

Corso di Laurea Magistrale in Ingegneria Informatica  
Ingegneria del Software

## **Progetto finale: AnaLab**

Fase di Elaborazione  
Iterazione 1

Giulia Silvestro

# Contents

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Caso d'uso UC5 : Inserisci nuovo test singolo</b>	<b>4</b>
2.1	Analisi . . . . .	4
2.1.1	Formato Dettagliato . . . . .	4
2.1.2	Modello di Dominio . . . . .	5
2.1.3	Diagramma di sequenza di sistema . . . . .	6
2.2	Progettazione . . . . .	6
2.2.1	Diagrammi di Interazione . . . . .	6
2.2.2	Diagramma delle Classi . . . . .	7
<b>3</b>	<b>Caso d'uso UC4: Inserisci nuovo test diagnostico</b>	<b>8</b>
3.1	Analisi . . . . .	8
3.1.1	Formato Dettagliato . . . . .	8
3.1.2	Modello di Dominio . . . . .	9
3.1.3	Diagramma di sequenza di sistema . . . . .	11
3.1.4	Contratti delle operazioni . . . . .	11
3.2	Progettazione . . . . .	12
3.2.1	Diagrammi di Interazione . . . . .	12
3.2.2	Diagramma delle Classi . . . . .	14
<b>4</b>	<b>Caso d'uso UC2: Inserisci nuova esenzione</b>	<b>16</b>
4.1	Analisi . . . . .	16
4.1.1	Modello di Dominio . . . . .	16
4.1.2	Diagramma di sequenza di sistema . . . . .	17
4.1.3	Contratti delle operazioni . . . . .	17
4.2	Progettazione . . . . .	18
4.2.1	Diagrammi di Interazione . . . . .	18
4.2.2	Diagramma delle Classi . . . . .	20
<b>5</b>	<b>Caso d'uso UC6: Inserisci nuovo paziente</b>	<b>21</b>
5.1	Analisi . . . . .	21
5.1.1	Formato Dettagliato . . . . .	21
5.1.2	Modello di Dominio . . . . .	22
5.1.3	Diagramma di sequenza di sistema . . . . .	23
5.1.4	Contratti delle operazioni . . . . .	24

5.2	Progettazione . . . . .	25
5.2.1	Diagrammi di Interazione . . . . .	25
5.2.2	Diagramma delle Classi . . . . .	28
<b>6</b>	<b>Implementazione</b>	<b>29</b>
6.1	Testing . . . . .	29
6.1.1	TestDiagnostico . . . . .	29

# 1 Introduzione

Conclusa la fase di ideazione, si passa alla fase di elaborazione. Scopo delle iterazioni seguenti sarà quello di: raffinare la Visione, implementare in maniera iterativa il software testandolo ad ogni iterazione, risolvere le problematiche relative ai rischi maggiori, identificare la maggior parte dei requisiti. Durante questa prima iterazione si è deciso di concentrarsi sulle seguenti attività:

- Progettazione dell'UC5: Inserisci nuovo test singolo,
- Progettazione dell'UC4: Inserisci nuovo test diagnostico,
- Progettazione dell'UC2: Inserisci nuova esenzione,
- Progettazione dell'UC6: Inserisci nuova paziente,
- Sviluppo del caso d'uso di avviamento, con la creazione di istanze delle classi identificate,
- Mantenere i dati momentaneamente solo in memoria principale.

L'idea è quella di iniziare la progettazione e l'implementazione del sistema e delle classi in sè che esistono a prescindere dalla prenotazione. I casi d'uso sono abbastanza semplici e simili tra loro, per questo motivo si è pensato di progettarli nella stessa iterazione. Ogni capitolo riguarderà la realizzazione di un caso d'uso nella sua interezza, dalla fase di analisi, a quella di progettazione, fino all'implementazione ed eventuali test.

## 2 Caso d'uso UC5 : Inserisci nuovo test singolo

### 2.1 Analisi

Prima di tutto, scriviamo il caso d'uso in formato dettagliato:

#### 2.1.1 Formato Dettagliato

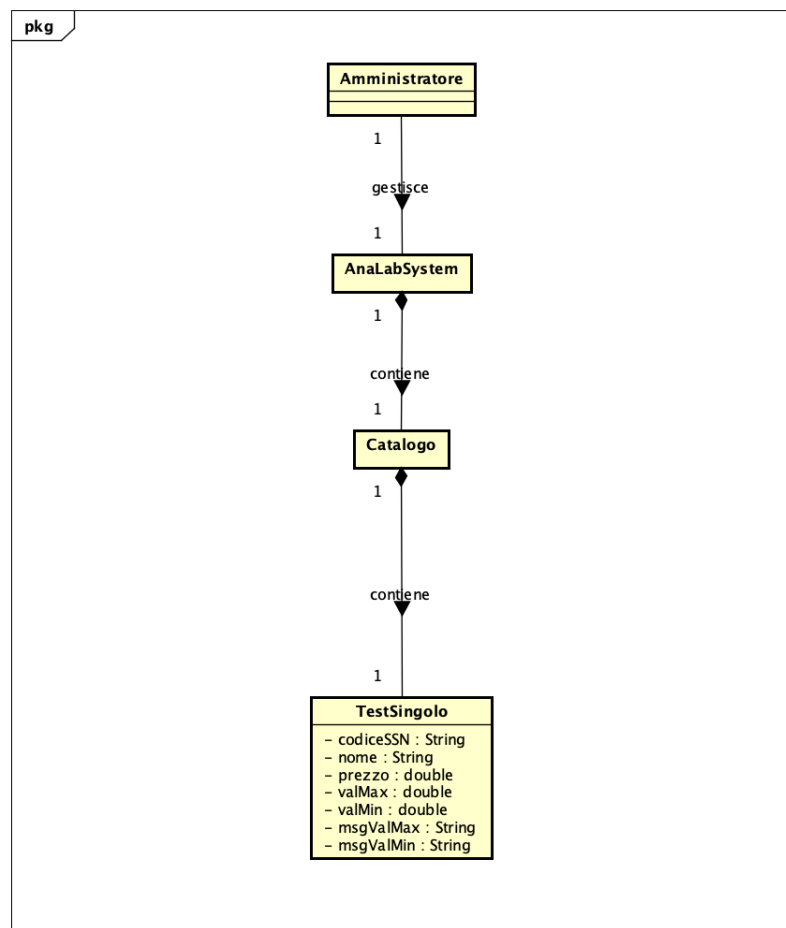
- **Nome:** Inserisci nuovo test diagnostico
- **Livello:** Obiettivo utente
- **Attore primario:** Amministratore
- **Parti interessate e interessi:**
  - **Paziente:** potrà prenotare quel test.
- **Pre-condizioni:**
  - L'amministratore conosce tutte le informazioni relative al test da inserire da inserire.
- **Post-condizioni:**
  - Il test è aggiunto correttamente al catalogo dei test.
- **Scenario principale di successo:**
  1. L'amministratore inizia la procedura di inserimento di un nuovo test diagnostico.
  2. L'amministratore inserisce tutte le informazioni del test: codice, nome, prezzo, valore massimo, valore minimo, messaggio superamento valore massimo, messaggio inferiorità valore minimo.
  3. L'amministratore conferma i dati immessi e il sistema aggiunge il nuovo test alla sezione dei test disponibili nel del Catalogo dei Servizi.
- **Estensioni:**
  -
- **Frequenza di ripetizioni:** Non frequente.

### 2.1.2 Modello di Dominio

Dalla descrizione del caso d'uso in formato dettagliato possiamo identificare le seguenti classi concettuali:

- **Amministratore**: attore primario che interagisce direttamente con il sistema,
- **AnaLabSystem**: il sistema AnaLab,
- **TestSingolo**: un test singolo che è possibile creare e successivamente aggiungere alla prenotazione,
- **Catalogo**: contiene una collezione di tutti i test singoli.

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:



Modello di dominio

Si è deciso di considerare la classe Catalogo che conterrà una collezione di TestSingolo, in modo che non sia AnaLabSystem a dover gestire tale responsabilità, in quanto si prevede che nella progettazione dei successivi casi d'uso AnaLabSystem avrà un numero elevato di responsabilità in quanto è stata scelta come facade controller (vedere diagrammi di interazione).

### 2.1.3 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC5 è il seguente:

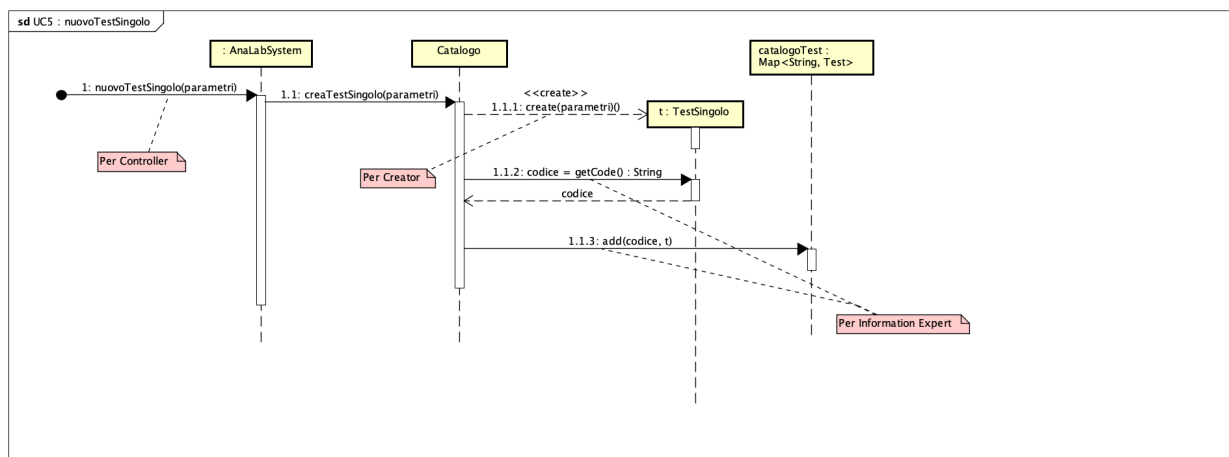


Diagramma di sequenza di sistema UC5

## 2.2 Progettazione

### 2.2.1 Diagrammi di Interazione

**nuovoTestSingolo** Parametri: (*codiceSSN : String, nome : String, prezzo : float, valMax : float, valMin : float, mexValMax : String, mexValMin : String*)



Diagrammi di Interazione: nuovoTestSingolo

## 2.2.2 Diagramma delle Classi

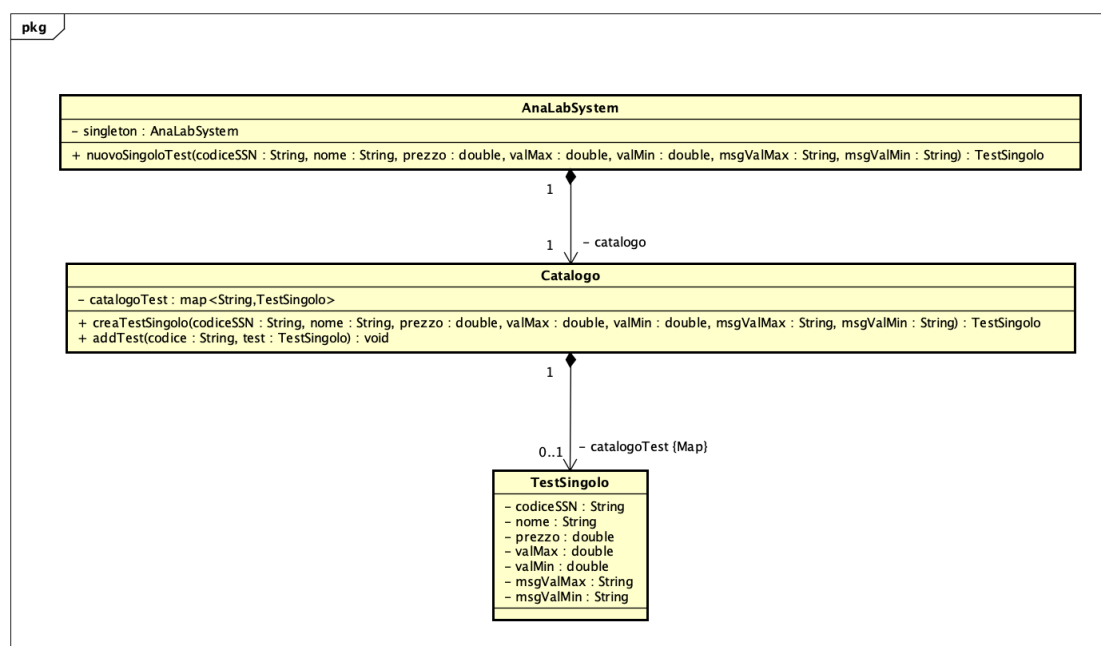


Diagramma delle Classi

E' stato scelto di usare il design pattern Singleton (GoF) per il controller, la classe AnaLabSystem. La scelta è guidata dalla necessita di assicurare che la classe AnaLabSystem abbia un'unica istanza.



## 3 Caso d'uso UC4: Inserisci nuovo test diagnostico

### 3.1 Analisi

Prima di tutto, scriviamo il caso d'uso in formato dettagliato:

#### 3.1.1 Formato Dettagliato

- **Nome:** Inserisci nuovo test diagnostico
- **Livello:** Obiettivo utente
- **Attore primario:** Amministratore
- **Parti interessate e interessi:**
  - **Paziente, Segretario:** potrà prenotare un gruppo di singoli test in un unico momento piuttosto che elencare ogni singolo test (esempio: Emocromo).
- **Pre-condizioni:**
  - L'amministratore conosce tutte le informazioni relative al test diagnostico da inserire.
- **Post-condizioni:**
  - Il test è aggiunto correttamente al catalogo di test.
- **Scenario principale di successo:**
  1. L'amministratore inizia la procedura di inserimento di una nuovo test diagnostico.
  2. L'amministratore inserisce il codice del test e il nome.
  3. L'amministratore indica un test singolo da associare al test diagnostico. Il sistema registra l'informazione.

Il passo 3 si ripete fino a quando l'amministratore indica di aver terminato

  4. Il sistema registra il nuovo test diagnostico e la aggiunge alla sezione dei test offerti del Catalogo dei Servizi.
- **Estensioni:**

– **3a. Un test inserito non esiste:**

1. Il sistema notifica l'amministratore che il test non esiste e non verrà aggiunto all'esenzione.

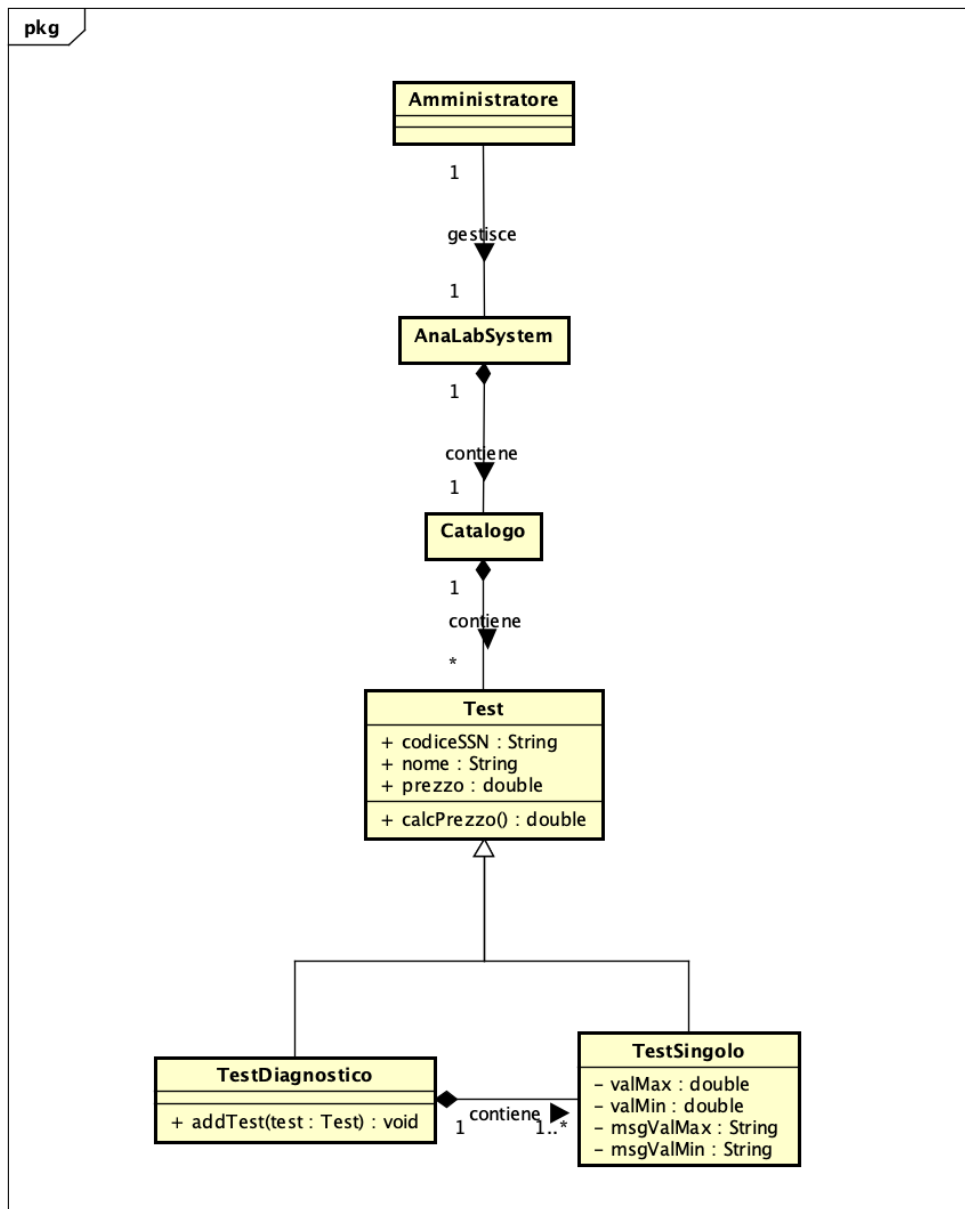
— **Frequenza di ripetizioni:** Non frequente.

### **3.1.2 Modello di Dominio**

Dalla descrizione del caso d'uso in formato dettagliato possiamo identificare le seguente classe concettuale aggiuntiva:

- **Test:** concetto generale di test,
- **TestDiagnostics:** un gruppo di test singoli (TestSingolo) che è possibile creare e successivamente aggiungere alla prenotazione,

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:



Modello di dominio

Si è deciso di generalizzare il concetto di **Test** e creare due diverse specificazioni: **TestSingolo** e **TestDiagnostico**. Sarà la generalizzazione **Test** che verrà (in seguito associata alla prenotazione). Questo è stato supportato dall'idea di utilizzare il pattern Composite nella definizione e gestione di **TestDiagnostico** (vedere la sezione di progettazione di UC4).

### 3.1.3 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC4 'è il seguente:

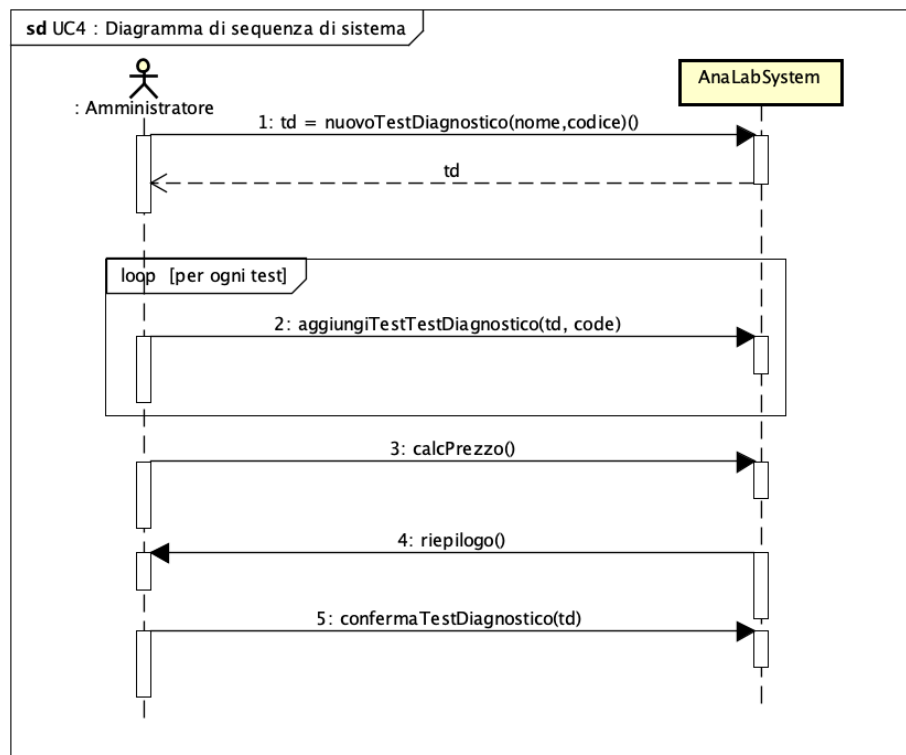


Diagramma di sequenza di sistema UC4

### 3.1.4 Contratti delle operazioni

nuovoTestDiagnostico(codice, nome)

Descrizione:	L'operazione consente di iniziare la creazione di un nuovo gruppo di test singoli.
Riferimenti:	Caso d'uso: UC4 Inserisci nuovo test diagnostico
Pre-condizioni:	-
Post-condizioni:	è stato crea creata una nuova istanza td di TestDiagnostico, gli attributi di td sono stati inizializzati correttamente

aggiungiTestTestDiagnostico(td, codice)

Descrizione:	L'operazione consente di aggiungere un test all'istanza td corrente.
Riferimenti:	Caso d'uso: UC4 Inserisci nuovo gruppo test
Pre-condizioni:	E' in corso la creazione di un nuovo gruppo di test (istanza td di TestDiagnostico)
Post-condizioni:	un test singolo è stato associata all'istanza td corrente tramite l'associazione "contiene"

calcPrezzo()

Descrizione:	L'operazione consente di calcolare il prezzo dell'istanza td corrente in base alla lista corrente di test singoli associati.
Riferimenti:	Caso d'uso: UC4 Inserisci nuovo gruppo test
Pre-condizioni:	E' in corso la creazione di un nuovo gruppo di test (istanza td di TestDiagnostico)
Post-condizioni:	è stato calcolato correttamente il prezzo dell'intero test diagnostico.

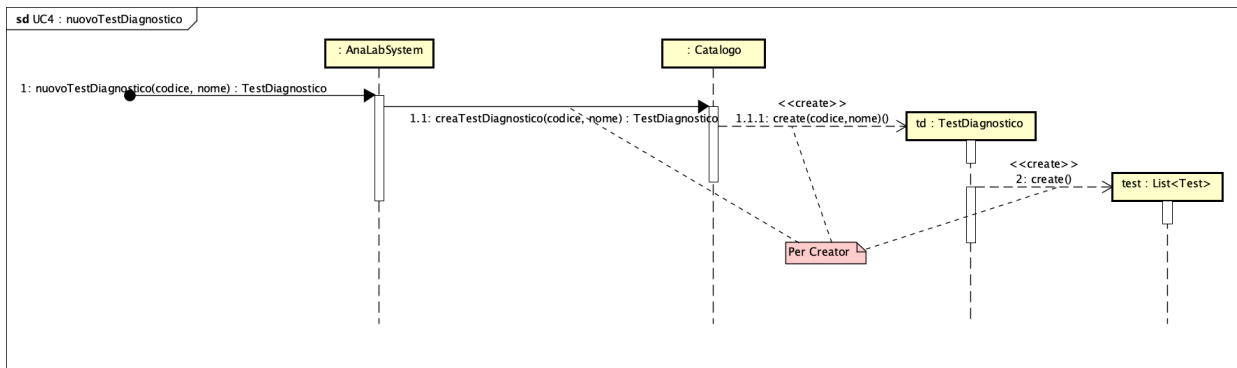
confermaTestDiagnostico(td)

Descrizione:	L'operazione consente di terminare e inserire un nuovo gruppo test nel catalogo dei test al termine dell'inserimento di tutte le informazioni.
Riferimenti:	Caso d'uso: UC4 Inserisci nuovo gruppo test
Pre-condizioni:	E' in corso la creazione di un nuovo gruppo test (istanza td di TestDiagnostico)
Post-condizioni:	è stata correttamente inserita l'istanza corrente td di TestDiagnostico nel catalogo dei test.

## 3.2 Progettazione

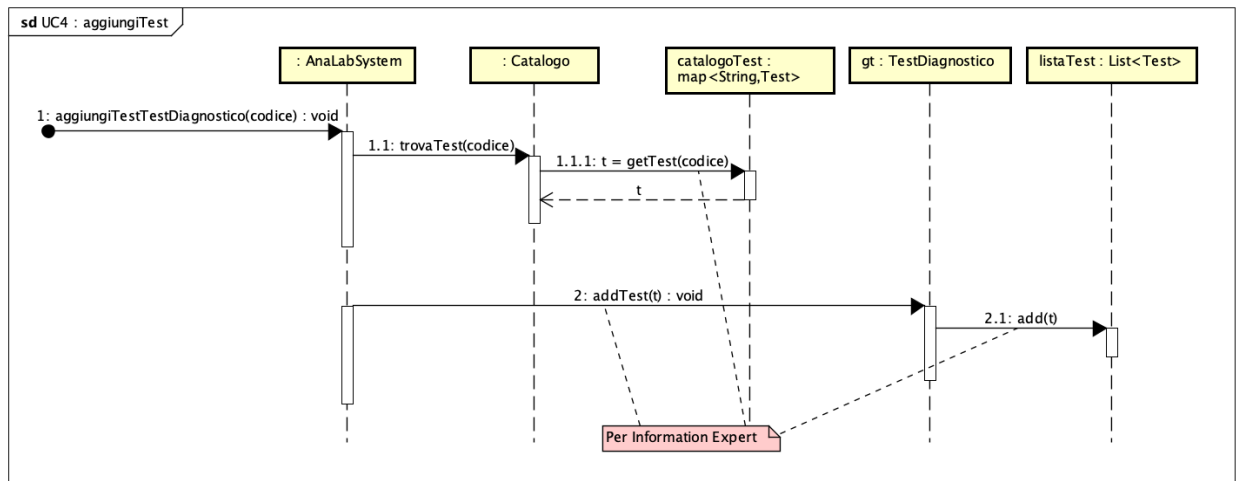
### 3.2.1 Diagrammi di Interazione

**nuovoTestDiagnostico** Parametri: (*codice* : String, *nome* : String)



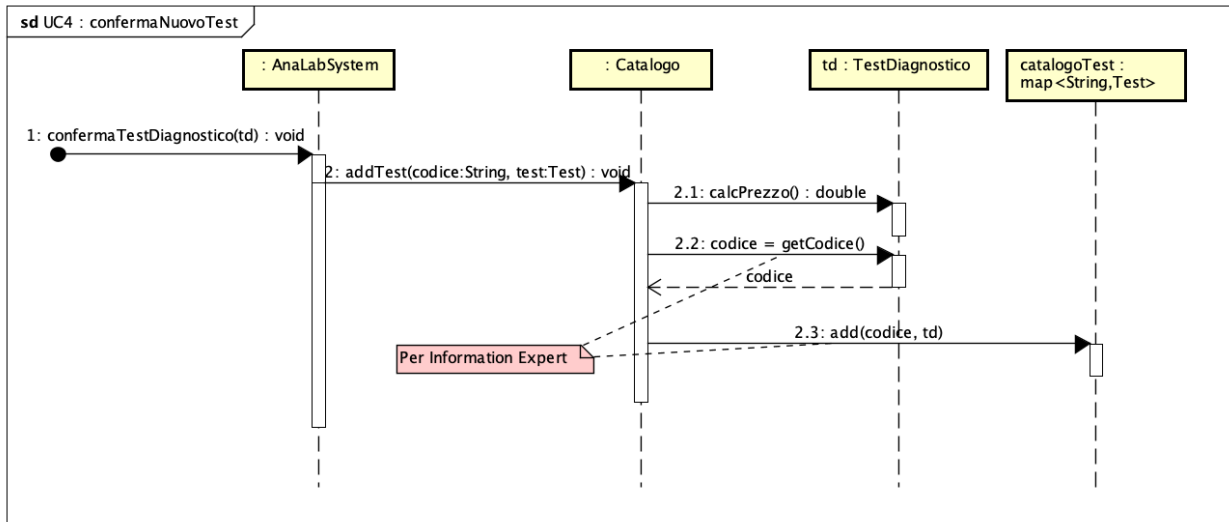
Diagrammi di Interazione: nuovoTestDiagnostico

**aggiungiTest** Parametri: (*codice* : String)



Diagrammi di Interazione: aggiungiTest

**confermaTestDiagnostico** Parametri: ()



Diagrammi di Interazione: confermaTestDiagnostico

### 3.2.2 Diagramma delle Classi

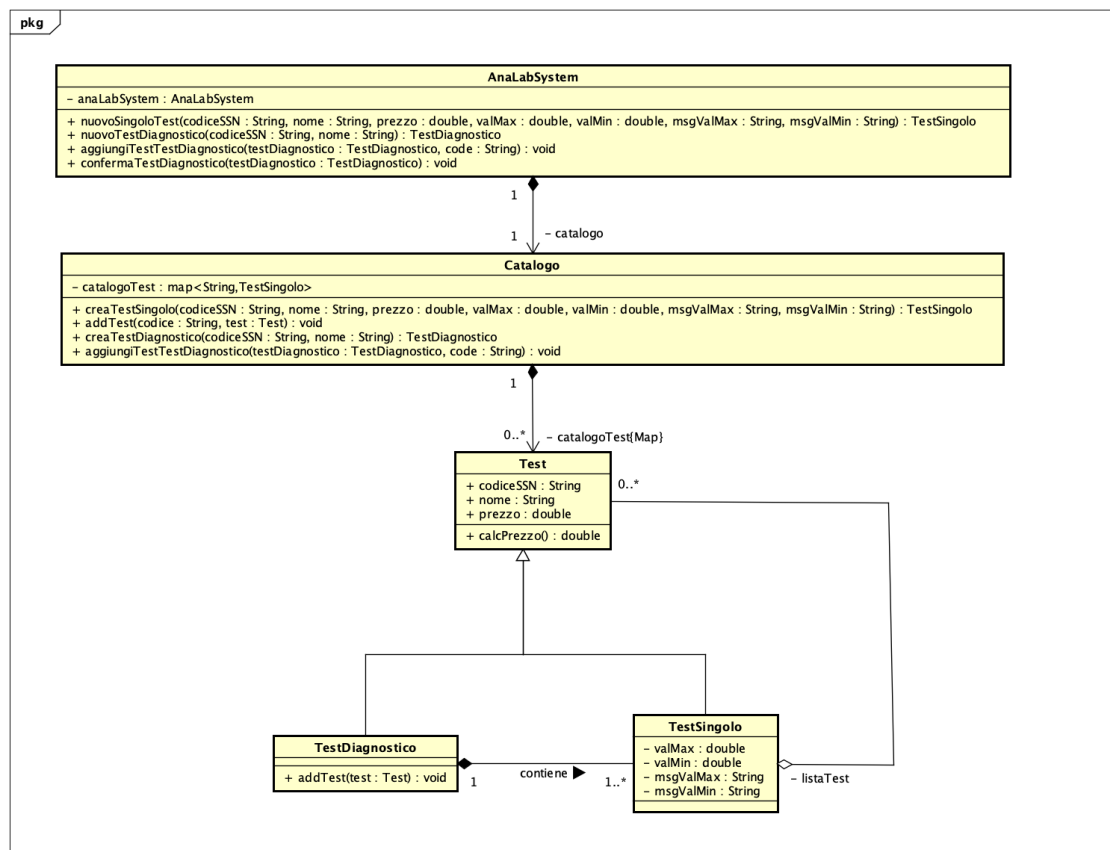


Diagramma delle Classi

E' stato scelto di usare il design pattern Composite (GoF) per gestire la classe TestDiagnostico. Tale scelta è nata dall'esigenza di trattare TestSingolo e TestDiagnostico nella stessa maniera (avere un unico catalogoTest che può contenere istanze di entrambe le classi), si prevede che anche per la gestione della prenotazione tale scelta sarà strategica. E' inoltre apparente l'adeguatezza di tale scelta, dato che TestDiagnostico è caratterizzato da una collezione di TestSingolo.



## 4 Caso d'uso UC2: Inserisci nuova esenzione

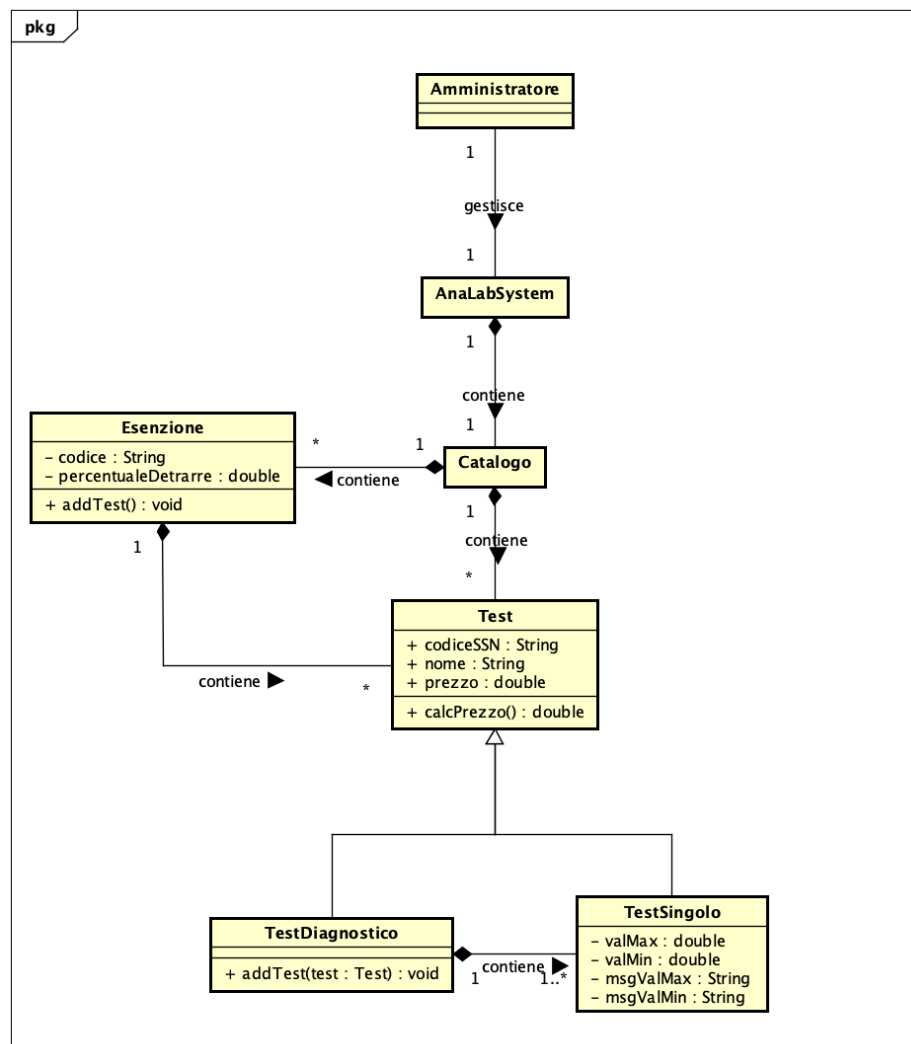
### 4.1 Analisi

#### 4.1.1 Modello di Dominio

Dalla descrizione del caso d'uso in formato dettagliato possiamo identificare le seguente classe concettuale aggiuntiva:

- **Esenzione** : una esenzione che è possibile creare e successivamente associare a un paziente.

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:



Modello di dominio

### 4.1.2 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC2 'e il seguente:

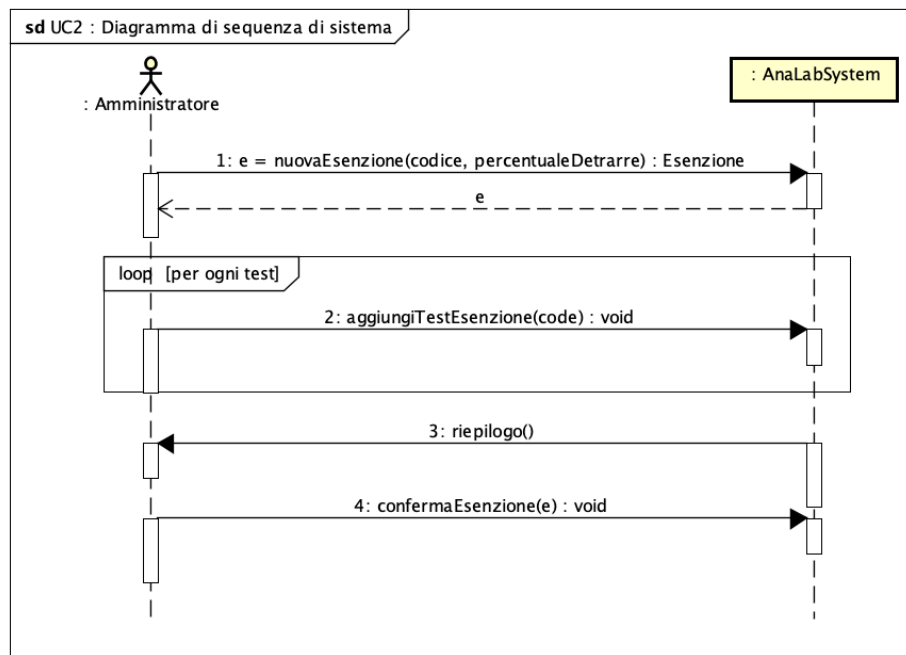


Diagramma di sequenza di sistema UC2

### 4.1.3 Contratti delle operazioni

nuovaEsenzione(codice, percentualeDetrarre)

Descrizione:	L'operazione consente di iniziare la creazione di una nuova esenzione.
Riferimenti:	Caso d'uso: UC2 Inserisci nuova esenzione
Pre-condizioni:	-
Post-condizioni:	è stato crea creata una nuova istanza e di Esenzione, gli attributi di e sono stati inizializzati correttamente

aggiungiTest(e, code)

Descrizione:	L'operazione consente di aggiungere un test all'istanza e corrente.
Riferimenti:	Caso d'uso: UC2 Inserisci nuova esenzione
Pre-condizioni:	E' in corso la creazione di una nuova esenzione (istanza e di Esenzione)
Post-condizioni:	un test è stata associata all'istanza e tramite l'associazione "contiene"

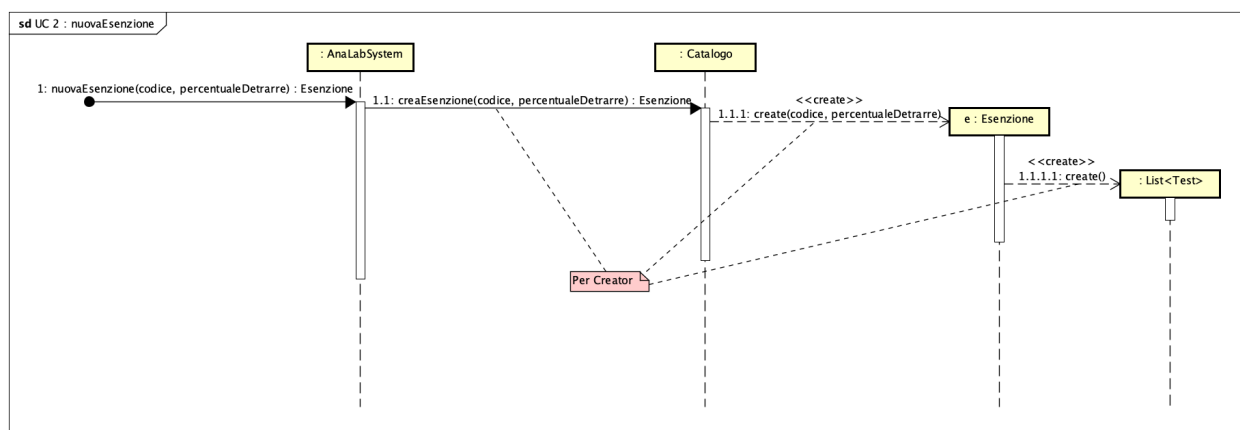
confermaNuovaEsenzione(e)

Descrizione:	L'operazione consente di terminare e inserire una nuova esenzione nel catalogo delle esenzioni al termine dell'inserimento di tutte le informazioni.
Riferimenti:	Caso d'uso: UC2 Inserisci nuova esenzione
Pre-condizioni:	E' in corso la creazione di una nuova esenzione (istanza e di Esenzione)
Post-condizioni:	è stata correttamente inserita l'istanza corrente e di Esenzione nel catalogo delle esenzioni.

## 4.2 Progettazione

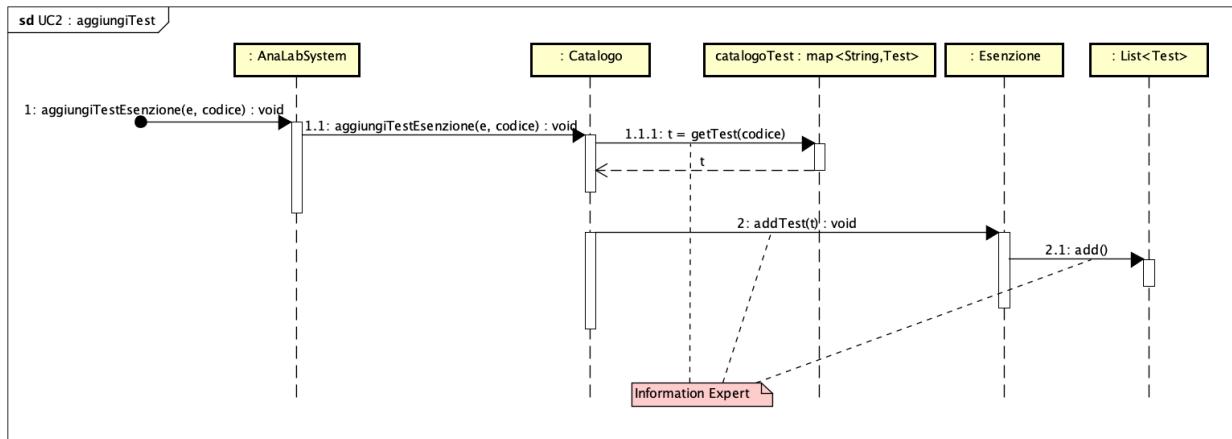
### 4.2.1 Diagrammi di Interazione

**nuovaEsenzione** Parametri: (*codice* : String, *nome* : String, *percentualeDetrarre* : double)



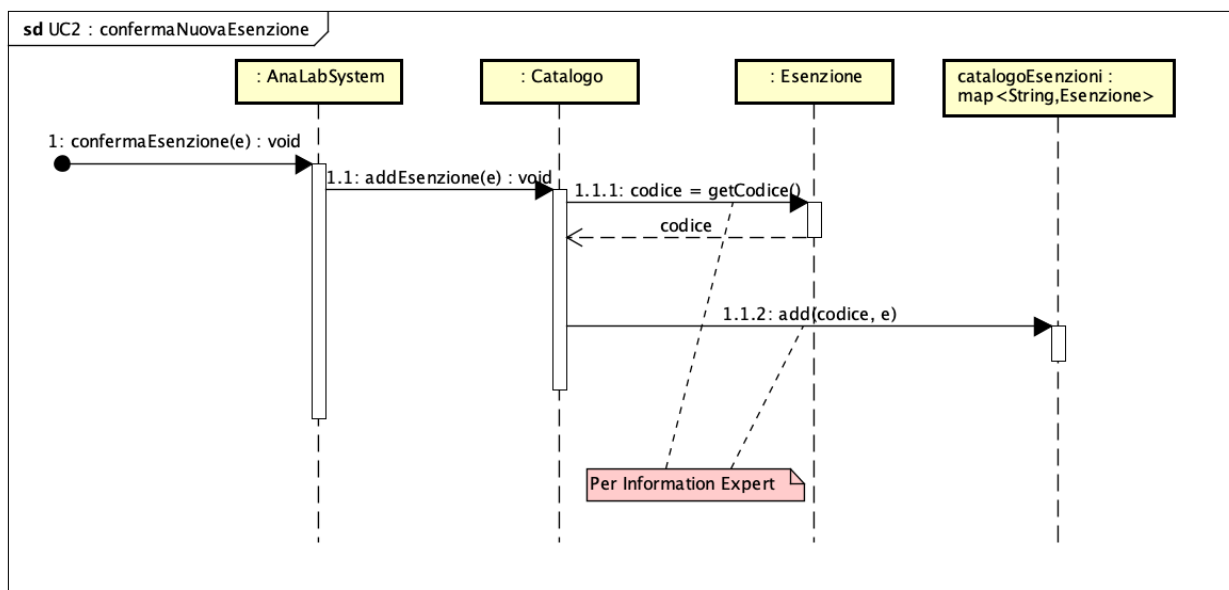
Diagrammi di Interazione: nuovaEsenzione

**aggiungiTest** Parametri: (*Esenzione e*, *String code*)



Diagrammi di Interazione: aggiungiTest

**confermaNuovaEsenzione** Parametri: ()



Diagrammi di Interazione: confermaNuovaEsenzione

## 4.2.2 Diagramma delle Classi

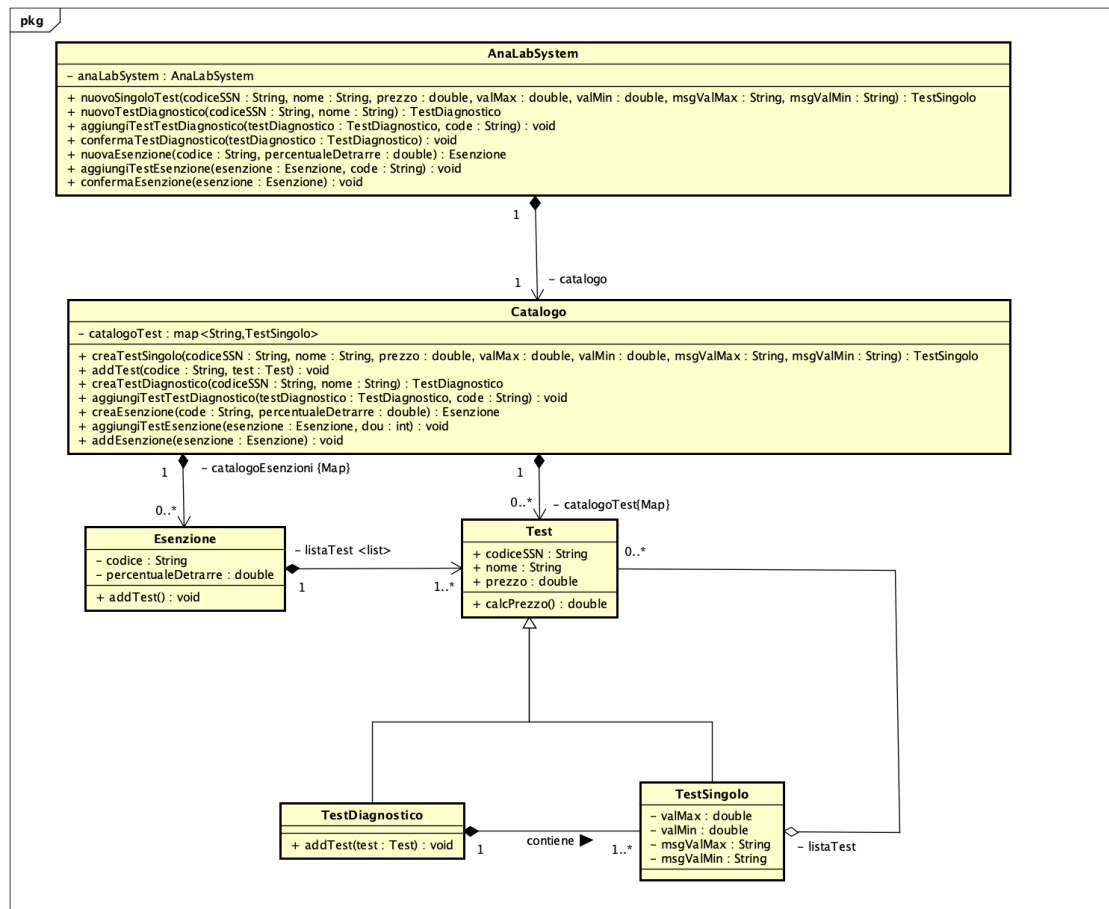


Diagramma delle Classi

## 5 Caso d'uso UC6: Inserisci nuovo paziente

### 5.1 Analisi

Prima di tutto, scriviamo il caso d'uso in formato dettagliato:

#### 5.1.1 Formato Dettagliato

- **Nome:** Inserisci nuovo paziente
- **Livello:** Obiettivo utente
- **Attore primario:** Segretario
- **Parti interessate e interessi:**
  - **Paziente:** potrà effettuare nuove prenotazioni in futuro senza dover ripetere le sue informazioni anagrafiche; potrà inoltre visualizzare l'esito dei test da lui prenotati, i quali saranno associati alle sue informazioni anagrafiche.
- **Pre-condizioni:**
  - Il paziente comunica al segretario le sue informazioni anagrafiche.
- **Post-condizioni:**
  - Il paziente è stato registrato correttamente nel sistema.
- **Scenario principale di successo:**
  1. Il segretario inizia la procedura di inserimento di un nuovo paziente.
  2. Il segretario inserisce tutte le informazioni del paziente: codice fiscale, nome, cognome.
  3. Il paziente comunica una sua esenzione. Il segretario registra l'informazione.  
Il passo 3 si ripete fino a quando il paziente indica di non avere altre esenzioni.
  4. Il sistema registra il nuovo cliente nel sistema.
- **Estensioni:**
  - **3a. Un'esenzione inserita non esiste:**
    1. Il sistema notifica il segretario che l'esenzione non è presente nel sistema.

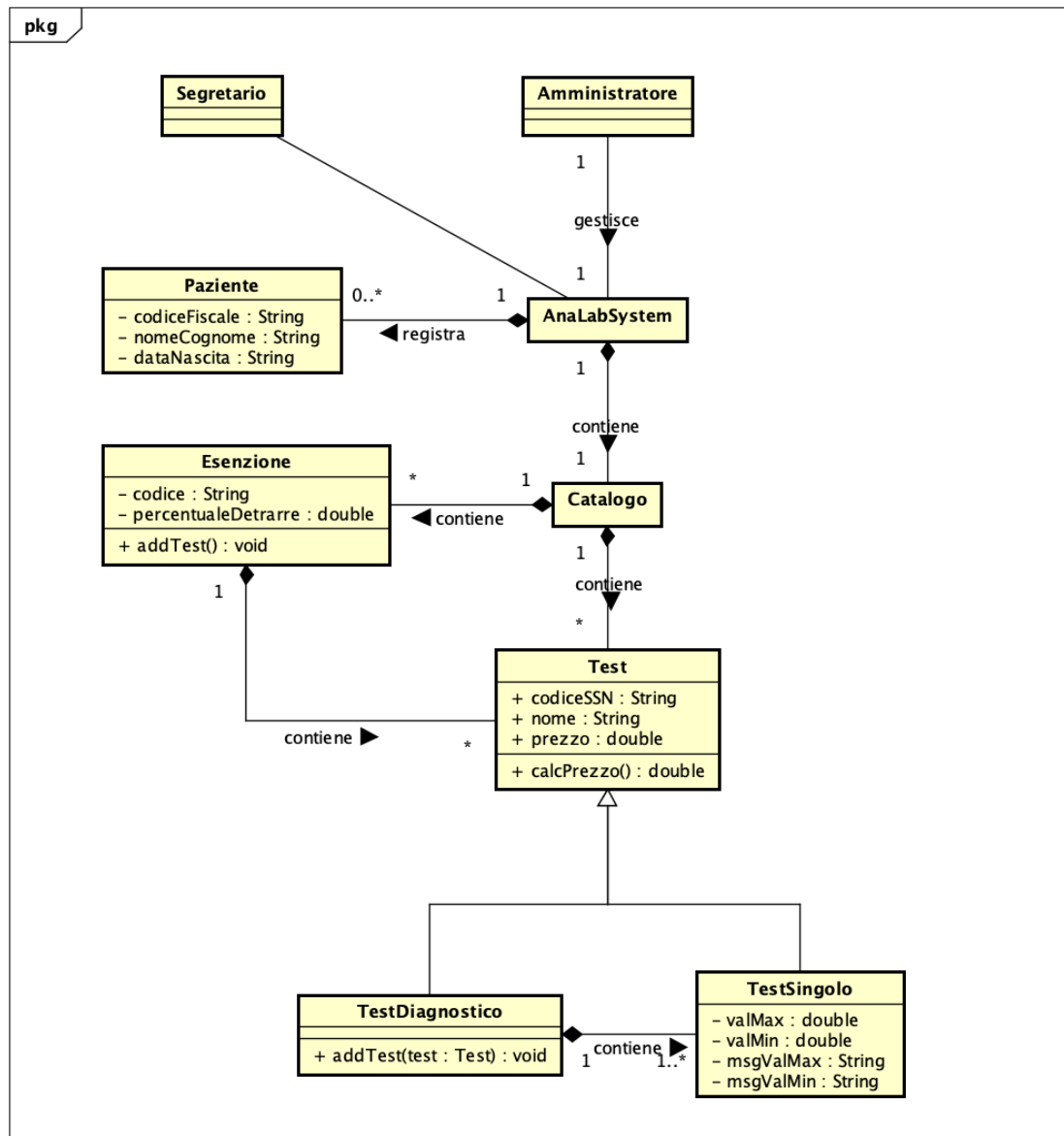
- **Frequenza di ripetizioni:** Frequente, soprattutto inizialmente, quando non ci sono molti pazienti registrati.

### 5.1.2 Modello di Dominio

Dalla descrizione del caso d'uso in formato dettagliato possiamo identificare le seguente classe concettuale aggiuntiva:

- **Segretario:** attore primario che interagisce direttamente con il sistema per registrare il paziente.
- **Paziente:** un paziente che è possibile registrare e successivamente associare a una prenotazione.

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:



Modello di dominio

### 5.1.3 Diagramma di sequenza di sistema

Il diagramma di sequenza di sistema per il caso d'uso UC6 è il seguente:



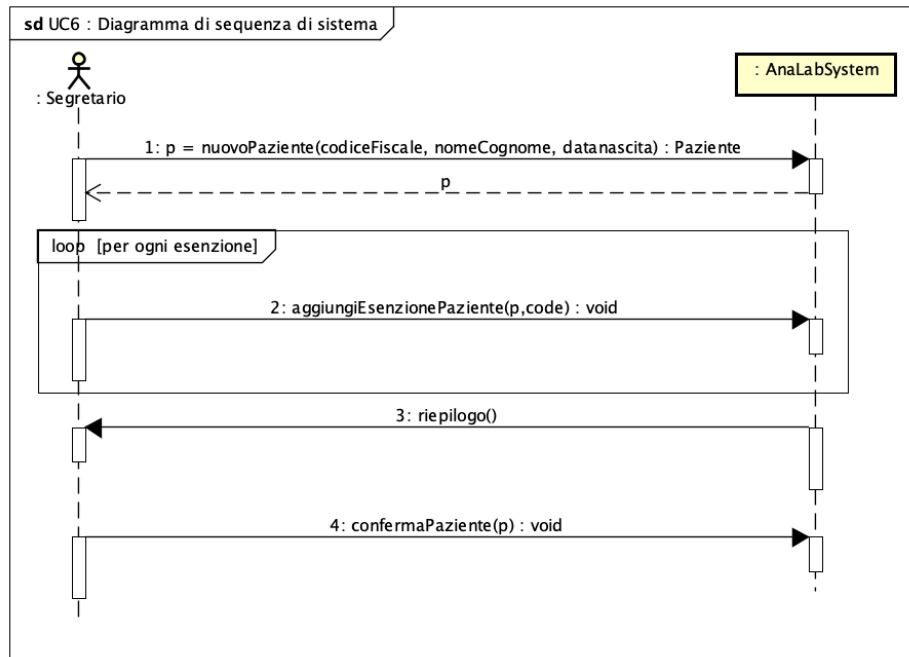


Diagramma di sequenza di sistema UC6

#### 5.1.4 Contratti delle operazioni

nuovoPaziente(codiceFiscale, nomeCognome, dataNascita)

Descrizione:	L'operazione consente di iniziare la registrazione di un nuovo paziente.
Riferimenti:	Caso d'uso: UC6 Inserisci nuovo paziente
Pre-condizioni:	-
Post-condizioni:	è stato creata una nuova istanza p di Paziente, gli attributi di tipo numerico e String di p sono stati inizializzati

### aggiungiEsenzionePaziente(p, codice)

Descrizione:	L'operazione consente di aggiungere un'esenzione all'istanza pat corrente (in fase di registrazione).
Riferimenti:	Caso d'uso: UC6 Inserisci nuovo paziente
Pre-condizioni:	E' in corso la registrazione di un nuovo utente (istanza p di Paziente)
Post-condizioni:	un'esenzione è stata associata all'istanza pat tramite l'associazione "detiene"

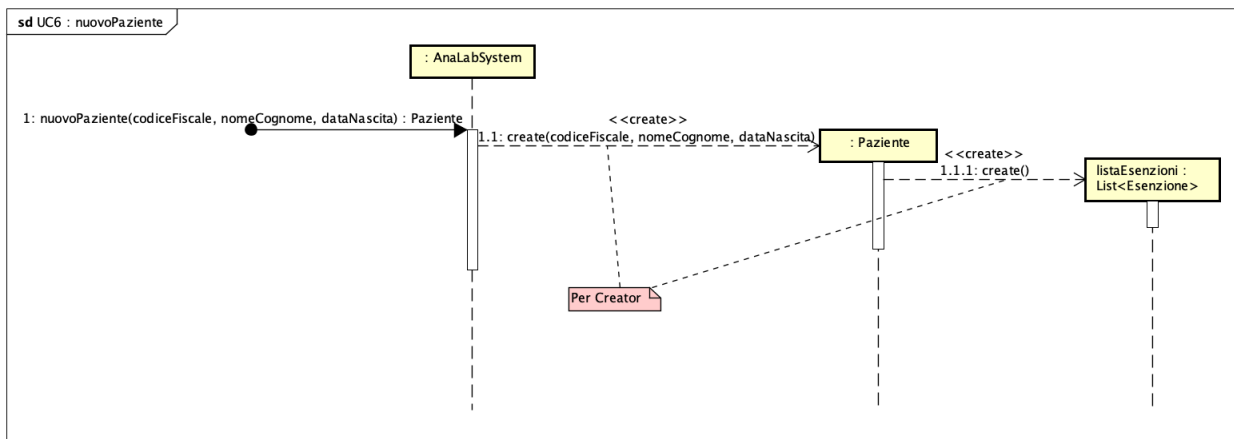
### confermaPaziente(p)

Descrizione:	L'operazione consente di terminare e registrare un nuovo paziente al termine dell'inserimento di tutte le sue informazioni ed esenzioni.
Riferimenti:	Caso d'uso: UC6 Inserisci nuovo paziente
Pre-condizioni:	E' in corso la registrazione di un nuovo paziente (istanza p di Paziente)
Post-condizioni:	è stata correttamente registrata l'istanza corrente p di Paziente nel sistema AnaLab

## 5.2 Progettazione

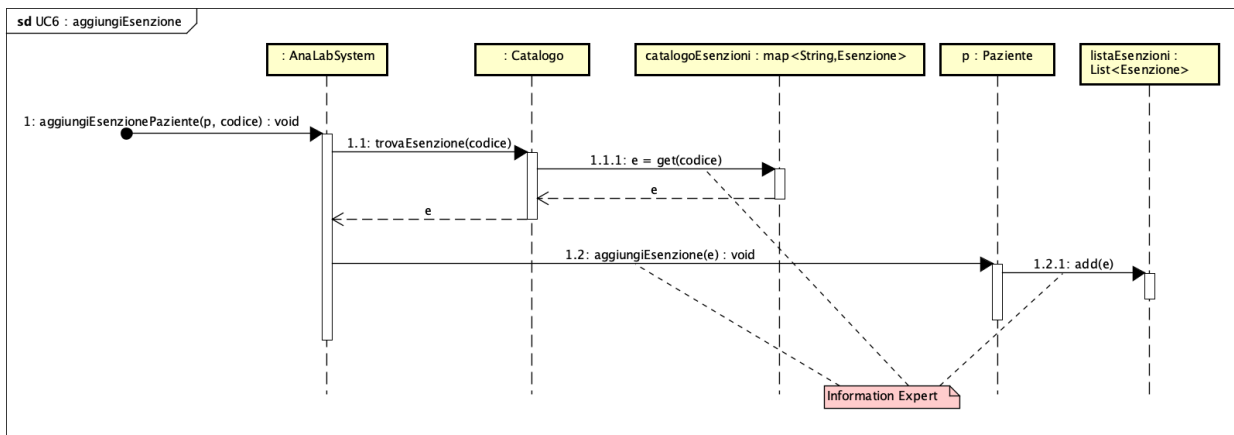
### 5.2.1 Diagrammi di Interazione

**nuovoPaziente** Parametri: (*codiceFiscale: String, nomeCognome: String, dataNascita: String*)



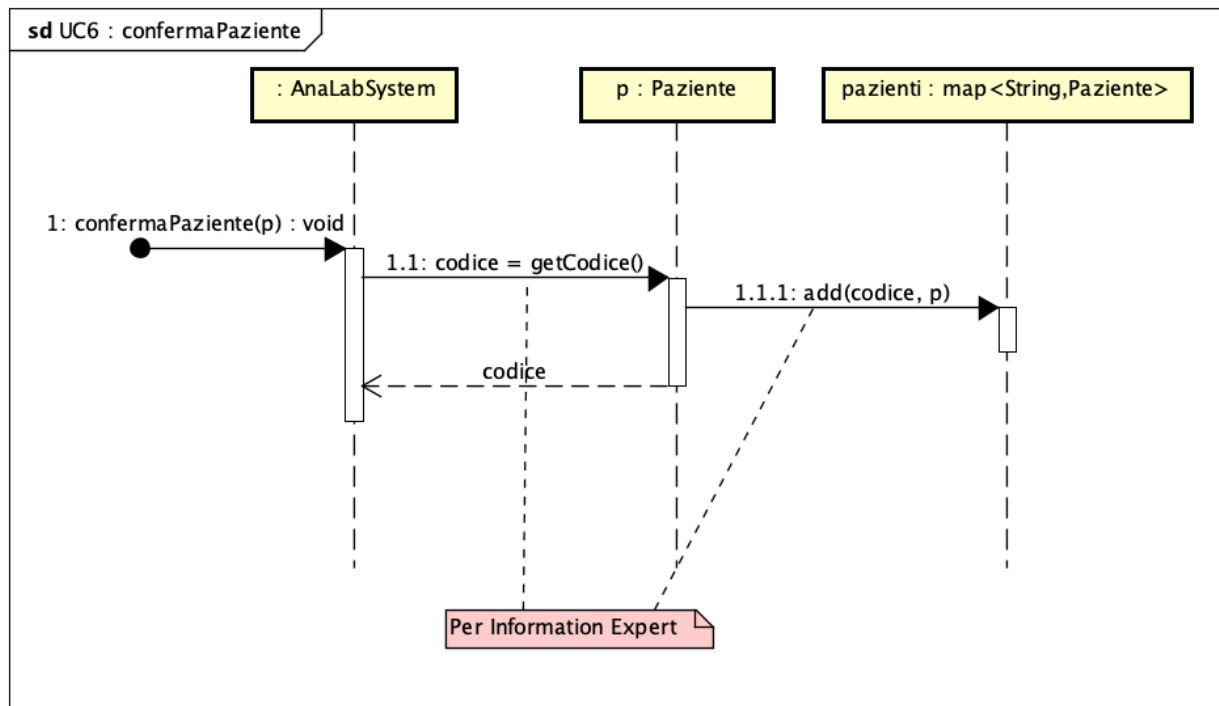
Diagrammi di Interazione: nuovoPaziente

**aggiungiEsenzione** Parametri: ( $p$  : Paziente, codice : String)



Diagrammi di Interazione: aggiungiEsenzione

**confermaPaziente** Parametri: ( $p$  : Paziente)



Diagrammi di Interazione: confermaPaziente

## 5.2.2 Diagramma delle Classi

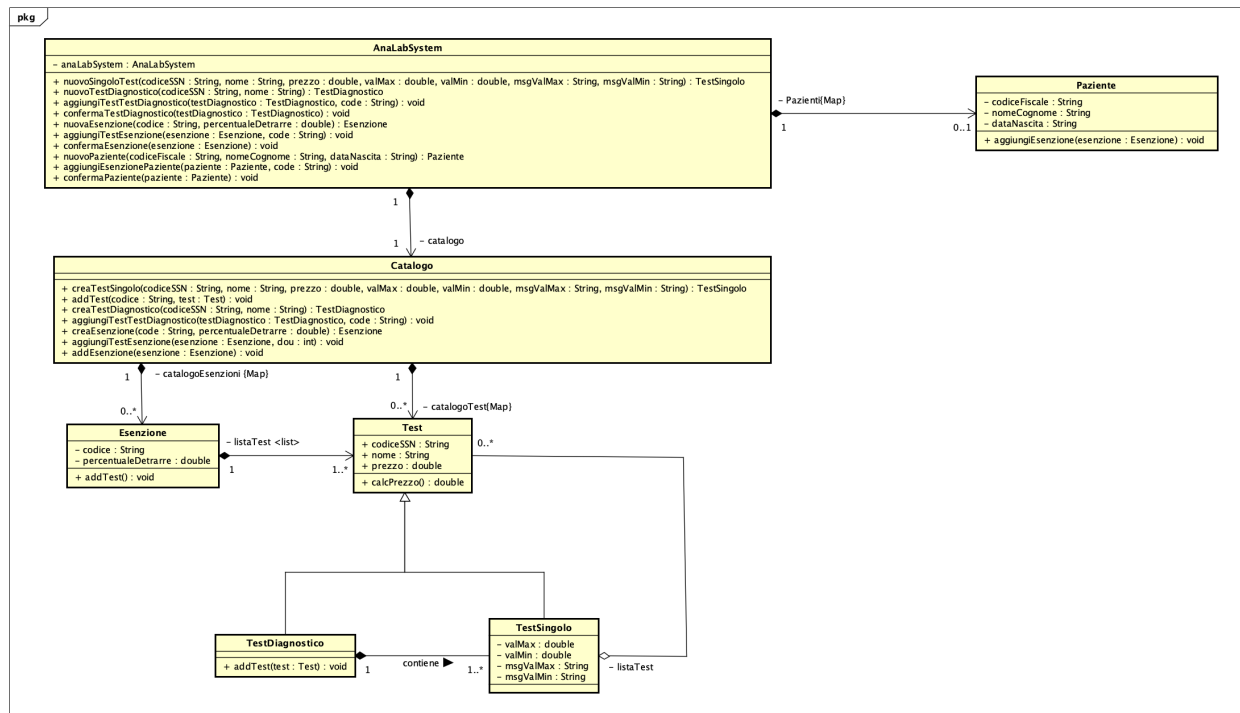


Diagramma delle Classi

## 6 Implementazione

Tutti i casi d'uso sono stati implementati e alcuni brevi test per valutare i metodi con una più alta probabilità di errore.

### 6.1 Testing

Si è scelto di implementare test a scatola nera, testando in particolare i metodi che calcolano i prezzi, i quali possono presentare le maggiori criticità. L'approccio utilizzato in questa iterazione è di bottom-up; si testano, quindi, i metodi più interni al sistema (metodi che operano solo sugli attributi pubblici e privati della classe stessa).

#### 6.1.1 TestDiagnostico

E' stato testato il metodo di *TestDiagnosticoTest testCalcPrezzo* che permette di calcolare il prezzo di un test diagnostico, sommando i prezzi dei singoli test in esso contenuti. E' stato creato un nuovo test diagnostico e si verifica che il prezzo sia quello aspettato.