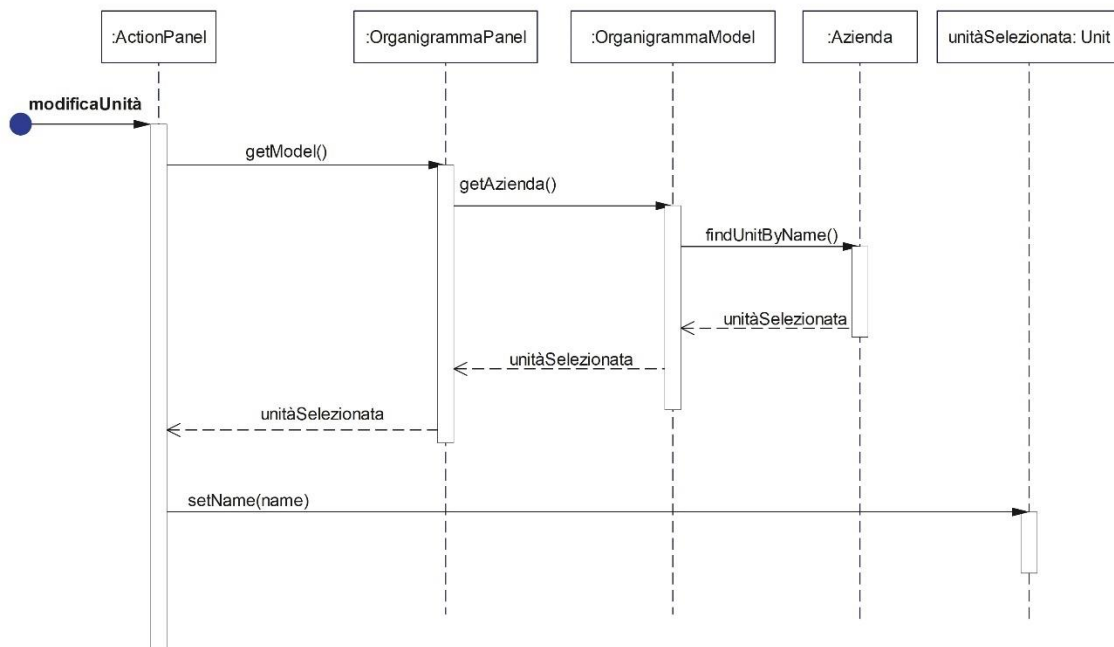
## Sequence Diagram

*Visione dinamica dell'applicazione*

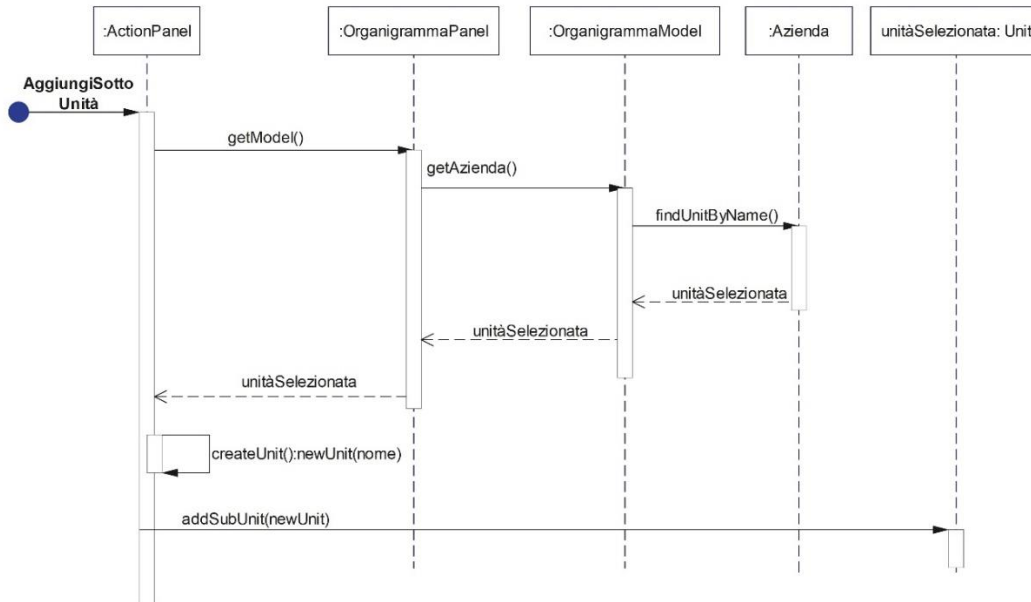
Di seguito sono riportati i diagrammi di sequenza che descrivono la collaborazione degli oggetti finalizzata all'esecuzione dei servizi offerti dall'applicazione.

*N.B: Per i primi diagrammi si specificherà anche l'interazione che avviene tra utente ed interfaccia grafica, per i successivi ci si limiterà alla collaborazione relativa alle sole entità coinvolte, trascurando la GUI.*

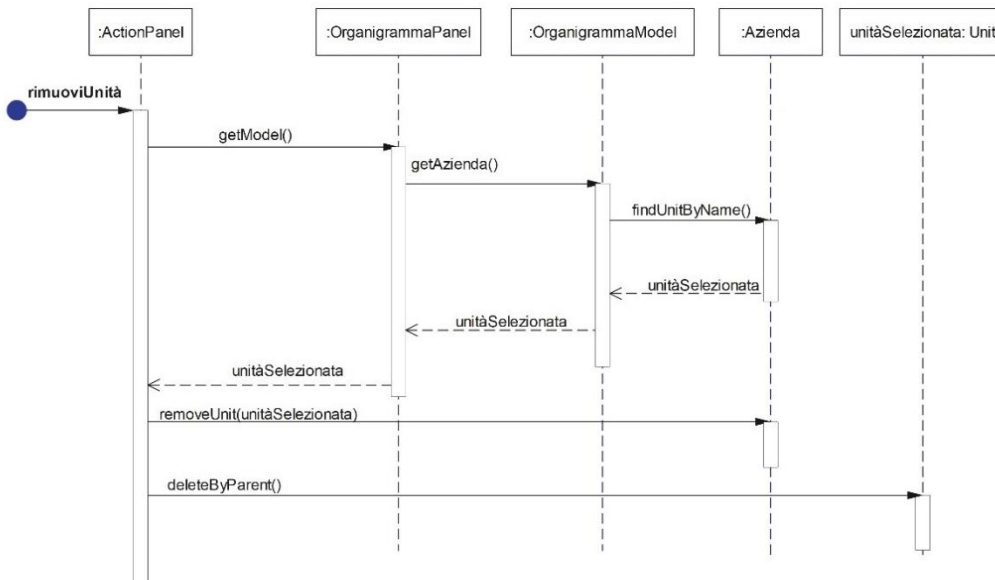
### Modifica di un'unità esistente



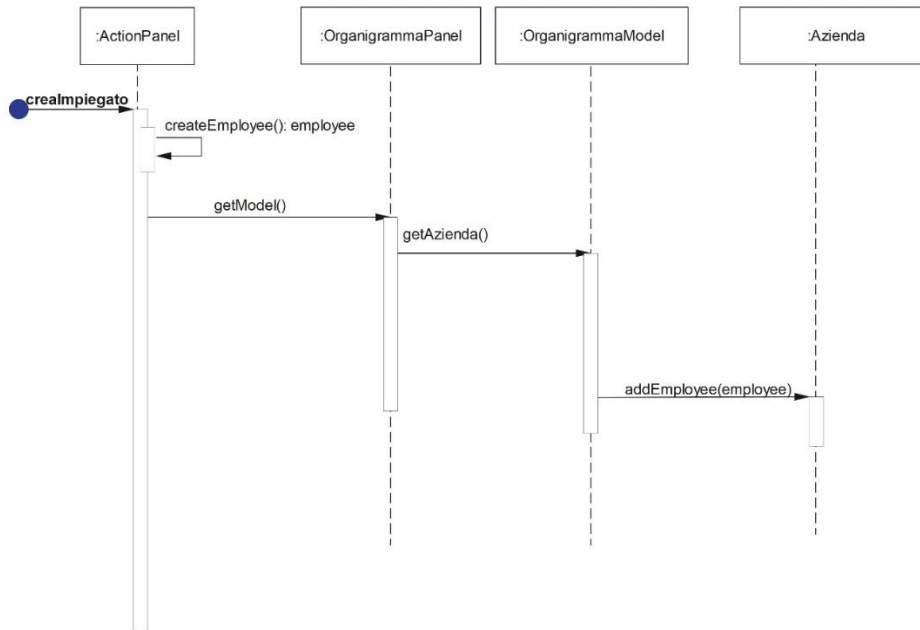
### Aggiunta di una sotto unità ad un'altra sotto unità



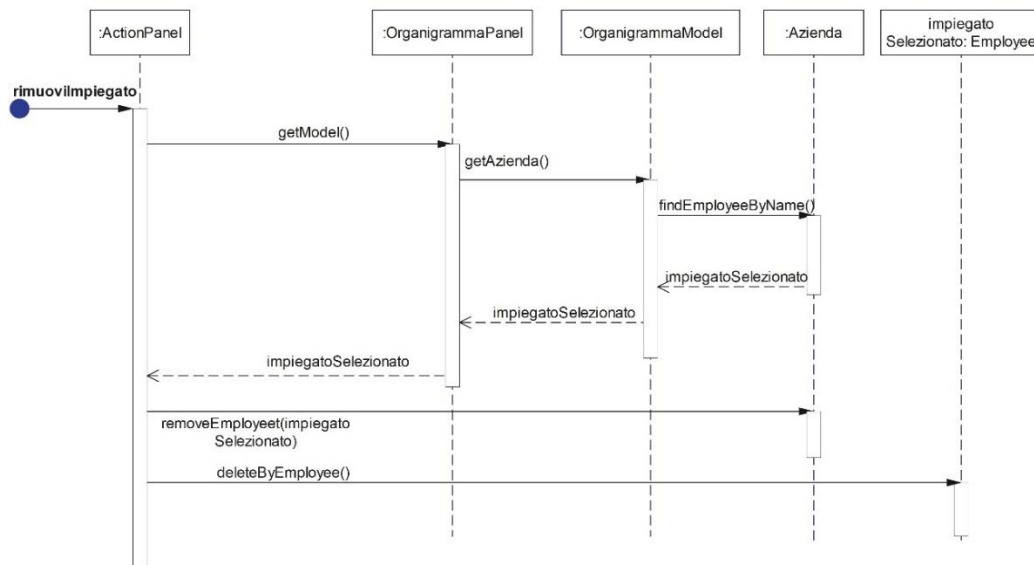
### Rimozione unità esistente



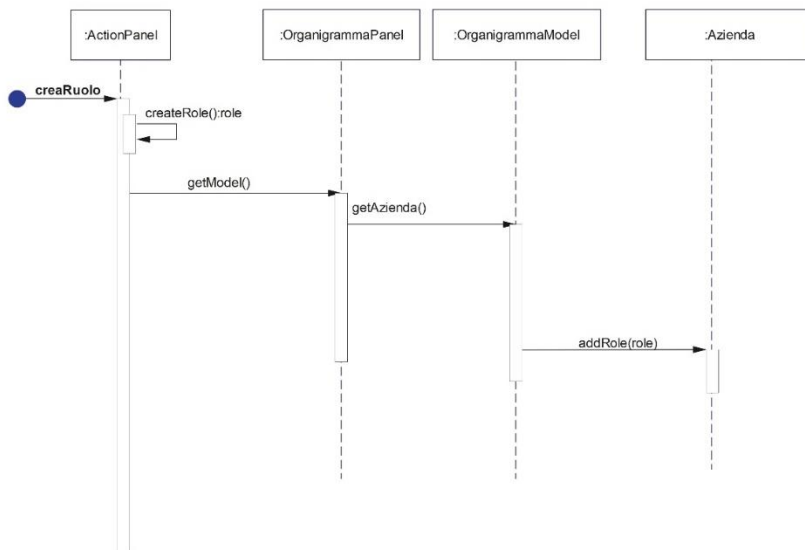
### Creazione Impiegato



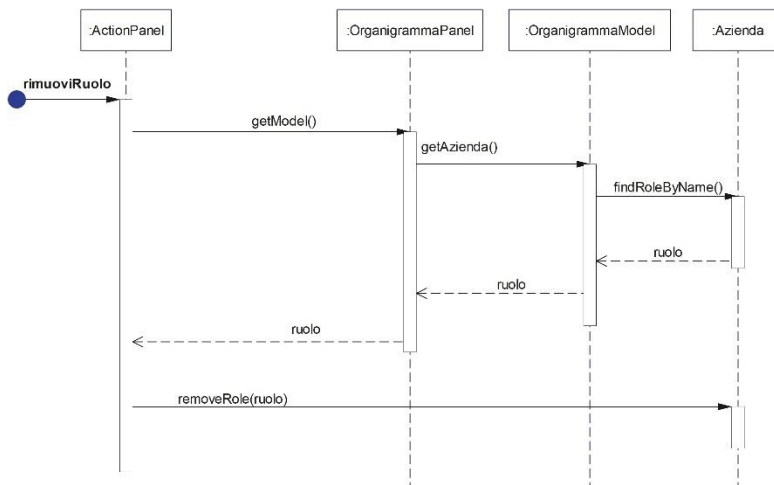
### Rimozione Impiegato



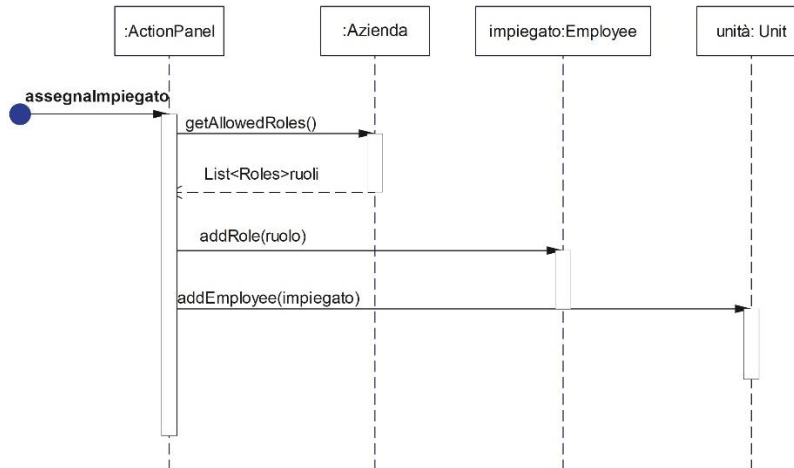
### Creazione di un ruolo



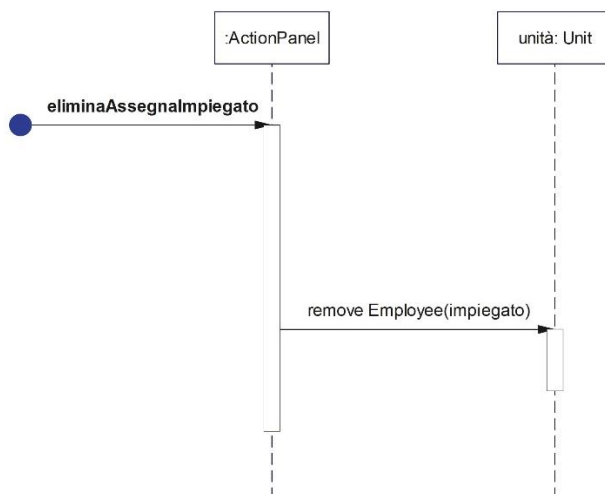
### Rimozione di un ruolo



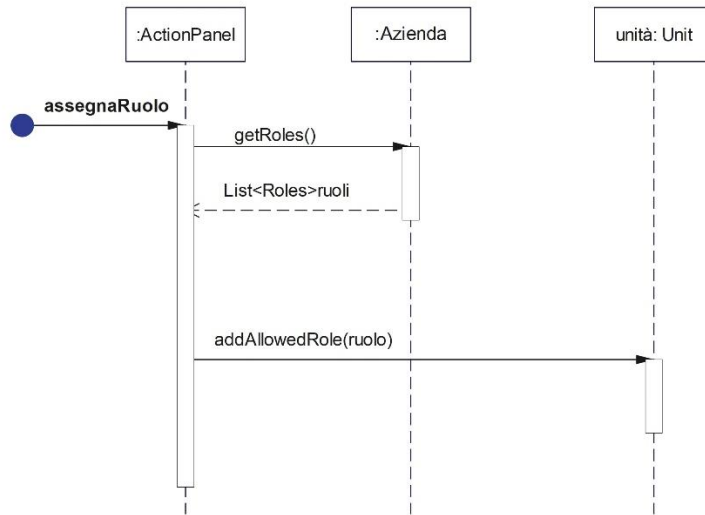
### Assegna impiegato ad unità



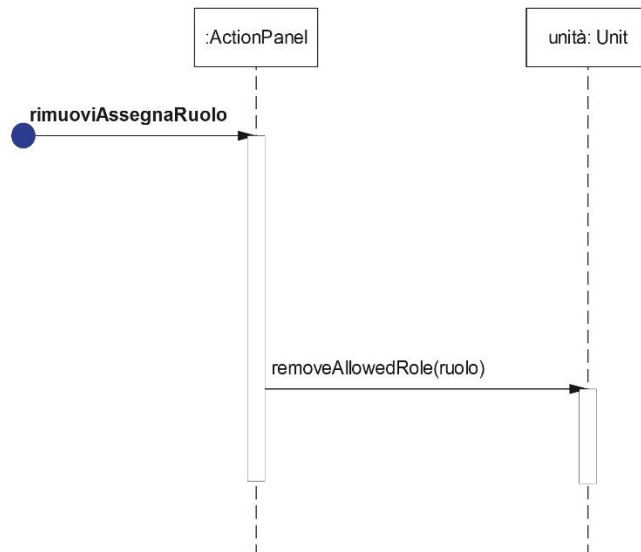
### Rimuovi assegnamento impiegato ad unità



**Assegna ruolo ad unità**

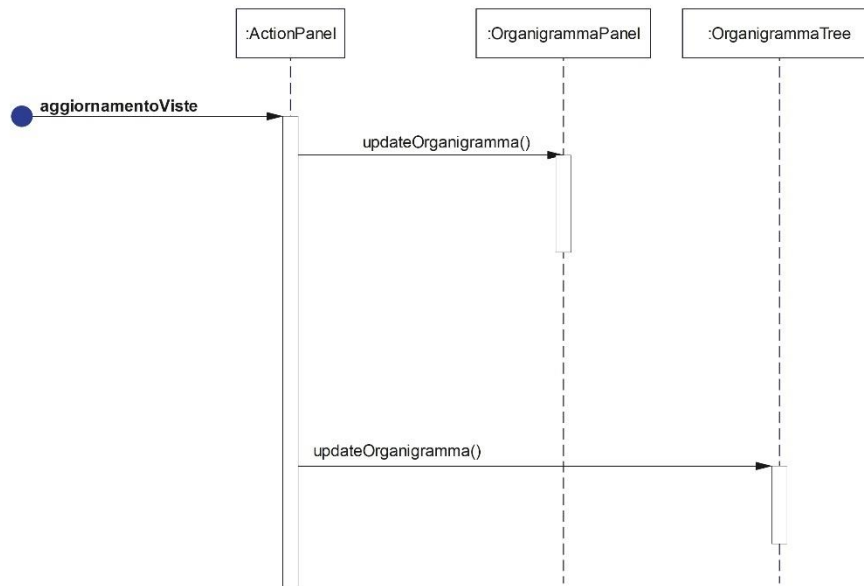


**Rimuovi assegnamento ruolo ad unità**



### Aggiornamento delle Viste

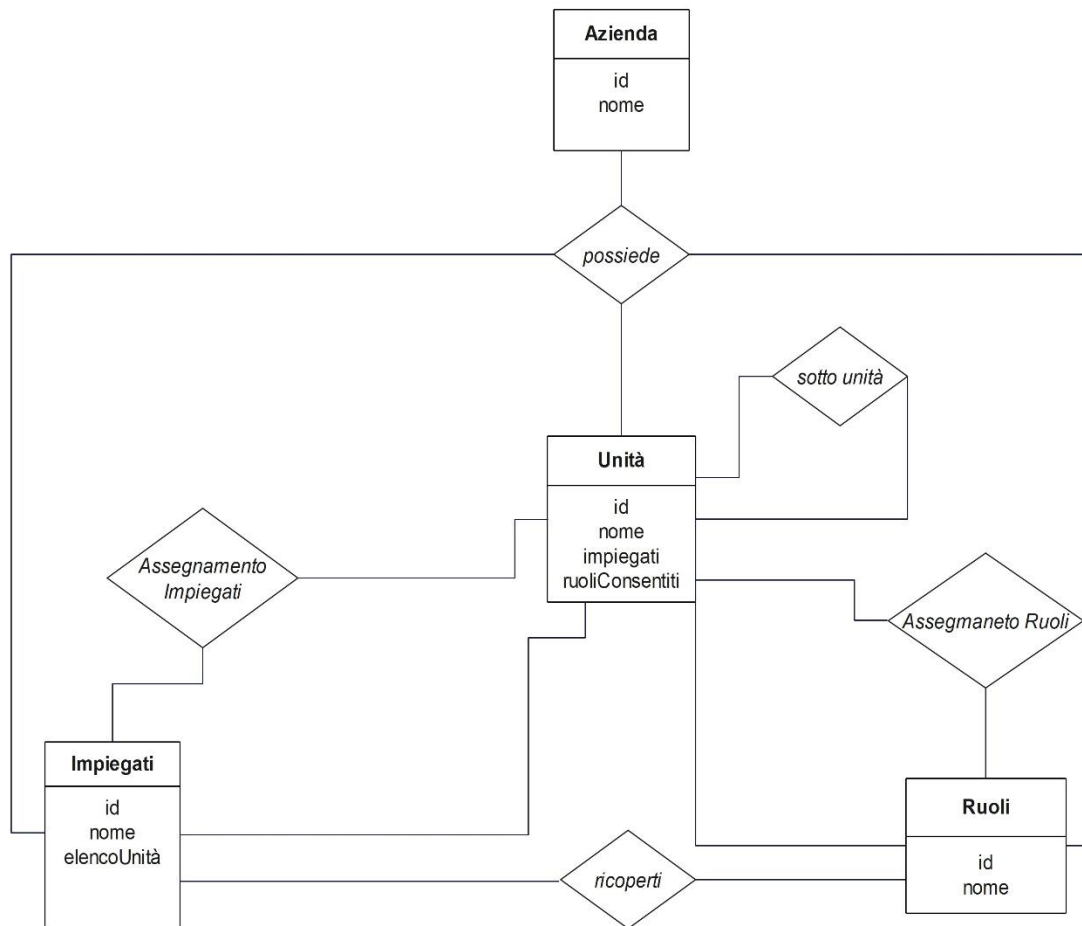
*Questa operazione viene eseguita ad ogni singola operazione effettuata sull'organigramma.*





## D. Dati e loro modellazione

L'applicazione è in grado di creare, rimuovere e gestire le entità dinamicamente a tempo d'esecuzione. Tuttavia, è possibile caricare uno stato interno, precedentemente salvato, al fine di operare direttamente su una base di dati già pronta.



Il modello entità-relazione fornito rappresenta un sistema aziendale con quattro entità principali:

- Azienda
- Impiegato
- Unità
- Ruolo

Queste entità sono connesse da diverse relazioni che definiscono le interazioni tra di loro.

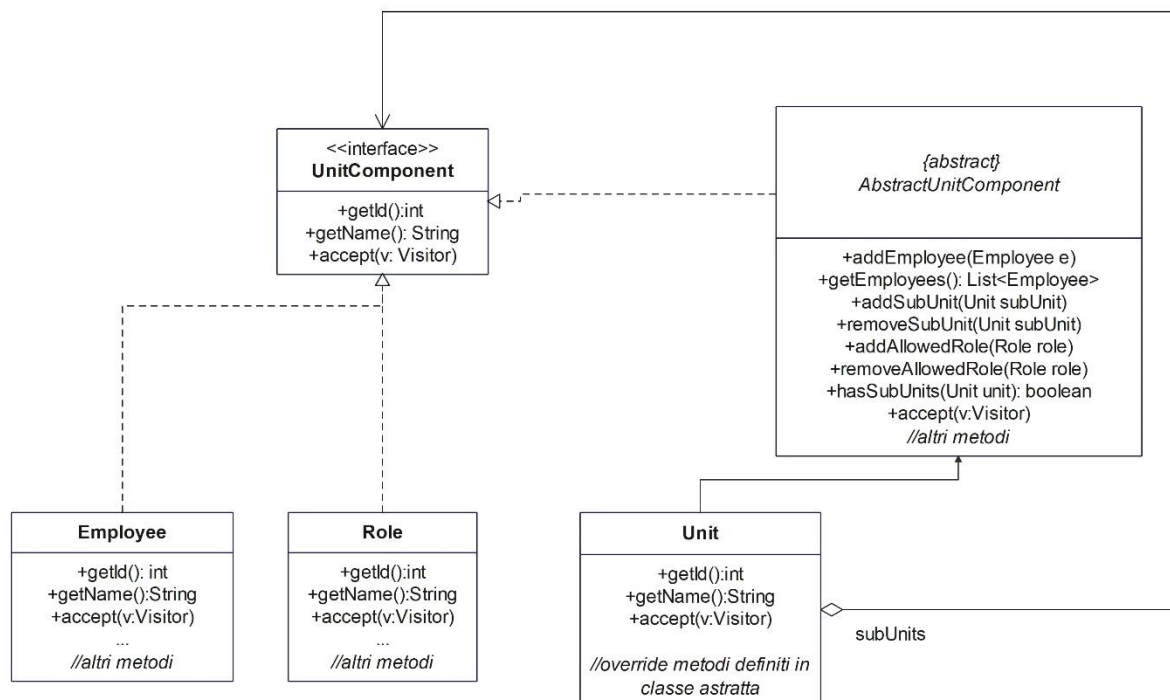
1. un'azienda può avere uno o più dipendenti:
    - a. un'azienda può avere molti dipendenti, ma un dipendente appartiene a un'unica azienda.
  2. un'azienda può avere una o più unità:
    - a. un'azienda può avere molte unità, ma un'unità appartiene a un'unica azienda.
  3. un'azienda può avere uno o più ruoli:
    - a. un'azienda può avere molti ruoli, ma un ruolo appartiene a un'unica azienda.
  4. un'unità può avere uno o più dipendenti.
  5. un'unità può permettere uno o più ruoli.
  6. un impiegato può ricoprire uno o più ruoli
-

## E. Scelte Progettuali (Design Decisions)

Di seguito saranno riportare le 5 scelte progettuali più significative finalizzate alla realizzazione del sistema software in esame: 4 in particolare includono ciascuna l'utilizzo di un design pattern specifico.

### Struttura Gerarchica Organigramma: Composite

Nella creazione della struttura ad albero finalizzata alla rappresentazione dell'organigramma aziendale, si è scelto di impiegare il pattern Composite, con lo scopo di gestire in maniera uniforme le unità, gli impiegati e i ruoli.



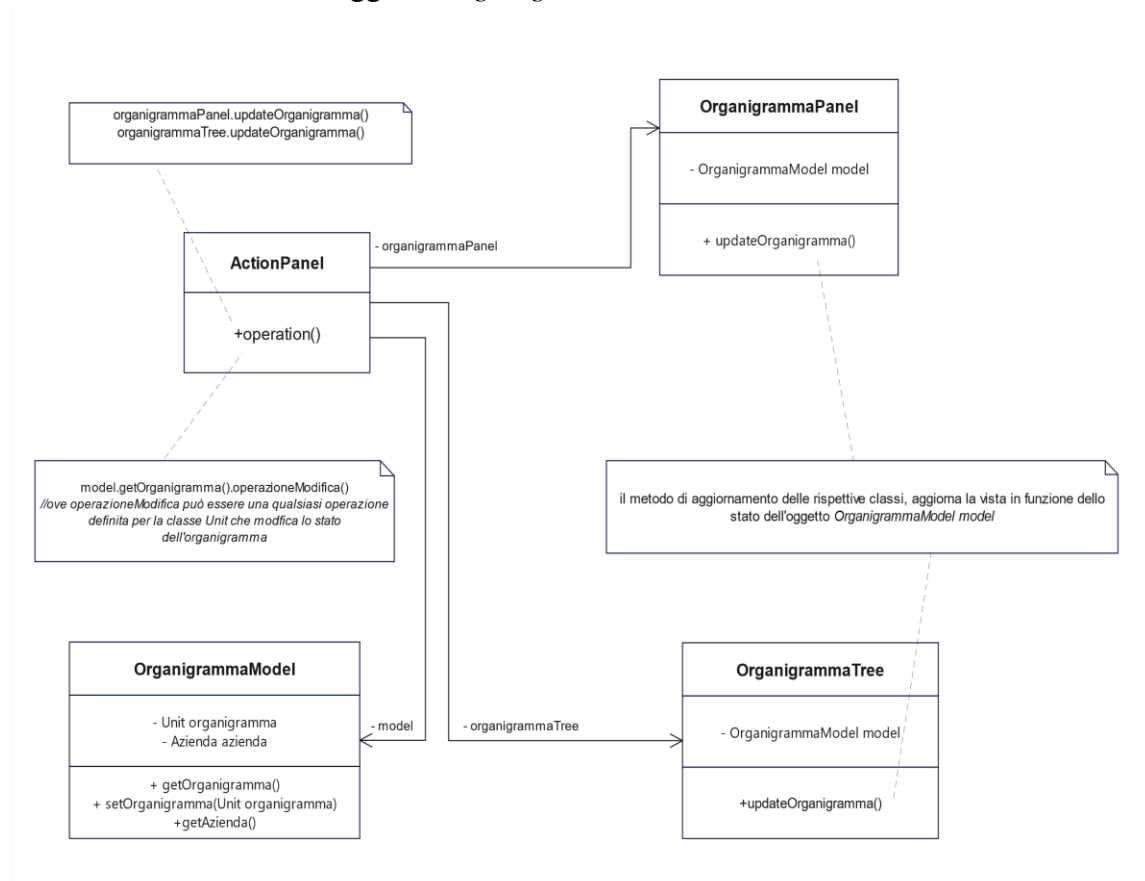
## Interfaccia Grafica: Observer

L'interfaccia grafica è stata realizzata impiegando il design pattern Observer al fine di definire degli oggetti osservatori, le viste, in grado di attendere le modifiche effettuate sullo stato dell'organigramma ed aggiornarsi in funzione del nuovo stato corrente.

La classe *Organigramma Model* rappresenta il soggetto osservato mentre le classi *Organigramma Panel* e *Organigramma Tree*, rispettivamente usate per la visualizzazione a rettangoli e ad albero, rappresentano gli osservatori.

A livello implementativo si è scelto, per motivi di pulizia del codice, di introdurre un'ulteriore classe *OrganigrammaAction* che funge da intermediario tra soggetto osservato e osservatori.

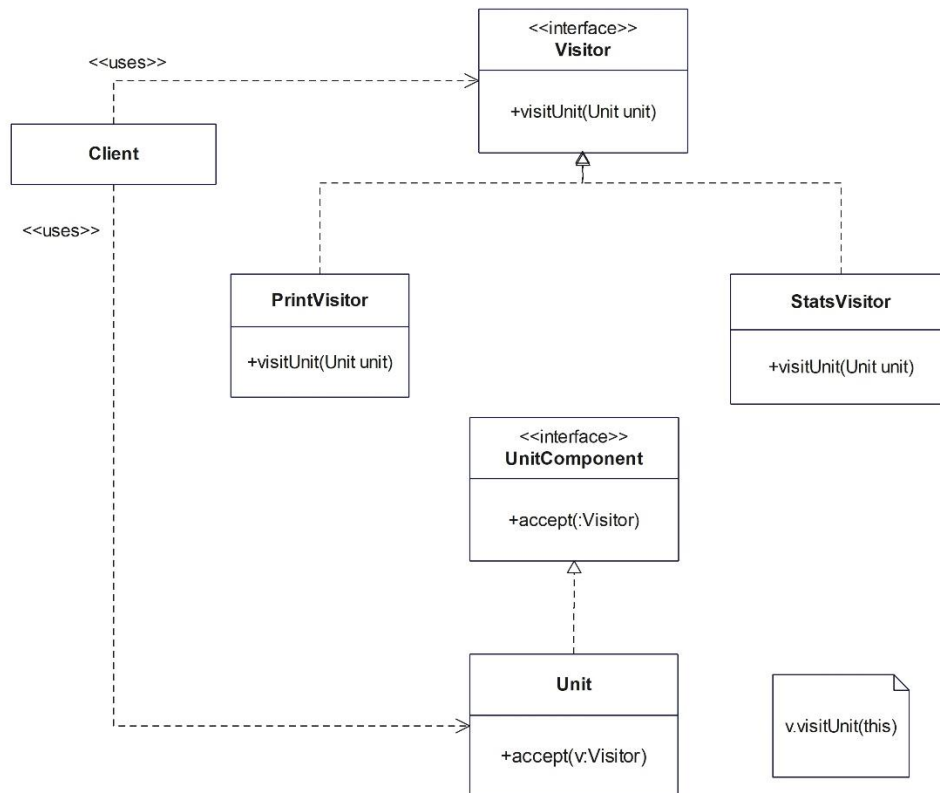
L'oggetto *ActionPanel* aggiorna lo stato di *OrganigrammaModel* e ne notifica la modifica alle due viste invocando i rispettivi metodi di aggiornamento, quest'ultimi costruiranno la visualizzazione dell'organigramma tenendo conto delle informazioni racchiuse all'interno dell'oggetto *OrganigrammaModel*



## Raccolta Informazioni Organigramma: Visitor

È stato utilizzato il pattern Visitor al fine di definire due visite distinte sull'albero delle unità dell'azienda:

1. *PrintVisitor*: per raccogliere informazioni sull'azienda e creare una stringa di output descrittiva:
  - a. in altre parole, si organizza una visita al fine di restituire una visualizzazione testuale dell'organigramma
2. *StatsVisitor*: per calcolare le statistiche totali sull'azienda.
  - a. Questo approccio consente di separare le operazioni di visita dalla struttura delle unità e dagli impiegati, consentendo una maggiore flessibilità e facilitando l'aggiunta di nuove operazioni o visite in futuro.



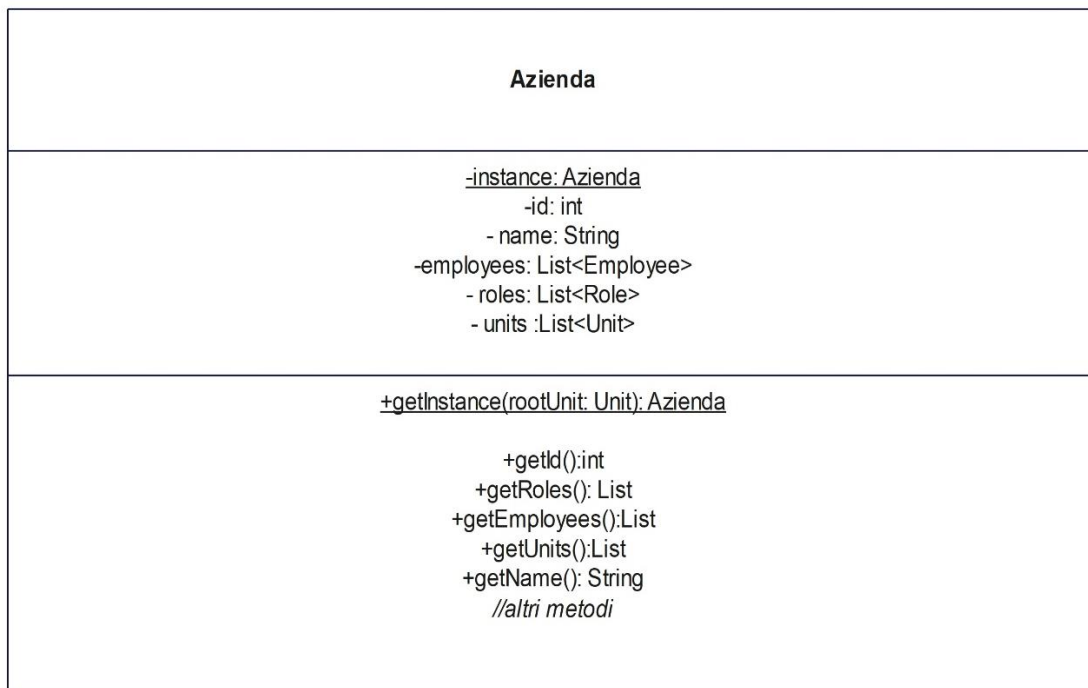
## Punto Accesso Globale per Informazioni Azienda: Singleton

La classe Azienda rappresenta il punto centrale d'accesso per l'organigramma e contiene informazioni su dipendenti, unità organizzative e ruoli dell'azienda.

La classe Azienda implementa il pattern Singleton per garantire che esista una sua sola istanza all'interno dell'applicazione.

Questo significa che, una volta istanziata, l'istanza singleton può essere accessibile da qualsiasi parte del codice, tramite il metodo *getInstance ()*.

In questo modo si evita la creazione di molteplici istanze che potrebbero portare a inconsistenze e problemi di sincronizzazione, molto utile per gestire la funzione di azzeramento organigramma e caricamento da memoria secondaria.



## Gestione persistenza dei dati

Nelle classi fornite, la persistenza dei dati è gestita utilizzando la libreria *com.fasterxml.jackson*.

La libreria jackson è ampiamente utilizzata per la serializzazione e deserializzazione di oggetti Java in formato JSON. Le annotazioni utilizzate nelle classi indicano come i dati devono essere serializzati o deserializzati.

1. *@JsonIdentityInfo(generator = ObjectIdGenerators.PropertyGenerator.class, property = "name")*:
  - a. Questa annotazione è stata utilizzata per gestire la serializzazione degli oggetti in modo da mantenere la loro identità unica.
  - b. Viene specificato il generatore di identità (*ObjectIdGenerators.PropertyGenerator.class*) e la proprietà utilizzata come identificatore (*property = "name"*). In questo caso, la proprietà "name" viene utilizzata per identificare gli oggetti durante la serializzazione.
2. *@JsonIdentityReference*:
  - a. Questa annotazione è stata utilizzata con lo scopo di gestire la deserializzazione degli oggetti facendo riferimento all'identità già serializzata invece di creare nuovi oggetti. In questo caso, viene utilizzata per fare riferimento agli oggetti Employee e Unit già serializzati durante la deserializzazione.
3. *@JsonIgnoreProperties("allowedRoleNames")*:
  - a. Questa annotazione è stata impiegata al fine di ignorare la proprietà "allowedRoleNames" durante la serializzazione e deserializzazione dell'oggetto Unit. Ciò significa che la proprietà non verrà inclusa nel JSON generato o considerata durante la deserializzazione.

L'utilizzo di *@JsonIdentityInfo* e *@JsonIdentityReference* consente di mantenere le relazioni tra gli oggetti durante la deserializzazione, evitando duplicati e creando riferimenti agli oggetti già esistenti.

Ciò aiuta a mantenere la consistenza dei dati e riduce la dimensione del JSON generato.

## F. Progettazione di Basso Livello

Di seguito si descrive secondo quali modalità, i concetti e le decisioni, prese durante la fase di progettazione ad alto livello, siano state tradotte in una rappresentazione più concreta ed implementativa del sistema.

Si descrivono i dettagli specifici dell'implementazione comprese le strutture dati, le interfacce, le classi e le relazioni che intercorrono tra di esse.

L'applicazione finalizzata alla gestione ed organizzazione di un organigramma aziendale è stata realizzata in linguaggio Java sfruttando l'ambiente di sviluppo integrato *IntelliJ IDEA*.

### Struttura codice Sorgente

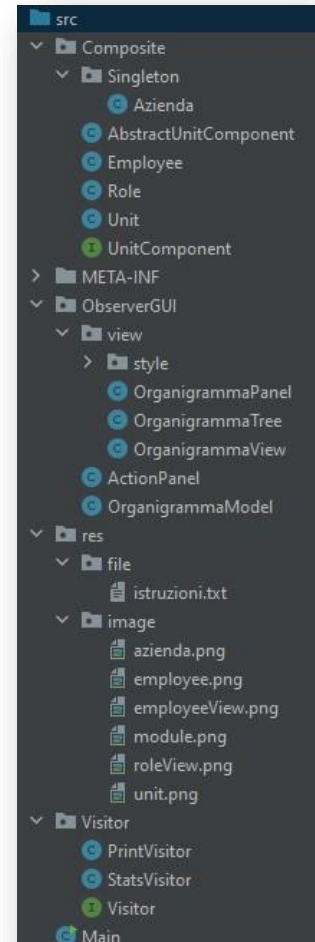
Per quanto riguarda l'organizzazione del codice sorgente, si è scelto di dividere in package la seguente gerarchia riportata in figura.

Le scelte progettuali hanno portato all'impiego di 4 Design Pattern e le classi costituenti l'applicazione sono state organizzate, in package, in funzione degli stessi.

### Azienda

La classe "Azienda" rappresenta un'entità aziendale nell'applicazione dell'organigramma aziendale:

- mantiene una lista di impiegati (employees), unità (units), ruoli (roles) e nomi utilizzati (nomiUtilizzati) nell'azienda.
- Viene utilizzata l'annotazione `@JsonIdentityInfo` per gestire la serializzazione e deserializzazione degli oggetti in formato JSON.
- implementa il pattern Singleton, che garantisce l'esistenza di una singola istanza dell'azienda all'interno dell'applicazione.
- fornisce metodi per aggiungere e rimuovere impiegati, ruoli e unità dall'azienda.
- È possibile recuperare l'elenco degli impiegati, delle unità, dei ruoli e dei nomi utilizzati attraverso i relativi metodi getter.
- offre metodi per la ricerca di unità, impiegati e ruoli in base al nome.





## **Interfaccia UnitComponent**

L'interfaccia "UnitComponent" rappresenta un componente di unità nell'applicazione dell'organigramma aziendale.

- Definisce due metodi: getName() per ottenere il nome del componente e accept(Visitor v) per accettare un visitatore che può eseguire operazioni specifiche sul componente.
- fornisce una base per la creazione di classi concreti di componenti di unità e supporta l'implementazione del pattern Visitor per l'elaborazione dei componenti.

## **AbstractUnitComponent**

La classe "AbstractUnitComponent" è una classe astratta che implementa l'interfaccia "UnitComponent".

- Fornisce implementazioni per i metodi dell'interfaccia.
- Questa classe viene estesa dalle classi concrete come "Unit" per fornire una base comune per le operazioni dell'organigramma aziendale.

## **Unit**

La classe "Unit" rappresenta un'unità nell'applicazione dell'organigramma aziendale.

- È un componente concreto che estende la classe astratta "AbstractUnitComponent".
- contiene attributi come il nome dell'unità, la lista di impiegati associati, la lista di sotto-unità, la lista di ruoli consentiti, l'unità genitore e una mappa che tiene traccia dei ruoli degli impiegati.
- fornisce metodi per gestire gli impiegati e le sotto-unità, aggiungere e rimuovere ruoli consentiti, e stampare la struttura dell'unità.
- Implementa l'interfaccia "UnitComponent" per consentire l'uso del pattern Visitor e offre override dei metodi toString() e accept(Visitor visitor).

## **Employee**

La classe "Employee" è una classe che implementa l'interfaccia "UnitComponent" e rappresenta un impiegato all'interno dell'organigramma aziendale.

- contiene attributi come il nome dell'impiegato, una lista di unità a cui l'impiegato è assegnato, una lista di ruoli che l'impiegato ricopre e un identificatore univoco.
- fornisce metodi per gestire l'aggiunta e la rimozione di unità e ruoli, nonché per l'eliminazione dell'impiegato.
- utilizza le annotazioni di Jackson per la gestione della serializzazione e deserializzazione degli oggetti.

## **Role**

La classe "Role" è una classe che implementa l'interfaccia "UnitComponent" e rappresenta un ruolo all'interno dell'organizzazione.

- contiene un attributo per il nome del ruolo e un identificatore univoco.
- fornisce metodi per ottenere e impostare il nome del ruolo, nonché per ottenere e impostare l'identificatore.
- ridefinisce il metodo toString() per restituire il nome del ruolo e il metodo equals() per confrontare due ruoli in base al nome.
- utilizza anche le annotazioni di Jackson per la gestione della serializzazione e deserializzazione degli oggetti.

## **OrganigrammaModel**

La classe "OrganigrammaModel" rappresenta il modello dell'organigramma all'interno dell'applicazione GUI. Essa gestisce i dati e la logica dell'organigramma.

- contiene due attributi: "organigramma" che rappresenta l'unità radice dell'organigramma e "azienda" che rappresenta l'azienda a cui l'organigramma appartiene.
- fornisce metodi per ottenere e impostare l'organigramma, nonché per ottenere l'azienda. Il costruttore della classe accetta un'unità radice e un'istanza di Azienda per inizializzare gli attributi.

## **OrganigrammaView**

La classe OrganigrammaView rappresenta la vista dell'organigramma di un'azienda e si occupa della visualizzazione grafica degli elementi dell'organigramma.

- ha variabili di istanza per la larghezza e l'altezza dello schermo, il modello dell'organigramma e gli elementi grafici come il frame principale, i pannelli, le tabelle e i bottoni.
- Il metodo inizializzaFrame() viene chiamato per impostare le dimensioni dello schermo e creare il frame principale.
  - Il metodo creazionePannelloPrincipale() viene chiamato per creare il pannello principale che ospita l'organigramma e il titolo.
- I metodi creazioneTabelleRuoliConsentiti(), creazioneTabellaInfoImpiegati() e creazioneTabellaRuoliCreati() vengono chiamati per creare le tabelle che mostrano rispettivamente i ruoli consentiti per ogni unità, le informazioni sugli impiegati dell'azienda e i ruoli creati nell'azienda.
- Il metodo gestionePannelloTabella() viene chiamato per gestire il pannello delle tabelle e i relativi bottoni.

- Il metodo `displayStatistics()` viene chiamato per visualizzare le statistiche sull'organigramma, come il totale delle unità, il totale degli impiegati e il totale dei ruoli.
- Il metodo `stampaInfoAzienda()` viene chiamato per stampare le informazioni sull'azienda, come i dettagli dell'organigramma.
- Il metodo `salvataggioDati()` viene chiamato per gestire il salvataggio dell'organigramma.
- Il metodo `gestioneActionBar()` viene chiamato per gestire la barra degli strumenti, che contiene i bottoni per le varie funzionalità dell'applicazione.
- Altri metodi come `ripristinoDati()`, `azzerOrganigramma()` e `helpButton()` gestiscono rispettivamente il ripristino dei dati, l'azzeramento dell'organigramma e il pulsante di aiuto.

### **OrganigrammaPanel**

La classe `OrganigrammaPanel` estende `JPanel`: è una classe di Swing che rappresenta un pannello all'interno di un'interfaccia grafica.

- Questa classe viene utilizzata per visualizzare l'organigramma a rettangoli
- ha diversi attributi, tra cui `screenWidth` e `screenHeight`, che rappresentano la larghezza e l'altezza dello schermo, rispettivamente.
- Altri attributi sono `model`, che rappresenta il modello dell'organigramma, e `selectedUnit`, che rappresenta l'unità attualmente selezionata.
- Il costruttore della classe accetta un oggetto `OrganigrammaModel`, che rappresenta il modello dell'organigramma inizializzato
- contiene anche diversi metodi:
  - Il metodo `updateOrganigramma` viene utilizzato per aggiornare la struttura dell'organigramma con un'unità aggiornata.
  - Viene richiamato il metodo `setOrganigramma` del modello per impostare l'unità aggiornata
  - Il metodo `drawUnit` viene utilizzato per disegnare un rettangolo con il nome dell'unità.
  - Viene anche disegnata una freccia che collega l'unità ai suoi sotto-unità.
  - Viene invocato il metodo `drawUnit` per disegnare l'organigramma a partire dall'unità radice memorizzata nel modello.

### **OrganigrammaTree**

La classe `OrganigrammaTree` estende `JPanel`, una classe di Swing che rappresenta un pannello all'interno di un'interfaccia grafica.

- Questa classe viene utilizzata per visualizzare l'organigramma come una struttura ad albero.
- ha diversi attributi, tra cui `organigrammaTree`, che rappresenta l'oggetto `JTree` utilizzato per visualizzare l'organigramma, `model`, che rappresenta il modello dell'organigramma, e `screenWidth` e `screenHeight`, che rappresentano la larghezza e l'altezza dello schermo.
- Il costruttore della classe accetta un oggetto `OrganigrammaModel`, che rappresenta il modello dell'organigramma
- Il metodo `updateOrganigramma` viene utilizzato per aggiornare la struttura dell'organigramma con un'unità aggiornata.
- Il metodo `paintComponent` viene sovrascritto per disegnare il contenuto del pannello, ma non contiene alcuna implementazione.
- Il metodo `createOrganigrammaTree` viene utilizzato per creare il `JTree` che rappresenta la struttura gerarchica dell'organigramma.

### ActionPanel

La classe `ActionPanel` estende `JPanel` e viene utilizzata per creare un pannello con bottoni e azioni correlate per l'interfaccia grafica di un organigramma.

- è responsabile di gestire le azioni dell'utente nell'organigramma e di comunicare con altre componenti dell'interfaccia grafica *per aggiornare lo stato dell'organigramma stesso*.
- Vengono creati i bottoni con stili personalizzati.
- Vengono aggiunti gli action listener ai bottoni per eseguire le azioni associate.
- Ogni azione esegue una serie di operazioni, come l'apertura di dialoghi di input/output per modificare l'organigramma o aggiungere/rimuovere elementi.
  - Dopo ogni azione, viene chiamato il metodo `inizializzaFrame()` sull'oggetto `view` per aggiornare l'interfaccia grafica.

### Visitor

L'interfaccia `Visitor` contiene due metodi, `visit(Unit unit)` e `visit(Employee employee)`, che devono essere implementati dalle classi che desiderano utilizzare il pattern `Visitor` per visitare oggetti di tipo `Unit` e `Employee`.

- Il metodo `visit(Unit unit)` viene utilizzato per visitare un oggetto di tipo `Unit`, consentendo di eseguire operazioni specifiche su di esso.
- Il metodo `visit(Employee employee)` viene utilizzato per visitare un oggetto di tipo `Employee`, consentendo di eseguire operazioni specifiche su di esso.

### **PrintVisitor**

La classe PrintVisitor implementa l'interfaccia Visitor e fornisce una visita specifica per gli oggetti di tipo Unit.

- ha un oggetto StringBuilder chiamato sb, che viene utilizzato per costruire una rappresentazione testuale della visita.

### **StatsVisitor**

La classe StatsVisitor implementa l'interfaccia Visitor e fornisce un modo per calcolare statistiche relative agli oggetti Unit presenti nella struttura gerarchica.

- ha tre variabili private: totalUnits, totalEmployees e totalRoles, che vengono utilizzate per tenere traccia del conteggio totale delle unità, degli impiegati e dei ruoli.
-

## G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs)

---

Di seguito si descrivono gli aspetti implementativi del progetto che soddisfano i requisiti funzionali e non funzionali definiti in fase di analisi dei requisiti all'inizio della progettazione.

Maggiori informazioni relative all'implementazione specifica delle componenti dell'applicazione, sono state esplicitate al punto inerente la *Progettazione di Basso Livello*.

### *Requisiti funzionali*

il progetto risponde ai requisiti funzionali specificati, precedentemente definiti:

#### *1. Modifiche allo stato dell'organigramma:*

Reso possibile dal corretto impiego del pattern Composite, dalla corretta implementazione dell'interfaccia grafica e l'implementazione delle classi Unit, Employee, Azienda, Role.

- Il progetto permette agli utenti di apportare modifiche all'organigramma esistente.
- Gli utenti possono aggiungere, rimuovere o modificare i membri dell'organigramma.
- Visualizzazione grafica dello stato corrente dell'organigramma:

*Il progetto fornisce una rappresentazione grafica dell'organigramma corrente.*

Reso possibile dall'utilizzo del design pattern Observer e l'implementazione delle classi: OrganigrammaPanel, OrganigrammaView, OrganigrammaModel, OrganigrammaTree e la classe adibita alla gestione della barra degli strumenti, ovvero ActionPanel.

- Gli utenti possono visualizzare l'organigramma in un formato visivamente intuitivo e comprensibile.

- La visualizzazione grafica mostra le relazioni tra i membri dell'organigramma, inclusi i loro ruoli e le gerarchie.

*Memorizzazione su memoria secondaria:*

Reso possibile grazie alla libreria *com.faster.xml*, approfondita al punto relativo alle *Scelte di Design*.

- Il progetto consente di memorizzare l'organigramma su una memoria secondaria
- I dati relativi all'organigramma vengono salvati in modo permanente e possono essere caricati in un secondo momento.

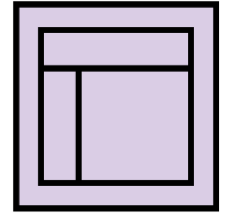
*Requisiti non funzionali*

1. *Prestazioni:* Il progetto è in grado di gestire un grande numero di unità, impiegati e ruoli senza degradare le prestazioni. La progettazione delle strutture dati e l'implementazione dei metodi è ottimizzata al fine di garantire una buona performance nel recupero e nella manipolazione delle informazioni. L'utilizzo del pattern Singleton consente di evitare sprechi di memoria nella creazione dell'istanza Azienda, struttura fondamentale dell'organigramma.
2. *Scalabilità e Manutenibilità:* Il progetto risulta essere progettato per essere scalabile, in modo che possa gestire l'aggiunta di nuove unità, impiegati e ruoli senza dover apportare modifiche significative al codice sorgente, l'impiego dei design pattern ha garantito ordine nella stesura del codice, e maggiore facilità nella comprensione dello stesso.
3. *Affidabilità:* Il sistema risulta affidabile e robusto, in grado di gestire correttamente situazioni anomale o errori durante l'esecuzione. Questo è stato raggiunto attraverso l'implementazione di meccanismi di gestione degli errori, la validazione dei dati di input e l'implementazione di un'interfaccia grafica in grado di tutelare l'utente da eventuali errori di inserimento.
4. *Sicurezza:* Il progetto garantisce persistenza delle informazioni dell'organigramma aziendale.
  - a. Ciò è stato raggiunto tramite l'adozione di meccanismi di salvataggio e caricamento dati implementati all'interno dell'applicazione stessa.
  - b. L'utilizzo di apposite librerie Java consente all'utente di memorizzare in formato json, su memoria secondaria, il proprio organigramma, e ripristinarlo in un secondo momento, attingendo sempre al file precedentemente salvato in memoria secondaria.

5. *Usabilità*: L'applicazione dell'organigramma aziendale risulta essere intuitiva e facile da usare per gli utenti. L'interfaccia utente è stata progettata in modo da consentire una navigazione semplice e una comprensione immediata delle funzionalità offerte.
- a. A basso livello, l'impiego del pattern Observer, ha garantito la realizzazione di un'interfaccia ordinata nel codice ed efficiente nell'esecuzione.
6. *Portabilità*: il progetto è stato implementato in linguaggi java, il cui codice sorgente può essere eseguito su diverse piattaforme hardware e sistemi operativi grazie alla Java Virtual Machine (JVM) e al bytecode.
-



## Appendix. Prototype



Di seguito si fornisce una breve descrizione del prototipo realizzato, costituente un'applicazione volta alla creazione di un organigramma aziendale

Si presenteranno le caratteristiche dell'applicazione progettate in accordo con i requisiti funzionali e non funzionali espressi in fase di progettazione.

### Descrizione

È possibile caricare un organigramma di esempio utilizzando l'opzione "Carica Organigramma" e selezionando il file "organigrammaExample.json" allegato al codice del progetto fornito.

### Visualizzazione

L'interfaccia grafica offre due tipi di visualizzazione dell'organigramma.

#### 1. Visualizzazione con Rettangoli:

- a. Nel pannello centrale, le unità sono rappresentate come rettangoli, collegati alle rispettive sotto unità tramite segmenti. In questa visualizzazione, vengono mostrate solo le unità, non gli impiegati. La struttura grafica si adatta dinamicamente all'inserimento ed eliminazione delle unità.

#### 2. Visualizzazione ad Albero:

- a. Nel pannello di sinistra, è possibile visualizzare la struttura dell'organigramma mediante una visualizzazione ad albero. Cliccando su una specifica unità, è possibile espandere il contenuto, se presente, e visualizzare le sotto unità e gli impiegati. Gli impiegati all'interno di un'unità sono visualizzati come figli dell'unità nella gerarchia, e nella loro etichetta sono indicati i ruoli che ricoprono all'interno dell'unità stessa.

#### 3. Barra degli Strumenti:

- a. La barra degli strumenti offre i seguenti comandi:
  - i. *Modifica Nome Unità*: Permette di modificare il nome di un'unità esistente.

- ii. *Aggiungi Sotto Unità*: Consente di definire una nuova sotto unità per un'unità esistente.
- iii. *Rimuovi Unità*: Permette di eliminare un'unità esistente. Non è possibile rimuovere un'unità che ha delle sotto unità (è necessario eliminare prima le sotto unità).
- iv. *Aggiungi Impiegato Azienda*: Permette di definire un nuovo impiegato nell'azienda. Dopo aver creato un impiegato, è possibile assegnarlo alle unità ricoprendo uno o più ruoli.
- v. *Rimuovi Impiegato Azienda*: Consente di rimuovere un impiegato dall'azienda. Una volta eliminato, l'impiegato non sarà più assegnato alle unità per cui ricopriva un determinato ruolo.
- vi. *Assegna Impiegato ad Unità*: Permette di assegnare un impiegato, con un determinato ruolo, a una specifica unità. È necessario che all'unità sia stato assegnato almeno un ruolo consentito per poter assegnare un impiegato.
- vii. *Rimuovi Impiegato da Unità*: Consente di eliminare l'assegnamento di un impiegato da un'unità. L'impiegato continuerà a far parte dell'azienda e potrà essere assegnato a un'altra unità.
- viii. *Crea Ruolo*: Permette di definire un nuovo ruolo all'interno dell'azienda. Dopo averlo creato, il ruolo può essere assegnato alle unità come ruolo consentito.
- ix. *Rimuovi Ruolo*: Consente di eliminare un ruolo definito all'interno dell'azienda. Eliminando un ruolo, gli impiegati assegnati con quel ruolo in determinate unità non saranno più assegnati a quel ruolo (se ricoprono solo quel ruolo nell'unità, non saranno più assegnati all'unità).
- x. *Assegna Ruolo ad Unità*: Permette di assegnare un ruolo a una specifica unità. Se un ruolo è assegnato a un'unità, l'impiegato può ricoprire quel ruolo all'interno dell'unità.
- xi. *Rimuovi Ruolo da Unità*: Consente di eliminare l'assegnamento di un ruolo da un'unità. Il ruolo non sarà più consentito in quell'unità, e gli impiegati non potranno più essere assegnati a quell'unità ricoprendo quel ruolo. Il ruolo continuerà a far parte dell'azienda e potrà essere assegnato ad altre unità.

### Pannello Laterale

Il pannello laterale offre le seguenti opzioni:

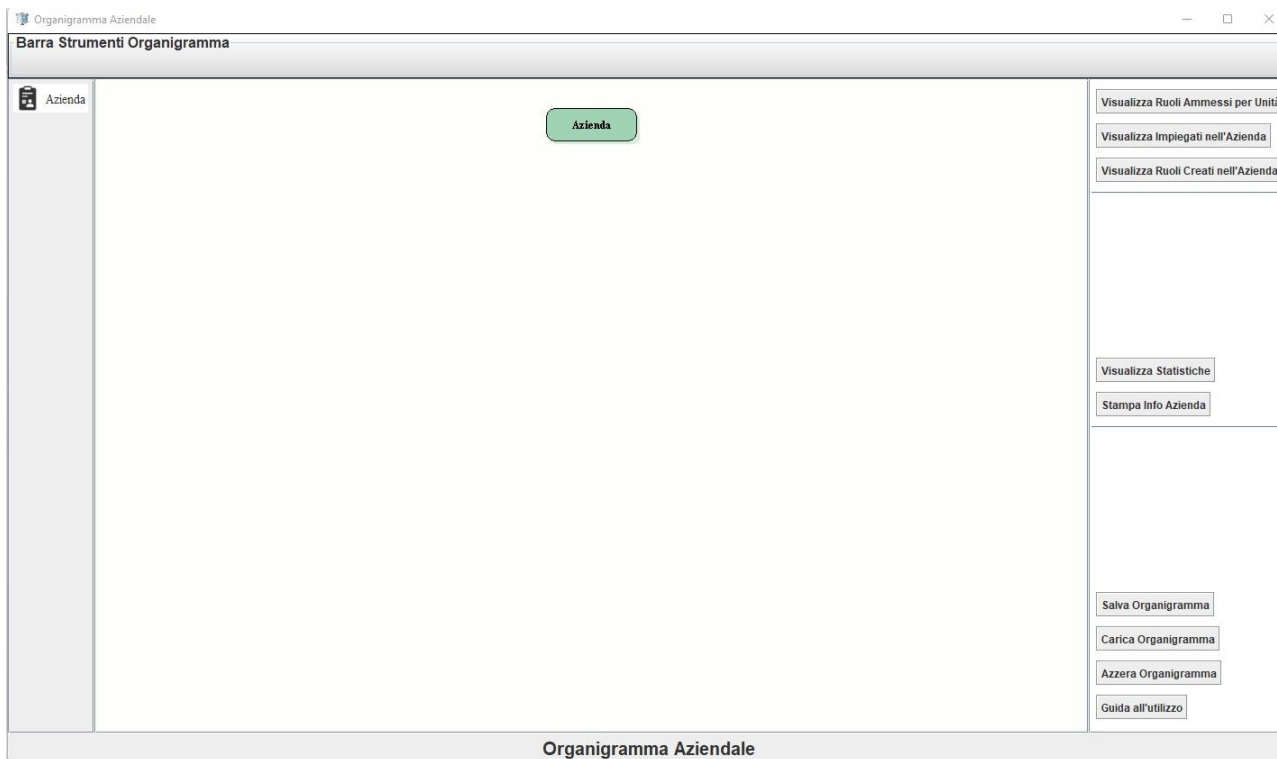
- 1. *Visualizza Ruoli Ammessi per Unità*: Mostra una tabella che associa ad ogni unità i ruoli consentiti in essa.
- 2. *Visualizza Impiegati nell'Azienda*: Mostra una tabella che associa ad ogni impiegato tutte le unità a cui è assegnato e tutti i ruoli che ricopre in esse.

3. *Visualizza Ruoli Creati nell'Azienda*: Mostra una tabella che visualizza tutti i ruoli creati all'interno dell'azienda.
4. *Visualizza Statistiche*: Apre una finestra che mostra le statistiche dell'organigramma, come il totale delle unità, impiegati e ruoli.
5. *Stampa Info Azienda*: Apre una finestra con una rappresentazione testuale dello stato corrente dell'organigramma.
6. *Salva Organigramma*: Permette di salvare lo stato corrente dell'organigramma in un file in formato .json, utilizzabile per caricare l'organigramma successivamente.
7. *Carica Organigramma*: Consente di caricare un organigramma precedentemente salvato in formato .json utilizzando l'opzione "Salva Organigramma".
8. *Azzera Organigramma*: Permette di ripristinare l'organigramma allo stato iniziale.
9. *Guida all'utilizzo*: Questa opzione apre una finestra contenente le istruzioni relative all'applicazione in esame.

## Avvio applicazione

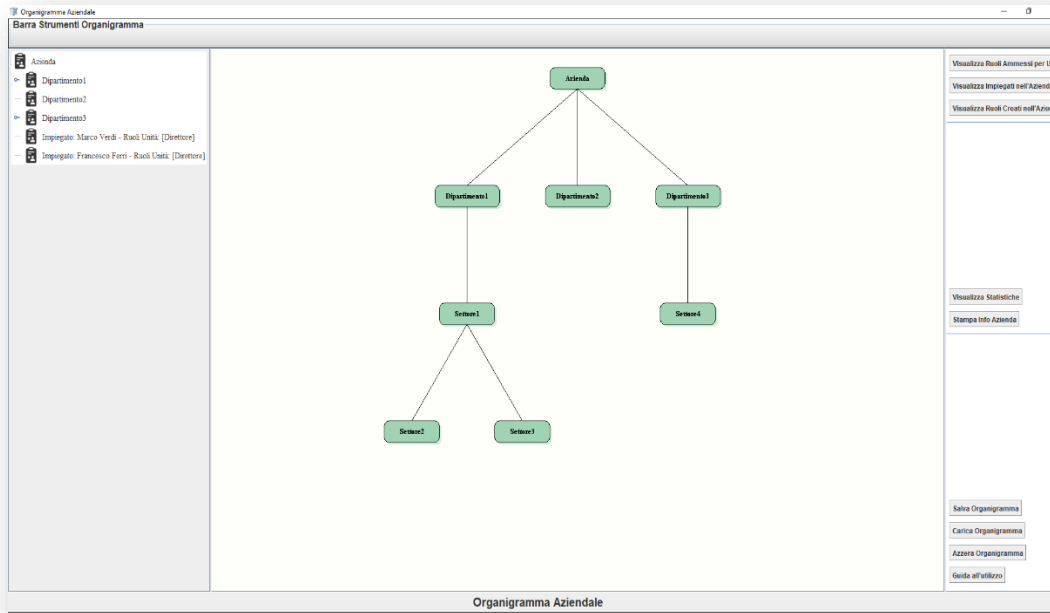
All'avvio l'applicazione si presenterà con una sola Unità creata, *Azienda*, con nessun ruolo o impiegato definito all'interno della struttura dell'organigramma.

*Per poter iniziare a lavorare è sufficiente modificare il nome dell'unica unità esistente e iniziare a definire a proprio piacimento, altre sotto unità, impiegati e ruoli.*

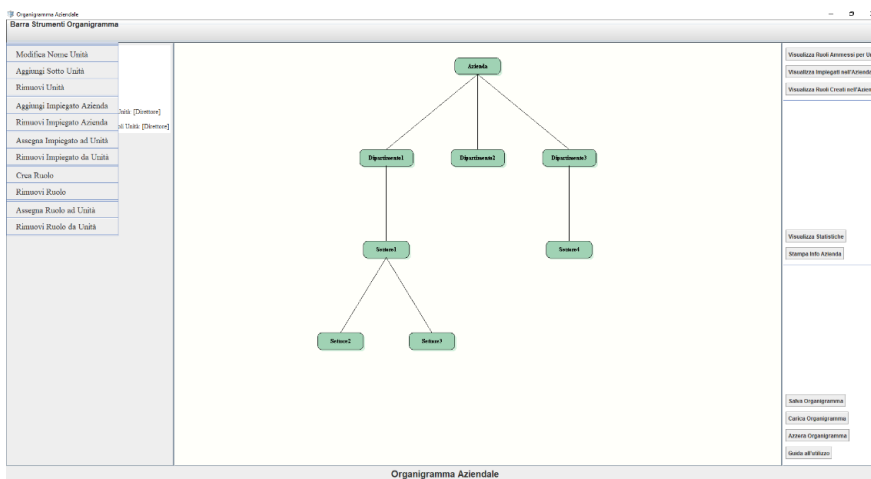


## Esempio creazione organigramma

La struttura si aggiorna dinamicamente in seguito all'aggiunta di unità, impiegati e ruoli.

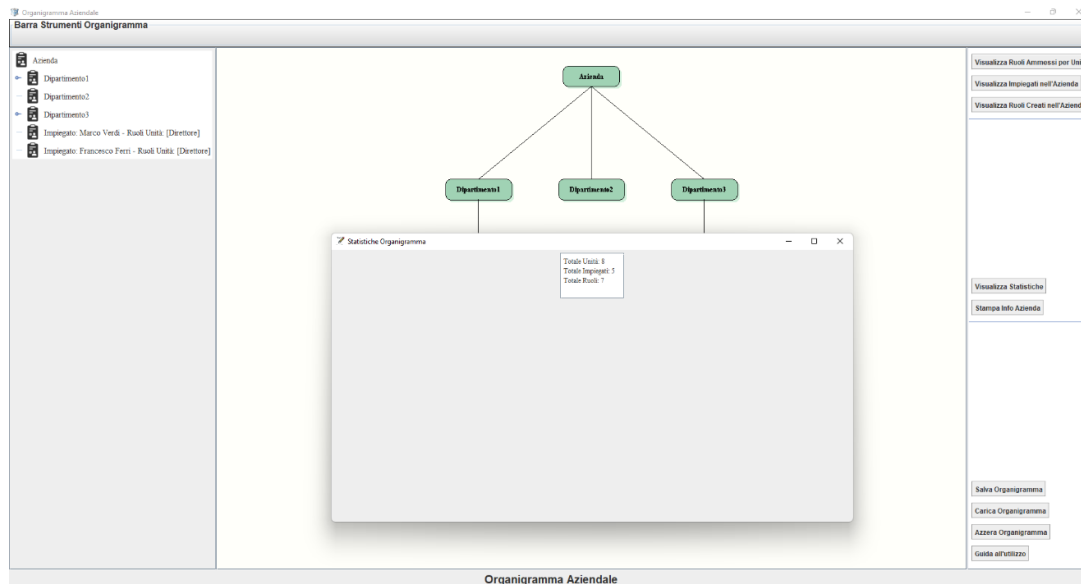


## Barra Strumenti Organigramma



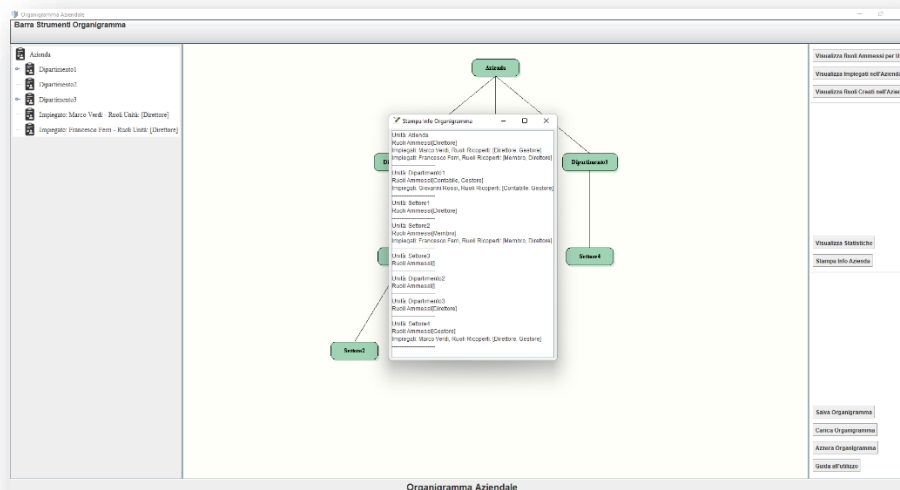
## Visualizzazione Statistiche dell'Organigramma

Le statistiche dell'organigramma possono essere visualizzate con una finestra apposita.

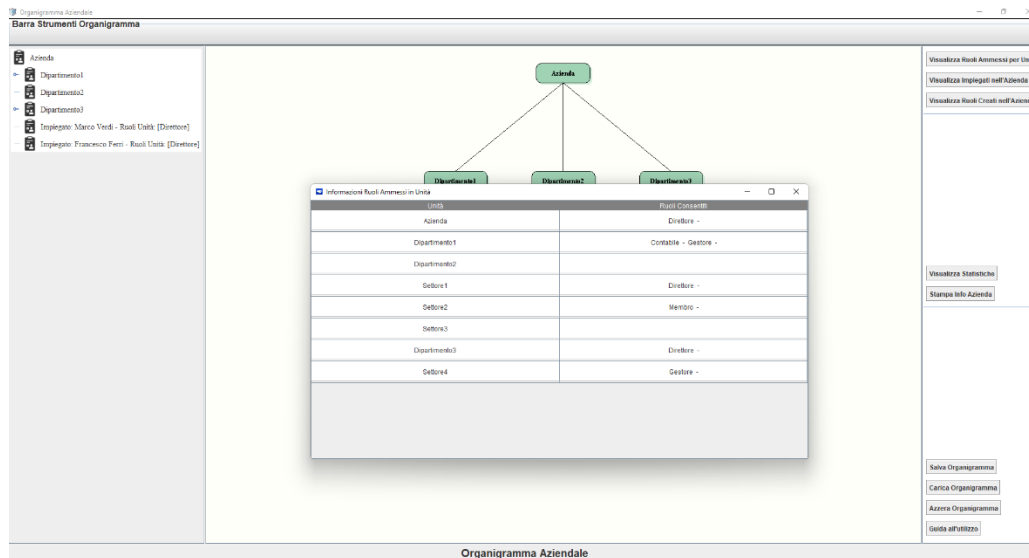


## Visualizzazione Testuale dell'Organigramma

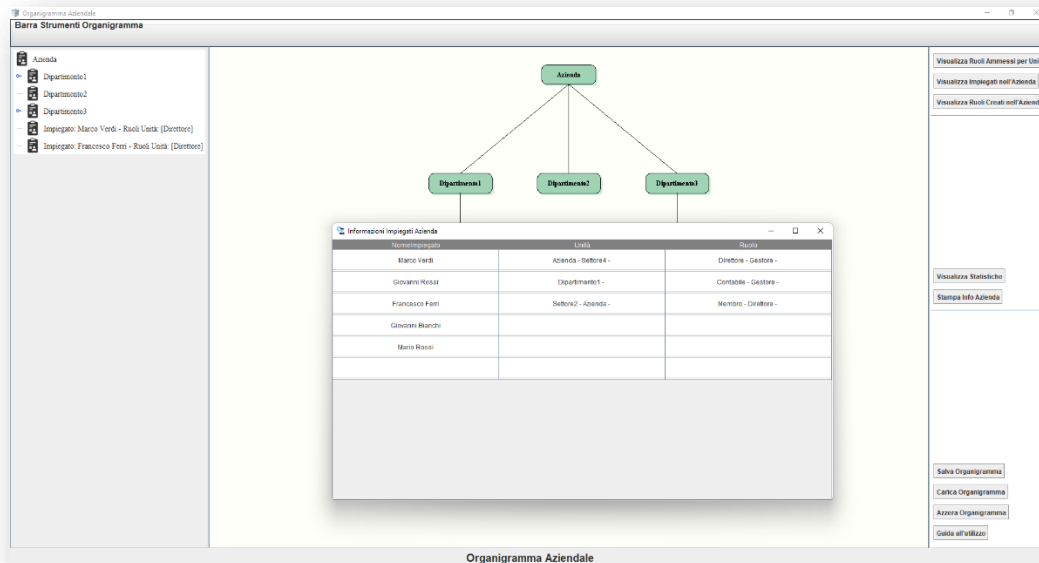
Una visualizzazione testuale della struttura dell'organigramma può essere osservata in una finestra apposita.



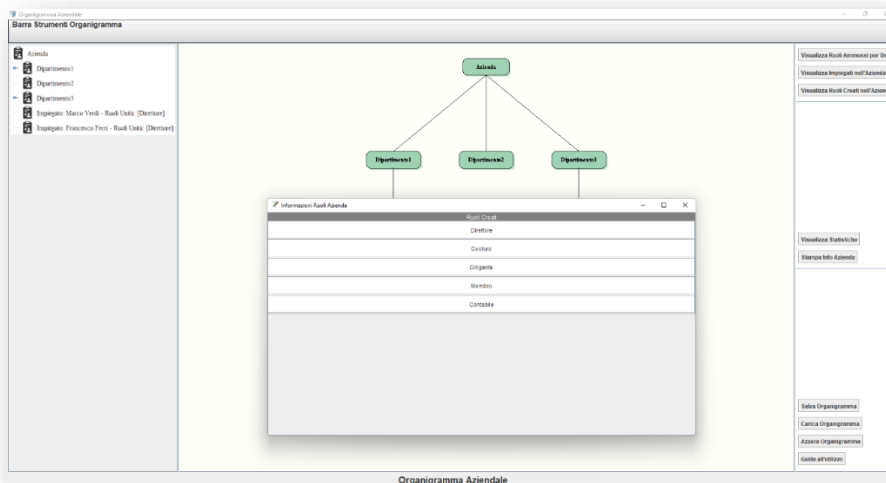
### Visualizzazione ruoli consentiti per ogni unità



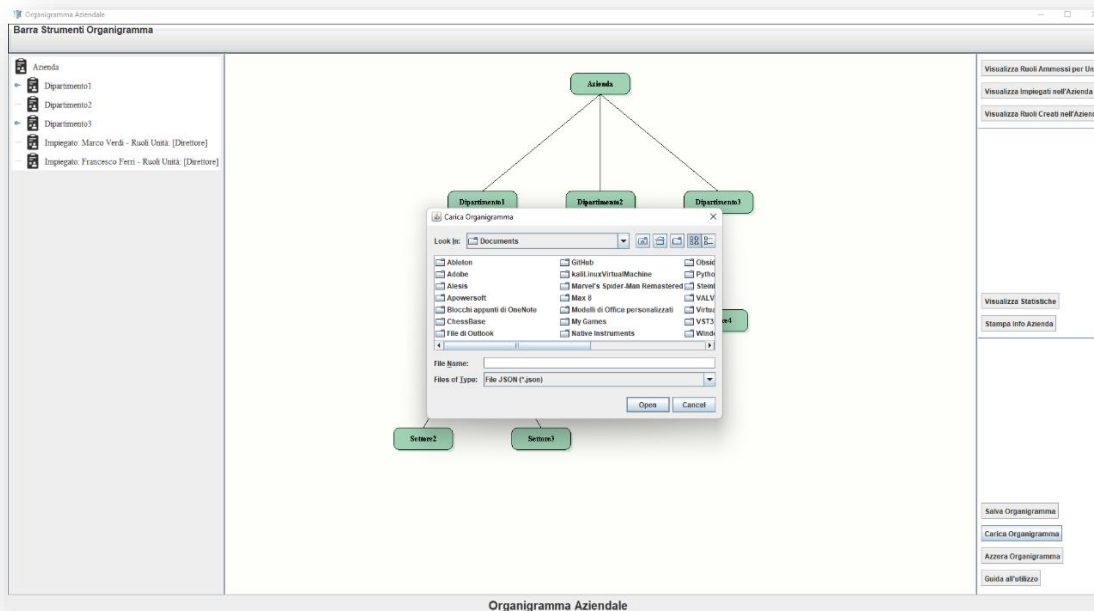
### Visualizzazione impiegati definiti nella struttura con eventuali assegnamenti in unità e ruoli ricoperti



### Visualizzazione ruoli definiti all'interno della struttura



### Visualizzazione finestra caricamento organigramma salvato esternamente *Analogo di finestra di salvataggio*



## Visualizzazione istruzioni utilizzo applicazione

