

# Mitro

μητρώο

Membri del team di sviluppo:

Amir Al Sadi 0000792816  
Jason Dellaluce 0000789710  
Federico Baldini 0000789767

# Indice

<b>Abstract</b>	4
<b>Analisi dei requisiti</b>	5
Requisiti del sistema	5
Analisi del dominio	7
Vocabolario	7
Sistemi esterni	9
Riconoscimento di altri sistemi dello stesso dominio	9
Analisi dei requisiti	10
Casi d'uso	14
Modello	14
Scenari	15
Analisi del Rischio	30
Tabella valutazione dei beni	30
Tabella minacce e controlli	30
Analisi Tecnologica della Sicurezza	31
Security Use Case & Misuse Case	33
Requisiti di Protezione dei Dati	37
<b>Analisi del Problema</b>	38
Analisi Documento dei Requisiti: Analisi delle Funzionalità	38
Tabella Funzionalità	38
GestioneAmministrazione: Tabella Informazioni/Flusso	38
GestioneClasse: Tabella Informazioni/Flusso	39
GestioneStudente: Tabella Informazioni/Flusso	40
Login: Tabella Informazioni/Flusso	41
ScritturaLog: Tabella Informazioni/Flusso	41
AnalisiLog: Tabella Informazioni/Flusso	42
Analisi Documento dei Requisiti: Analisi dei Vincoli	43
Tabella Vincoli	43
Analisi Documento dei Requisiti: Analisi delle Interazioni	44
Tabella Maschere	44
Tabella Sistemi Esterni	45
Analisi Ruoli e Responsabilità	46
Tabella Ruoli	46
Amministratore: Tabella Ruolo-Informazioni	47
Professore: Tabella Ruolo-Informazioni	48
Studente: Tabella Ruolo-Informazioni	48
GestoreSicurezza: Tabella Ruolo-Informazioni	49
Scomposizione del Problema	50

Tabella Scomposizione Funzionalità	50
GestioneAmministrazione: Tabella Sotto-Funzionalità	50
GestioneClasse: Tabella Sotto-Funzionalità	51
GestioneStudente: Tabella Sotto-Funzionalità	51
Creazione Modello del Dominio	52
Architettura Logica: Struttura	54
Diagramma dei package	54
Diagramma delle classi: Amministratore	55
Diagramma delle classi: Professore	56
Diagramma delle classi: Studente	57
Diagramma delle classi: Login	58
Diagramma delle classi: Log	59
Architettura Logica: Interazione	61
Diagramma di sequenza: Autenticazione avvenuta con successo	61
Diagramma di sequenza: Registrazione Utente	62
Diagramma di sequenza: Registrazione Classe	63
Diagramma di sequenza: Visualizzazione Log e Anomalie	64
Architettura Logica: Comportamento	65
Piano del Lavoro	66
Funzionalità del prototipo	66
Tempi di rilascio	67
Sviluppi futuri	67
Piano del Collaudo	68
<b>Progettazione</b>	<b>70</b>
Progettazione architettonica	70
Requisiti non funzionali	70
Scelte tecnologiche	70
Scelta dell'architettura	70
Cliente	71
Servitore	71
Persistenza	72
Considerazioni relative alle tecnologie utilizzate	72
Progettazione di Dettaglio	74
Struttura	74
Package mitro.model	74
Package mitro.controller	76
Package mitro.persistenza	77
Package mitro.controller.amministratore	78
Package mitro.controller.professore	79
Package mitro.controller.studente	80
Package mitro.controller.login	81

Package mitro.controller.log	82
Package mitro.view.amministratore	83
Package mitro.view.professore	84
Package mitro.view.studente	85
Package mitro.view.login	86
Package mitro.view.log	87
Interazioni	88
Diagramma di Sequenza: Autenticazione avvenuta con successo	88
Diagramma di Sequenza: Registrazione Utente	89
Diagramma di Sequenza: Registrazione Classe	90
Diagramma di Sequenza: Visualizzazione Log e Anomalie	91
Diagramma di Sequenza: Inserimento Calendario di Attività	92
Diagramma di sequenza: Registrazione Voto	93
Comportamento	94
Progettazione della Persistenza	95
Diagramma Entità-Relazione	95
Formato dei file di Log	96
Considerazioni su protezione e sicurezza	97
Progettazione del Collaudo	98
Progettazione per il Deployment	101
<b>Deployment</b>	<b>102</b>
Artefatti	102
Deployment Type-Level	102

# Abstract

Il progetto mira a realizzare una infrastruttura efficiente e scalabile in grado di fornire servizi a istituti scolastici di livello secondario.

Si intende realizzare un registro elettronico capace di interfacciare i professori, gli studenti e l'amministrazione, garantendo adeguate restrizioni degli accessi nel rispetto delle norme sulla privacy.

Le funzionalità previste emulano quindi quelle fornite dai tradizionali registri cartacei, ormai in disuso, con l'introduzione di servizi aggiuntivi.

L'esigenza di questi prodotti è stata sollevata dagli utenti ultimi (professori e studenti) che lamentano l'inadeguatezza di molti concorrenti del settore.

Si intende interagire attivamente con gli enti scolastici del territorio al fine di sviluppare adeguatamente struttura ed interfacce grafiche.

# Analisi dei requisiti

## Requisiti del sistema

Id. Requisito	Requisito	Tipo
R1F	Gli utenti del sistema devono potersi autenticare ed utilizzare il servizio.	Funzionale
R2F	Il sistema di autenticazione deve prevedere l'utilizzo di una coppia di credenziali chiamate username e password entrambe alfanumeriche, robuste e di scarsa intuibilità.	Funzionale
R3F	Il sistema deve prevedere tre distinti tipi di utente registrati che possono usufruire del servizio chiamati studente, professore, amministrazione.	Funzionale
R4F	Ognuno dei diversi utenti del sistema deve avere privilegi personalizzati di accesso ai dati, restrizioni, limiti sulla modifica degli stessi, e interfacce grafiche differenziate.	Funzionale
R5F	Il sistema deve permettere di organizzare le attività svolte nella scuola come lezioni, verifiche, interrogazioni. Studenti e professori interessati devono poter consultare un calendario grafico dei loro impegni.	Funzionale
R6F	Il sistema deve permettere ai professori di registrare le presenze nelle attività scolastiche. Professori e studenti devono poter consultare uno storico delle presenze nelle quali sono coinvolti.	Funzionale
R7F	I professori devono essere in grado di inserire e visionare valutazioni per ogni studente durante le proprie lezioni. Ogni studente deve poter visionare lo storico delle proprie valutazioni per ogni materia.	Funzionale
R8F	Studenti e professori devono poter ottenere visualizzazioni statistiche grafiche personalizzate su valutazioni e presenze nelle quali sono coinvolti.	Funzionale
R9F	L'amministrazione deve poter comunicare collettivamente o privatamente con studenti e professori.	Funzionale
R10F	L'amministrazione deve registrare, visualizzare, modificare le informazioni degli utenti del sistema.	Funzionale
R1NF	Il servizio deve garantire alte performance per le funzionalità di registrazione dei dati.	Non Funzionale
R2NF	Deve essere previsto un adeguato sistema di archiviazione	Non

	dei dati.	Funzionale
R3NF	Le interfacce grafiche devono essere minimali e di facile utilizzo.	Non funzionale
R4NF	Tutti gli username devono essere univoci per ogni utente e non devono contenere informazioni che permettano di dedurre il possessore di tale credenziale.	Non funzionale
R5NF	Il sistema deve prevedere misure di sicurezza in caso di compromissione di un utente, e deve operarsi ad impedire le stesse.	Non funzionale
R6NF	Il sistema deve garantire isolamento dei dati personali degli utenti, modulando adeguatamente l'accesso agli stessi secondo le norme di privacy.	Non funzionale
R7NF	Il sistema deve prevedere un metodo di registrazione di tutte le attività tecniche, mantenendo uno storico utile alla diagnostica e alla notifica.	Non funzionale
R8NF	Il servizio deve essere installato su dispositivi informatici di proprietà della scuola, la quale sarà responsabile del mantenimento degli stessi in funzione della qualità del prodotto.	Non funzionale

# Analisi del dominio

## Vocabolario

Voce	Definizione	Sinonimi
Servizio	Attività offerta dal sistema.	
Dispositivo	Apparecchiatura elettronica attraverso la quale il sistema può essere eseguito ed utilizzato.	Terminale, Macchina
Autenticazione	Processo attraverso il quale un utente del sistema viene riconosciuto ed identificato mediante l'inserimento delle proprie credenziali.	Accesso, Login
Credenziali	Insieme di parole necessarie ad un utente per essere identificato dal sistema ed accedervi.	
Username	Parola alfanumerica che identifica l'utente all'interno del sistema.	
Password	Parola alfanumerica accoppiata ad un Username necessaria all'identificazione ed accesso nel sistema da parte di un utente.	
Scuola	Istituzione ed infrastruttura formativa.	
Utente	Persona utilizzatrice del servizio.	Account, Profilo
Studente	Persona iscritta al percorso formativo scolastico, che può aver raggiunto o meno la maggior età.	
Professore	Persona che pratica l'insegnamento di una o più materie all'interno della scuola.	Insegnante
Amministrazione	Persona incaricata alla gestione ed organizzazione delle attività e strutture scolastiche.	Segreteria, Presidenza
Privilegio	Insieme di azioni ed informazioni concesse ad un determinato utente.	
Interfaccia grafica	Metodo visuale per l'interazione facilitata con il sistema attraverso i dispositivi supportati, destinato agli utenti.	GUI, Maschera, Schermata
Visualizzazione statistica	Rappresentazione grafica di un dato statistico calcolato su una collezione di informazioni.	Statistiche
Attività scolastica	Insieme di azioni svolte da insegnanti e studenti	

	all'interno della scuola durante un'ora.	
Lezione	Attività che prevede l'insegnamento di una materia da parte di un professore ad una classe, ha inizio in una certa data ed ora, ha una durata limitata.	
Verifica	Attività in cui tutta la classe o un suo sottosinsieme viene sottoposta ad una prova scritta e/o orale relativa alla materia di insegnamento della lezione stessa.	
Interrogazione	Attività in cui un sottosinsieme della classe viene sottoposto ad una prova orale relativa alla materia di insegnamento della lezione stessa.	
Materia	Tematica di insegnamento relativa ad un qualsiasi ambito disciplinare.	
Classe	Insieme di studenti, identificata con un certo nome e unica solo nell'anno scolastico di riferimento.	
Presenza	Il trovarsi fisicamente nel luogo previsto da un'attività scolastica da parte di uno studente o un professore.	
Assenza	Il non trovarsi fisicamente nel luogo previsto da un'attività scolastica da parte di uno studente o un professore.	
Valutazione	Quantità numerica compresa tra 1 e 10 che indica la performance di uno studente in una verifica o interrogazione, attribuita dal professore relativo alla materia in questione.	Voto
Calendario	Elenco delle attività a cui un utente è tenuto a partecipare.	Planner
Storico	Collezione di elementi o attività passate di un utente.	
Comunicazione	Documento digitale di carattere informativo.	Circolare
Registrazione dati	Atto di inserimento di una determinata informazione all'interno del sistema.	
Sistema di archiviazione	Insieme di tecnologie software ed hardware atte al contenimento, mantenimento, organizzazione dei dati del sistema.	Database
Attività tecnica	Processo o azione svolta dal sistema stesso nei suoi meccanismi interni.	

Compromissione	Rubare le credenziali ad un utente permettendo ad un contraffattore di essere identificato dal sistema a nome della vittima.	
Annotazione	Nota testuale inseribile da un docente all'interno del registro, visibile dagli studenti della classe riferita ad una materia e lezione. Può essere una nota generica, quindi può anche essere indicazione di verifica o interrogazione.	Nota

## Sistemi esterni

Il Sistema è costituito da un applicativo che consente l'accesso differenziato a tutti gli Utenti. Tramite un unico servitore software installato su un dispositivo o sistema di proprietà e responsabilità della scuola, ogni Utente accede in remoto al Sistema usando i propri codici identificativi. La scelta di delegare all'istituto scolastico il possesso del servitore su cui installare il Sistema consente allo stesso di organizzare le proprie risorse in modo da decidere i propri standard di performance e spese.

## Ricognizione di altri sistemi dello stesso dominio

L'esperienza passata in qualità di utenti finali di questo genere di servizio, insieme ai commenti di diversi docenti raccolti in fase di intervista, ci hanno permesso di concludere che i software concorrenti presenti attualmente sul mercato si rivelano spesso inefficaci e presentano difetti strutturali. Tali prodotti non sono inoltre particolarmente numerosi, ed adottarli risulta spesso una spesa economica rilevante per gli istituti formativi.

La primissima carenza rilevabile riguarda generalmente la performance. In genere infatti tali servizi sono costruiti secondo un modello centralizzato che permette ai gestori un pieno controllo dell'hardware ospitante e un opportuno sistema di licenze per gli abbonamenti. Tale scelta porta tuttavia a drastici cali di velocità e scomode attese nei momenti in cui l'utenza intende collegarsi al sistema contemporaneamente, come per esempio accade durante gli appelli mattutini nelle classi.

L'introduzione di una gestione interna e separata del servizio stesso, da noi proposta, permetterebbe alle singole scuole di occuparsi della disponibilità del servizio in base alla propria esigenza, non dipendendo sotto questo aspetto da enti esterni.

Questa scelta, inoltre, ridurrebbe i costi del software stesso.

Un'ulteriore aspetto tipicamente lamentato è la sovrabbondanza di funzionalità dei software e delle interfacce coinvolte. La nostra proposta progettuale intende produrre un software minimale e modulabile, introducendo quindi inizialmente solo le funzionalità fondamentali previste in una scuola permettendo al contempo l'aggiunta di componenti aggiuntive secondo le esigenze future dei consumatori. Tra le priorità fondamentali di questo modello vi è infine sicuramente la semplicità di utilizzo, particolarmente desiderata dagli utenti.

## Analisi dei requisiti

### Requisito R1F

- Gli Utenti del sistema devono potersi autenticare ed utilizzare il servizio.

Da questo requisito possiamo evincere che:

- Tutti gli Utenti si autenticano nel sistema per utilizzarne i servizi

### Requisito R2F

- Il sistema di Autenticazione deve prevedere l'utilizzo di una coppia di credenziali chiamate username e password entrambe alfanumeriche, robuste e di scarsa intuibilità.

Da questo requisito possiamo evincere che:

- Ogni Utente ottiene l'accesso al sistema introducendo le sue credenziali di accesso: username e password
- Username e password sono stringhe alfanumeriche

### Requisito R3F

- Il Sistema deve prevedere tre distinti tipi di utente registrati che possono usufruire del servizio chiamati Studente, Professore, Amministrazione.

Da questo requisito possiamo evincere che:

- Esistono tre entità che possono utilizzare il servizio: Studente, Professore e Amministrazione.

### Requisito R4F

- Ognuno dei diversi Utenti del Sistema deve avere privilegi personalizzati di Accesso ai dati, restrizioni, limiti sulla modifica degli stessi, e interfacce grafiche differenziate.

Da questo requisito possiamo evincere che:

- Ogni attore in gioco ha responsabilità diverse nell'interazione con il sistema.
- Le interfacce di accesso al Sistema sono differenziate in base al ruolo rivestito.

### Requisito R5F

- Il Sistema deve permettere di organizzare le attività svolte nella scuola come lezioni, verifiche, interrogazioni. Studenti e Professori interessati devono poter consultare un Calendario grafico dei loro impegni.

Da questo requisito possiamo evincere che:

- Esiste un Calendario che organizza le attività della Classe.
- I dati presenti nel Calendario sono visibili a Studenti e Professori.
- Ognuno dei due Utenti visiona solamente le attività del Calendario nelle quali è coinvolto.

- Tramite Annotazioni, potranno essere specificate nel registro della Classe, insieme alle note generiche, le attività sopra menzionate.

## Requisito R6F

- Il sistema deve permettere ai professori di registrare le presenze nelle attività scolastiche. Professori e studenti devono poter consultare uno storico delle presenze nelle quali sono coinvolti.

Da questo requisito possiamo evincere che:

- Esiste la possibilità di registrare ed ottenere uno storico delle Presenze nelle Attività Scolastiche.
- Uno Studente deve visionare un sommario delle attività alle quali ha o non ha partecipato.
- Un Professore deve visionare un sommario delle attività alle quali gli Studenti delle sue Classi hanno o non hanno presenziato.

## Requisito R7F

- I Professori devono essere in grado di inserire e visionare Valutazioni per ogni Studente durante le proprie lezioni. Ogni Studente deve poter visionare lo storico delle proprie Valutazioni per ogni Materia.

Da questo requisito possiamo evincere che:

- All'interno del Sistema si tiene traccia delle Valutazioni.
- Un Professore deve poter inserire e visionare solo le Valutazioni relative alle proprie Lezioni.
- Uno Studente deve poter visionare le proprie Valutazioni.

## Requisito R8F

- Studenti e Professori devono poter ottenere visualizzazioni statistiche grafiche personalizzate su valutazioni e presenze nelle quali sono coinvolti.

Da questo requisito possiamo evincere che:

- Esiste la possibilità di ottenere statistiche su Valutazioni e Presenze, per Studenti e Professori.

## Requisito R9F

- L'Amministrazione deve poter comunicare collettivamente o privatamente con studenti e professori.

Da questo requisito possiamo evincere che:

- L'Amministrazione comunica con ogni Studente o Professore tramite Comunicazioni.
- Le Comunicazioni possono essere indirizzate ad uno Studente o un Professore, singolarmente o in gruppo.

## Requisito R10F

- L'Amministrazione deve registrare, visualizzare, modificare le informazioni degli utenti del sistema.

Da questo requisito possiamo evincere che:

- L'Amministrazione ha potere di registrare ogni Professore e ogni Studente.
- L'Amministrazione ha potere di poter visualizzare e modificare le informazioni di Studenti e Professori.
- L'Amministrazione svolge un ruolo chiave nel sistema e risulta dunque un Account molto sensibile a eventuali attacchi o compromissioni.

## Requisito R1NF

- Il servizio deve garantire alte performance per le funzionalità di registrazione dei dati. Nasce l'esigenza di dover fornire un sistema veloce e performante, implementabile ad esempio con un sistema di gestione locale dei dati e degli account (Server interno).

## Requisito R2NF

- Deve essere previsto un adeguato sistema di archiviazione dei dati.

Serve l'implementazione di un sistema di archiviazione dei dati (per esempio un Database).

## Requisito R3NF

- Le interfacce grafiche devono essere minimali e di facile utilizzo.

L'implementazione delle interfacce grafiche deve avere come obiettivo principale la facilità di fruizione e utilizzo di essa.

## Requisito R4NF

- Tutti gli username devono essere univoci per ogni utente e non devono contenere informazioni che permettano di dedurre il possessore di tale credenziale.

E' necessario garantire nel sistema completa indipendenza fra username dell'utente e identità dell'utente. Questo requisito ci fornisce, fra le altre cose, un criterio di sicurezza da dover garantire per evitare la conoscenza esterna di dati sensibili.

## Requisito R5NF

- Il sistema deve prevedere misure di sicurezza in caso di compromissione di un utente, e deve operarsi ad impedire le stesse.

Il sistema deve essere modellato prendendo in considerazione la possibilità che vengano compromessi o attaccati gli Utenti. Dunque, il sistema dovrà il più possibile garantire all'Utente che queste situazioni non si presentino, specialmente nel caso dell'Amministrazione.

## Requisito R6NF

- Il sistema deve garantire isolamento dei dati personali degli utenti, modulando adeguatamente l'accesso agli stessi secondo le norme di privacy.

Il sistema deve garantire la gestione dei dati sensibili consentendo solo a chi di interesse di poterli utilizzare.

## Requisito R7NF

- Il sistema deve prevedere un metodo di registrazione di tutte le attività tecniche, mantenendo uno storico utile alla diagnostica e alla notifica.

Vengono introdotti nel Sistema dei componenti utili alla Diagnostica e alla Notifica. Questo può servire al Sistema per poter intercettare situazioni anomale (e.g. tante modifiche di informazioni degli Utenti lato Amministrazione potrebbe essere sintomo di attacco). Inoltre, la Diagnostica può essere utile ad un personale tecnico per dare spiegabilità a comportamenti anomali del Sistema.

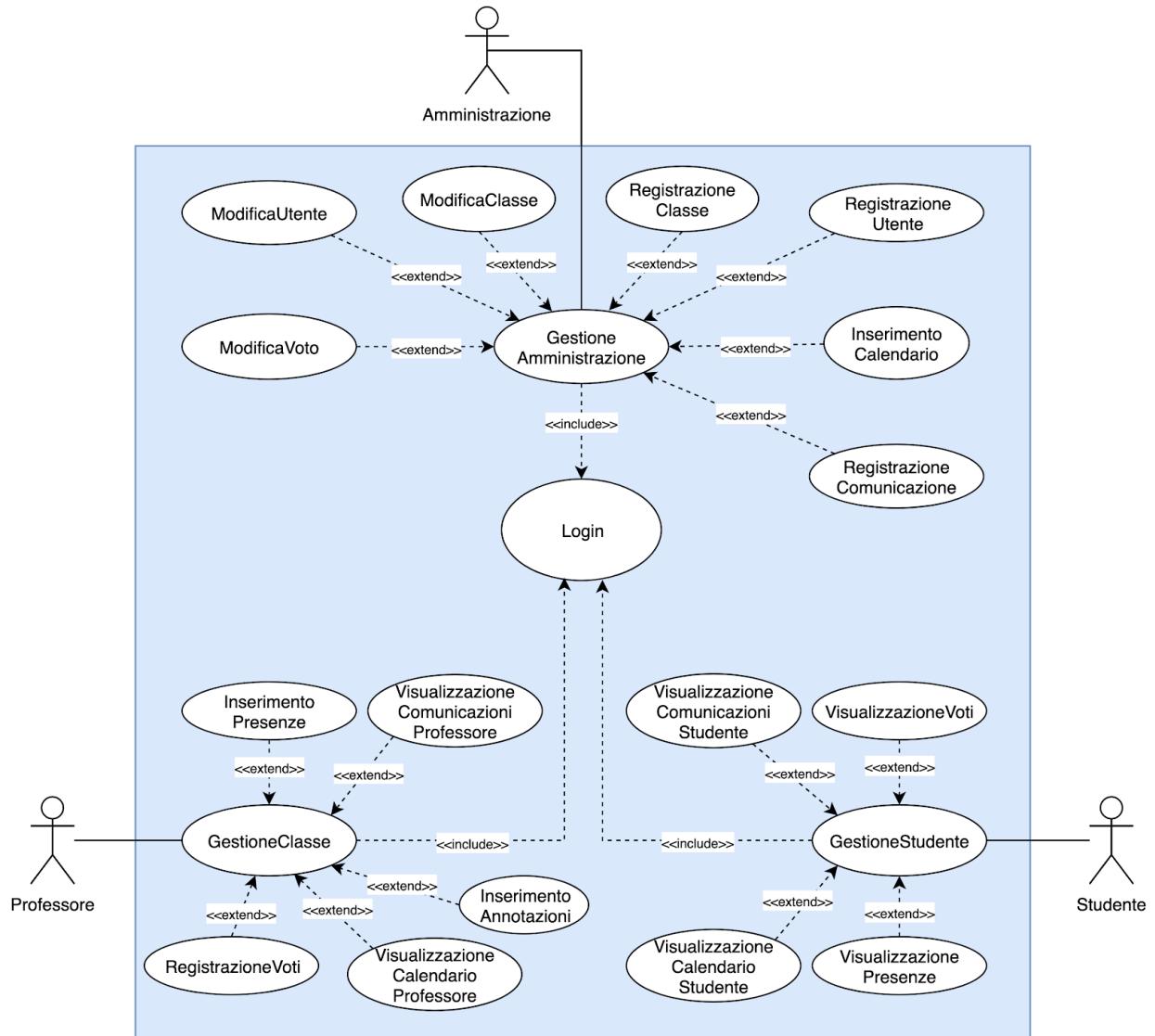
## Requisito R8NF

- Il servizio deve essere installato su dispositivi informatici di proprietà della scuola, la quale sarà responsabile del mantenimento degli stessi in funzione della qualità del prodotto.

Come accennato in precedenza, il Sistema deve garantire un servizio quanto più veloce ed efficiente. Per questo motivo, l'installazione di esso all'interno di un dispositivo interno alla Scuola, risulta una scelta verso questa direzione. Le componenti utili ai clienti (Studenti e Professori) devono essere installabili su Dispositivi a piacimento.

# Casi d'uso

## Modello



## Scenari

<b>Titolo</b>	Login
<b>Descrizione</b>	Modalità di accesso ed autenticazione nel sistema da parte dell'Utente.
<b>Attori</b>	Amministrazione, Professore, Studente
<b>Relazioni</b>	GestioneAmministrazione, GestioneClasse, GestioneStudente
<b>Precondizioni</b>	L'Utente deve essere registrato nel Sistema.
<b>Postcondizioni</b>	L'Utente è autenticato nel Sistema.
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Sistema presenta un'interfaccia di Accesso all'Utente.</li> <li>2. L'Utente inserisce le proprie Credenziali nell'interfaccia.</li> <li>3. L'Utente preme il bottone di Accesso.</li> <li>4. Il Sistema verifica validità e correttezza delle Credenziali.</li> <li>5. Viene presentata all'Utente l'interfaccia principale adatta a lui.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Credenziali non valide</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di accesso all'utente.</li> </ol> <p><i>Utente non registrato nel sistema</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di accesso all'utente.</li> </ol>
<b>Requisiti non funzionali</b>	<ul style="list-style-type: none"> <li>• La password digitata nell'apposito spazio per la credenziale non deve essere leggibile esplicitamente in modo da prevenirne il furto.</li> <li>• Intuitività e semplicità di utilizzo di interfaccia ed avvisi.</li> </ul>
<b>Punti aperti</b>	

<b>Titolo</b>	GestioneAmministrazione
<b>Descrizione</b>	Collezione delle attività possibili ad un Utente Amministrativo usando la sua apposita interfaccia.
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	ModificaUtente, ModificaVoto, ModificaClasse, Registrazione Utente, InserimentoCalendario, RegistrazioneComunicazione, RegistrazioneClasse
<b>Precondizioni</b>	
<b>Postcondizioni</b>	L'Utente Amministrativo è in grado di svolgere una qualsiasi azione di sua competenza dalla sua interfaccia personalizzata.

<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue il Login.</li> <li>2. Il Sistema presenta all'utente la sua schermata principale nella quale può eseguire tutte le attività di sua competenza.</li> </ol>
<b>Scenari alternativi</b>	<i>Autenticazione fallita</i> <ol style="list-style-type: none"> <li>1. Il Sistema notifica l'Utente la ragione del fallimento.</li> <li>2. Il Sistema ripresenta all'Utente l'interfaccia che gli permetta di ritentare l'Autenticazione.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	GestioneClasse
<b>Descrizione</b>	Collezione delle attività possibili ad un Professore usando la sua apposita interfaccia
<b>Attori</b>	Professore
<b>Relazioni</b>	RegistrazioneVoti, InserimentoPresenze, InserimentoAnnotazioni, VisualizzazioneComunicazioniProfessore, VisualizzazioneCalendarioProfessore
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Il Professore è in grado di svolgere una qualsiasi azione di sua competenza dalla sua interfaccia personalizzata
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login.</li> <li>2. Il Sistema presenta all'utente la sua schermata principale nella quale può eseguire tutte le attività di sua competenza.</li> </ol>
<b>Scenari alternativi</b>	<i>Autenticazione fallita</i> <ol style="list-style-type: none"> <li>1. Il Sistema notifica l'Utente la ragione del fallimento.</li> <li>2. Il Sistema ripresenta all'Utente l'interfaccia che gli permetta di ritentare l'Autenticazione.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	GestioneStudente
<b>Descrizione</b>	Collezione delle attività possibili ad uno Studente usando la sua apposita interfaccia

<b>Attori</b>	Studente
<b>Relazioni</b>	VisualizzazioneVoti, VisualizzazionePresenze, VisualizzazioneComunicazioniStudente, VisualizzazioneCalendarioStudente
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Lo Studente è in grado di svolgere una qualsiasi azione di sua competenza dalla sua interfaccia personalizzata
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Lo Studente esegue il Login.</li> <li>2. Il Sistema presenta all'utente la sua schermata principale nella quale può eseguire tutte le attività di sua competenza.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Autenticazione fallita</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema notifica l'Utente la ragione del fallimento.</li> <li>2. Il Sistema ripresenta all'Utente l'interfaccia che gli permetta di ritentare l'Autenticazione.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	ModificaUtente
<b>Descrizione</b>	Modifica delle informazioni relative a uno Studente o a un Professore
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	L'Utente da modificare deve essere registrato nel Sistema
<b>Postcondizioni</b>	L'Utente modificato deve poter riconoscere i cambiamenti effettuati a suo carico
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Modifica Utente" nella sua interfaccia.</li> <li>3. L'Utente Amministrativo indica se l'Utente da modificare è un Professore o uno Studente nell'apposita cella.</li> <li>4. L'Utente Amministrativo sceglie quale Utente modificare dalla lista che ottiene in una nuova tendina.</li> <li>5. L'Utente Amministrativo può cercare all'interno della tendina l'Utente da modificare.</li> <li>6. L'Utente Amministrativo preme un bottone invio che genera il caricamento dei dati dell'Utente.</li> <li>7. L'Utente Amministrativo può:</li> </ol>

	<p>A. Modificare i dati dell'Utente</p> <p>B. Rigenerare le Credenziali di un Utente, tramite un apposito pulsante.</p> <p>8. L'Utente Amministrativo preme un pulsante per confermare le modifiche.</p> <p>9. Nel caso venga scelta l'Opzione B, dopo aver premuto il pulsante di conferma, viene mostrato a video il contenuto delle nuove Credenziali generate.</p>
<b>Scenari alternativi</b>	<i>Inserimento di dati non validi all'interno delle caselle di testo</i> 1. Il sistema presenta un avviso di errore all'Utente. 2. Il sistema ripresenta l'interfaccia di modifica all'Utente Amministrativo.
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	ModificaVoto
<b>Descrizione</b>	Modifica della Valutazione relativa ad uno Studente
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	L'Amministrazione deve rilevare una richiesta verbale di Modifica del Voto da parte di un Professore
<b>Postcondizioni</b>	Professore e Studente devono poter riconoscere i cambiamenti effettuati a loro carico
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Modifica Voto" nella sua interfaccia.</li> <li>3. L'Utente Amministrativo sceglie il nome del Professore e la Classe di riferimento per il cambiamento del voto.</li> <li>4. L'Utente Amministrativo preme il bottone di conferma della sua scelta.</li> <li>5. L'Utente Amministrativo preme un bottone invio che genera il caricamento della tabella dei voti dell'Utente.</li> <li>6. L'Utente Amministrazione visualizza una tabella al cui interno sono registrati i voti per data.</li> <li>7. A questo punto, l'Utente Amministrativo sceglie che voto modificare e attua la modifica scegliendo dall'opportuno menù a tendina il voto da inserire.</li> <li>8. L'Utente Amministrativo preme il pulsante di conferma delle modifiche.</li> </ol>

<b>Scenari alternativi</b>	
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	RegistrazioneUtente
<b>Descrizione</b>	Inserimento delle informazioni relative a uno Studente o a un Professore all'interno del Sistema
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	L'Utente da registrare deve essere un Professore o uno Studente, deve essere parte della Scuola di riferimento, e non ancora registrato nel Sistema
<b>Postcondizioni</b>	L'Utente deve risultare regolarmente iscritto e deve poter fare Accesso al Sistema
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Registrazione Utente" nella sua interfaccia.</li> <li>3. L'Utente Amministrativo riempie le caselle dei dati anagrafici generici per ogni Utente (nome, cognome, data di nascita,...).</li> <li>4. L'Utente Amministrativo può scegliere di registrare:           <ol style="list-style-type: none"> <li>A. Un Professore.</li> <li>B. Uno Studente.</li> </ol> </li> <li>5. L'Utente Amministrativo preme un bottone invio che genera il caricamento dei dati dell'Utente.</li> <li>6. Dopo aver premuto il pulsante di conferma, viene mostrato a video il contenuto delle Credenziali Utente generate.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Inserimento di dati non validi all'interno delle caselle di testo</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	RegistrazioneClasse
<b>Descrizione</b>	Inserimento delle informazioni relative a una Classe all'interno del Sistema.
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	L'Utente da registrare deve essere uno Studente, deve essere parte della Scuola di riferimento, e deve essere registrato nel Sistema.
<b>Postcondizioni</b>	L'Utente deve risultare appartenente a quella Classe.
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Registrazione Classe" nella sua interfaccia.</li> <li>3. L'Utente Amministrativo riempie le caselle dei dati generici relativi alla classe (nome,anno accademico,ecc...).</li> <li>4. L'Utente Amministrativo inserisce nella Classe una lista di Studenti appartenenti al Sistema.</li> <li>5. L'Utente Amministrativo preme un bottone invio che genera il caricamento della Classe nel Sistema.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Inserimento di una Classe con nome già esistente</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol> <p><i>Inserimento di una Classe in un anno accademico diverso da quello corrente</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol> <p><i>Inserimento dello stesso alunno per 2 volte nella lista della Classe</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	ModificaClasse
<b>Descrizione</b>	Modifica delle informazioni relative a una Classe all'interno del Sistema.
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione

<b>Precondizioni</b>	La Classe da modificare deve essere correttamente registrata nel sistema.
<b>Postcondizioni</b>	Gli Utenti appartenenti a quella Classe devono poter visualizzare le modifiche apportate.
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Modifica Classe" nella sua interfaccia.</li> <li>3. L'Utente Amministrativo modifica le caselle dei dati generici relativi alla classe (nome, anno accademico, ecc...).</li> <li>4. L'Utente Amministrativo modifica la lista di Studenti della Classe.</li> <li>5. L'Utente Amministrativo preme un bottone invio che genera l'aggiornamento della Classe nel Sistema.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Assegnamento alla Classe si un nome già esistente</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol> <p><i>Assegnamento alla Classe di un anno accademico diverso da quello corrente</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol> <p><i>Inserimento dello stesso alunno per 2 volte nella lista della Classe</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione all'Utente Amministrativo.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	InserimentoCalendario
<b>Descrizione</b>	Registrazione del Calendario settimanale in vigore per tutto l'Anno Scolastico
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Tutti gli Utenti devono visualizzare il Calendario una volta che esso è stato generato.
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L'Utente Amministrativo esegue Login.</li> <li>2. L'Utente Amministrativo seleziona la sezione "Inserimento</li> </ol>

	<p>Calendario” nella sua interfaccia.</p> <ol style="list-style-type: none"> <li>3. L’Utente Amministrativo sceglie una Classe tramite un menù di scelta.</li> <li>4. L’Utente Amministrativo inserisce il Calendario settimanale di una Classe.</li> <li>5. L’Utente Amministrativo preme un pulsante di conferma dei dati inseriti.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Inserimento di orario settimanale incompleto</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all’Utente.</li> <li>2. Il sistema ripresenta l’interfaccia di registrazione del Calendario all’Utente Amministrativo.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell’interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	RegistrazioneComunicazione
<b>Descrizione</b>	Inserimento di una Comunicazione dedicata a un gruppo di Studenti e/o Professori
<b>Attori</b>	Amministrazione
<b>Relazioni</b>	GestioneAmministrazione
<b>Precondizioni</b>	La segreteria deve aver deliberato una Comunicazione di qualsiasi genere di interesse di Professori e/o Studenti
<b>Postcondizioni</b>	Gli Utenti interessati alla comunicazione devono ricevere nella casella dedicata una nuova Comunicazione
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. L’Utente Amministrativo esegue Login.</li> <li>2. L’Utente Amministrativo seleziona la sezione “Registrazione Comunicazione” nella sua interfaccia.</li> <li>3. L’Utente Amministrativo carica nell’ apposita casella di inserimento una Comunicazione (e.g. formato PDF, testuale, altri).</li> <li>4. L’Utente Amministrativo aggiunge nell’apposito menù a tendina la Classe/Professore/Studente/Tutti interessato/i.</li> <li>5. In una apposita casella sottostante compare la lista di Utenti che riceveranno la Comunicazione.</li> <li>6. Una volta aggiunti tutti i destinatari, l’Utente Amministrativo preme un pulsante di conferma per inviare la Comunicazione.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Non è stato inserito un contenuto di Comunicazione</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all’Utente.</li> <li>2. Il sistema ripresenta l’interfaccia di registrazione della Comunicazione all’Utente Amministrativo.</li> </ol>

	<p><i>Non è stato inserito un contenuto di formato giusto</i></p> <ol style="list-style-type: none"> <li>1. Il sistema presenta un avviso di errore all'Utente.</li> <li>2. Il sistema ripresenta l'interfaccia di registrazione della Comunicazione all'Utente Amministrativo.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	RegistrazioneVoti
<b>Descrizione</b>	Registrazione di Valutazioni relative ad uno o più Studenti di una Classe in una determinata attività scolastica
<b>Attori</b>	Professore
<b>Relazioni</b>	GestioneClasse
<b>Precondizioni</b>	<p>Il Professore utilizzatore e gli Studenti relativi alle Valutazioni devono essere propriamente registrati nel Sistema.</p> <p>Al momento dell'inserimento l'attività relativa alle Valutazioni deve essere avvenuta o in fase di svolgimento, ed il Professore utilizzatore deve esserne il referente.</p>
<b>Postcondizioni</b>	Le Valutazioni sono registrate nel Sistema, gli Studenti ed il Professore coinvolti potranno visionarla negli storici presenti nelle rispettive interfacce
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login</li> <li>2. Il Professore seleziona la sezione "Registra Valutazioni" nella sua interfaccia.</li> <li>3. Il Professore sceglie una classe ed una relativa lezione dalle apposite liste di selezione</li> <li>4. Il Sistema presenta la lista degli Studenti presumibilmente presenti nella lezione selezionata</li> <li>5. Il Professore inserisce la valutazione numerica nei campi di inserimento relativi agli Studenti interessati presenti nella lista</li> <li>6. Il Professore preme un bottone di conferma per rendere permanenti le Valutazioni inserite</li> </ol>
<b>Scenari alternativi</b>	<p><i>Il Professore non inserisce alcuna valutazione ma preme ugualmente il bottone di conferma</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema notifica il Professore dell'inappropriatezza della sua azione</li> </ol> <p><i>Il Professore inserisce erroneamente dati scorretti e preme il bottone di conferma</i></p> <ol style="list-style-type: none"> <li>1. Il Professore deve contattare verbalmente l'Amministrazione richiedendo un alteramento o eliminazione dei dati inseriti in precedenza.</li> </ol>

<b>Requisiti non funzionali</b>	Responsività dell’interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	InserimentoAnnotazioni
<b>Descrizione</b>	Registrazione di Annotazioni relative ad uno o più Studenti di una Classe in una determinata Lezione
<b>Attori</b>	Professore
<b>Relazioni</b>	GestioneClasse
<b>Precondizioni</b>	Il Professore utilizzatore e gli Studenti che visualizzano le Annotazioni devono essere propriamente registrati nel Sistema. Il Professore che inserisce la nota deve poterle inserire solo nelle Lezioni di sua competenza, in una qualsiasi data a sua scelta.
<b>Postcondizioni</b>	Le Annotazioni sono registrate nel Sistema, gli Studenti potranno visionarle negli storici presenti nelle rispettive interfacce.
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login</li> <li>2. Il Professore seleziona la sezione “Registra Annotazioni” nella sua interfaccia.</li> <li>3. Il Professore sceglie una classe ed una relativa lezione dalle apposite liste di selezione.</li> <li>4. Il Sistema presenta un menù in cui inserire un’Annotazione e nella quale sono presenti le Annotazioni che ha già inserito in un unico campo di testo modificabile.</li> <li>5. Il Professore inserisce una Annotazione nel relativo campo di inserimento, decidendo se modificare l’Annotazione inserita o aggiungere del nuovo testo dopo il testo già inserito.</li> <li>6. Il Professore preme un bottone di conferma per rendere permanenti le modifiche alle Annotazioni.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Il Professore non inserisce alcuna Annotazione ma preme ugualmente il bottone di conferma.</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema notifica il Professore dell’inappropriatezza della sua azione.</li> </ol> <p><i>Il Professore inserisce erroneamente dati scorretti e preme il bottone di conferma.</i></p> <ol style="list-style-type: none"> <li>1. Il Professore può modificare successivamente l’Annotazione relativa a quella lezione.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell’interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	InserimentoPresenze
<b>Descrizione</b>	Registrazione delle Presenze relative ad uno o più Studenti di una Classe in una determinata attività scolastica
<b>Attori</b>	Professore
<b>Relazioni</b>	GestioneClasse
<b>Precondizioni</b>	Il Professore utilizzatore e gli Studenti relativi alle Presenze devono essere propriamente registrati nel Sistema. Al momento dell'inserimento l'attività relativa alle Presenze deve essere avvenuta o in fase di svolgimento, ed il Professore utilizzatore deve esserne il referente.
<b>Postcondizioni</b>	Le Presenze sono registrate nel Sistema, gli Studenti ed il Professore coinvolti potranno visionarle negli storici presenti nelle rispettive interfacce
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login</li> <li>2. Il Professore seleziona la sezione "Registra Presenze" nella sua interfaccia.</li> <li>3. Il Professore sceglie una classe ed una relativa lezione dalle apposite liste di selezione</li> <li>4. Il Sistema presenta la lista degli Studenti presumibilmente presenti nella lezione selezionata</li> <li>5. Il Professore spunta la casella relativa ad ogni Studente nella lista che è (o è stato) fisicamente presente alla lezione indicata</li> <li>6. Il Professore preme un bottone di conferma per rendere permanenti le Valutazioni inserite</li> </ol>
<b>Scenari alternativi</b>	<p><i>Il Professore compila erroneamente la lista delle presenze e preme il pulsante di conferma</i></p> <ol style="list-style-type: none"> <li>1. Il Professore può ricompilare lo stesso modulo di Presenze indicando la stessa coppia Classe-Lezione, che una volta confermato andrà a sovrascrivere i dati errati inseriti in precedenza.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazioneComunicazioniProfessore
<b>Descrizione</b>	Visualizzazione una o più Comunicazioni indirizzate ad un Professore da parte dell'Amministrazione
<b>Attori</b>	Professore

<b>Relazioni</b>	GestioneClasse
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Il Professore è in grado di aprire e leggere le Comunicazioni ricevute
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login.</li> <li>2. Il Professore seleziona la sezione "Comunicazioni" nella sua interfaccia.</li> <li>3. Il Sistema presenta al Professore una lista delle Comunicazioni indirizzate a lui, ordinate per data di ricezione.</li> <li>4. Il Professore sceglie la Comunicazione alla quale è interessato e preme un bottone per aprirla.</li> </ol>
<b>Scenari alternativi</b>	<i>Il Professore non ha ricevuto comunicazioni dall'Amministrazione</i> <ol style="list-style-type: none"> <li>1. Il Sistema presenta al Professore una lista vuota.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazioneCalendarioProfessore
<b>Descrizione</b>	Visualizzazione delle Attività scolastiche alle quali un Professore dovrà presenziare, e a tutte quelle in cui ha dovuto essere presente in passato
<b>Attori</b>	Professore
<b>Relazioni</b>	GestioneClasse
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Il Professore è in grado di conoscere tutte le Attività Scolastiche che lo competono
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Il Professore esegue il Login.</li> <li>2. Il Professore seleziona la sezione "Calendario" nella sua interfaccia.</li> <li>3. Il Sistema presenta al Professore l'elenco delle attività di sua competenza future e passate, organizzate per data ed ora.</li> </ol>
<b>Scenari alternativi</b>	<i>Non vi sono Attività Scolastiche relative al Professore</i> <ol style="list-style-type: none"> <li>1. Il Sistema mostra al Professore un elenco vuoto.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazioneComunicazioniStudente
<b>Descrizione</b>	Visualizzazione una o più Comunicazioni indirizzate ad uno Studente da parte dell'Amministrazione
<b>Attori</b>	Studente
<b>Relazioni</b>	GestioneStudente
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Lo Studente è in grado di aprire e leggere le Comunicazioni ricevute
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Lo Studente esegue il Login.</li> <li>2. Lo Studente seleziona la sezione "Comunicazioni" nella sua interfaccia.</li> <li>3. Il Sistema presenta allo Studente una lista delle Comunicazioni indirizzate a lui, ordinate per data di ricezione.</li> <li>4. Lo Studente sceglie la Comunicazione alla quale è interessato e preme un bottone per aprirla.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Lo Studente non ha ricevuto comunicazioni dall'Amministrazione</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema presenta allo Studente una lista vuota.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazioneCalendarioStudente
<b>Descrizione</b>	Visualizzazione delle Attività scolastiche alle quali uno Studente dovrà presenziare, e a tutte quelle in cui ha dovuto essere presente in passato
<b>Attori</b>	Studente
<b>Relazioni</b>	GestioneStudente
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Lo Studente è in grado di conoscere tutte le Attività Scolastiche che lo competono
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Lo Studente esegue il Login.</li> <li>2. Lo Studente seleziona la sezione "Calendario" nella sua interfaccia.</li> <li>3. Il Sistema presenta allo Studente l'elenco delle attività di sua competenza future e passate, organizzate per data ed ora.</li> </ol>

<b>Scenari alternativi</b>	Non vi sono Attività Scolastiche relative allo Studente 1. Il Sistema mostra al Professore un elenco vuoto.
<b>Requisiti non funzionali</b>	Responsività dell’interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazioneVoti
<b>Descrizione</b>	Visualizzazione dello Storico delle Valutazioni che uno Studente può aver conseguito nelle Attività Scolastiche in cui era presente
<b>Attori</b>	Studente
<b>Relazioni</b>	GestioneStudente
<b>Precondizioni</b>	
<b>Postcondizioni</b>	Lo Studente è in grado di conoscere tutte le Valutazione alle quali era interessato, e di visionare opportune statistiche relative alle stesse
<b>Scenario principale</b>	<ol style="list-style-type: none"> <li>1. Lo Studente esegue il Login.</li> <li>2. Dalla sua schermata principale, lo Studente seleziona una attività relativa alla sua Classe.</li> <li>3. Il Sistema presenta allo Studente una lista di tutte le Valutazioni inerenti all’attività selezionata.</li> <li>4. Lo Studente preme un elemento della lista di Valutazioni per mostrarne le relative statistiche.</li> </ol>
<b>Scenari alternativi</b>	<p><i>Lo Studente non è partecipe di una Classe</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema presenta una lista di attività vuota.</li> </ol> <p><i>Lo Studente non ha conseguito Valutazioni per l’attività selezionata</i></p> <ol style="list-style-type: none"> <li>1. Il Sistema presenta all’Utente una lista di Valutazioni vuota.</li> </ol>
<b>Requisiti non funzionali</b>	Responsività dell’interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

<b>Titolo</b>	VisualizzazionePresenze
<b>Descrizione</b>	Visualizzazione dello Storico delle Presenze e Assenze dello Studente nelle Attività Scolastiche che lo competono
<b>Attori</b>	Studente
<b>Relazioni</b>	GestioneStudenti

<b>Precondizioni</b>	
<b>Postcondizioni</b>	Lo Studente è in grado di conoscere tutte le Presenze e Assenze alle quali era interessato
<b>Scenario principale</b>	<p>1. Lo Studente esegue il Login.</p> <p>2. Il Sistema presenta allo Studente, nella sua schermata principale, una lista ordinata per data e ora di tutte le Presenze e Assenze che egli ha effettuato.</p>
<b>Scenari alternativi</b>	<p><i>Lo Studente non è partecipe di una Classe</i></p> <p>1. Il Sistema presenta una lista vuota.</p> <p><i>Non vi sono Attività Scolastiche relative allo Studente</i></p> <p>1. Il Sistema presenta una lista vuota.</p>
<b>Requisiti non funzionali</b>	Responsività dell'interfaccia e facilità di esplorazione
<b>Punti aperti</b>	

## Analisi del Rischio

Tabella valutazione dei beni

Bene	Valore	Esposizione
Sistema Informativo	<b>Alto</b> Archivia e gestisce le informazioni dell'intero sistema scolastico	<b>Alto</b> Perdita di immagine e finanziaria dati i costi di ripristino del sistema
Informazioni relative agli Studenti	<b>Alto</b> Dati sensibili degli Studenti, tra cui dati personali e valutazioni individuali	<b>Alto</b> Perdita di immagine in caso di divulgazione di dati critici; situazione peggiore se i dati sono di un minorenne
Informazioni relative ai Professori	<b>Alto</b> Dati sensibili dei Professori e parziali dati sensibili degli Studenti	<b>Alto</b> Perdita di immagine in caso di divulgazione di dati o corruzione dei dati
Informazioni relative all'Amministrazione	<b>Alto</b> Accesso, creazione e modifica dei dati sensibili degli altri utenti	<b>Molto Alto</b> Perdita di immagine in caso di divulgazione dei dati; Elevato rischio di corruzione del sistema, la compromissione porta ad una perdita finanziaria

Tabella minacce e controlli

Minaccia	Probabilità	Controllo	Fattibilità
Furto credenziali relative ad uno Studente	Media	<ul style="list-style-type: none"> <li>• Log delle operazioni</li> <li>• Generazione di password da parte dell'Amministrazione</li> </ul>	Basso costo implementativo  Basso costo implementativo
Furto credenziali relative ad un Professore	Alta	<ul style="list-style-type: none"> <li>• Log delle operazioni</li> <li>• Generazione di password da parte dell'Amministrazione</li> </ul>	Basso costo implementativo  Basso costo implementativo
Furto credenziali relative ad un	Bassa	<ul style="list-style-type: none"> <li>• Log delle operazioni</li> </ul>	Basso costo implementativo

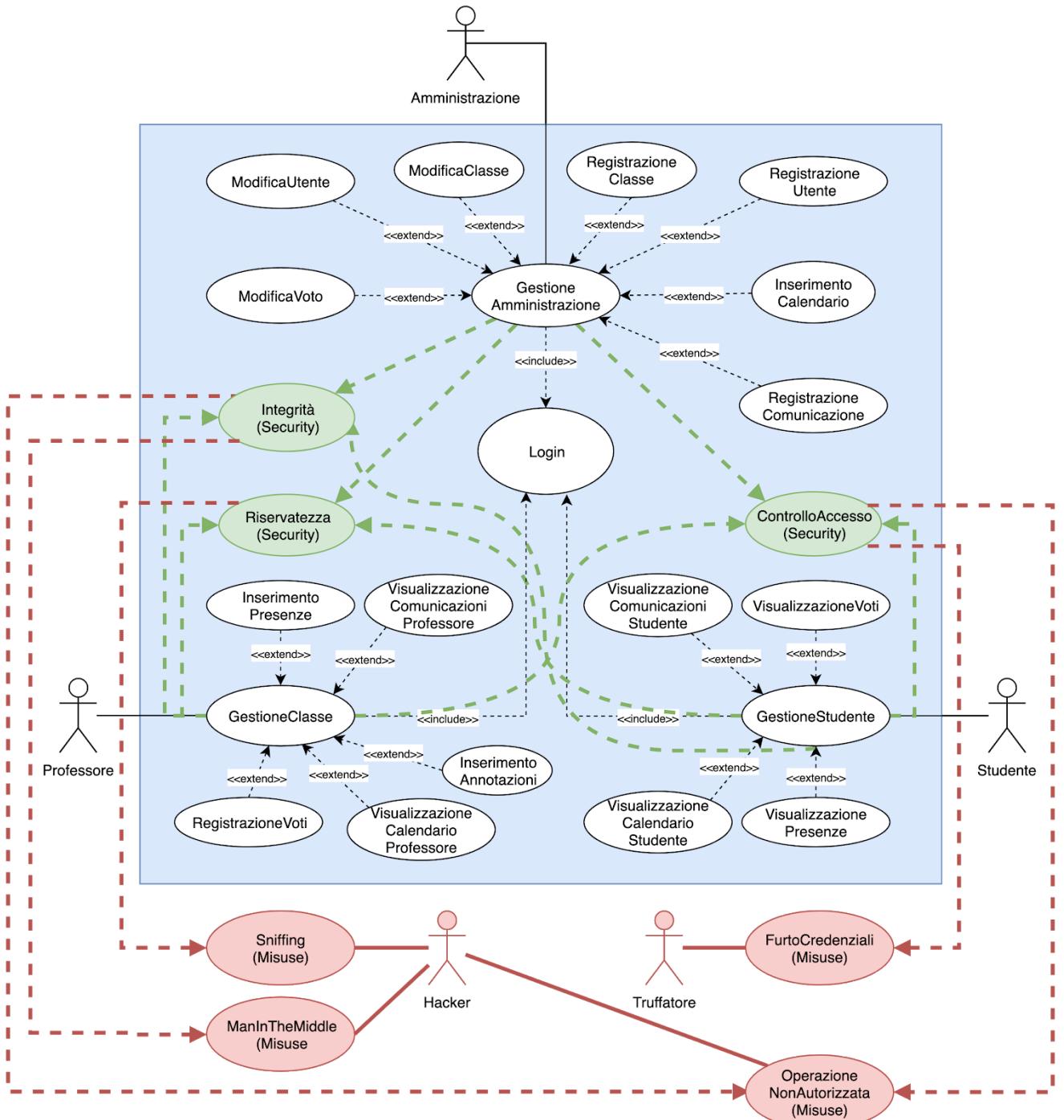
Amministratore		<ul style="list-style-type: none"> <li>• Accesso solo da macchine sicure</li> <li>• Generazione di password da parte del Sistema</li> </ul>	<p>Basso costo di realizzazione, la postazione può essere lasciata incustodita</p> <p>Basso costo implementativo</p>
Intercettazione comunicazioni	Alta	<ul style="list-style-type: none"> <li>• Cifratura delle comunicazioni</li> </ul>	Costo variabile a seconda della tecnologia scelta
Man In The Middle	Media	<ul style="list-style-type: none"> <li>• Cifratura delle comunicazioni</li> <li>• Sistema di autenticazione</li> </ul>	<p>Costo variabile a seconda della tecnologia scelta</p> <p>Costo variabile a seconda della tecnologia scelta</p>
DoS	Bassa	Il sistema viene ospitato dinamicamente in base al numero massimo di utilizzatori tra tutti gli utenti	Basso costo, non vi è modo di prevenire con certezza un attacco DoS

## Analisi Tecnologica della Sicurezza

Tecnologia	Vulnerabilità
Cifratura delle comunicazioni	<p>La cifratura delle comunicazioni presenta vulnerabilità sia sul lato della riservatezza dei dati sia sull'autenticazione dei soggetti in gioco.</p> <p>In linea generale per la riservatezza dei dati viene utilizzato un sistema a chiave simmetrica che presenta le seguenti vulnerabilità:</p> <ul style="list-style-type: none"> <li>• Memorizzazione della chiave: l'archiviazione sicura della chiave deve essere adeguata, al fine di prevenire il furto;</li> <li>• Tempo di vita della chiave: maggiori informazioni cifrate con la stessa chiave e più materiale ha l'attaccante per individuare un collegamento tra i vari messaggi cifrati;</li> <li>• Lunghezza della chiave: essendo tutte le chiavi attaccabili con un</li> </ul>

	<p>sistema a forza bruta, maggiore è la lunghezza della chiave e maggiore è la difficoltà (e il costo) per individuarla;</p> <p>Mentre per l'autenticazione degli individui interessati a comunicare tra di loro viene utilizzato solitamente un sistema a chiave asimmetrica con le seguenti vulnerabilità:</p> <ul style="list-style-type: none"> <li>• Memorizzazione della chiave privata: l'archiviazione sicura della chiave deve essere adeguata, al fine di prevenire il furto;</li> </ul>
Autenticazione tramite credenziali	<ul style="list-style-type: none"> <li>• L'utente decide volontariamente di rivelare le proprie credenziali</li> <li>• L'utente tramite l'inganno rivela le proprie credenziali</li> </ul>
Architettura Client/Server	<ul style="list-style-type: none"> <li>• Intercettazione delle comunicazioni</li> <li>• DoS</li> <li>• Man In The Middle</li> </ul>

## Security Use Case & Misuse Case



<b>Titolo</b>	Integrità	
<b>Descrizione</b>	I dati inseriti nel Sistema e presentati agli Utenti devono essere corretti ed autentici.	
<b>Misuse Case</b>	ManInTheMiddle, OperazioneNonAutorizzata	
<b>Relazioni</b>	GestioneAmministrazione, GestioneClasse, GestioneStudente	
<b>Precondizioni</b>	<ol style="list-style-type: none"> <li>1. L'attaccante ha i mezzi fisici o informatici per intercettare le comunicazioni tra Sistema e Utenti o per autenticarsi come un Utente vittima.</li> <li>2. L'attaccante ha i mezzi per leggere e modificare i messaggi della comunicazione.</li> <li>3. L'attaccante ha i mezzi per spedire il messaggio modificato al destinatario.</li> </ol>	
<b>Postcondizioni</b>	Il Sistema deve notificare i tentativi di manomissione delle comunicazioni e controllare che i dati trasmessi abbiano validità rispetto al dominio applicativo.	
<b>Scenario principale</b>	<b>Sistema</b>	<b>Attaccante</b>
	Il Sistema si occupa di proteggere i messaggi scambiati con gli Utenti e di registrarli in un log.	
		Intercetta uno o più messaggi scambiati, ma non riuscendo a rimuovere i meccanismi di protezione manda al destinatario un nuovo messaggio.
<b>Scenario di attacco avvenuto con successo</b>	Il Sistema riconosce la non autenticità del messaggio ricevuto, quindi notifica e registra un tentativo di attacco.	
	<b>Sistema</b>	<b>Attaccante</b>
	Il Sistema si occupa di proteggere i messaggi scambiati con gli Utenti e di registrarli in un log.	
		Intercetta uno o più messaggi scambiati e riesce a rimuovere i meccanismi di protezione. Successivamente modifica i dati contenuti nel messaggio, riapplica i meccanismi di protezione, e lo manda al destinatario.

	Il Sistema elabora il messaggio ed agisce di conseguenza. Registra il messaggio in un log e ne controlla il contenuto sintattico e semantico in modo da non compromettere la correttezza dei dati persistenti, quando possibile.	
	Il Sistema analizza periodicamente il log alla ricerca di pattern di accesso o interazione atipici, in modo da segnalare eventuali attacchi.	

<b>Titolo</b>	Riservatezza	
<b>Descrizione</b>	I dati inseriti nel Sistema e presentati agli Utenti devono essere visionabili solo dal diretto interessato.	
<b>Misuse Case</b>	Sniffing	
<b>Relazioni</b>	GestioneAmministrazione, GestioneClasse, GestioneStudente	
<b>Precondizioni</b>	L'attaccante ha i mezzi fisici o informatici per intercettare o assistere alle comunicazioni tra Sistema e Utenti.	
<b>Postcondizioni</b>	Il Sistema deve aver reso esternamente illeggibili i dati scambiati nella comunicazione, ed aver registrato tutti i messaggi che la costituiscono.	
<b>Scenario principale</b>	Sistema	Attaccante
	Il Sistema si occupa di proteggere i messaggi scambiati con gli Utenti e di registrarli in un log.	
<b>Scenario di attacco avvenuto con successo</b>		Intercetta o prende visione di uno o più messaggi scambiati, ma non riesce a rimuovere i meccanismi di protezione.
	Sistema	Attaccante
		Intercetta o prende visione di uno o più messaggi scambiati, riesce a rimuovere i meccanismi di protezione e a leggere il contenuto.

<b>Titolo</b>	ControlloAccesso	
<b>Descrizione</b>	L'accesso ai servizi del Sistema deve essere controllato.	
<b>Misuse Case</b>	FurtoCredenziali, OperazioneNonAutorizzata	
<b>Relazioni</b>	GestioneAmministrazione, GestioneClasse, GestioneStudente	
<b>Precondizioni</b>	L'attaccante ha i mezzi fisici o informatici per venire a conoscenza delle Credenziali di accesso di un Utente vittima, o a parti di esse.	
<b>Postcondizioni</b>	Il Sistema deve aver registrato e tenuto traccia dei tentativi di accesso e delle azioni compiute dall'Utente.	
<b>Scenario principale</b>	<i>Sistema</i>	<i>Attaccante</i>
		Attua uno o più tentativi per accedere al Sistema fingendosi l'Utente vittima, per poi interagire con i servizi a suo nome.
	Controlla le credenziali immesse e blocca l'accesso nel caso tali credenziali risultino errate dopo un certo numero di tentativi.	
<b>Scenario di attacco avvenuto con successo</b>	<i>Sistema</i>	<i>Attaccante</i>
		Attua uno o più tentativi per accedere al Sistema fingendosi l'Utente vittima, per poi interagire con il servizi a suo nome.
	Controlla le credenziali immesse, le quali risultano corrette e consentono l'accesso.	
		Naviga nelle interfacce disponibili all'Utente vittima e può eventualmente tentare di interagire con funzionalità alle quali egli non dovrebbe aver accesso per mancanza dei privilegi necessari.
	Il Sistema tiene traccia in un log di tutte le azioni e richieste effettuate dall'Utente vittima, e blocca l'accesso alle funzionalità che non gli sono consentite.	
	Il Sistema analizza periodicamente il log alla ricerca di pattern di accesso o interazione	

	atipici, in modo da segnalare eventuali attacchi.	
--	---	--

## Requisiti di Protezione dei Dati

Dall'analisi del rischio si può concludere che il requisito R7NF può essere specificato dai seguenti punti:

1. Creazione di un log nel quale tracciare:
  - a. Tutte le azioni che avvengono nel Sistema.
  - b. Tutti i messaggi scambiati durante le comunicazioni con gli Utenti.
2. Gestione di un sistema di analisi periodica del log per:
  - a. Rilevare azioni atipiche o sospette avvenute nel Sistema.
  - b. Rilevare discrepanze nei messaggi scambiati con gli Utenti.

Si evince inoltre la necessità di due nuovi requisiti non funzionali:

1. Tutti i dati memorizzati nel Sistema devono essere protetti.
2. Tutti i dati scambiati nelle comunicazioni con il Sistema devono essere protette.

Si ritiene infine che i requisiti già introdotti in precedenza, insieme a quelli già citati, siano sufficienti per un'opportuna protezione dei dati.

**Nota:** Le specificazioni del requisito R7NF implica la presenza di un nuovo tipo di Utente unico nel Sistema, di qui in poi identificato come **GestoreSicurezza**, che è esclusivamente incaricato dell'analisi e visione dei Log.

# Analisi del Problema

## Analisi Documento dei Requisiti: Analisi delle Funzionalità

Tabella Funzionalità

Funzionalità	Tipo	Grado Complessità	Requisiti Collegati
Gestione Amministrazione	Memorizzazione e gestione dati	Complessa	R3F, R4F, R9F, R10F
GestioneClasse	Interazione con l'esterno, memorizzazione e gestione dati	Complessa	R3F, R4F, R5F, R6F, R7F, R8F
GestioneStudente	Interazione con l'esterno, gestione dati	Complessa	R3F, R4F, R5F, R6F, R7F, R8F
Login	Interazione con l'esterno, gestione dati	Semplice	R1F, R2F
ScritturaLog	Memorizzazione dati	Semplice	R7NF
AnalisiLog	Gestione dati	Complessa	R7NF

GestioneAmministrazione: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Nome Utente	Semplice	Alto	Input	Non più di 50 caratteri
Cognome Utente	Semplice	Alto	Input	Non più di 50 caratteri
Username Utente	Semplice	Molto alto	Input	Non più di 32 caratteri
Password Utente	Semplice	Molto alto	Input	Non più di 32 caratteri
Tipo Utente	Semplice	Medio	Input	
Email Utente	Semplice	Alto	Input	Non più di 256 caratteri

Indirizzo Utente	Semplice	Alto	Input	Non più di 128 caratteri
Recapito telefonico Utente	Semplice	Alto	Input	Non più di 16 caratteri
Data di nascita Utente	Semplice	Alto	Input	Non più di 10 caratteri
Voto	Composto	Alto	Input	
Classe	Composto	Medio	Input	
Tipo Attività Scolastica	Semplice	Medio	Input	Non più di 50 caratteri
Identificativo Classe Attività Scolastica	Semplice	Basso	Input	
Identificativo Professore Attività Scolastica	Semplice	Basso	Input	
Data Attività Scolastica	Semplice	Medio	Input	Non più di 10 caratteri
Ora Attività Scolastica	Semplice	Medio	Input	Non più di 5 caratteri
Comunicazione	Composto	Alto	Input	Non più di 5 MegaByte
Identificativo Utente destinatario Comunicazione	Semplice	Alto	Input	
Data Comunicazione	Semplice	Medio	Input	Non più di 10 caratteri
Ora Comunicazione	Semplice	Medio	Input	Non più di 5 caratteri
Oggetto Comunicazione	Semplice	Alto	Input	Non più di 100 caratteri

## GestioneClasse: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Nome	Semplice	Alto	Output	Non più di 50 caratteri
Cognome	Semplice	Alto	Output	Non più di 50 caratteri
Classe	Composto	Medio	Output	
Presenza	Composto	Alto	Input	
Voto	Composto	Alto	Input	
Annotazione	Semplice	Medio	Input	Non più di 1000 caratteri
Tipo Attività Scolastica	Semplice	Medio	Output	Non più di 50 caratteri
Data Attività Scolastica	Semplice	Medio	Output	Non più di 10 caratteri
Ora Attività Scolastica	Semplice	Medio	Output	Non più di 5 caratteri
Comunicazione	Composto	Alto	Output	Non più di 5 MegaByte
Data Comunicazione	Semplice	Medio	Output	Non più di 10 caratteri
Ora Comunicazione	Semplice	Medio	Output	Non più di 5 caratteri
Oggetto Comunicazione	Semplice	Alto	Output	Non più di 100 caratteri

## GestioneStudente: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Nome	Semplice	Alto	Output	Non più di 50

				caratteri
Cognome	Semplice	Alto	Output	Non più di 50 caratteri
Classe	Composto	Medio	Output	
Presenza	Composto	Alto	Output	
Voto	Composto	Alto	Output	
Annotazione	Semplice	Medio	Output	Non più di 1000 caratteri
Tipo Attività Scolastica	Semplice	Medio	Output	Non più di 50 caratteri
Data Attività Scolastica	Semplice	Medio	Output	Non più di 10 caratteri
Ora Attività Scolastica	Semplice	Medio	Output	Non più di 5 caratteri
Comunicazione	Composto	Alto	Output	Non più di 5 MegaByte
Data Comunicazione	Semplice	Medio	Output	Non più di 10 caratteri
Ora Comunicazione	Semplice	Medio	Output	Non più di 5 caratteri
Oggetto Comunicazione	Semplice	Alto	Output	Non più di 100 caratteri

### Login: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Username	Semplice	Molto alto	Input	Non più di 32 caratteri
Password	Semplice	Molto alto	Input	Non più di 32 caratteri

### ScritturaLog: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Data	Semplice	Medio	Input	Non più di 10 caratteri
Ora	Semplice	Medio	Input	Non più di 5 caratteri
Utente attore	Semplice	Alto	Input	
Azione eseguita	Composto	Alto	Input	
Messaggio comunicazione	Composto	Molto alto	Input	

### AnalisiLog: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione / privacy	Input / output	Vincoli
Azione eseguita	Composto	Alto	Output	
Messaggio comunicazione	Composto	Molto alto	Output	
Anomalia	Composto	Medio	Output	

## Analisi Documento dei Requisiti: Analisi dei Vincoli

Tabella Vincoli

Requisito	Categorie	Impatto	Funzionalità
Velocità memorizzazione dati (R1NF)	Tempo di risposta	Cercare di migliorare	Gestione Amministrazione, GestioneClasse
Protezione dei dati (R2NF, R4NF, R6NF)	Sicurezza	Aumentano la privacy dei dati a scapito del tempo di risposta	Gestione Amministrazione, GestioneClasse, Login
Controllo Accessi (R5NF)	Sicurezza	Aumentano la privacy dei dati a scapito del tempo di risposta e dell'usabilità	Gestione Amministrazione, GestioneClasse, GestioneStudente, Login, ScritturaLog, AnalisiLog
Controllo Azioni (R7NF)	Sicurezza	Aumentano la privacy dei dati a scapito del tempo di risposta	Gestione Amministrazione, GestioneClasse, GestioneStudente, Login, ScritturaLog, AnalisiLog
Semplice utilizzo delle interfacce (R3NF)	Usabilità	Cercare di migliorare	Gestione Amministrazione, GestioneClasse, GestioneStudente, Login

# Analisi Documento dei Requisiti: Analisi delle Interazioni

Tabella Maschere

Maschera	Informazioni	Funzionalità
Home Amministratore	Messaggio benvenuto e scelta funzionalità	GestioneAmministrazione
View Modifica Utente	Lista Professori e Studenti, dati relativi all'Utente selezionato	GestioneAmministrazione
View Modifica Voto	Lista Professori e Classi, Valutazioni relative agli Studenti selezionati	GestioneAmministrazione
View Modifica Classe	Lista Studenti della Classe, informazioni sulla Classe selezionata	GestioneAmministrazione
View Registrazione Utente	Lista Professori e Studenti, dati relativi all'Utente inserito	GestioneAmministrazione
View Inserimento Calendario	Lezioni disposte per orario e Professori che partecipano ad una determinata Lezione	GestioneAmministrazione
View Registrazione Comunicazione	Contenuto da comunicare, oggetto della Comunicazione e lista degli Utenti	GestioneAmministrazione
View Registrazione Classe	Lista Studenti, lista delle Classi, informazioni sulla Classe creata	GestioneAmministrazione
Home Professore	Messaggio benvenuto, Calendario delle Attività.	GestioneClasse
View Registrazione Presenze	Lista delle Classi, Attività, Studenti e Presenze	GestioneClasse
View Registrazione Voti	Lista delle Classi, Attività, Studenti e Voti	GestioneClasse
View Registrazione Annotazione	Calendario delle Attività, informazioni riguardanti una verifica, un'interrogazione o una Nota generica.	GestioneClasse
View Storico Professore	Lista degli Studenti, relative statistiche di Presenze e Valutazioni degli Studenti	GestioneClasse
Home Studente	Messaggio benvenuto, Calendario delle Attività, Valutazioni, lista delle proprie Presenze ed Assenze, statistiche.	GestioneStudente

View Comunicazioni	Comunicazioni indirizzate allo Studente o Professore.	GestioneClasse, GestioneStudente
Home Log	Messaggio di benvenuto e scelta delle funzionalità.	AnalisiLog
View Log	Data, ora, operazione eseguita.	AnalisiLog
View Anomalie	Elenco delle azioni anomale registrate.	AnalisiLog
View Login	Username, password.	Login

## Tabella Sistemi Esterni

Analizzando i requisiti del sistema, abbiamo concluso che non vi sono sistemi software esterni con cui l'applicativo ha necessità di interfacciarsi.

# Analisi Ruoli e Responsabilità

Tabella Ruoli

Ruolo	Responsabilità	Maschere	Riservatezza	Numerosità
Amministratore	Gestione di tutte le informazioni relative a Professori e Studenti registrati nel Sistema, alle Attività Scolastiche, e alle Comunicazioni.	Home Amministratore, View Modifica Utente, View Modifica Voto, View Modifica Classe, View Registrazione Utente, View Inserimento Calendario, View Registrazione Classe, View Registrazione Comunicazione, View Login	Molto alto	Il numero di componenti dell'Amministrazione è fissato dall'istituto, considerando l'alternanza dei turni di lavoro e dei giorni di riposo. Il numero deve essere il minimo necessario.
Professore	Visualizzazione di tutte le informazioni relative al singolo Professore, alle Attività Scolastiche che lo interessano, agli Studenti coinvolti in queste ultime. Gestione di Presenze, Voti, e Annotazioni relativi alle Classi nelle quali svolge le Attività Scolastiche.	Home Professore, View Registrazione Presenze, View Registrazione Voti, View Registrazione Annotazione, View Storico Professore, View Comunicazioni, View Login	Alto	Il numero di Professori è determinato dall'organico dell'istituto formativo, e limitato unicamente dalle risorse del sistema.
Studente	Visualizzazione di tutte le informazioni relative al singolo Studente, alle Attività Scolastiche che lo	Home Studente, View Comunicazioni, View Login	Alto	Il numero di Studenti è determinato dagli iscritti nell'istituto formativo, e limitato

	interessano, ai Professori coinvolti in queste ultime.			unicamente dalle risorse del Sistema.
Gestore Sicurezza	Visualizzazione dei Log.	Home Log, View Log, View Anomalie, View Login	Medio	2 o 3 persone, considerando l'alternanza dei turni di lavoro e dei giorni di riposo. L'Utente registrato nel sistema rimane tuttavia unico.

### Amministratore: Tabella Ruolo-Informazioni

Informazione	Tipo di accesso
Nome Utente	Lettura/scrittura
Cognome Utente	Lettura/scrittura
Username Utente	Scrittura
Password Utente	Scrittura
Tipo Utente	Lettura/scrittura
Email Utente	Lettura/scrittura
Indirizzo Utente	Lettura/scrittura
Recapito telefonico Utente	Lettura/scrittura
Data di nascita Utente	Lettura/scrittura
Voto	Lettura/scrittura
Classe	Lettura/scrittura
Tipo Attività Scolastica	Lettura/scrittura
Identificativo Classe Attività Scolastica	Lettura/scrittura
Identificativo Professore Attività Scolastica	Lettura/scrittura
Data Attività Scolastica	Lettura/scrittura
Ora Attività Scolastica	Lettura/scrittura
Comunicazione	Lettura/scrittura

Identificativo Utente destinatario Comunicazione	Lettura/scrittura
Data Comunicazione	Lettura/scrittura
Ora Comunicazione	Lettura/scrittura
Oggetto Comunicazione	Lettura/scrittura
Username	Scrittura
Password	Scrittura

### Professore: Tabella Ruolo-Informazioni

Informazione	Tipo di accesso
Nome	Lettura
Cognome	Lettura
Classe	Lettura
Presenza	Lettura/scrittura
Voto	Lettura/scrittura
Annotazione	Lettura/scrittura
Tipo Attività Scolastica	Lettura
Data Attività Scolastica	Lettura
Ora Attività Scolastica	Lettura
Comunicazione	Lettura
Data Comunicazione	Lettura
Ora Comunicazione	Lettura
Oggetto Comunicazione	Lettura
Username	Scrittura
Password	Scrittura

### Studente: Tabella Ruolo-Informazioni

Informazione	Tipo di accesso
Nome	Lettura

Cognome	Lettura
Classe	Lettura
Presenza	Lettura
Voto	Lettura
Annotazione	Lettura
Tipo Attività Scolastica	Lettura
Data Attività Scolastica	Lettura
Ora Attività Scolastica	Lettura
Comunicazione	Lettura
Data Comunicazione	Lettura
Ora Comunicazione	Lettura
Oggetto Comunicazione	Lettura
Username	Scrittura
Password	Scrittura

### GestoreSicurezza: Tabella Ruolo-Informazioni

Informazione	Tipo di accesso
Data	Lettura
Ora	Lettura
Utente attore	Lettura
Azione eseguita	Lettura
Messaggio comunicazione	Lettura
Anomalia	Lettura
Username	Scrittura
Password	Scrittura

## Scomposizione del Problema

Tabella Scomposizione Funzionalità

Funzionalità	Scomposizione
GestioneAmministrazione	ModificaUtente, ModificaVoto, ModificaClasse, RegistrazioneUtente, InserimentoCalendario, RegistrazioneComunicazione, RegistrazioneClasse
GestioneClasse	InserimentoPresenze, RegistrazioneVoti, VisualizzazioneComunicazioniProfessore, InserimentoAnotazioni, VisualizzazioneCalendarioProfessore
GestioneStudente	VisualizzazioneVoti, VisualizzazioneComunicazioniStudente, VisualizzazioneCalendarioStudente, VisualizzazionePresenze

GestioneAmministrazione: Tabella Sotto-Funzionalità

Sotto-funzionalità	Sotto-funzionalità	Legame	Informazioni
ModificaUtente	Registrazione Utente	Non si possono modificare le informazioni di un utente non registrato.	Identificativo Utente
RegistrazioneClasse	Registrazione Utente	Non si possono inserire Studenti non registrati in una Classe.	Identificativo Studente
ModificaClasse	RegistrazioneClasse	Non si possono modificare Classi inesistenti.	Identificativo Classe, Identificativo Studenti della Classe
Inserimento Calendario	RegistrazioneClasse	Non si possono coinvolgere Classi inesistenti in Attività Scolastiche.	Identificativo Classe
Registrazione Comunicazione	Registrazione Utente	Non si possono inviare Comunicazioni ad	Identificativo Utente destinatario

		Utenti non registrati.	
ModificaVoto	Registrazione Utente	Non si può modificare un Voto di uno Studente non registrato.	Identificativo Studente

### GestioneClasse: Tabella Sotto-Funzionalità

Sotto-funzionalità	Sotto-funzionalità	Legame	Informazioni
RegistrazioneVoti	Inserimento Presenze	Non si può assegnare un Voto ad uno Studente non presente nella relativa Attività Scolastica.	Identificativo Studente
Inserimento Annotazioni	Visualizzazione Calendario Professore	Un Professore non può inserire Annotazioni per un'Attività Scolastica che non gli è possibile visionare.	Identificativo Attività Scolastica

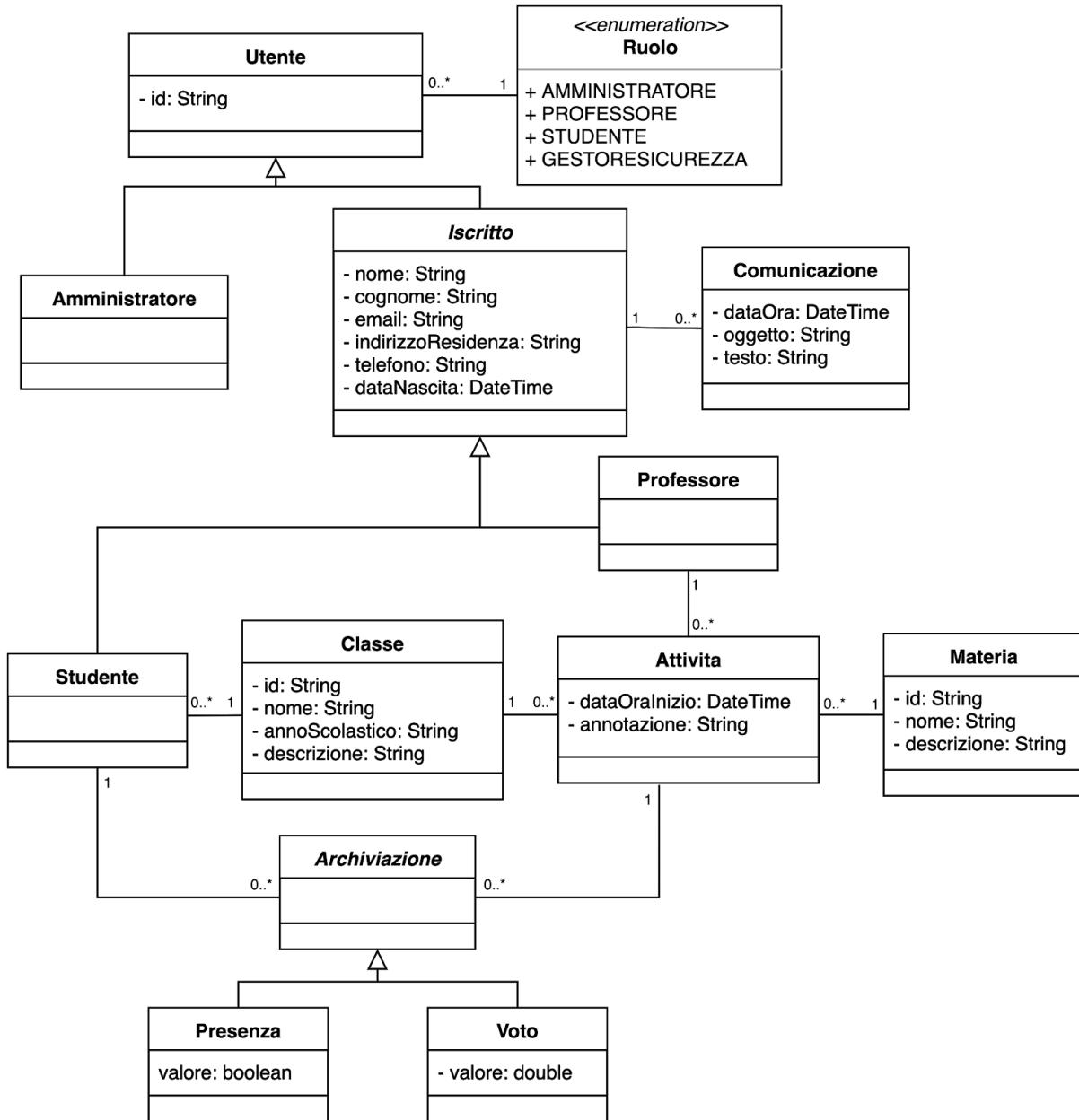
### GestioneStudente: Tabella Sotto-Funzionalità

Sotto-funzionalità	Sotto-funzionalità	Legame	Informazioni
Visualizzazione Presenze	Visualizzazione Calendario Studente	Uno Studente può visionare una Presenza (o Assenza) solo se gli è possibile visionare la relativa Attività Scolastica.	Identificativo Attività Scolastica
VisualizzazioneVoti	Visualizzazione Presenze	Uno Studente può visionare un Voto relativo ad un'Attività Scolastica solo se egli vi era presente.	Identificativo Attività Scolastica

## Creazione Modello del Dominio

Il concetto di **Ruolo** identifica il tipo di posizione che un certo attore, ed il suo relativo Utente registrato, ricopre all'interno dei servizi del Sistema.

1. **AMMINISTRATORE**: Ruolo annesso agli Amministratori
2. **PROFESSORE**: Ruolo annesso ai Professori
3. **STUDENTE**: Ruolo annesso agli Studenti
4. **GESTORESICUREZZA**: Ruolo annesso ad un Gestore della Sicurezza



Si introduce inoltre la nozione di Utente **Iscritto** inteso come entità astratta che accomuna gli utilizzatori del servizio, ossia Studenti e Professori. Di tale tipo di Utente vengono raccolti dati

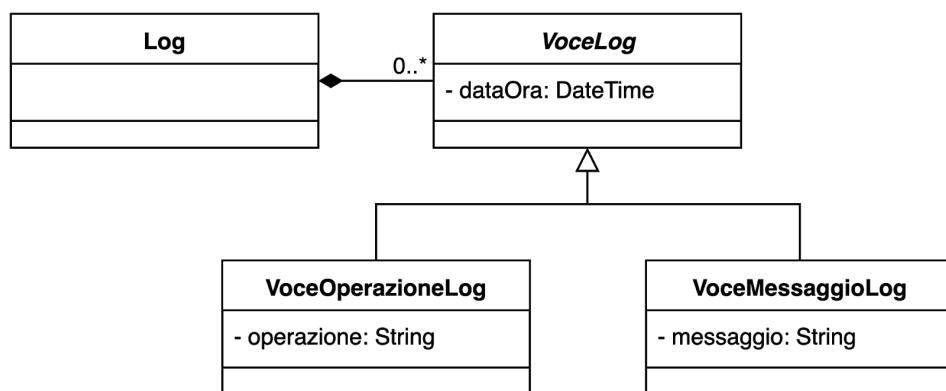
anagrafici, ed esso è l'unico possibile destinatario di una Comunicazione.

L'entità astratta **Archiviazione** accomuna invece tutti gli elementi del Sistema che sono relazionati ad un certo Studente in una determinata Attività, rendendo facilmente possibile l'aggiunta di componenti aggiuntive simili in sviluppi futuri.

La classe **Attività** modella tutti gli eventi relativi ad una determinata classe per i quali un certo Professore svolge il ruolo di referente, di cui lezioni, gite, verifiche, interrogazioni. Mediante la **Materia** e l'annotazione interna è possibile esprimere ciò che è stato svolto in passato o che si farà in futuro. Infine, ogni attività ha una durata costante di un'ora.

Non è stata modellata alcuna classe per il Calendario in quanto esso è stato riconosciuto come semplice collezione astratta di Attività, gestita e voluta al livello delle interfacce utente e quindi non propriamente parte delle entità del dominio.

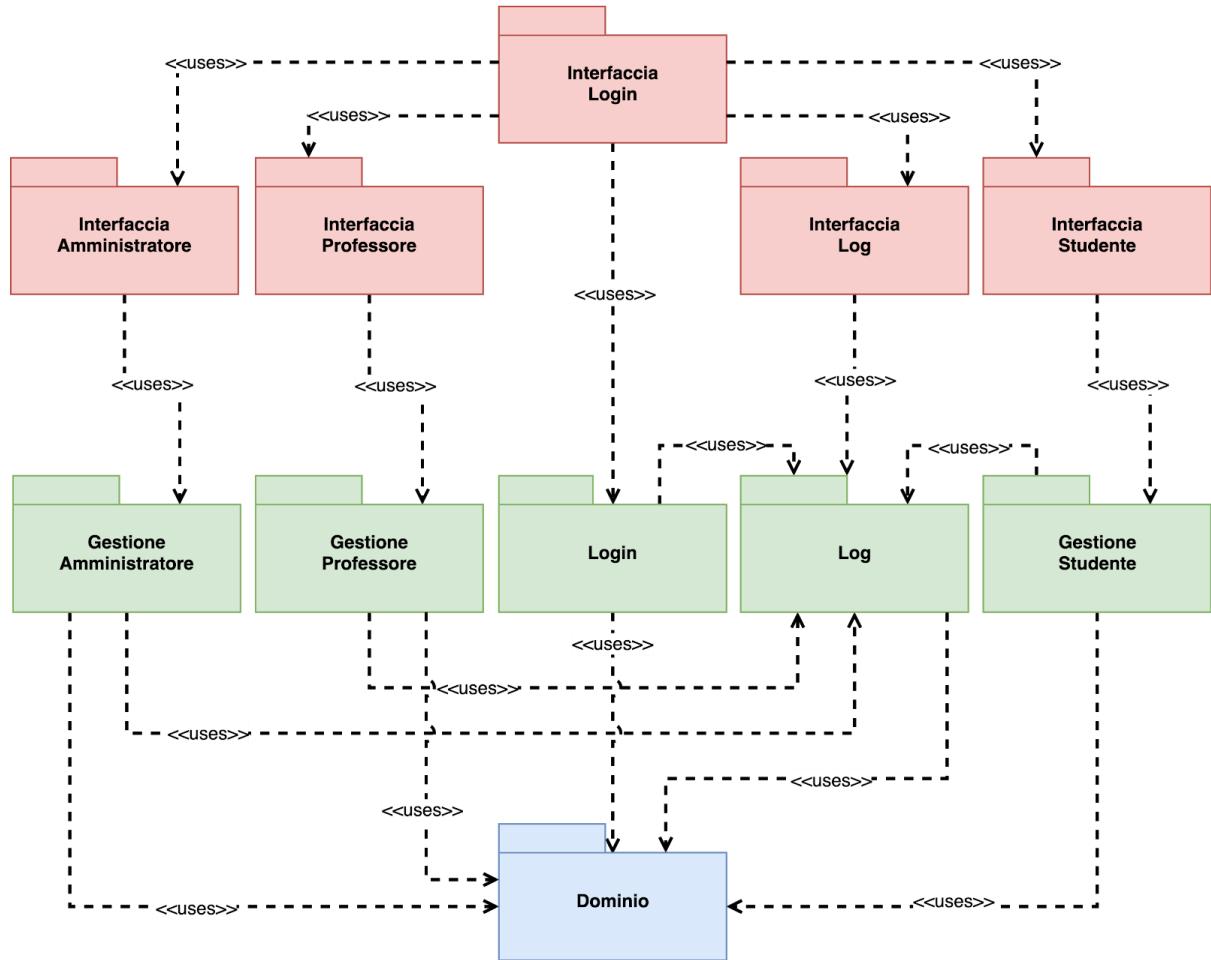
Le entità Studente, Classe, Attività, ed Archiviazione sono partecipi di un ciclo di relazioni, fenomeno che può risultare critico e compromettere la consistenza informativa del Sistema. Derivando direttamente dall'Analisi dei Requisiti, il modello viene considerato a questo livello sufficiente, tuttavia tale criticità dovrà essere opportunamente contenuta e controllata durante la fase di Progettazione.



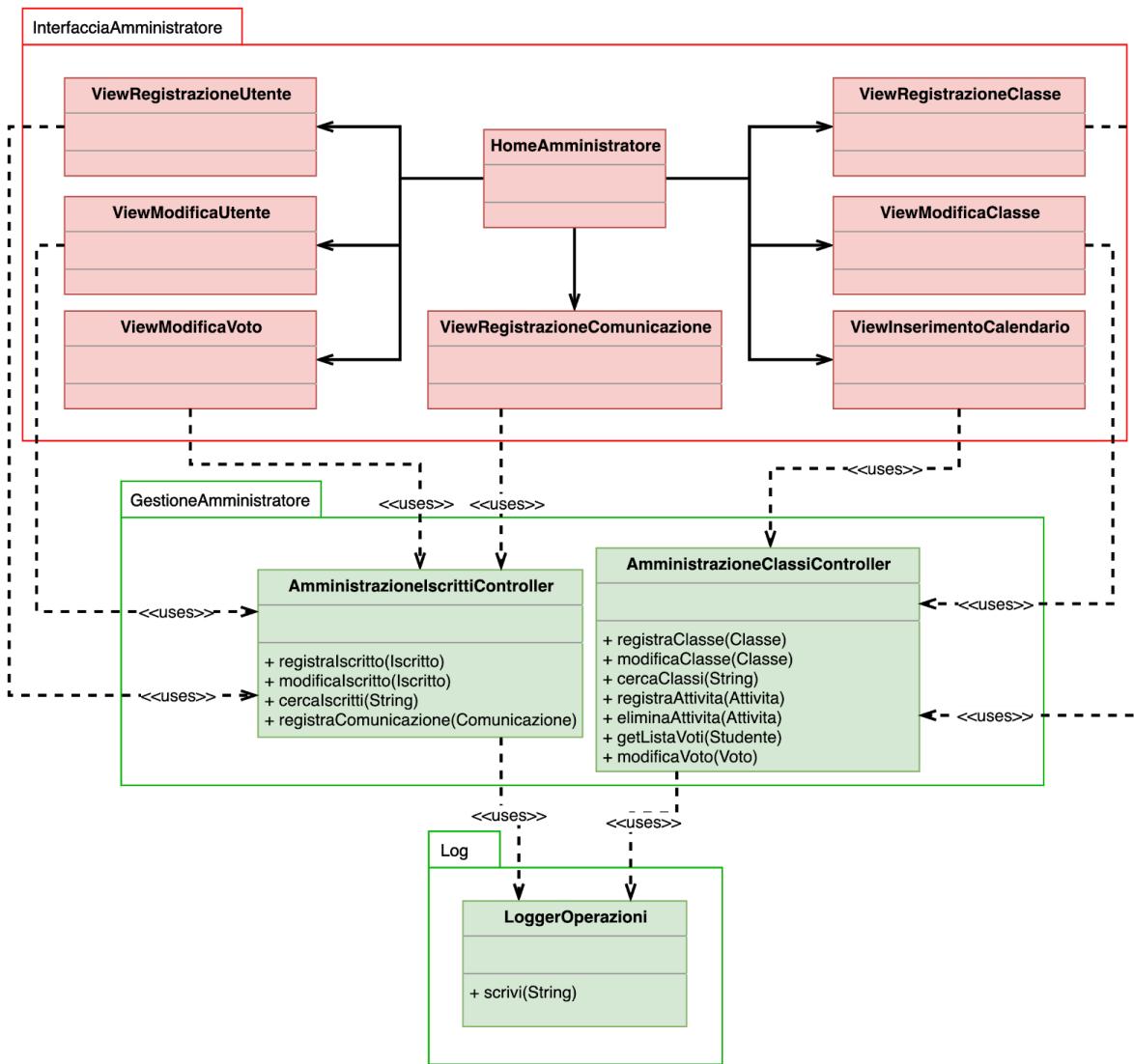
Infine, il soprastante diagramma delle classi rappresenta la parte del modello del dominio relativa alla gestione dei Log e delle voci contenute, differenziando le operazioni interne al servizio dai messaggi scambiati durante l'interazione con gli utilizzatori.

# Architettura Logica: Struttura

Diagramma dei package



## Diagramma delle classi: Amministratore



**AmministracionIscrittiController** è l'entità incaricata di gestire in maniera persistente gli Utenti del Sistema secondo i privilegi consentiti ad un Utente Amministratore, il quale è esclusivamente responsabile degli Iscritti.

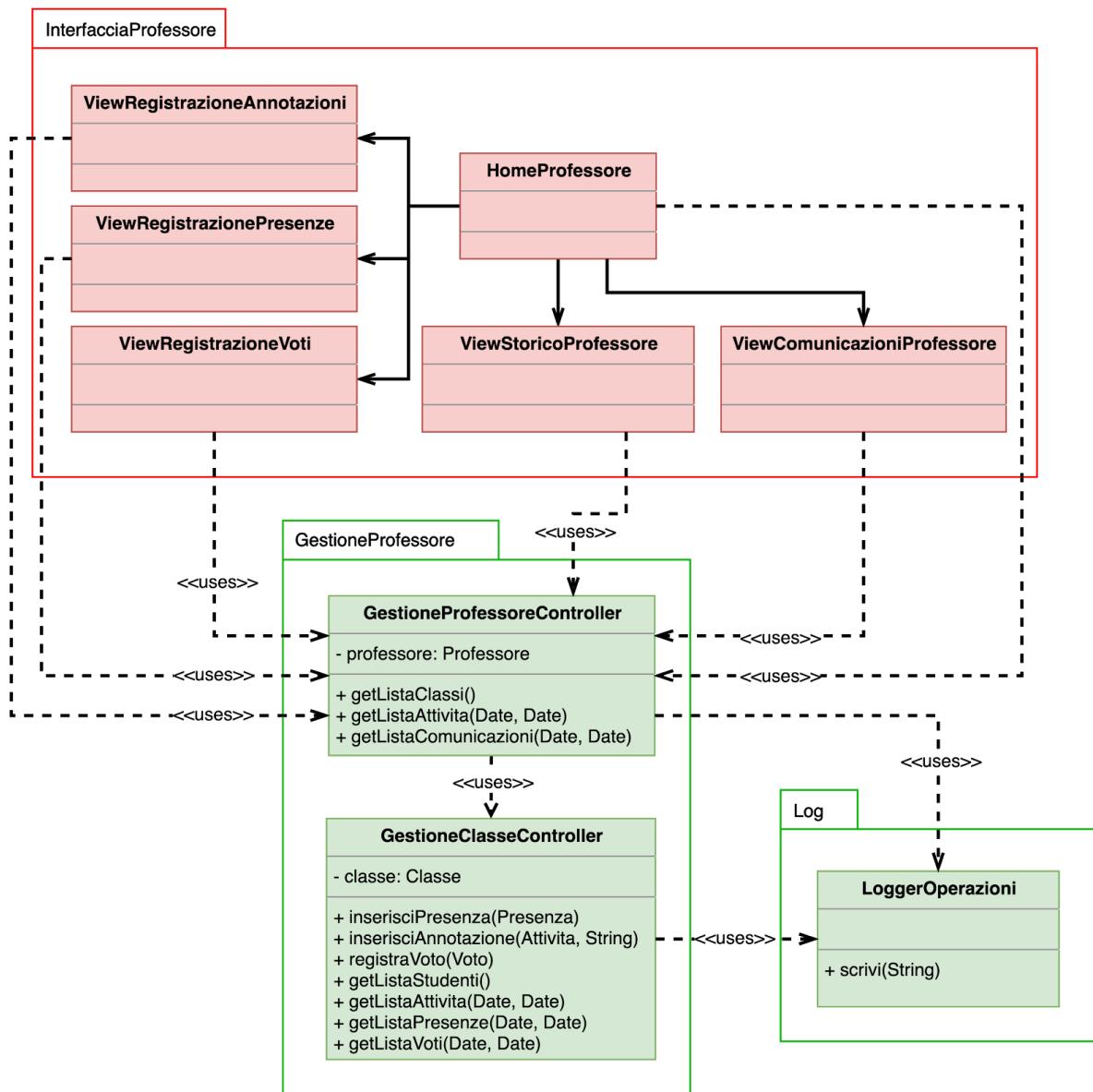
In maniera analoga, **AmministrazioneClassiController** è la componente mediante cui un Amministratore gestisce le informazioni persistenti relative alle Classi.

Ogni operazione interna deve essere registrata nel Log tramite l'entità **LoggerOperazioni**.

Nell'ambito della persistenza, alcuni termini qui introdotti godono di una semantica particolare atta a permettere la differenziazione di privilegi richiesta nei requisiti, e verranno opportunamente utilizzati in varie successive componenti del sistema.

- **Registrazione:** Aggiunta di un nuovo elemento persistente, solo se non già presente.
- **Modifica:** Modifica di un elemento persistente, solo se già presente.
- **Inserimento:** Aggiunta di un elemento persistente, sovrascrivendolo se già presente.

## Diagramma delle classi: Professore

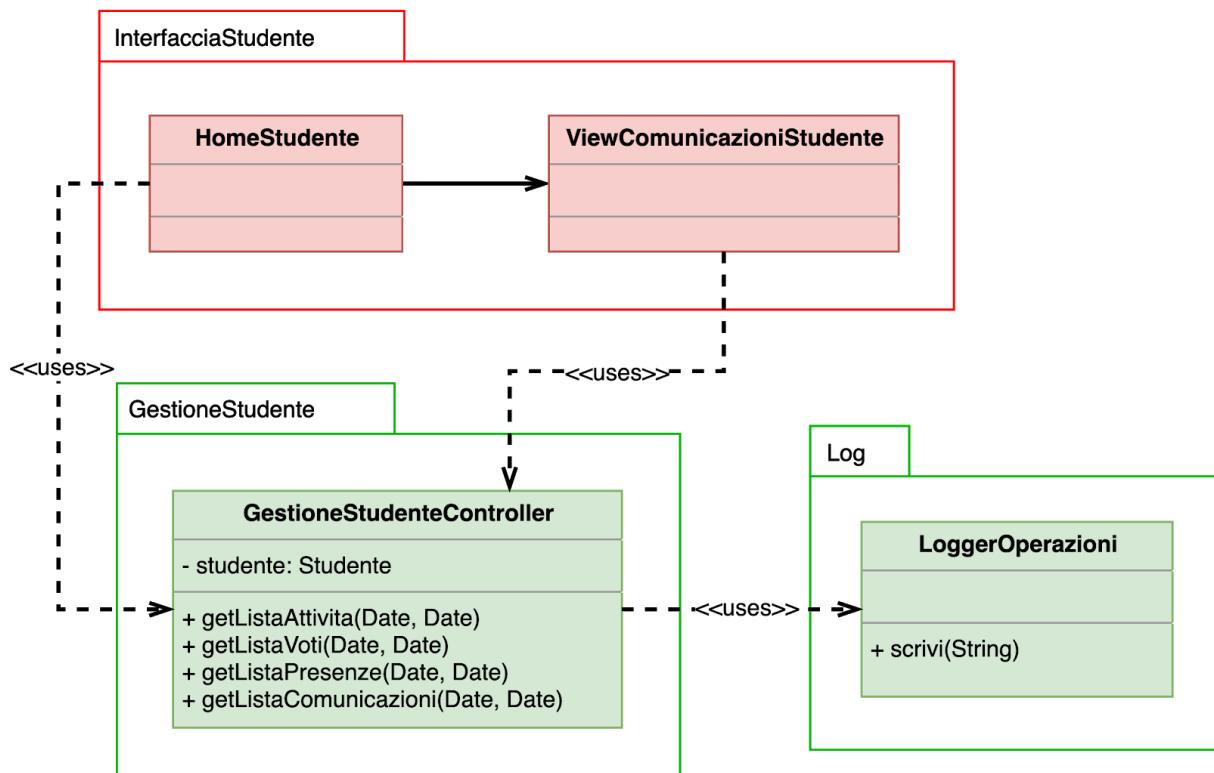


**GestioneProfessoreController** è l'entità che si occupa di gestire il profilo di un Professore, permettendogli di ottenere informazioni e dati che gli competono.

**GestioneClasseController** invece si occupa di gestire le informazioni relative ad una singola Classe, permettendo anche inserimento e registrazione di Presenze e Voti.

Ogni operazione interna deve essere registrata nel Log tramite l'entità **LoggerOperazioni**.

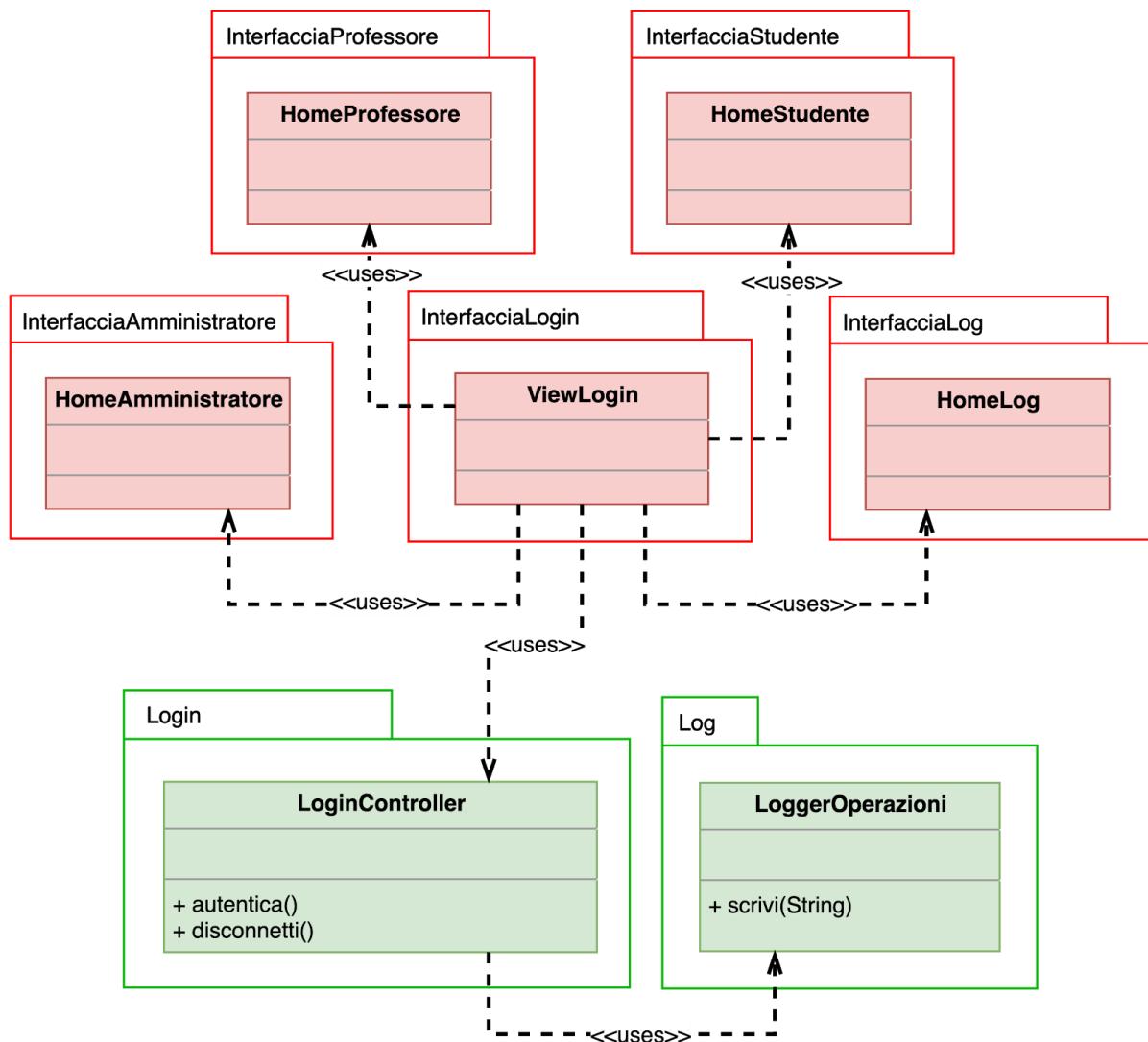
## Diagramma delle classi: Studente



**GestioneStudenteController** è l'entità che si occupa di gestire il profilo di uno Studente, permettendogli di ottenere informazioni e dati che gli competono.

Ogni operazione interna deve essere registrata nel Log tramite l'entità **LoggerOperazioni**.

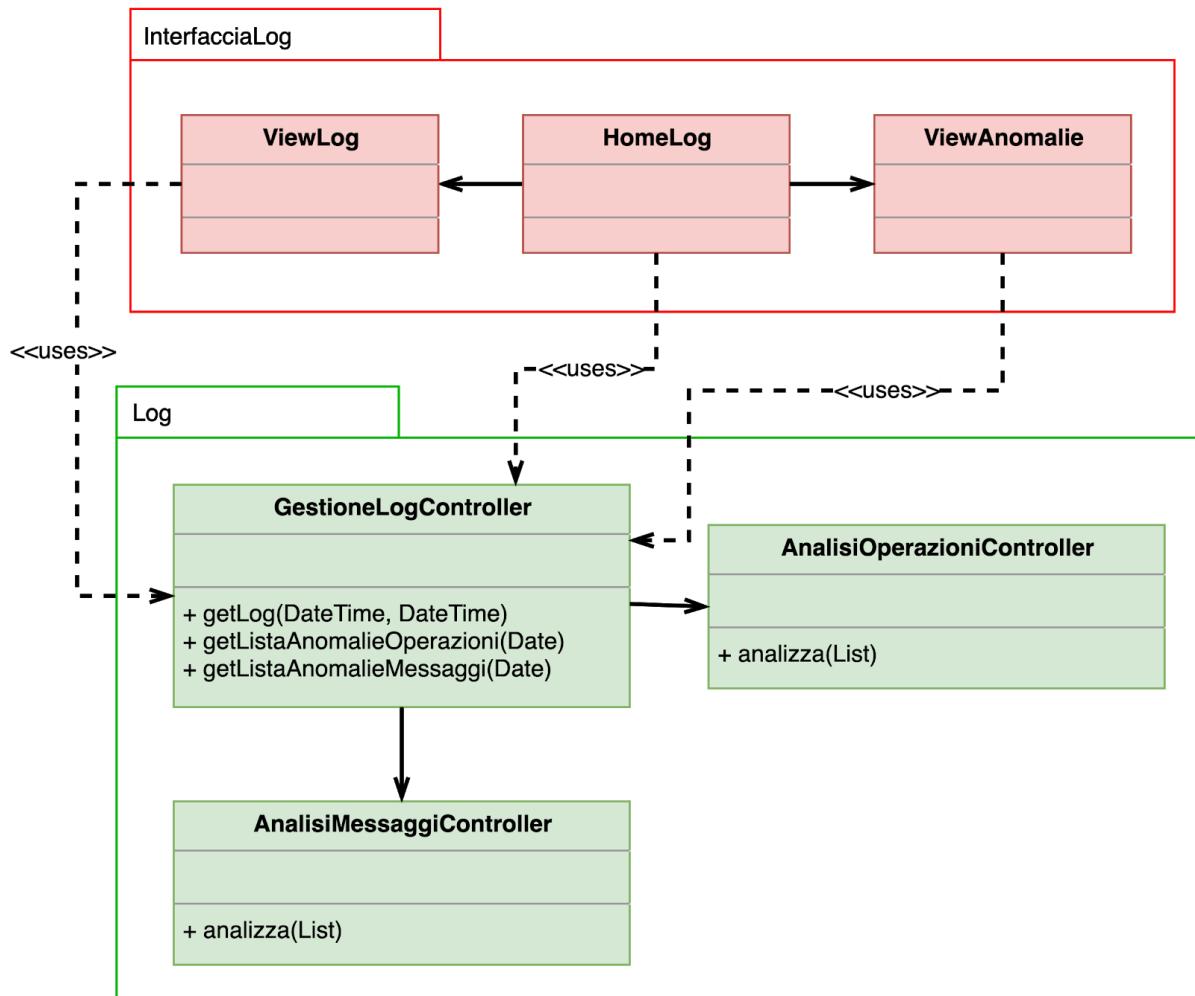
## Diagramma delle classi: Login



**LoginController** gestisce la sessione di un utente all'interno del sistema, verificando che le Credenziali fornite siano valide e mantenendo in memoria i dati dell'utilizzatore per tutta la durata della sua interazione con il Sistema.

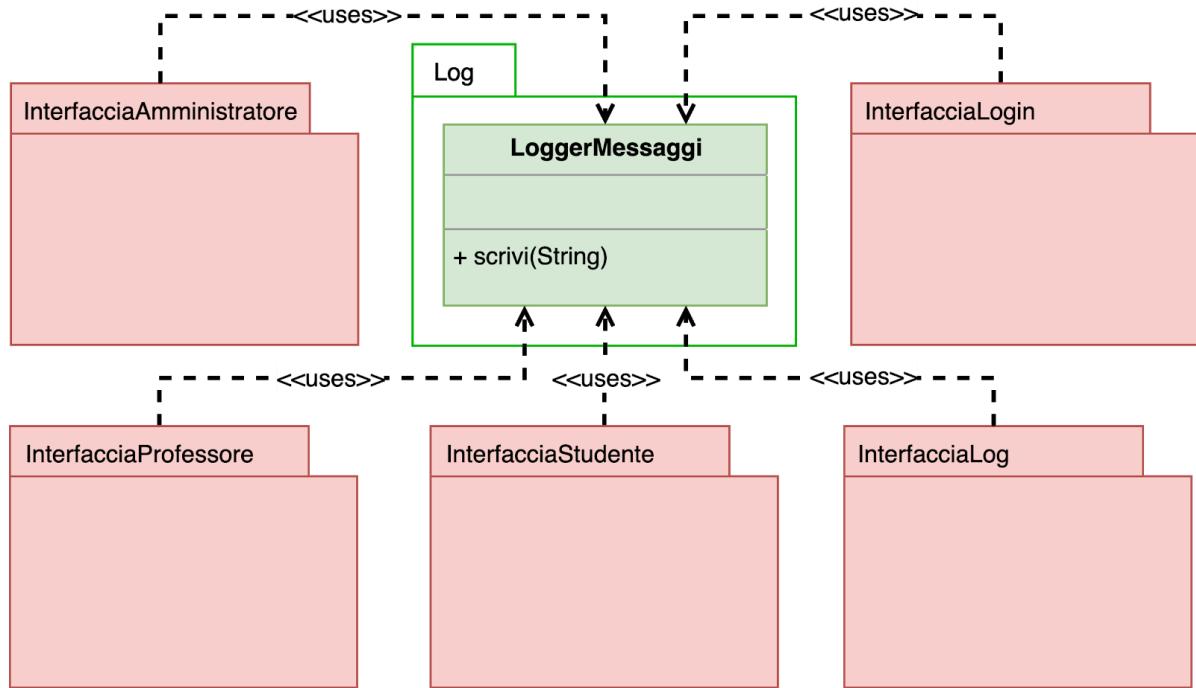
Mediante il sistema di mantenimento di sessione, l'interfaccia di autenticazione riconosce l'utilizzatore del Sistema e gli presenta la schermata a lui destinata in base al suo Ruolo. Ogni operazione interna deve essere registrata nel Log tramite l'entità **LoggerOperazioni**.

## Diagramma delle classi: Log



**GestioneLogController** è la componente del Sistema che gestisce l'interazione con i Log e permette di visionarne il contenuto.

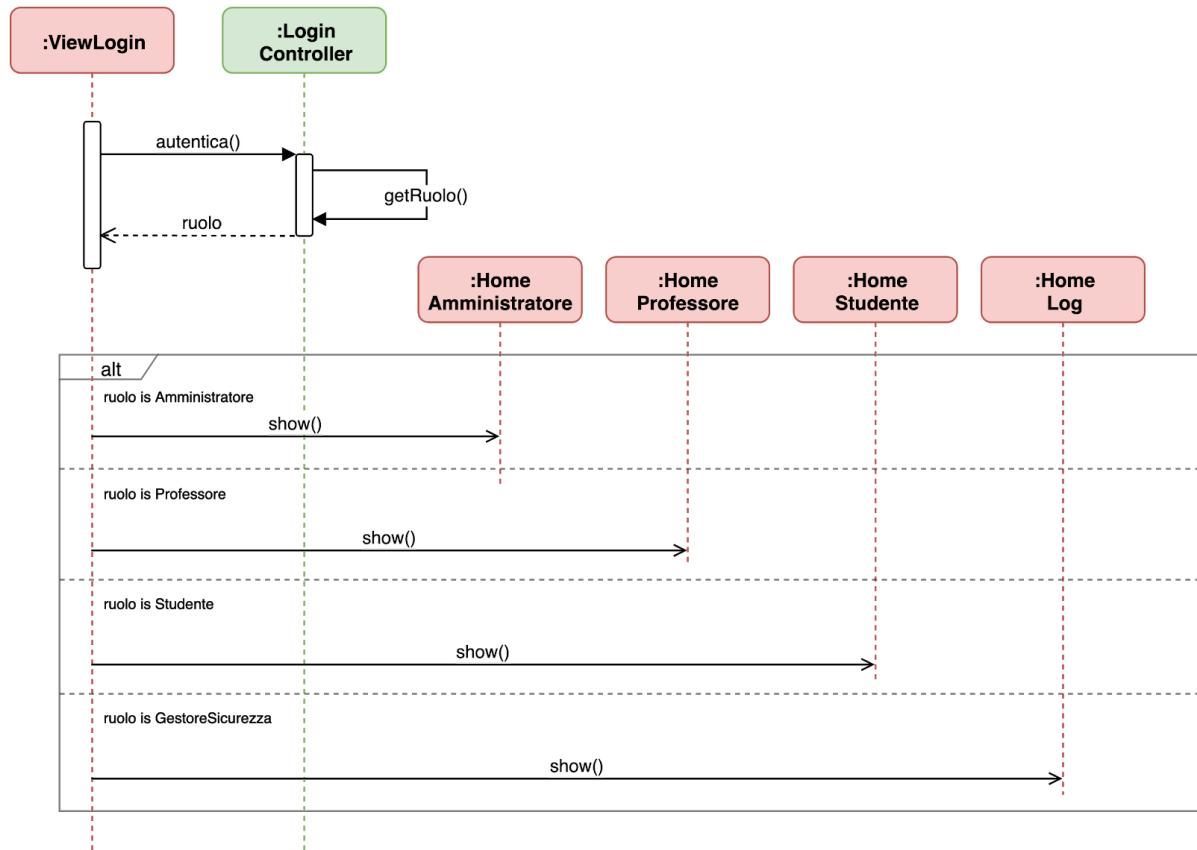
Mediante **AnalisiMessaggiController** e **AnalisiOperazioniController**, è possibile effettuare un'analisi sul contenuto dei Log al fine di riconoscere e notificare l'esistenza di anomalie nelle operazioni del Sistema o nei messaggi scambiati con gli Utenti.



Mediante l'entità **LoggerMessaggi**, ogni interfaccia registra tutti i messaggi scambiati con gli Utenti utilizzatori del Servizio.

# Architettura Logica: Interazione

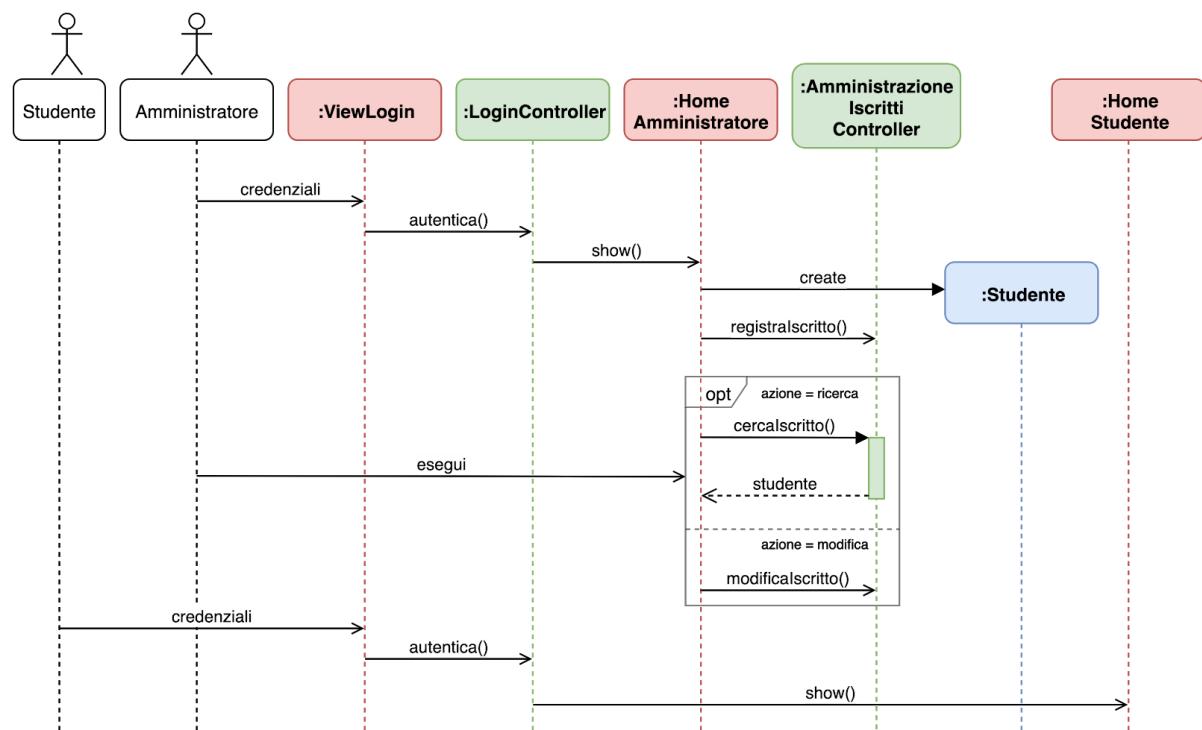
Diagramma di sequenza: Autenticazione avvenuta con successo



## Diagramma di sequenza: Registrazione Utente

Dal seguente diagramma di sequenza si evince la necessità di una relazione d'ordine tra la registrazione di un Utente nel Sistema da parte dell'Amministratore e la successiva autenticazione di un utilizzatore mediante le Credenziali dello stesso.

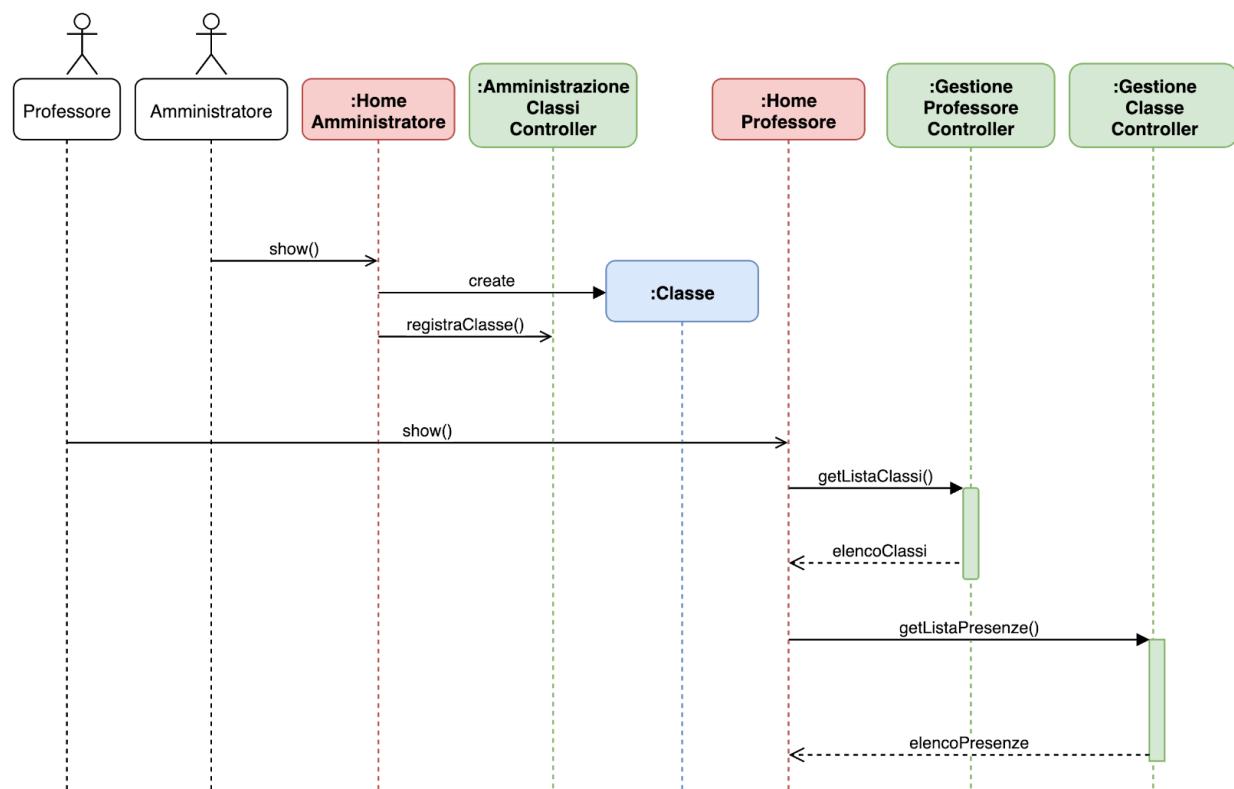
L'attore Studente è stato utilizzato a scopo esemplificativo, e l'esempio è del tutto analogo nel caso dei Professori.



## Diagramma di sequenza: Registrazione Classe

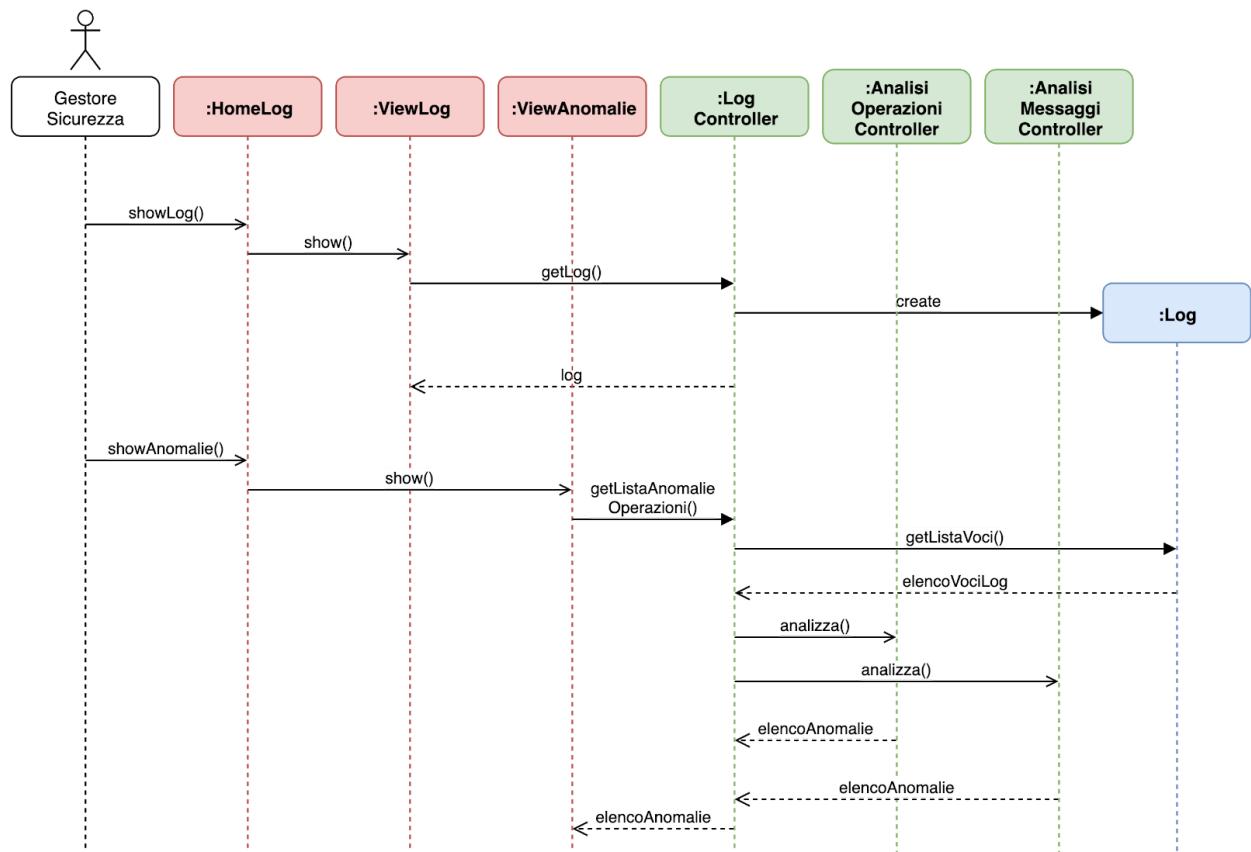
Dal seguente diagramma di sequenza si evince la necessità di una relazione d'ordine tra la registrazione di una nuova Classe nel Sistema da parte dell'Amministratore ed il successivo accesso alla stessa di un Professore o uno Studente.

Nel caso dello Studente l'esempio è pressoché analogo, tuttavia esso godrà di privilegi differenziati sull'accesso alle Classi.



## Diagramma di sequenza: Visualizzazione Log e Anomalie

Solamente un Gestore della Sicurezza gode della possibilità di accedere alle interfacce di visualizzazione dei Log.



## Architettura Logica: Comportamento

Studiando le strutture introdotte nell'Analisi del Problema, non si riconoscono entità che richiedano la presenza di un diagramma di stato.

Inoltre, i requisiti del Sistema non impongono comportamenti specifici in alcuna delle entità e classi introdotte.

## Piano del Lavoro

Lo sviluppo del progetto è stato suddiviso nelle varie componenti del team, secondo quanto indicato nella seguente tabella:

Package	Progetto	Sviluppo
Dominio	Al Sadi, Dellaluce, Baldini	Dellaluce, Al Sadi
GestioneAmministrazione	Al Sadi, Dellaluce, Baldini	Baldini, Dellaluce
GestioneProfessore	Al Sadi, Dellaluce, Baldini	Baldini, Al Sadi
GestioneStudente	Al Sadi, Dellaluce, Baldini	Baldini, Al Sadi
Login	Al Sadi, Dellaluce, Baldini	Dellaluce, Baldini
Log	Al Sadi, Dellaluce, Baldini	Dellaluce, Al Sadi
InterfacciaAmministratore	Al Sadi, Dellaluce, Baldini	Baldini, Dellaluce
InterfacciaProfessore	Al Sadi, Dellaluce, Baldini	Baldini, Al Sadi
InterfacciaStudente	Al Sadi, Dellaluce, Baldini	Baldini, Al Sadi
InterfacciaLogin	Al Sadi, Dellaluce, Baldini	Dellaluce, Baldini
InterfacciaLog	Al Sadi, Dellaluce, Baldini	Dellaluce, Al Sadi

## Funzionalità del prototipo

La prima versione del software, la quale sarà costituita da un semplice prototipo dimostrativo, comprenderà le seguenti funzionalità:

- Interfacce grafiche semplici e dimostrative
- Interfaccia e Gestione del sistema di Login
- Interfacce e Gestione di tutte le funzionalità destinate agli Studenti
- Interfacce e Gestione di tutte le funzionalità destinate ai Professori
- Interfacce e Gestione delle funzionalità di Registrazione Utenti e Registrazione Comunicazioni destinate agli Amministratori
- Interfacce e Gestione della funzionalità di visualizzazione del Log del Gestore della Sicurezza, senza gli algoritmi di analisi delle anomalie
- Persistenza basilare, che deve rispettare i canoni strutturali imposti dalla progettazione ma funzionare su tecnologie semplificative
- Non è prevista l'integrazione di sistemi crittografici o di protezione, ma il sistema deve essere predisposto per ospitarne realizzazioni future.

## Tempi di rilascio

I tempi di rilascio previsti sono i seguenti:

- Progettazione entro due settimane dalla data odierna.
- Sviluppo delle singole parti con annessi test unitari entro due settimane dalla fine della progettazione.
- Integrazione e collaudo del sistema entro una settimana dal termine dello sviluppo.

## Sviluppi futuri

Il committente ha richiesto un prodotto minimale e modulare, capace di ospitare facilmente possibili funzionalità aggiuntive e garantendo agevole sostituibilità delle componenti.

Si richiede perciò al team di progettazione di considerare tali richieste e di costruire un sistema flessibile agli aggiornamenti.

Tra le possibili aggiunte da attuare negli sviluppi futuri vi è per esempio la possibilità di registrare note disciplinari.

## Piano del Collaudo

Per garantire un corretto e consistente funzionamento del sistema ogni componente oggetto dello sviluppo deve risultare idonea ad una serie di test unitari di collaudo.

Per non caricare il documento, e a scopo dimostrativo, verranno successivamente elencati esempi di test su alcune classi del dominio.

```
[Test Fixture]
public class TestStudente
{
    private Studente _studente;

    [SetUp]
    public void StudenteSetUp() {
        _studente = new Studente("Amir", "Dellaluce",
                               "example@hotmail.it",
                               "Via mozart 11", "3338001900",
                               new DateTime(1997, 7, 10));
    }

    [TestGetter]
    public void TestMethod() {
        Assert.That(_studente.getNome(), Is.EqualTo("Amir"));
        Assert.That(_studente.getCognome(), Is.EqualTo("Dellaluce"));
        Assert.That(_studente.getEmail(), Is.EqualTo("example@hotmail.it"));
        Assert.That(_studente.getIndirizzoResidenza(), Is.EqualTo("Via mozart 11"));
        Assert.That(_studente.getTelefono(), Is.EqualTo("3338001900"));
        Assert.That(_studente.getDataNascita(), Is.EqualTo(new DateTime(1997, 7, 10)));
    }

    [TestSetter]
    public void TestMethod() {
        _studente.setNome("Mario");
        Assert.That(_studente.getNome(), Is.EqualTo("Mario"));
        _studente.setCognome("Rossi");
        Assert.That(_studente.getCognome(), Is.EqualTo("Rossi"));
        _studente.setEmail("example@gmail.it");
        Assert.That(_studente.getEmail(), Is.EqualTo("example@gmail.it"));
        _studente.setIndirizzoResidenza("Via Ca 10");
        Assert.That(_studente.getIndirizzoResidenza(), Is.EqualTo("Via Ca 10"));
        _studente.setTelefono("3384441789");
        Assert.That(_studente.getTelefono(), Is.EqualTo("3384441789"));
        _studente.setDataNascita(new DateTime(2000, 2, 6));
        Assert.That(_studente.getDataNascita(), Is.EqualTo(new DateTime(2000, 2, 6)));
    }
}
```

```

[Test Fixture]
public class TestProfessore
{
    private Professore _professore;

    [SetUp]
    public void ProfessoreSetUp() {
        _professore = new Professore("Federico", "Verdini",
            "federico.verdini@gmail.it",
            "Via Re 1", "3667189123",
            new DateTime(1976, 5, 1));
    }

    [TestGetter]
    public void TestMethod() {
        Assert.That(_professore.getNome(), Is.EqualTo("Federico"));
        Assert.That(_professore.getCognome(), Is.EqualTo("Verdini"));
        Assert.That(_professore.getEmail(), Is.EqualTo("federico.verdini@gmail.it"));
        Assert.That(_professore.getIndirizzoResidenza(), Is.EqualTo("Via Re 1"));
        Assert.That(_professore.getTelefono(), Is.EqualTo("3667189123"));
        Assert.That(_professore.getDataNascita(), Is.EqualTo(new DateTime(1976, 5, 1)));
    }

    [TestSetter]
    public void TestMethod() {
        _professore.setNome("Luca");
        Assert.That(_professore.getNome(), Is.EqualTo("Luca"));
        _professore.setCognome("Verdi");
        Assert.That(_professore.getCognome(), Is.EqualTo("Verdi"));
        _professore.setEmail("luca.verdi@gmail.it");
        Assert.That(_professore.getEmail(), Is.EqualTo("luca.verdi@gmail.it"));
        _professore.setIndirizzoResidenza("via torre 36");
        Assert.That(_professore.getIndirizzoResidenza(), Is.EqualTo("via torre 36"));
        _professore.setTelefono("3384469837");
        Assert.That(_professore.getTelefono(), Is.EqualTo("3384469837"));
        _professore.setDataNascita(new DateTime(2000, 2, 6));
        Assert.That(_professore.getDataNascita(),
            Is.EqualTo(new DateTime(1985, 9, 12)));
    }
}

```

# Progettazione

## Progettazione architetturale

### Requisiti non funzionali

Durante la fase di Analisi del Problema sono emersi dai requisiti non funzionali i seguenti vincoli al sistema:

- Disponibilità e tempo di risposta
- Protezione dei dati
- Usabilità

Il vincolo sulla protezione dei dati inficia sul tempo di risposta, poiché i vari meccanismi di cifratura adottati per aumentare la sicurezza del sistema possono considerevolmente impattare sulle prestazioni complessive; al fine di mantenere un buon livello di sicurezza e dei tempi di risposta che non influenzino l'usabilità si dovrà trovare il corretto equilibrio di progettazione. Ciononostante è doveroso sottolineare l'importanza del vincolo sulla protezione dei dati in quanto, come si evince dalla tabella della valutazione dei beni, i dati in possesso del sistema hanno un elevato valore informativo. Perciò è corretto concludere che un adeguato livello di protezione è da considerarsi preferibile rispetto ad un tempo di risposta massimo.

La disponibilità può anche essere pregiudicata da una scarsa flessibilità del software alla sostituzione di componenti difettose o obsolete, perciò è richiesto l'introduzione di diversi meccanismi di astrazione e separazione in fase di progettazione.

L'usabilità non interferisce con i vincoli sopra citati, essa infatti avrà un impatto solo sulle interfacce grafiche, le quali dovranno permettere un utilizzo intuitivo e un rapido ritrovamento delle informazioni cercate.

### Scelte tecnologiche

La natura distribuita del dominio applicativo ed il desiderio di accedere al servizio da più dispositivi diversi ci hanno motivati a progettare il sistema come una Web Application. Per via della richiesta di costi ridotti indichiamo inoltre come tecnologia di sviluppo il linguaggio Java nella sua Enterprise Edition. Infatti, un artefatto sviluppato in Java può facilmente essere installato in maniera intercambiabile su una qualsiasi macchina Server, senza vincolare gli enti scolastici a sistemi operativi specifici o a licenze costose.

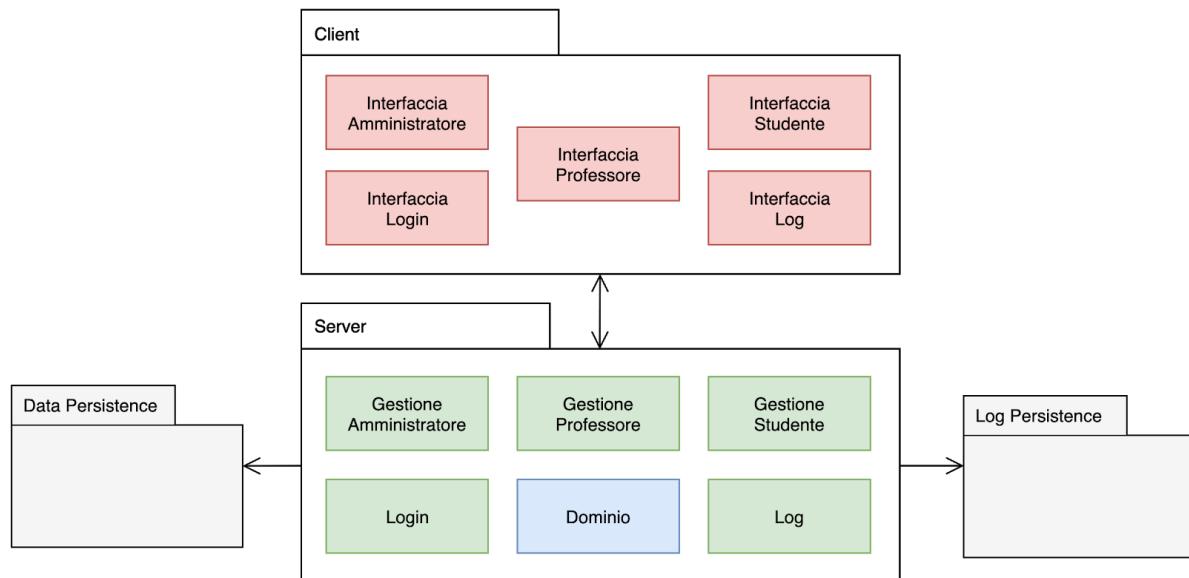
Per le stesse ragioni la scelta del Web Container di preferenza è ricaduta su Apache Tomcat, il quale è open-source e distribuito con una licenza libera che permette l'uso commerciale.

### Scelta dell'architettura

La struttura più idonea per progettare il sistema è quella di un'architettura Client-Server a tre livelli. Il pattern architettonico scelto come riferimento è il Model-View-Controller. Dove possibile, è stata dedicata attenzione al rispetto dei Design-Principles Single Responsibility, Open/Closed, Dependency Inversion, Design By Contract. La struttura dei package è,

all'infuori della nomenclatura, analoga a quella presentata in fase di analisi la quale rispecchia gli Acyclic Dependencies Principle e Stable Dependencies Principle.

Nel diagramma sottostante viene riportata la struttura logica del sistema:



## Cliente

Nell'ottica di una struttura a Web Application, i clienti sono sviluppati come **thin-client**.

Di fatto, essi sono costituiti da un semplice livello di presentazione della pagina Web fornita dinamicamente dal servitore in seguito ad una richiesta HTTP. La comunicazione tra clienti e servitori avviene su un canale TLS, in modo da garantire privacy e sicurezza durante lo scambio di messaggi.

Il progetto non si occupa perciò di strutturare la natura dei clienti, i quali possono essere costituiti da un qualsiasi Web Browser disponibile sul mercato. Tuttavia l'applicazione deve tenere conto del tipo di dispositivo dal quale il cliente richiede il servizio, in modo da presentare interfacce adeguate e ottimizzate per browser su desktop, smartphone, o tablet.

## Servitore

Il progetto prevede lo sviluppo di un unico applicativo Server dedicato alla logica di business di tutti i servizi del sistema. Esso deve tenere traccia dello stato di sessione delle conversazioni con i clienti, gestire la loro autenticazione, costruire e presentare interfacce grafiche dinamiche, comunicare con il livello di persistenza per lo scambio di dati.

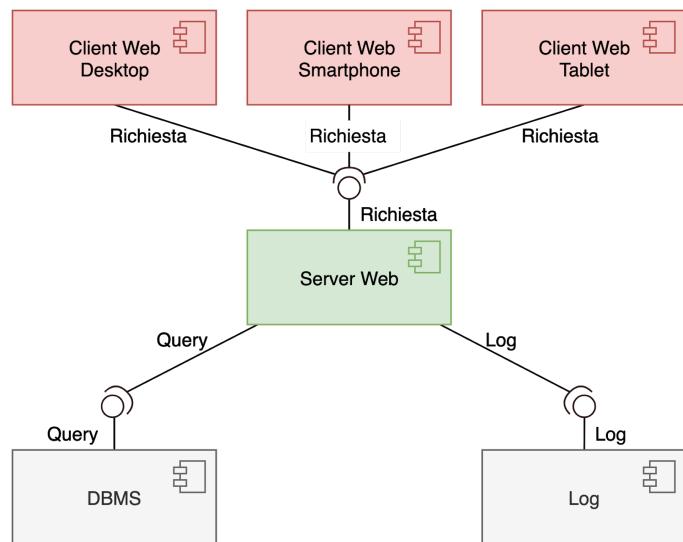
Un Server Web è di sua natura facilmente scalabile per via delle interazioni stateless previste dal protocollo HTTP, perciò deve essere prevista la possibilità di replicare varie istanze di servitori su macchine diverse per aumentare la disponibilità del servizio in caso di esigenza. In tal caso, andranno adottati opportuni sistemi di coordinamento per mantenere lo stato delle conversazioni con i clienti in maniera distribuita.

## Persistenza

Per la gestione della persistenza è previsto un server dedicato nel quale sarà installato un DBMS in grado di gestire la base di dati del sistema scolastico. Tale scelta è stata motivata dal considerevole volume di dati gestiti dal sistema, i quali necessitano di organizzazione ed efficiente gestione.

Per la scrittura dei Log è stato invece scelto di utilizzare alcuni file locali sulle macchine in cui le diverse istanze dei Server Web sono installate. In tal modo è possibile favorire la velocità di scrittura, la quale non deve impattare sulle prestazioni del sistema, ed al contempo distribuire l'occupazione di memoria dei Log suddividendoli su diverse macchine responsabili.

Nella figura sottostante è riportata l'Architettura del Sistema organizzata attraverso un diagramma dei componenti:



## Considerazioni relative alle tecnologie utilizzate

La suddivisione delle componenti dell'applicazione su più livelli permette una più efficiente gestione del carico lavorativo e una migliore manutenibilità. La possibilità di replicare i servitori e di gestire la persistenza dei dati in macchine esterne rappresenta infatti parzialmente un'applicazione del **Dependency Inversion Principle**, in quanto le componenti dipendono le une dalle altre mediante l'interazione con interfacce.

La scelta del linguaggio Java e di Apache TomCat come Web Container direzionano il modo in cui il livello di presentazione e lo stato delle interazioni con l'utente sono gestite. Infatti le interfacce utente saranno presumibilmente realizzate con le tecnologie Servlet e Java Server Pages, e si dovrà tenere conto della riutilizzabilità concorrente di uno stesso servitore per più clienti diversi. Inoltre, l'interazione con i DBMS Server dovrà avvenire mediante l'utilizzo della libreria JDBC.

Infine, la sicurezza è favorita sotto numerosi aspetti. In primo luogo la suddivisione architetturale in più livelli permette un maggior controllo sulle interazioni con i clienti e le

informazioni con essi scambiate. Inoltre l'architettura della Web Application deresponsabilizza i clienti da ogni operazione della logica di business, munendoli di un semplice livello di presentazione e gestendo i dati sensibili esclusivamente nei Servitori. La possibilità di instaurare connessioni mediante protocollo TLS permette oltretutto un alto grado di protezione nella trasmissione delle informazioni con i clienti, mitigando diverse vulnerabilità emerse in fase di analisi. Infine, l'architettura nasconde agli utilizzatori esterni posizione e struttura di tutti i livelli di persistenza.

Analizzando attentamente quanto appena discusso, si può dedurre che i rischi rimasti sui quali porre particolare attenzione durante la progettazione sono sinteticamente i seguenti:

Rischio	Descrizione	Possibili soluzioni
Accesso al contenuto del database ed esposizione delle informazioni	Se le misure di protezione dei DBMS dovessero per qualunque ragione risultare insufficienti, un attaccante potrebbe accedere al contenuto delle basi di dati e leggerne le informazioni.	Una possibile soluzione consisterebbe nella cifratura dei dati inseriti nella base di dati. Nel caso la cifratura della totalità degli elementi risultasse troppo costosa, si potrebbe considerare di proteggere almeno i dati più sensibili.
Accesso al contenuto dei Log ed esposizione delle informazioni	Essendo i Log scritti e salvati su dei file nelle macchine dei Servitori, alcuni attacchi fortemente mirati potrebbero portare all'accesso nel sistema in cui il servizio è installato e alla conseguente lettura delle voci registrate.	Una possibile soluzione consisterebbe nella cifratura delle voci del Log. Risulta necessario assicurarsi che la tecnologia di protezione scelta non impatti le performance, perciò sarebbe in ogni caso necessario studiare un compromesso di progettazione.
Manomissione di un certificato pubblico per il protocollo TLS	Un attaccante potrebbe intromettersi nella fase iniziale della comunicazione e sostituire il certificato pubblico del servitore con il proprio, rendendo possibile un attacco Man in the Middle.	La soluzione migliore è quella di basarsi su un ente terzo fidato da cui ottenere certificati sempre validi. In aggiunta, potrebbe essere strategico mantenere nei clienti delle copie dei certificati raccolte durante le interazioni con il servizio, in modo da rendere possibili controlli incrociati all'inizio di nuove conversazioni.
Mancanza di controlli nelle implementazioni dei servitori	Nel caso in cui l'implementazione trascurasse i controlli sugli input dei servitori o sulla modulazione dei privilegi, la consistenza del sistema potrebbe essere messa a rischio.	Risulta fondamentale controllare sempre ogni input delle richieste effettuate dai clienti, e generalmente orientare l'implementazione verso la sicurezza.

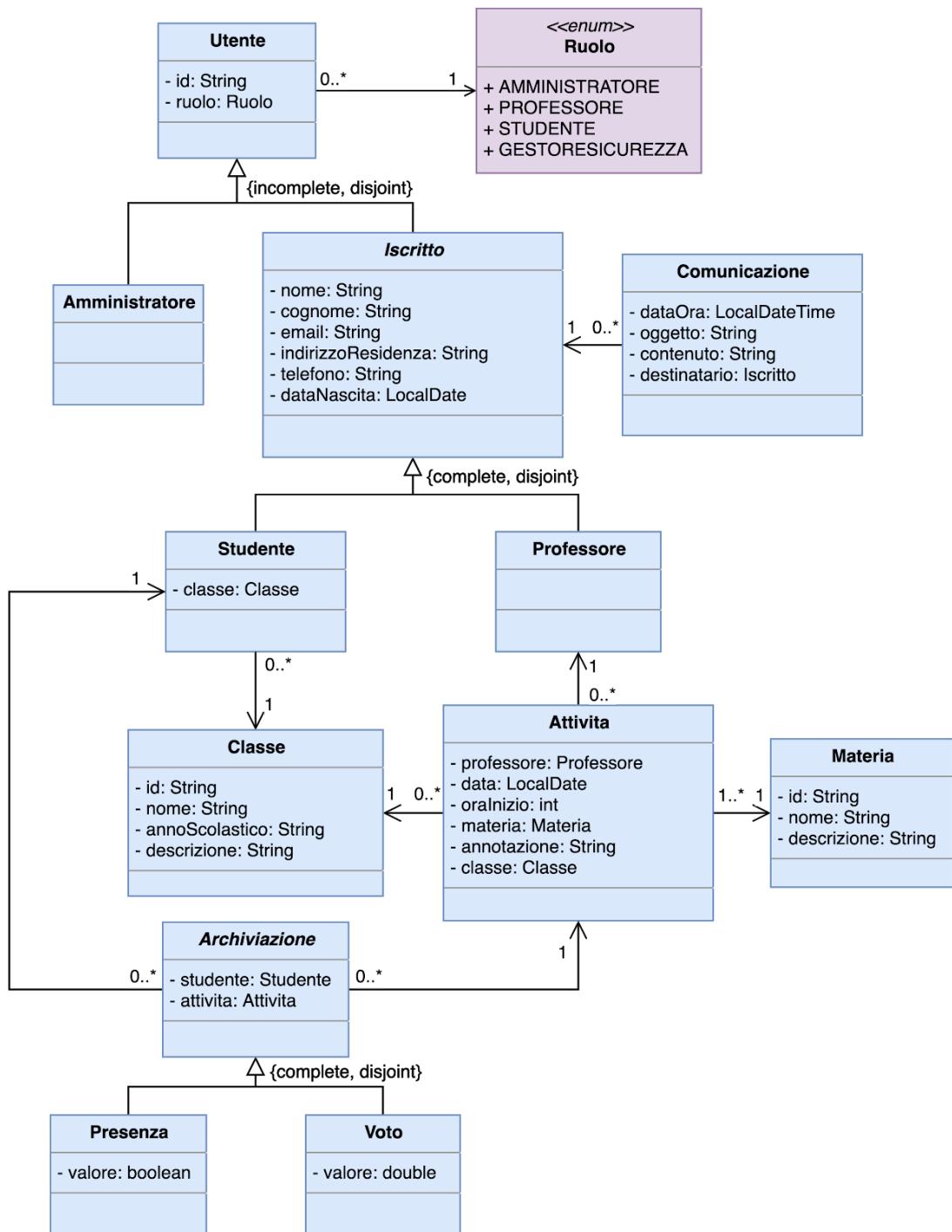
# Progettazione di Dettaglio

## Struttura

Package mitro.model

In questo package sono raccolte tutte le classi relative al dominio dell'applicazione.

Il diagramma delle classi non risulta particolarmente distante da quello evinto in fase di analisi.



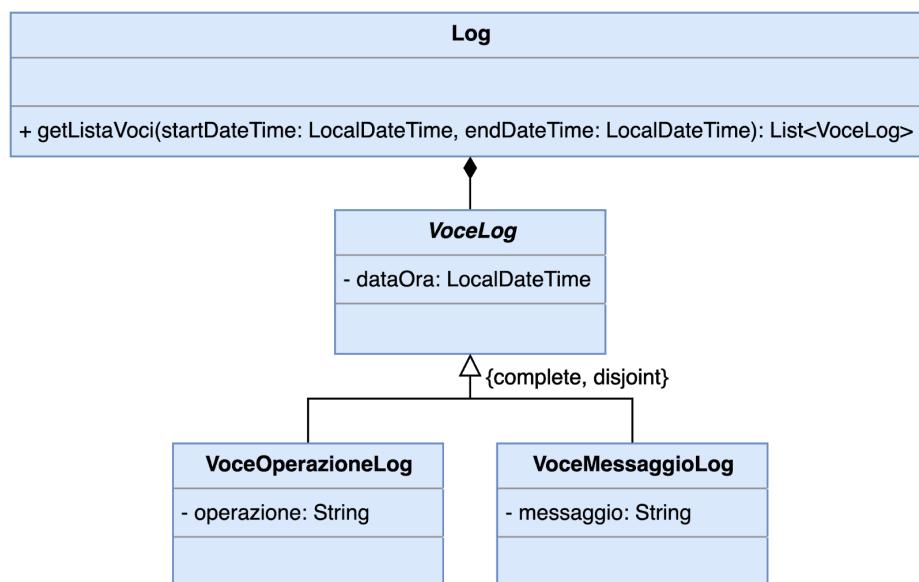
È possibile notare un cambiamento nella classe Attivita. Infatti l'ora di inizio delle attività scolastiche è stata codificata come un numero intero, in quanto convenzionalmente lo svolgimento delle stesse avviene esclusivamente allo scoccare di una nuova fascia oraria.

L'enumerativo Ruolo è ora un attributo della classe Utente, ed è fondamentale per caratterizzare la posizione che un certo cliente ricopre all'interno del sistema in fase di autenticazione e di mantenimento della sessione. Esiste la possibilità di utilizzare classi Utente dotate semplicemente di un Ruolo in modo da autenticare clienti che non sono rappresentati da una classe del dominio dedicata, come nel caso del Gestore della Sicurezza. Classi utente più specifiche, come Studente e Professore, devono avere un Ruolo costante coerente alla loro posizione nel sistema, e non permetterne la modifica dall'esterno.

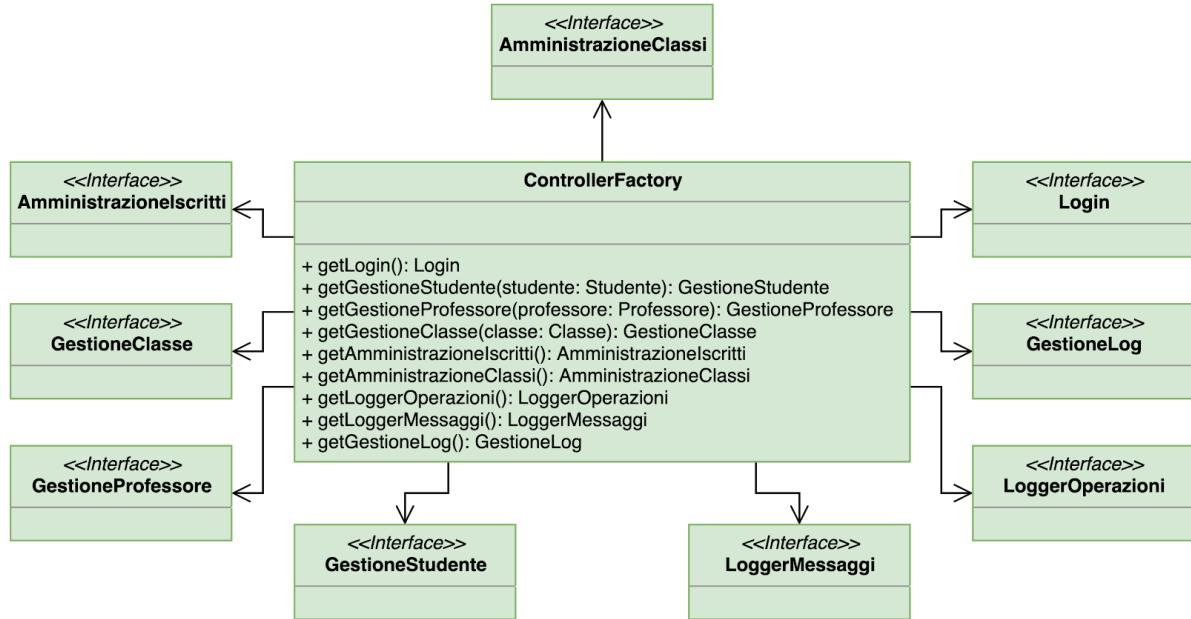
Come già reso noto durante l'analisi, le entità Archiviazione, Studente, Classe, e Attività presentano un ciclo di relazioni. Il problema è stato arginato manipolando la navigabilità tra le classi e portando Archiviazione ad essere l'unico componente capace di portare inconsistenze informative. Infatti, il Sistema deve assicurarsi che un'Archiviazione si riferisca ad uno Studente se e solo se questo è partecipe della Classe in cui l'Attività annessa deve svolgersi. Successivamente alla creazione dell'Archiviazione lo Studente potrebbe essere riassegnato a Classi diverse, per esempio all'inizio di un nuovo anno scolastico. Tale cambiamento non è da considerarsi inconsistente, e porterebbe le varie Archiviazioni precedentemente registrate ad essere considerate come uno "storico" della partecipazione di un determinato Studente nelle Classi degli anni passati.

Le classi del dominio sono intenzionalmente pensate come Java Bean, al fine di favorirne la persistenza basata sulla libreria JDBC.

Infine, il sottostante diagramma delle classi rappresenta il modello del dominio del Log.



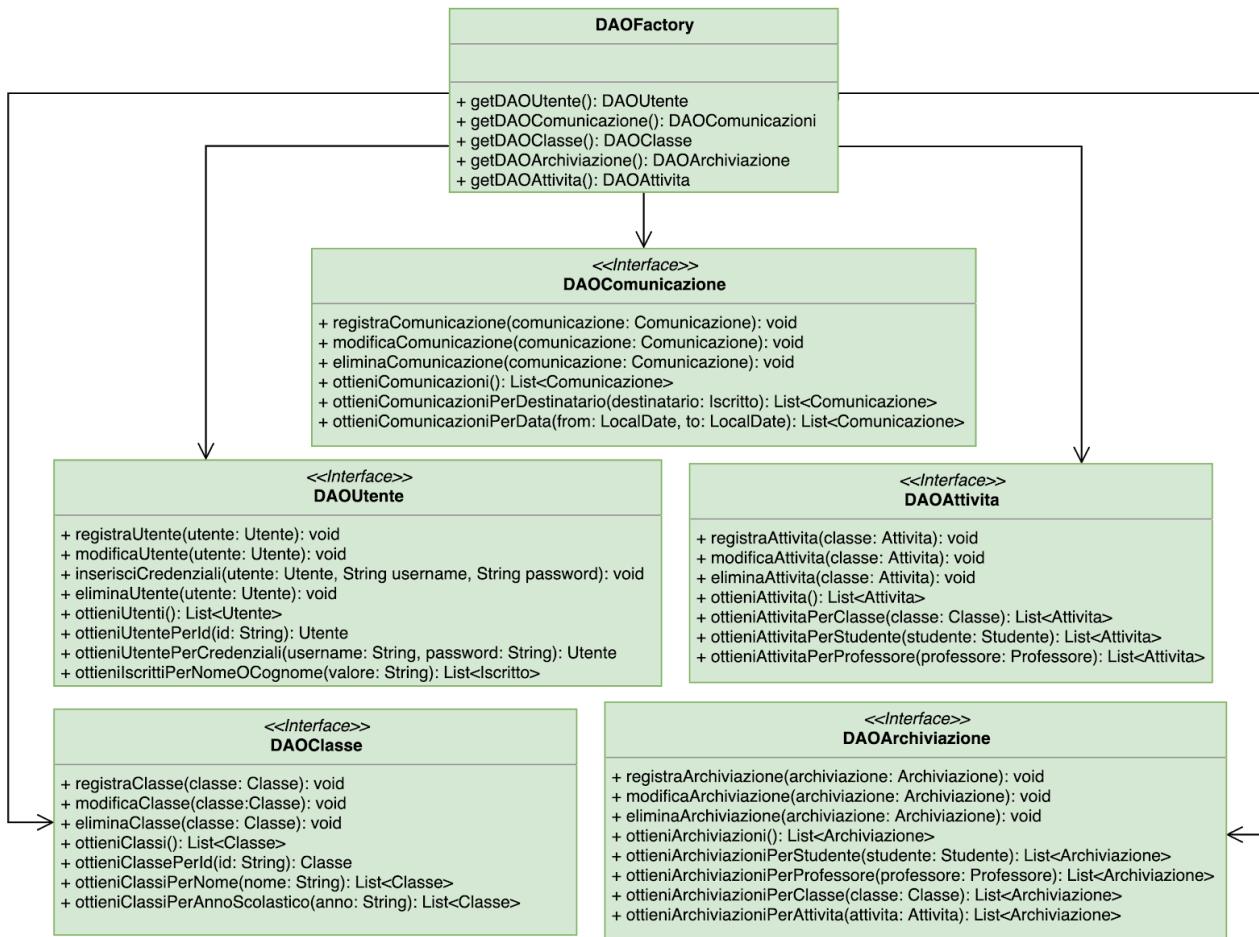
## Package mitro.controller



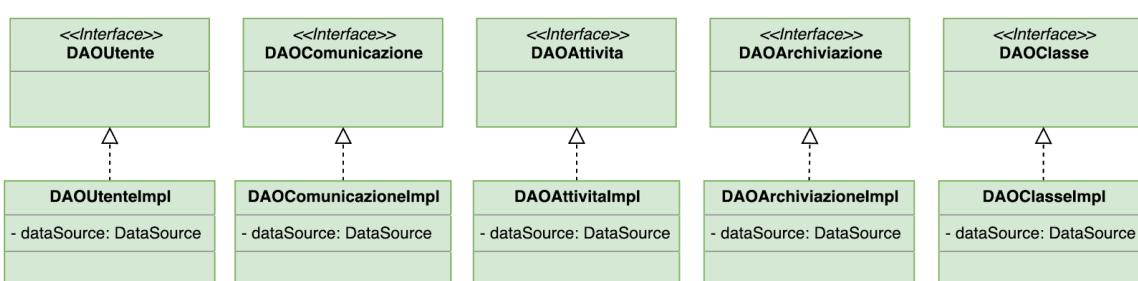
Ogni controllore della logica di business è stato definito sotto forma di interfaccia, in modo da applicare **The Dependency Inversion Principle**. Infatti in ogni costrutto sono esposti solamente i metodi fondamentali che ne caratterizzano la semantica ed il comportamento. In tal modo è possibile ospitare nel sistema implementazioni multiple, ridurre le dipendenze interne tra le componenti, e favorire l'elasticità del sistema alla sostituzione di realizzazioni date. Infine, la suddivisione in interfacce favorisce il collaudo delle varie componenti del sistema.

**ControllerFactory** è la classe incaricata della costruzione, configurazione, e gestione di tutti i controllori del sistema nelle loro varie implementazioni, applicando il design-pattern **Factory**. Risulta perciò opportuno che tutte le classi esterne al package ed interessate ad accedere ai controllori utilizzino `ControllerFactory`.

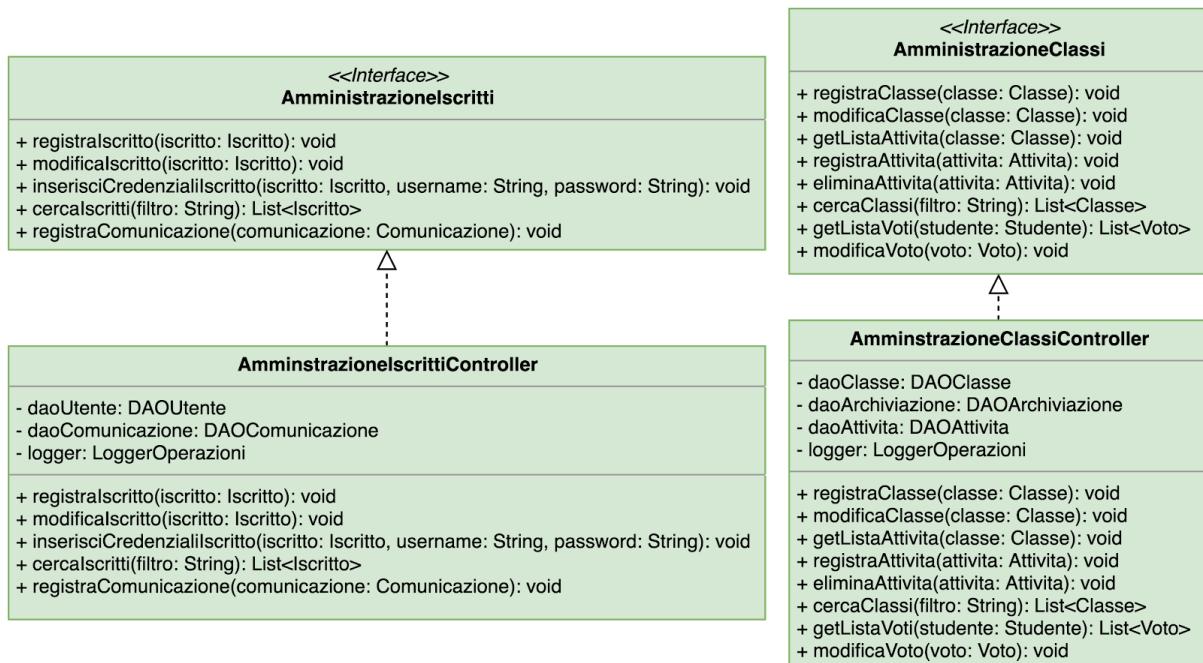
## Package mitro.persistenza



In modo da favorire una corretta separazione tra la logica di business ed il sistema di persistenza dei dati, è stato deciso di applicare il pattern **Data Access Object (DAO)**. Ogni classe fondamentale ai fini della persistenza è quindi gestita da un'opportuna interfaccia che espone i metodi per la creazione, rimozione, aggiornamento, e ricerca nella base di dati. Ogni interfaccia esprime anche tutti i metodi di ricerca che potrebbero godere di importanti ottimizzazioni in base alla realizzazione, cercando di prevedere anche le necessità che potrebbero insorgere negli sviluppi futuri. La separazione concettuale fornita dalle interfacce favorisce nuovamente la sostituibilità delle componenti, il collaudo, e non obbliga la logica di business ad una tecnologia di persistenza prefissata. Nuovamente, il pattern Factory è stato applicato per nascondere esternamente le implementazioni. **DAOAttività** è anche incaricato della persistenza delle Materie, in quanto esse sono strettamente legate alle Attività.



## Package mitro.controller.amministratore



Questo package si occupa della gestione dei controllori relativi alle azioni consentite ad un cliente Amministratore. Per effettuare tutte le operazioni relative alla persistenza delle classi del dominio, ogni realizzazione delle interfacce controller deve basarsi sugli oggetti DAO esposti in precedenza. Allo stesso modo, la scrittura delle voci del Log relativi alle operazioni di sistema deve avvenire mediante l'utilizzo dell'interfaccia LoggerOperazioni.

### AmministrazioneIscritti

Questa classe si occupa della registrazione e della gestione degli Utenti nel sistema. Come già indicato in fase di analisi, un Amministratore deve avere la possibilità di registrare e modificare i dati relativi a Professori e Studenti, accomunati dalla classe astratta Iscritto. La modifica dei dati dell'utente è stata separata dalla modifica delle credenziali, in quanto esse costituiscono un dato particolarmente sensibile e non devono essere mantenute all'interno delle classi.

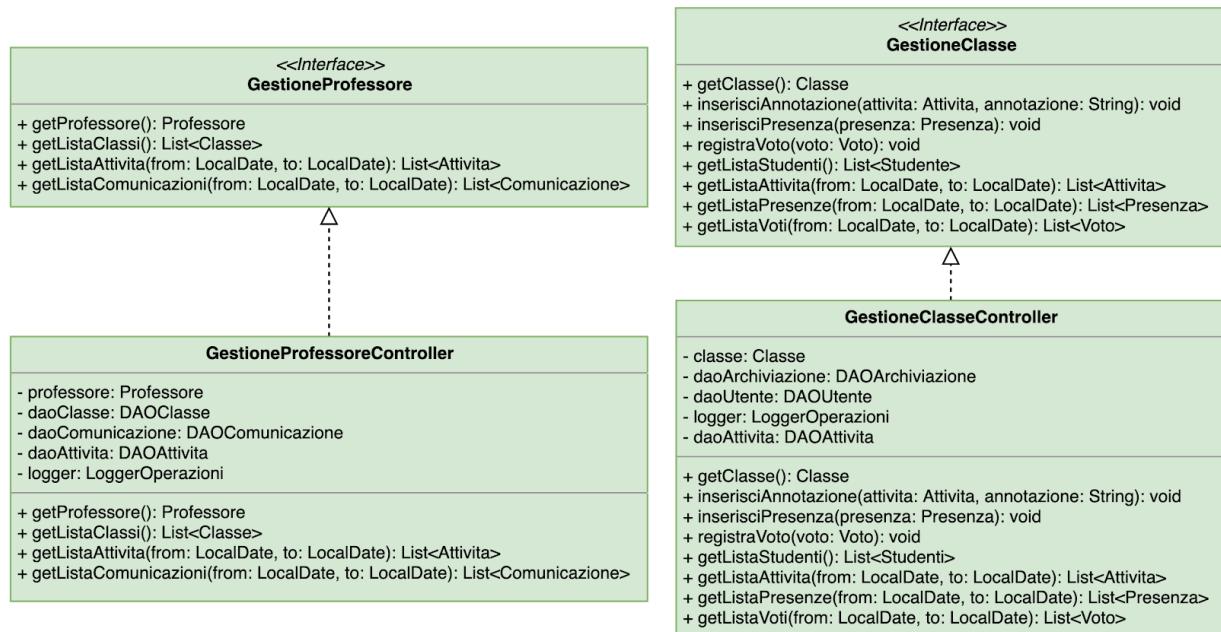
Non sono previste funzionalità che permettano l'eliminazione di un Utente dal sistema. Questo è dovuto da una mancanza di tale esigenza nei requisiti del software ed inoltre risulta una scelta coerente nel rispetto delle norme di archiviazione di sistemi scolastici, i quali tengono traccia di tutte le persone che hanno avuto un ruolo negli istituti nel corso degli anni.

### AmministrazioneClassi

Questa classe si occupa della registrazione e della gestione delle Classi scolastiche nel sistema. Essendo le Attività e i Voti legati ad una singola Classe, anche essi vengono gestiti in questo controllore. La registrazione o la rimozione delle Attività scolastiche costituiscono due operazioni critiche per la consistenza informativa del Sistema.

I metodi **registraAttivita** ed **eliminaAttivita** devono perciò soddisfare da contratto (**Design By Contract**) la seguente precondizione: **“All’istante di registrazione o eliminazione di un’Attività, lo svolgimento della stessa deve essere previsto in data ed ora futuri.”**

## Package mitro.controller.professore



Questo package si occupa della gestione dei controllori relativi alle azioni consentite ad un cliente Professore. Per effettuare tutte le operazioni relative alla persistenza delle classi del dominio, ogni realizzazione delle interfacce controller deve basarsi sugli oggetti DAO esposti in precedenza. Allo stesso modo, la scrittura delle voci del Log relativi alle operazioni di sistema deve avvenire mediante l'utilizzo dell'interfaccia `LoggerOperazioni`.

### GestioneProfessore

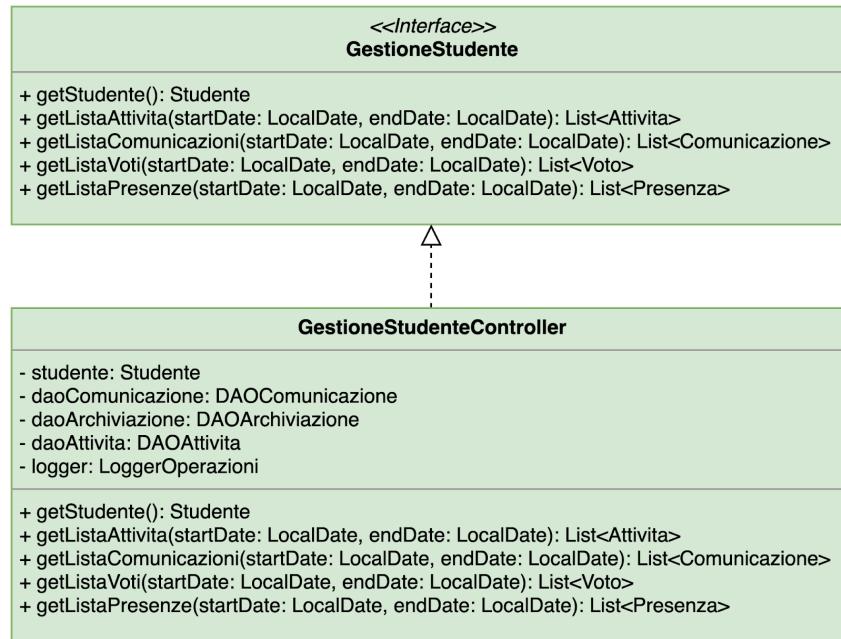
Questa classe si occupa di gestire e mostrare le informazioni relative ad un singolo Professore. Oltre alla lettura dei propri dati Utente, un Professore gode quasi esclusivamente di privilegi di visualizzazione di dati di sua responsabilità come la lista delle Attività di cui è referente, delle Classi in cui esse si svolgono, e delle Comunicazioni ad egli indirizzate.

### GestioneClasse

Questa classe si occupa di gestire e mostrare le informazioni relative ad una singola Classe, secondo quanto possibile ai privilegi di un Professore. Si rende possibile ottenere la lista degli Studenti partecipanti alla Classe, delle Attività svolte in essa, e delle Archiviazioni registrate. La registrazione (o l'inserimento) di una qualsiasi Archiviazione è un'azione critica nei confronti della consistenza informativa del Sistema.

I metodi `registraVoto` ed `inserisciPresenza` devono perciò soddisfare da contratto (**Design By Contract**) la seguente precondizione: **All'istante dell'inserimento o della registrazione, l'Archiviazione deve riferirsi ad uno Studente che risulta partecipante alla Classe in cui si svolge la relativa Attività.**

## Package mitro.controller.studente

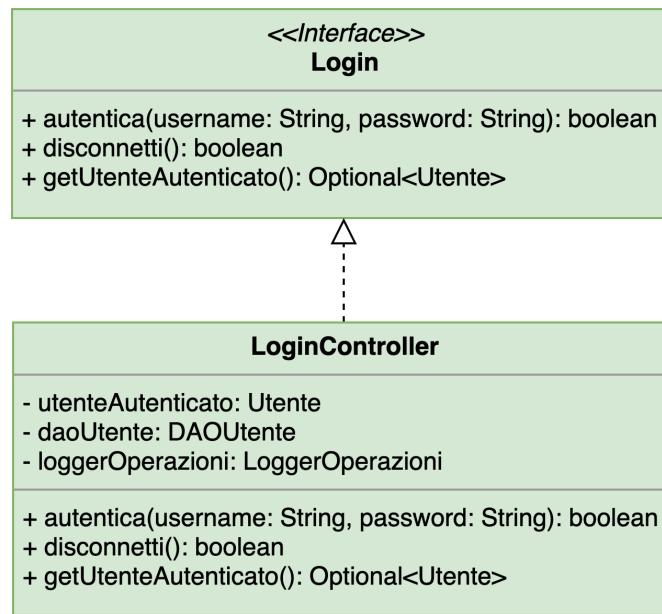


Questo package si occupa della gestione dei controllori relativi alle azioni consentite ad un cliente Amministratore. Per effettuare tutte le operazioni relative alla persistenza delle classi del dominio, ogni realizzazione delle interfacce controller deve basarsi sugli oggetti DAO esposti in precedenza. Allo stesso modo, la scrittura delle voci del Log relativi alle operazioni di sistema deve avvenire mediante l'utilizzo dell'interfaccia LoggerOperazioni.

### GestioneStudente

Questa classe si occupa di gestire e mostrare le informazioni relative ad un singolo Studente. Oltre alla lettura dei propri dati Utente, uno Studente gode quasi esclusivamente di privilegi di visualizzazione di dati di sua responsabilità come la Classe a cui è iscritto, la lista delle Attività in cui deve partecipare, le proprie Archiviazioni, e le Comunicazioni ad egli indirizzate.

## Package mitro.controller.login



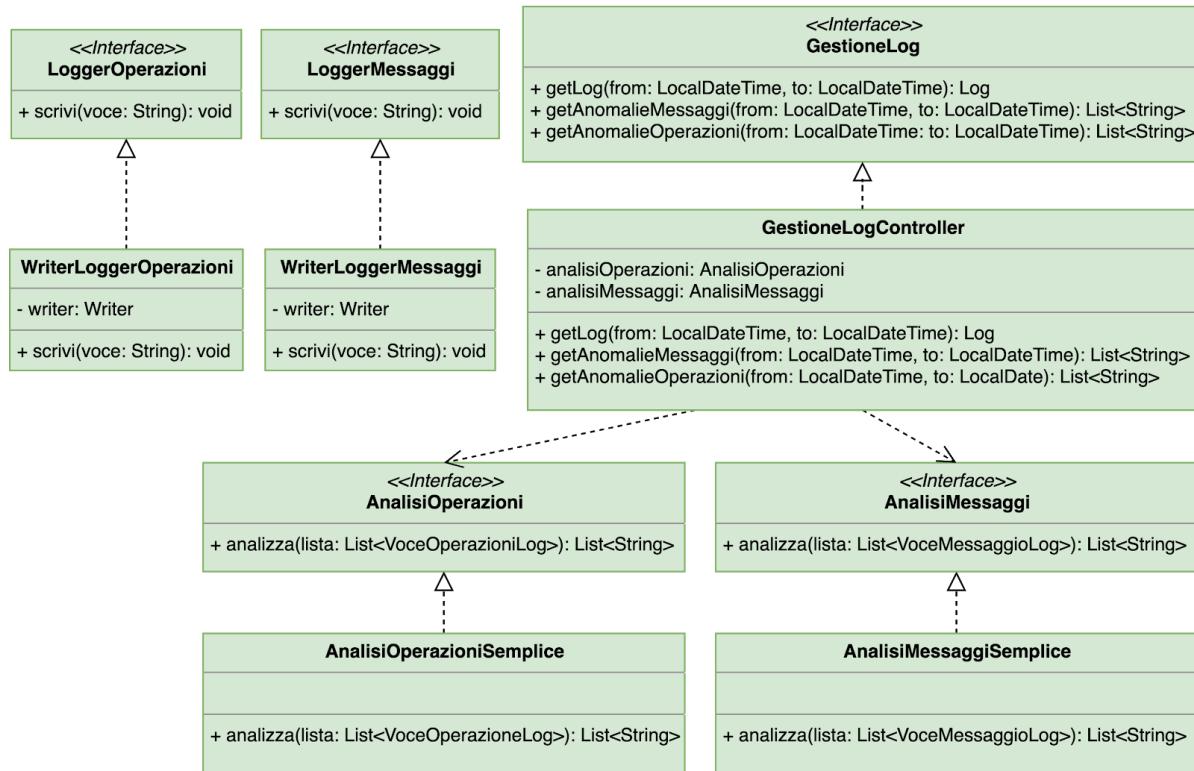
Questo package si occupa di organizzare le implementazioni del sistema di controllo per l'autenticazione dei clienti nel sistema.

### Login

Questa interfaccia definisce i metodi fondamentali che un qualsiasi controller per l'autenticazione deve rispettare. Le credenziali sono passate al metodo **autentica** come argomento, e non devono essere preservate come dato interno di alcuna classe in modo da salvaguardarne la sicurezza. Mediante l'interfaccia `DAOUtente`, le varie implementazioni possono utilizzare il metodo **ottieneUtentePerCredenziali** per interrogare la persistenza circa la correttezza delle credenziali e la loro associazione ad un `Utente` da usare per autenticare la sessione. Mediante il metodo **disconnetti**, è infine possibile deautenticare la sessione con il cliente. Entrambe i metodi restituiscono un valore booleano, in modo da verificare il successo dell'operazione. Invocando il metodo **getUtenteAutenticato** è possibile ottenere l'`Utente` associato al cliente durante la sessione di utilizzo del servizio. Il valore restituito utilizza il costrutto `Optional` di Java, in modo da favorire un controllo sicuro del dato desiderato e verificarne la presenza. Infine, anche in questo caso tutte le operazioni devono registrare nel Log mediante l'interfaccia `LoggerOperazioni`.

Nella fase di implementazione dovrà essere previsto un sistema centrale che gestisca le autenticazioni avvenute nel sistema, in modo da impedire l'accesso simultaneo di diversi clienti con lo stesso `Utente`.

## Package mitro.controller.log



Questo package si occupa di organizzare le implementazioni del sistema di controllo per la gestione dei Log. Le interfacce proposte possono avere diverse realizzazioni e configurazioni, la cui cura è nuovamente responsabilità della classe ControllerFactory.

### LoggerOperazioni

Questa interfaccia espone esclusivamente il metodo **scrivi**, e deve essere utilizzata da tutti i controller per registrare voci nel Log. Nascondere all'esterno la realizzazione del sistema di scrittura, permette implementazioni multiple e lascia aperta la possibilità di modificare il sistema di persistenza del Log. L'implementazione proposta `WriterLoggerOperazioni` si basa infatti sulla classe `Writer` del linguaggio Java, la quale può scrivere intercambiabilmente su file, connessioni remote, o sullo standard output. Tale approccio favorirebbe il collaudo, permettendo di verificare a run-time il corretto utilizzo dei metodi di scrittura del Log.

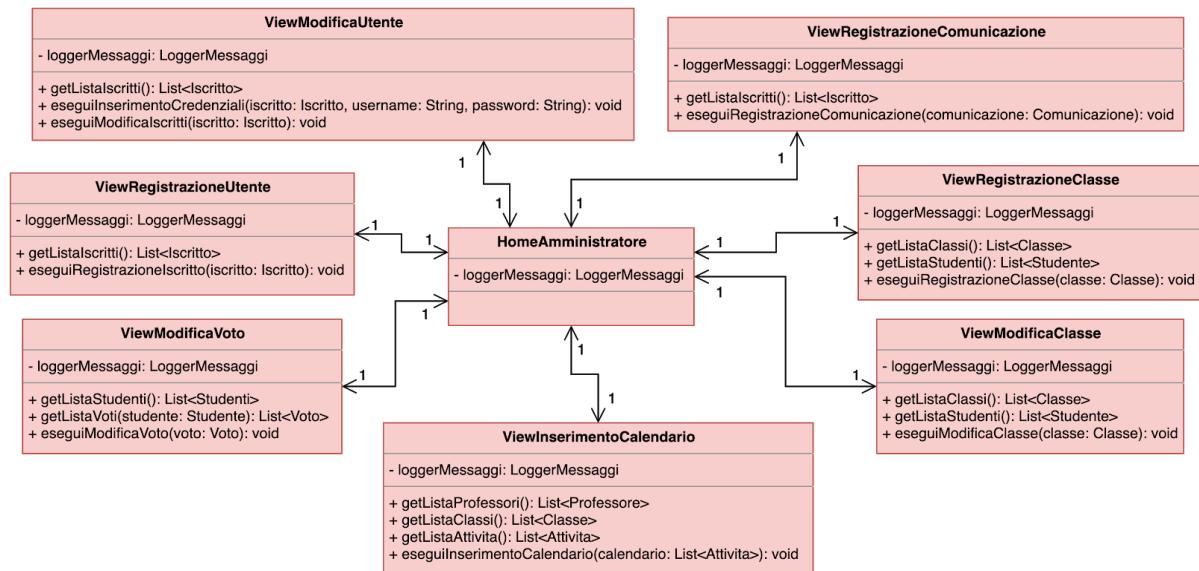
### LoggerMessaggi

Questa interfaccia è del tutto analoga alla precedente, ma specifica per i messaggi scambiati dalle conversazioni con i clienti.

### GestioneLog

Questa interfaccia espone i metodi fondamentali per l'accesso e l'analisi dei Log da parte di un Gestore della Sicurezza. Per l'analisi delle voci finalizzata al riconoscimento di anomalie, è stato applicato il design-pattern **Strategy**. Gli algoritmi di analisi sono perciò delegati alle varie realizzazioni delle interfacce `AnalisiOperazioni` ed `AnalisiMessaggi`.

## Package mitro.view.amministratore



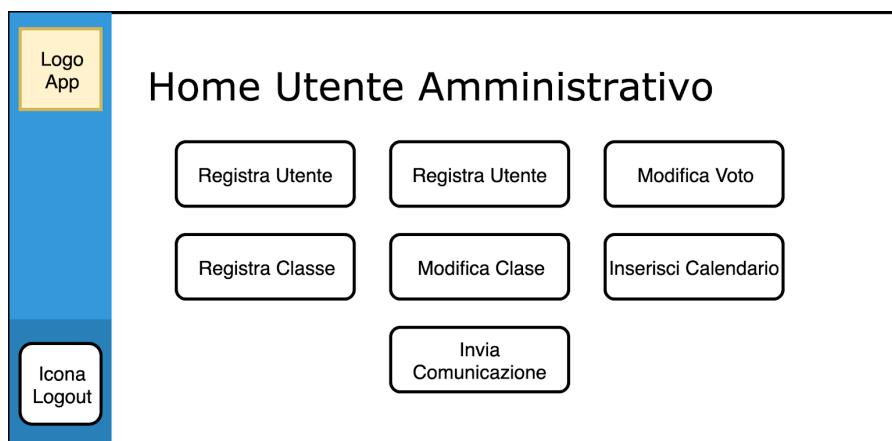
Nel diagramma sopra rappresentato, sono evidenziate tutte le interfacce disponibili riguardanti i casi d'uso dei clienti Amministratori. Ogni view deve mettere a disposizione form adeguati che permettano l'inserimento dei dati utili per lo svolgimento del suo compito.

Tutta la logica di business delle operazioni richieste deve essere affidata ad opportuni controller, e tutti i messaggi scambiati nelle conversazioni con i clienti devono essere scritte nel Log attraverso l'interfaccia LoggerMessaggi.

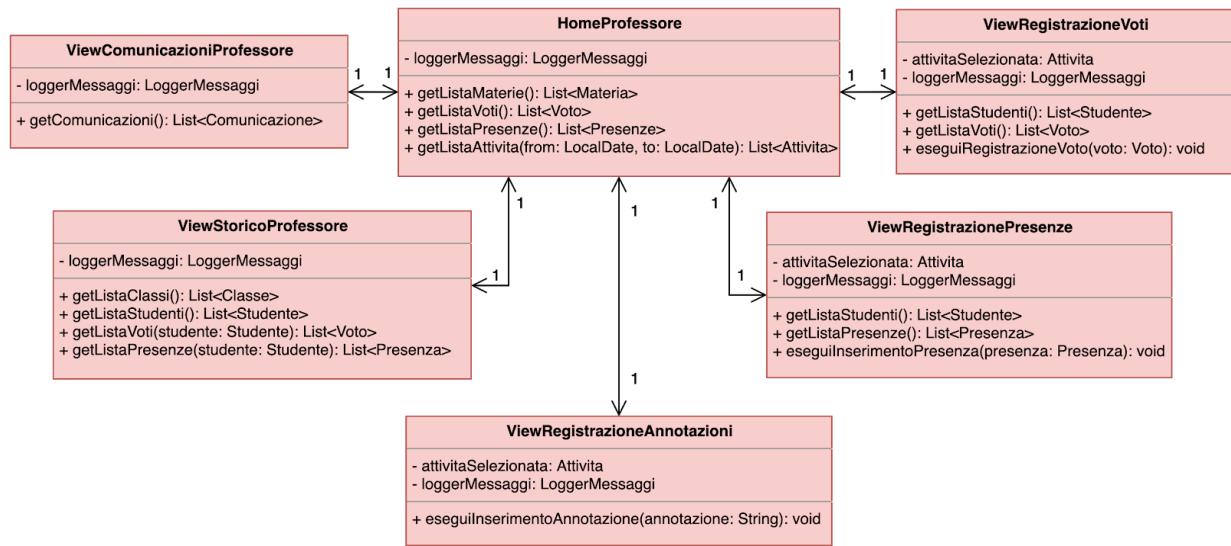
Tutte le view sono contenute all'interno dell'applicativo dei servitori ed ai clienti ne viene solo presentata una rappresentazione grafica sotto forma di pagina Web dinamica.

Nella figura sottostante è possibile vedere il formato di realizzazione grafica che si desidera ottenere per la Home dell'Amministratore, dalla quale egli deve poter accedere tutti i servizi consentiti. Ogni sotto-interfaccia specifica, deve infine prevedere un sistema che permetta la redirectazione alla Home o la deautenticazione.

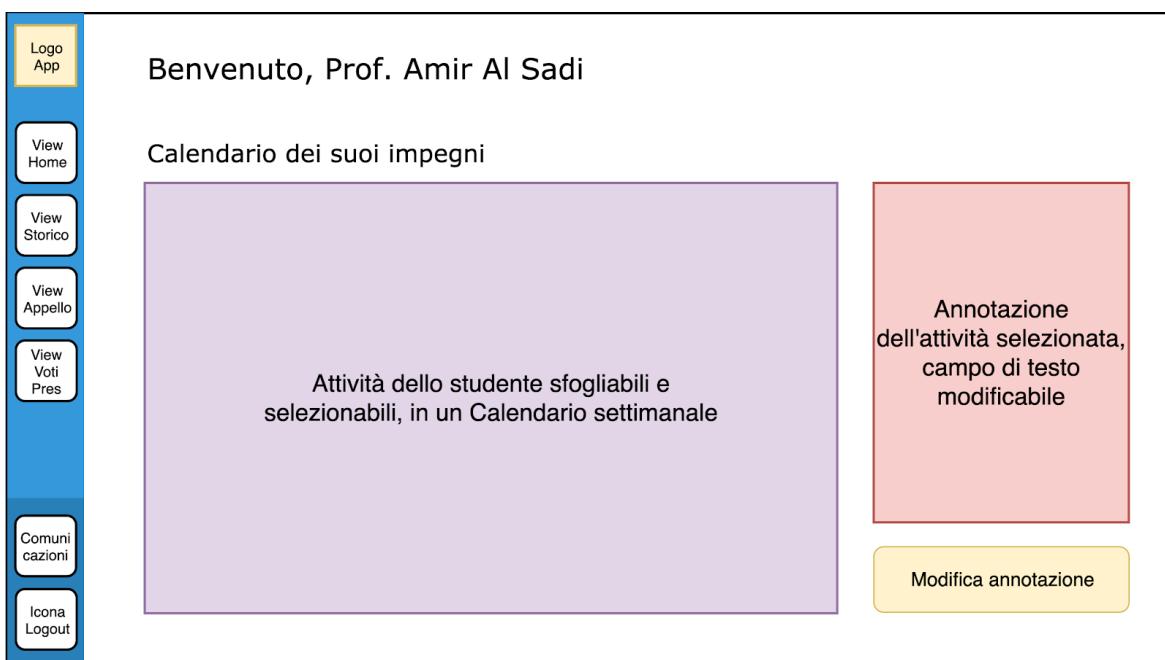
ViewRegistrazioneUtente e ViewModificaUtente devono prevedere un sistema di generazione automatica di Username e Password, in modo da astenere gli Amministratori dalla scelta di credenziali potenzialmente insicure qualora se ne volessero inserire di nuove nel sistema.



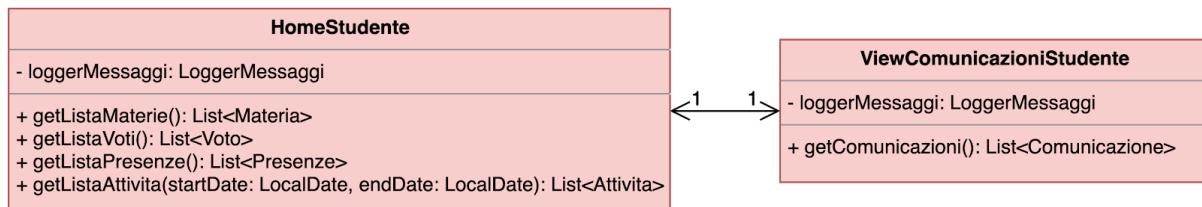
## Package mitro.view.professore



Nel diagramma sopra rappresentato, sono evidenziate tutte le interfacce disponibili riguardanti i casi d'uso dei clienti Professori. Tutta la logica di business delle operazioni richieste deve essere affidata ad opportuni controller, e tutti i messaggi scambiati nelle conversazioni con i clienti devono essere scritte nel Log usando l'interfaccia `LoggerMessaggi`. Tutte le view sono contenute all'interno dell'applicativo dei servitori ed ai clienti ne viene solo rappresentata una rappresentazione grafica sotto forma di pagina Web dinamica. Nella figura sottostante è possibile vedere il formato di realizzazione grafica che si desidera ottenere per la Home del Professore, dalla quale egli deve poter accedere a tutti i servizi consentiti. La Home deve inoltre mostrare il Calendario delle Attività in cui il Professore dovrà presenziare, e un form che permetta di modificarne le Annotazioni.



## Package mitro.view.studente



Nel diagramma sopra rappresentato, sono evidenziate tutte le interfacce disponibili riguardanti i casi d'uso dei clienti Studenti. Tutta la logica di business delle operazioni richieste deve essere affidata ad opportuni controller, e tutti i messaggi scambiati nelle conversazioni con i clienti devono essere scritte nel Log attraverso l'interfaccia LoggerMessaggi. Tutte le view sono contenute all'interno dell'applicativo dei servitori ed ai clienti ne viene solo rappresentata una rappresentazione grafica sotto forma di pagina Web dinamica. Nella figura sottostante è possibile vedere il formato di realizzazione grafica che si desidera ottenere per la Home dello Studente, la quale deve mostrare mostrare il Calendario delle Attività, la lista delle Materie e dei Voti, grafici e statistiche relative al rendimento scolastico. Per ogni Attività contenuta nel Calendario è possibile visionare le Presenze e le Annotazioni, qualora queste fossero disponibili.

The screenshot shows the student home page with the following components:

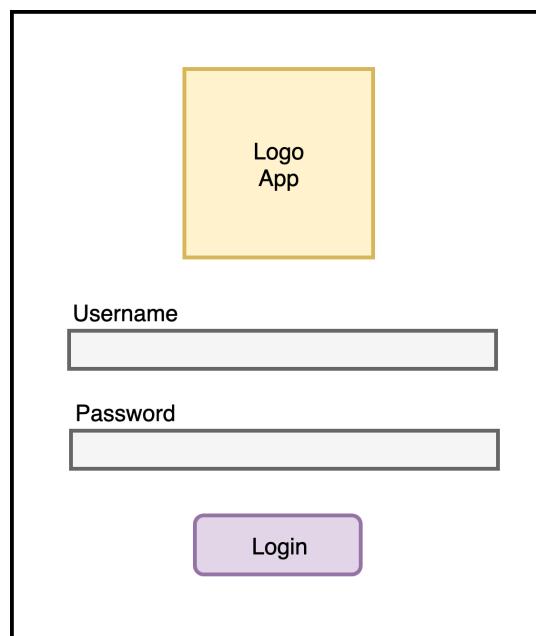
- Header:** Benvenuto, Federico Baldini
- Top Right:** Totale ore di assenza: 24
- Left Sidebar:**
  - Logo App
  - View Home
  - Comunicazioni
  - Icona Logout
- Middle Content:**
  - Attività:** Attività dello studente sfogliabili e selezionabili, in un Calendario settimanale. In ogni Attività è possibile visionare la Presenza.
  - Materie:** Lista delle materie (Matematica, Italiano, Scienze, **Fisica**, Filosofia, Inglese, Informatica, Educazione fisica, Francese, Spagnolo, Storia dell'arte).
  - Voti:** Voti della materia (Table showing votes for Fisica).
  - Statistics:**
    - Grafico voti della materia (Media 7.44)
    - Grafico voti tutte le materie (Media 7.82)

## Package mitro.view.login

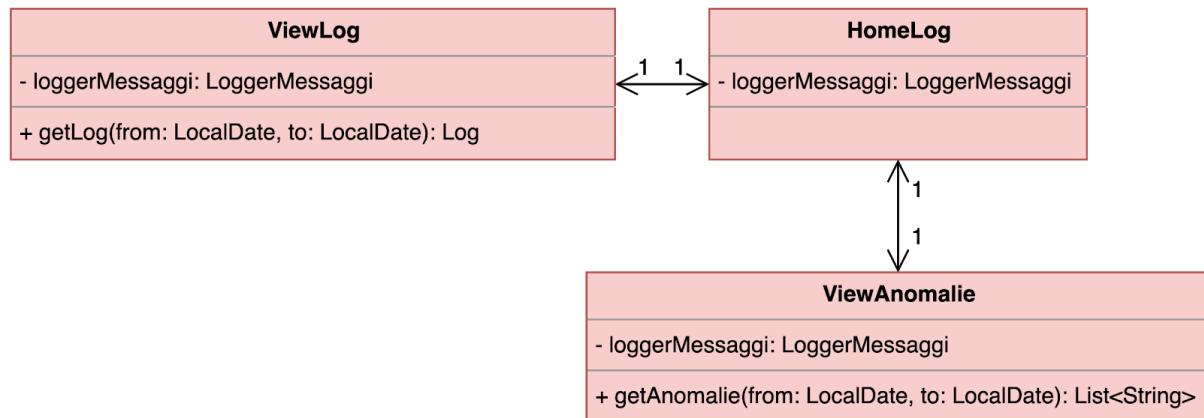
ViewLogin
- loggerMessaggi: LoggerMessaggi
+ eseguiAutenticazione(username: String, password: String): boolean
+ eseguiDisconnessione(): boolean

L'interfaccia ViewLogin mette a disposizione un form nel quale il cliente deve inserire le proprie credenziali per autenticarsi. Successivamente, egli deve essere reindirizzato all'interfaccia Home di sua competenza.

Il campo Password deve risultare illeggibile all'esterno, in modo da aumentare il grado di sicurezza dell'utenza.



## Package mitro.view.log

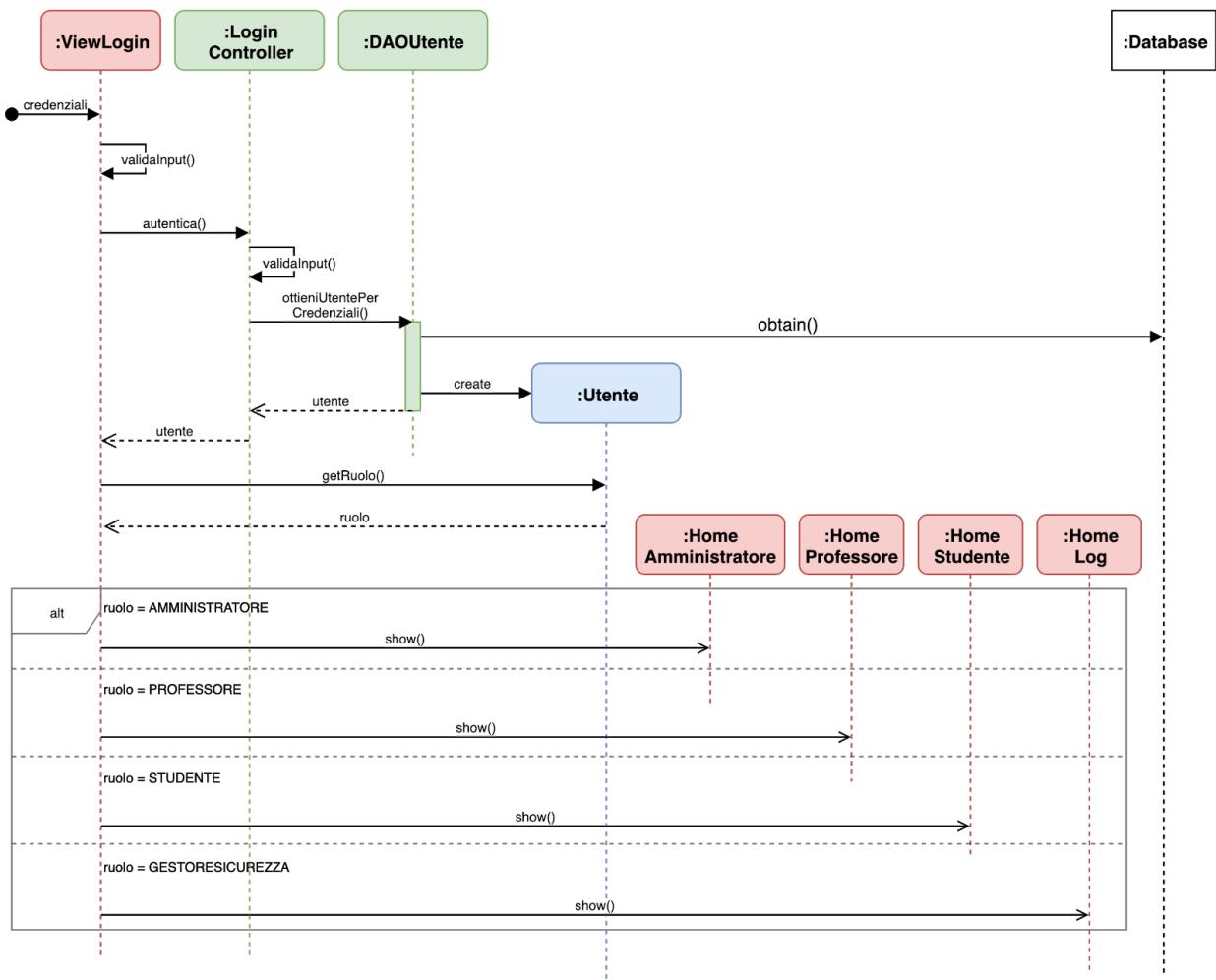


Nel diagramma sopra rappresentato, sono evidenziate tutte le interfacce disponibili riguardanti i casi d'uso del Gestore della Sicurezza. Tutta la logica di business delle operazioni richieste deve essere affidata ad opportuni controller, e tutti i messaggi scambiati nelle conversazioni con i clienti devono essere scritte nel Log usando l'interfaccia LoggerMessaggi. Tutte le view sono contenute all'interno dell'applicativo dei servitori ed ai clienti ne viene solo presentata una rappresentazione grafica sotto forma di pagina Web dinamica. Nella figura sottostante è possibile vedere il formato di realizzazione grafica che si desidera ottenere per la Home del Gestore della Sicurezza la quale deve permettere ricerca e visualizzazione dei Log in determinati intervalli temporali, e l'esecuzione di analisi mirate sugli stessi per il riconoscimento di anomalie di sistema.

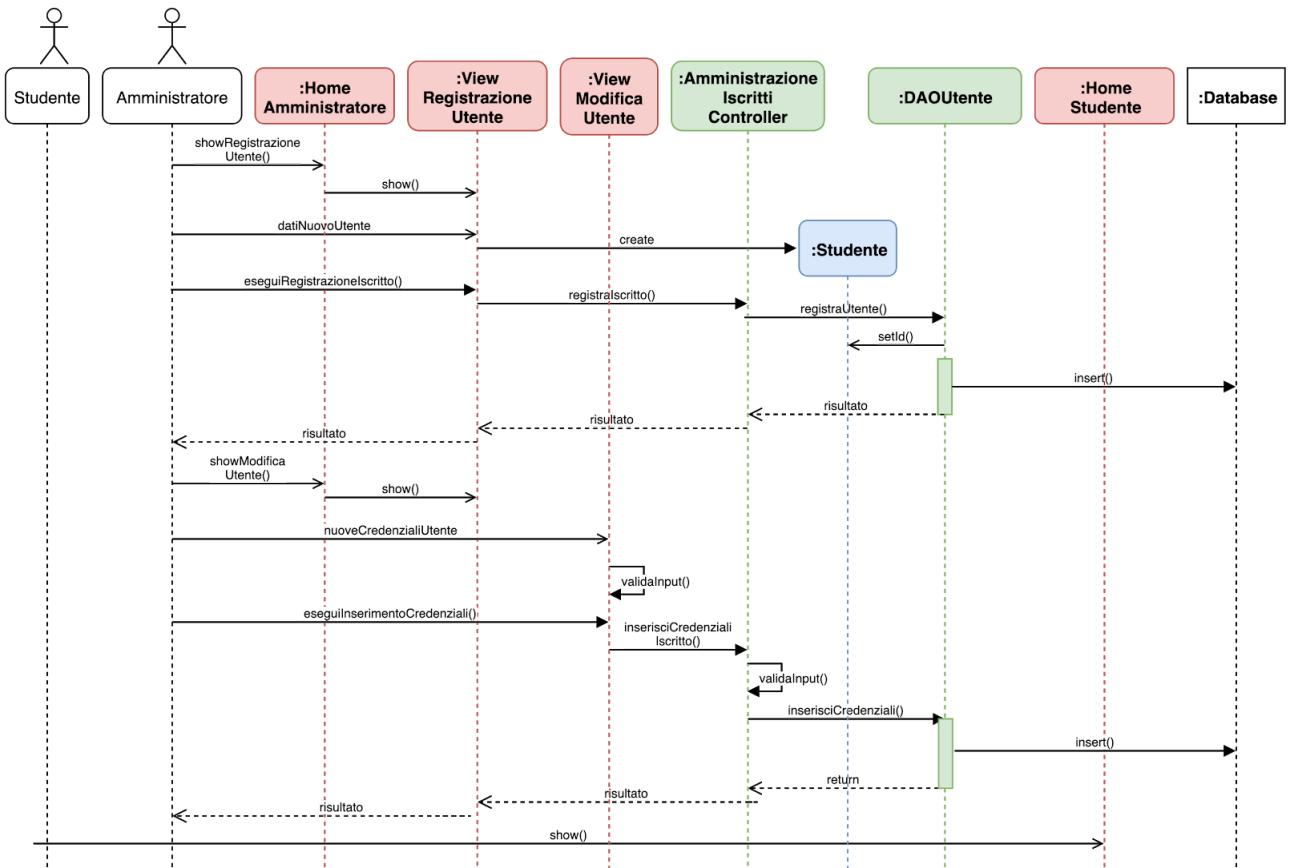
## Interazioni

Verranno in questa sezione mostrate delle rivisitazioni dei diagrammi di sequenza esposti in fase di analisi, approfonditi secondo le novità di dettaglio introdotte nella progettazione.

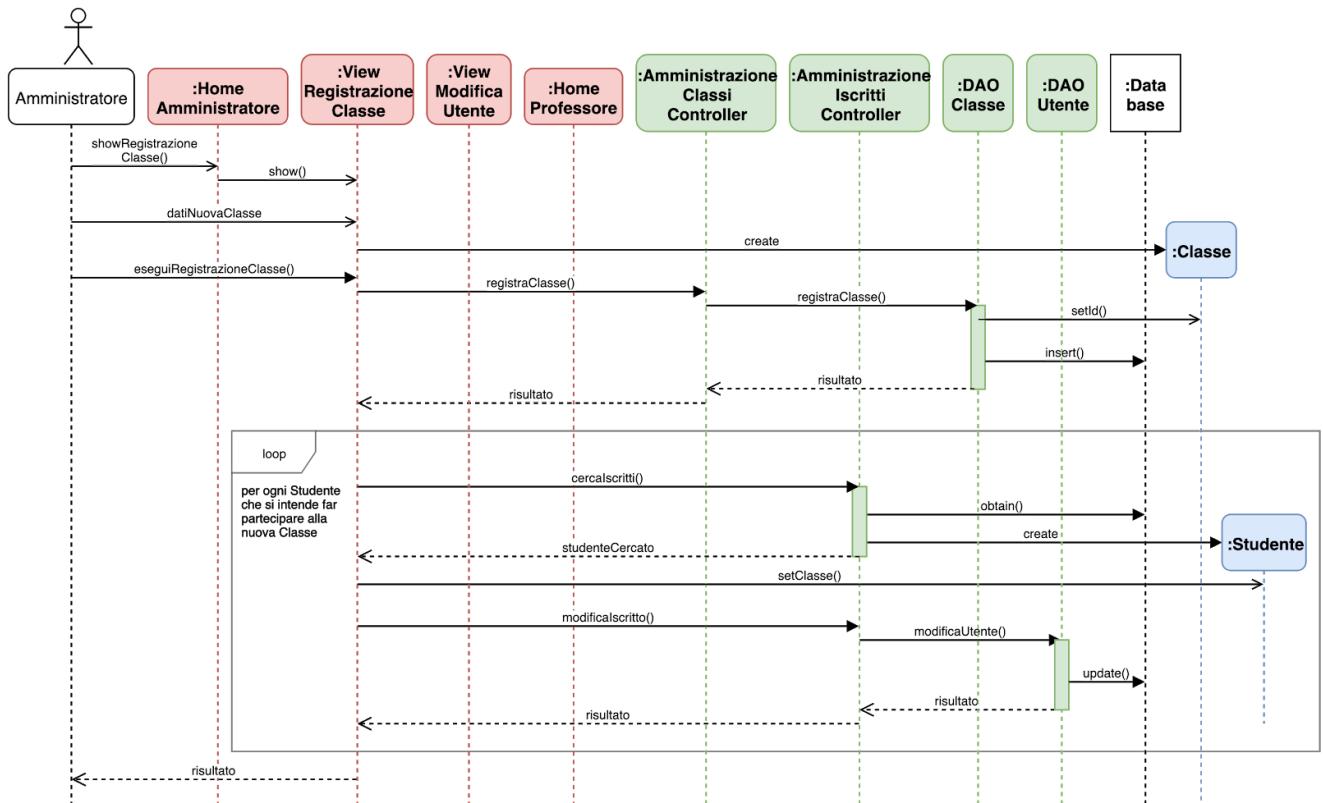
Diagramma di Sequenza: Autenticazione avvenuta con successo



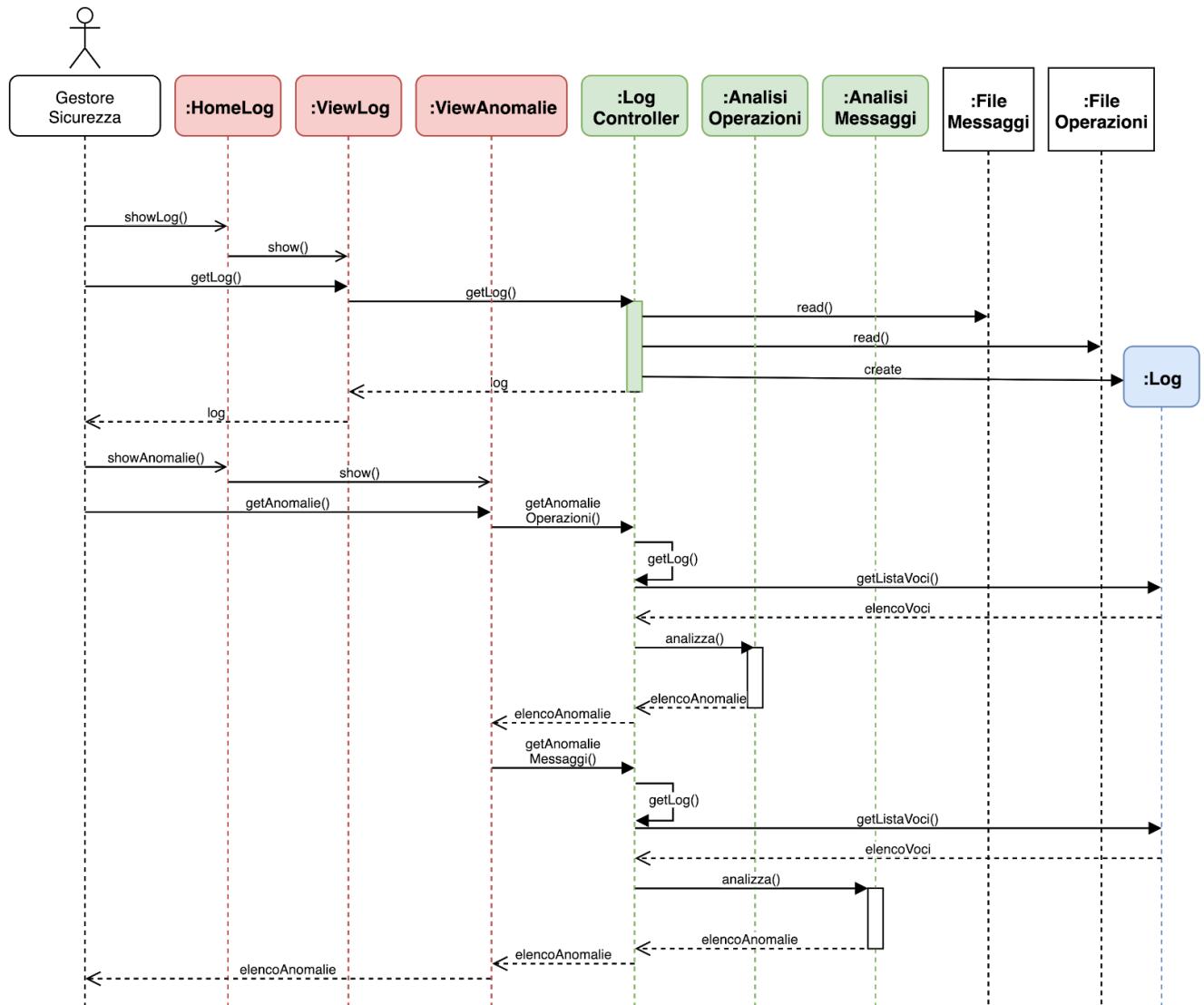
## Diagramma di Sequenza: Registrazione Utente



## Diagramma di Sequenza: Registrazione Classe



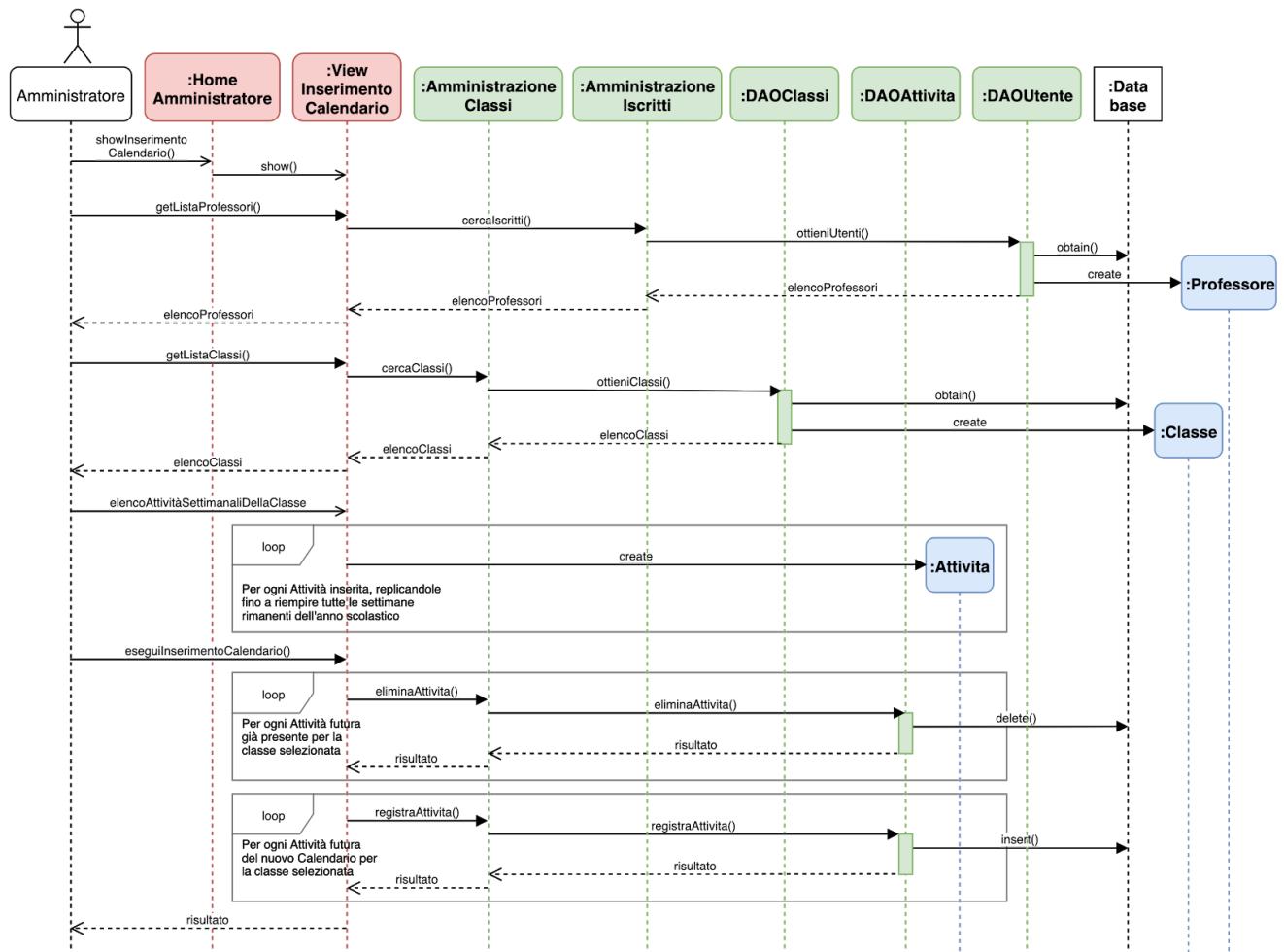
## Diagramma di Sequenza: Visualizzazione Log e Anomalie



## Diagramma di Sequenza: Inserimento Calendario di Attività

Questa interazione, non presente in fase di analisi, mette in luce la metodologia astratta con cui l'inserimento del Calendario delle Attività di una Classe deve avvenire.

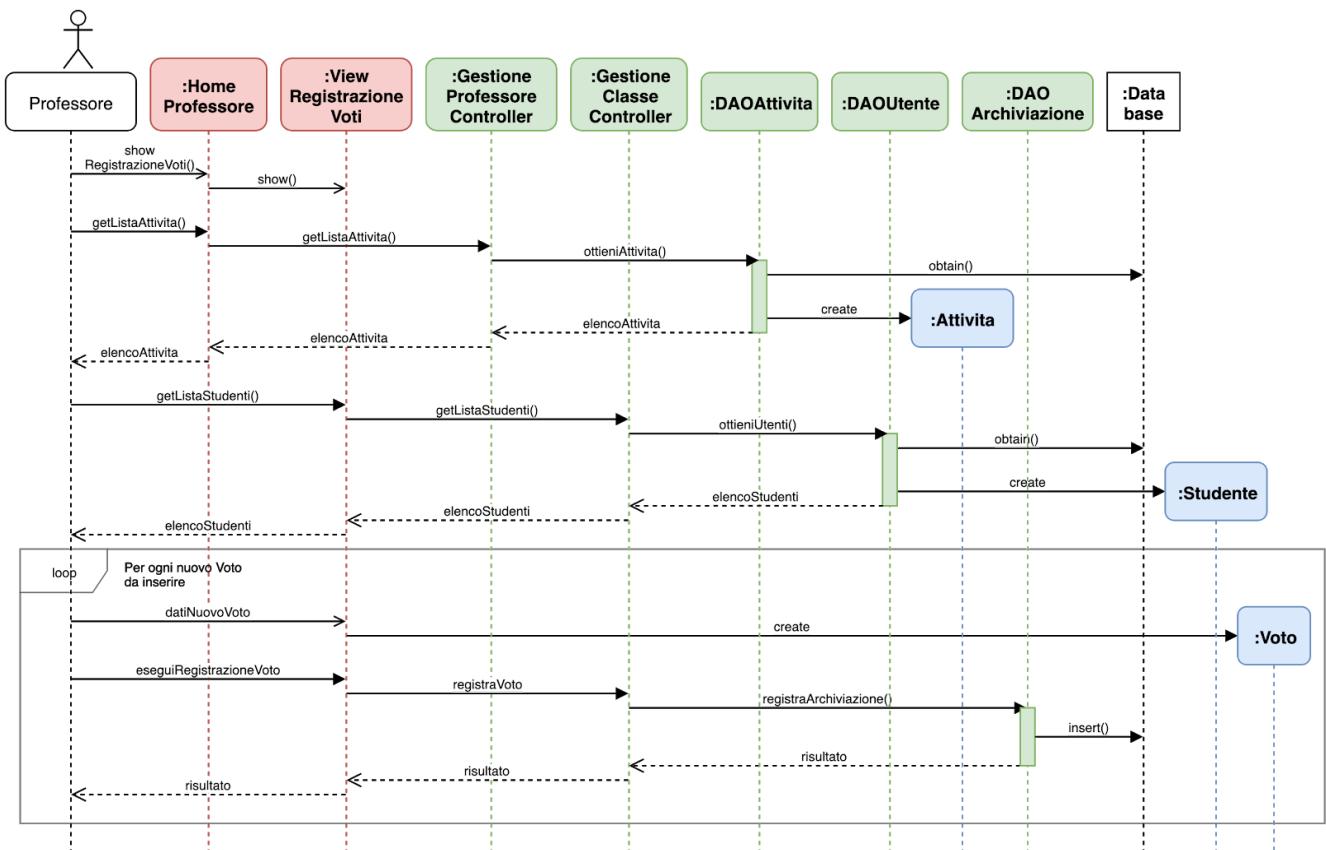
Il cambiamento del Calendario costituisce un alteramento di tutti gli eventi futuri che dovranno periodicamente essere verificati nella Classe, pertanto devono essere prima eliminate tutte le voci già esistenti nel sistema. Come discusso in precedenza non si deve, da contratto, tentare di eliminare Attività datate in momenti passati rispetto all'istante di inserimento.



## Diagramma di sequenza: Registrazione Voto

Questa interazione, non presente in fase di analisi, mette in luce la metodologia astratta con cui la registrazione di un nuovo Voto per uno Studente deve avvenire.

Nel caso dell'inserimento multiplo di Voti per più studenti, è necessario eseguire un ciclo iterativo su tutti i partecipanti della Classe interessati, nascondendo al cliente la complessità dell'operazione.



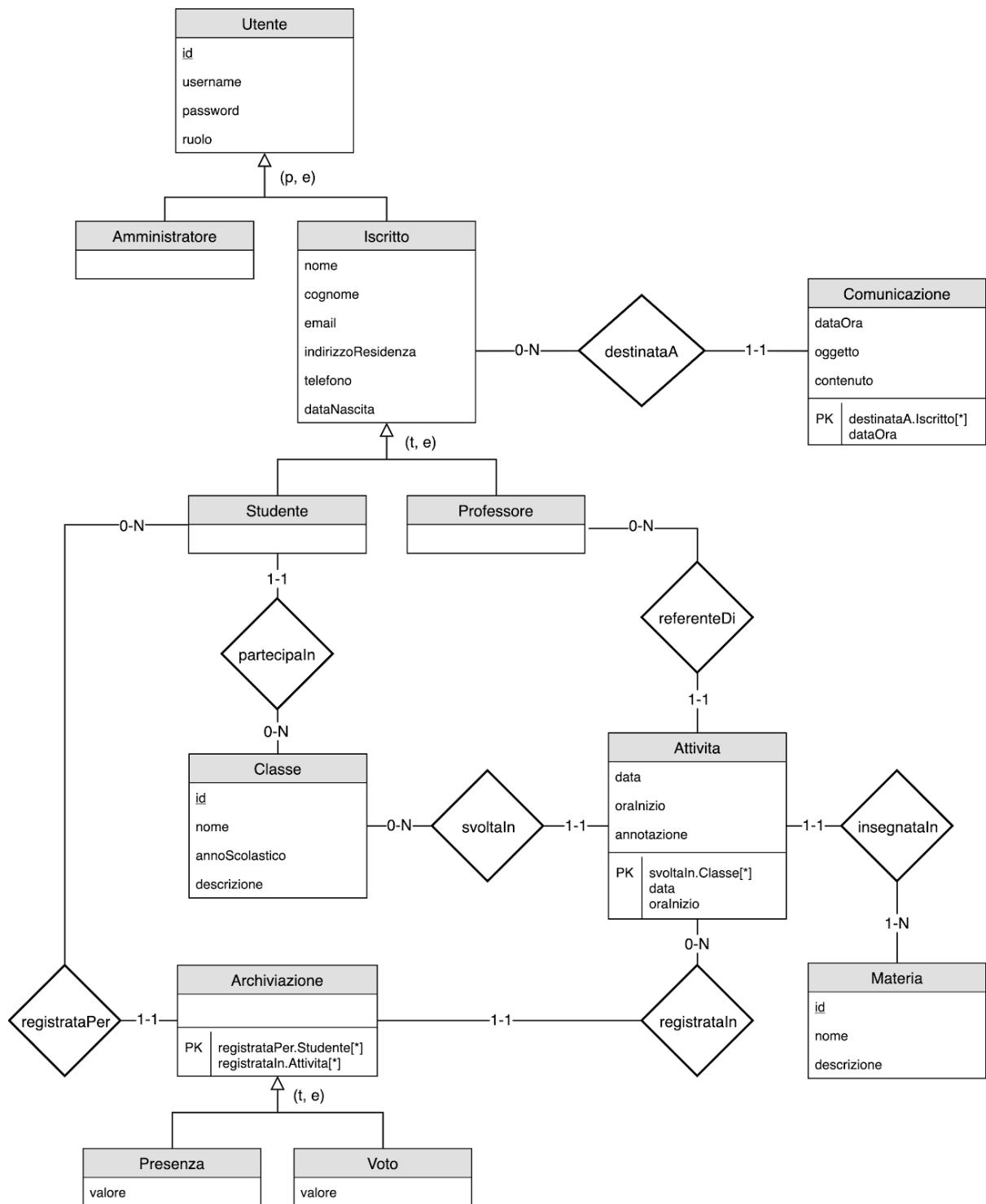
## Comportamento

Come riconosciuto in sede di Analisi del Problema, non sono presenti entità che richiedano la progettazione specifica di un comportamento.

Le realizzazioni degli algoritmi di analisi delle anomalie nei Log, le quali possono essere molteplici per via del Design-Pattern **Strategy**, sono delegate alla fase di implementazione.

# Progettazione della Persistenza

## Diagramma Entità-Relazione



Il diagramma Entità-Relazione sopra riportato rispecchia in maniera pressoché fedele il modello del dominio riconosciuto in fase di analisi. Sono state mantenute le gerarchie nelle entità Utente ed Archiviazione, le quali devono essere propriamente trattate in fase di realizzazione.

### Entità

- **Utente (e gerarchia):** Un Utente deve essere identificato univocamente (primary key) dal suo Id. Inoltre, il campo Username deve essere unico all'interno del sistema.
- **Comunicazione:** Una Comunicazione deve essere identificata univocamente (primary key) dall'Id dell'Utente destinatario e dalla DataOra di emissione.
- **Classe:** Una Classe deve essere identificata univocamente (primary key) dal suo Id.
- **Attività:** Un'Attività deve essere identificata univocamente (primary key) dall'Id della Classe in cui è svolta, dalla Data ed Ora di inizio.
- **Materia:** Una Materia deve essere identificata univocamente (primary key) dal suo Id. La presenza di questa entità nel modello è dovuta al fatto che la stessa Materie può essere associata a più Attività.
- **Archiviazione (e gerarchia):** Un'Archiviazione deve essere identificata univocamente (primary key) dall'Id dello Studente interessato, e dalla Attività in cui è stata registrata.

### Relazioni

- **DestinataA:** Ogni Comunicazione può essere associata ad un solo Iscritto; un Iscritto può essere associato a zero o più Comunicazioni.
- **ReferenteDi:** Un'Attività può essere associata ad un solo Professore; un Professore può essere associato a zero o più Attività.
- **Partecipaln:** Uno Studente può essere associato ad una sola Classe; una Classe può essere associata a zero o più Studenti.
- **Svoltaln:** Un'Attività può essere associata ad una sola Classe; una Classe può essere associata a zero o più Attività.
- **Insegnataln:** Un'Attività può essere associata ad una sola Materia; una Materia può essere associata a una o più Attività.
- **Registrataln:** Un'Archiviazione può essere associata ad una sola Attività; un'Attività può essere associata a zero o più Archiviazioni.
- **RegistrataPer:** Un'Archiviazione può essere associata ad un solo Studente; uno Studente può essere associato a zero o più Archiviazioni.

### Formato dei file di Log

Il file di Log delle Operazioni deve contenere tutte le azioni richieste da un cliente al Sistema. L'Utente richiedente dei servizi deve essere registrato nelle voci del Log con unica eccezione del controller Login, il quale potrebbe trattare di operazioni richiesta da un cliente non ancora autenticato.

<timestamp> <id-utente> <operazione>

Il file di Log dei Messaggi deve contenere tutte le parti delle conversazioni remote intraprese con i clienti del sistema. Quando possibile è opportuno registrare anche l'attore che ha inoltrato il messaggio, e specificare di volta in volta se esso tratta una ricezione o un invio.

```
<timestamp> <invio-ricezione> <attore> <messaggio>
```

## Considerazioni su protezione e sicurezza

Diverse osservazioni sono già state discusse nel paragrafo “Considerazioni relative alle tecnologie utilizzate” del capitolo di Progettazione Architetturale.

Sia nel caso del Log che in quello del Database, la soluzione preferibile nei termini della Sicurezza sarebbe la cifratura di tutti i dati inseriti. Tuttavia andrebbe di volta in volta considerato l'impatto di tale scelta sulle prestazioni del sistema, andando ad effettuare compromessi in fase di realizzazione.

Nel Database, per esempio, si potrebbe pensare di cifrare solo determinate entità o porzioni delle stesse in base alla criticità dell'informazione. Per esempio sarebbe opportuno proteggere tutti i dati relativi ad un Utente (all'infuori dell'identificativo), mentre potrebbero essere lasciate leggibili le Archiviazioni le quali non hanno significato specifico se non in relazione alla persona a cui sono riferite. In ogni caso, le credenziali utente devono essere cifrate. Per le password è in particolare opportuno utilizzare sistemi di Hashing, in modo da evitarne la leggibilità in possibili attacchi mirati.

Per il Log potrebbe essere applicato lo stesso principio. Considerando il fatto che le voci verranno codificate in file locali, si potrebbero anche considerare i sistemi di protezione del File System forniti dal sistema operativo su cui i servitori sono installati.

## Progettazione del Collaudo

Partendo da quanto citato in fase di analisi, è necessario sviluppare numerosi test unitari che verifichino la correttezza delle realizzazioni di tutte le classi del sistema. Avendo adottato il linguaggio Java in fase di progettazione, si indica di utilizzare la libreria JUnit per sviluppare tutti i test.

Come discusso in precedenza, l'utilizzo del pattern DAO nella gestione della persistenza favorisce particolarmente il collaudo dei controller, i quali possono essere dotati di realizzazioni "Mock" che mantengono i dati registrati in una collezione temporanea.

Più in generale, l'introduzione di numerose interfacce permette di collaudare ogni componente singolarmente, senza curarsi della dipendenza stretta a realizzazioni particolari.

Onde evitare una stesura eccessivamente lunga del documento, verranno qui riportati solo alcuni test caratteristici sui controller della logica di business dell'applicazione:

```
public class AmministrazioneIscrittiControllerTest {

    @Test
    void testIscritti() {
        /* Costruzione controller con DAO Mock */
        AmministrazioneIscrittiController controller =
            new AmministrazioneIscrittiController();
        controller.setDAOUtente(new MockListDAOUtente());

        /* Costruzione Iscritto d'esempio */
        Iscritto iscritto = new Iscritto();
        iscritto.setNome("Amir");
        iscritto.setCognome("Al Sadi");

        /* Test registrazione di un Iscritto */
        List<Iscritto> tutti = controller.cercaIscritti("");
        assertEquals(tutti.size(), 0);
        controller.registraIscritto(iscritto);
        tutti = controller.cercaIscritti("");
        assertEquals(tutti.size(), 1);
        assertEquals(tutti.get(0).getNome(), iscritto.getNome());
        assertEquals(tutti.get(0).getCognome(), iscritto.getCognome());

        /* Test modifica di un Iscritto */
        iscritto.setNome("Jason");
        iscritto.setCognome("Dellaluce");
        controller.modificaIscritto(iscritto);
        tutti = controller.cercaIscritti("");
        assertEquals(tutti.size(), 1);
        assertEquals(tutti.get(0).getId(), iscritto.getId());
        assertEquals(tutti.get(0).getNome(), iscritto.getNome());
        assertEquals(tutti.get(0).getCognome(), iscritto.getCognome());
    }
}
```

```

public class AmministrazioneClassiControllerTest {

    @Test
    void testClassi() {
        /* Costruzione controller con DAO Mock */
        AmministrazioneClassiController controller =
            new AmministrazioneClassiController();
        controller.setDAOClasse(new MockListDAOClasse());

        /* Costruzione classe d'esempio */
        Classe classe = new Classe();
        classe.setNome("5B");
        classe.setDescrizione("Esempio");

        /* Test registrazione di una Classe */
        List<Classe> tutti = controller.cercaClassi("");
        assertEquals(tutti.size(), 0);
        controller.registraClasse(classe);
        tutti = controller.cercaClassi();
        assertEquals(tutti.size(), 1);
        assertEquals(tutti.get(0).getNome(), classe.getNome());
        assertEquals(tutti.get(0).getDescrizione(), classe.getDescrizione());

        /* Test modifica di una Classe */
        classe.setNome("3C-LA");
        classe.setDescrizione("Esempio2");
        controller.modificaClasse(classe);
        tutti = controller.cercaClassi();
        assertEquals(tutti.size(), 1);
        assertEquals(tutti.get(0).getId(), classe.getId());
        assertEquals(tutti.get(0).getNome(), classe.getNome());
        assertEquals(tutti.get(0).getDescrizione(), classe.getDescrizione());
    }

    @Test
    void testAttivita() {
        /* Costruzione controller con DAO Mock */
        AmministrazioneClassiController controller =
            new AmministrazioneClassiController();
        DAOAttivita daoAttivita = new MockListDAOAttivita();
        controller.setDAOAttivita(daoAttivita);

        /* Costruzione Attività d'esempio */
        Classe classe = new Classe();
        classe.setNome("5B");
        controller.registraClasse(classe);
        Professore professore = new Professore();
        professore.setNome("Gianni");
        Attivita attivita = new Attivita();
        attivita.setProfessore(professore);
    }
}

```

```

attivita.setClasse(classe);
attivita.setData(LocalDate.now().plusDays(2));
attivita.setOraInizio(12);
attivita.setMateria(new Materia());

/* Test registrazione di una Attività futura */
controller.registraAttivita(attivita);
List<Attivita> tutti = controller.getListaAttivita(classe);
assertEquals(tutti.size(), 1);
assertEquals(tutti.get(0), attivita);

/* Test eliminazione di una Attività futura */
controller.eliminaAttivita(attivita);
tutti = controller.getListaAttivita(classe);
assertEquals(tutti.size(), 0);

/* Test registrazione di una Attività passata */
attivita.setData(LocalDate.now().minusDays(2));
try {
    controller.registraAttivita(attivita);
    fail("Attività riferita al passato!");
}
catch(Exception e) { }

/* Test eliminazione di una Attività passata */
daoAttivita.registraAttivita(attivita);
try {
    controller.eliminaAttivita(attivita);
    fail("Attività riferita al passato!")
}
catch(Exception e) { }
}
}

```

## Progettazione per il Deployment

In ogni interfaccia grafica presentata ai clienti del sistema deve essere inclusa una stringa che indichi l'attuale versione del software.

Mitro - <nome-versione>

I clienti non devo avere controllo sulla configurazione attuale del sistema, la quale è totalmente a carico dei servitori. Essendo il progetto ideato come Web Application, la fase di aggiornamento deve essere gestita nelle macchine dei servitori con adeguata frequenza ed adottando gli strumenti forniti da Apache Tomcat, in modo da non compromettere la disponibilità del servizio.

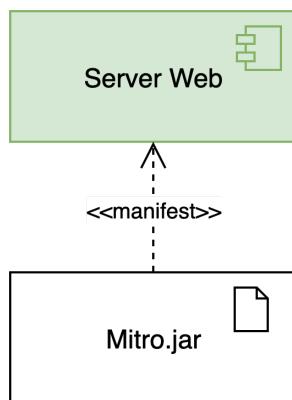
L'installazione del servizio avviene in due principali fasi:

1. **Installazione del DBMS:** Si intende installare il DBMS su un sistema hardware distaccato dai servitori, in modo da garantire sicurezza e performance. Questa fase del deployment può avvenire manualmente o mediante un ulteriore applicativo. Durante l'installazione del DBMS dovranno essere create le strutture iniziali capaci di ospitare la persistenza del servizio, e dovranno essere registrati gli Utenti Amministratori ed il Gestore della Sicurezza.
2. **Installazione dell'applicativo sui Server:** Si intende installare l'applicativo su almeno un dispositivo servitore, contemplando la possibilità di replicare il servizio su più unità. Al rilascio di nuove versioni del software, dovrà essere eseguito l'aggiornamento su tutti gli eventuali servitori in maniera manuale o automatica.

# Deployment

La natura distribuita del servizio prevede la possibilità di replicare il servitore su più nodi. Tuttavia l'applicazione rimane autocontenuta, con un unico artefatto che comprende tutte le funzionalità.

## Artefatti



## Deployment Type-Level

