

# Iterazione 1

## 1.1. Introduzione

Passiamo alla prima fase di elaborazione. Lo scopo delle iterazioni seguenti sarà quello di:

- Raffinare la visione.
- Implementare in maniera iterativa il nucleo dell'architettura del software.
- Risolvere le problematiche relative ai rischi maggiori.
- Identificare la maggior parte dei requisiti e la portata.
- Fornire delle stime più realistiche del piano di lavoro e delle risorse complessive.

Durante questa prima iterazione i requisiti scelti su cui concentrarsi sono i seguenti:

- Implementare gli scenari di successo dei primi casi d'uso:
  - UC2: iscrizione ad un corso di studio.
  - UC5: caricamento di contenuti.
  - UC6: caricamento di appunti.
- Implementare i casi d'uso di startup per gestire l'inizializzazione di questa specifica iterazione.

### 1.1.1 Modello dei casi d'uso

#### UC2: ricerca ed iscrizione ad un corso

<b>Nome</b>	UC2: ricerca di un corso studio
<b>Portata</b>	Gestione sistema di StudyHub
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Studente
<b>Parti interessate e interessi</b>	<p>Studente alunno: vuole avere la possibilità di iscriversi ad un corso di studio dal quale ricavare documenti di studio e lezioni.</p> <p>Studente docente: fornisce un corso a cui iscriversi (eventualmente dietro compenso) e desidera avere l'elenco degli studenti iscritti e condividere documenti e lezioni con loro</p>
<b>Pre-condizioni</b>	Lo studente è autenticato nel sistema. Il corso deve esistere.
<b>Garanzia di successo</b>	Lo studente è iscritto al corso selezionato. Il sistema registra lo studente.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'utente accede a StudyHub e seleziona "Ricerca corso".</li><li>2. Una volta inseriti i dati relativi al corso stesso, il sistema mostra i corsi disponibili sulla base delle informazioni inserite.</li><li>3. Il cliente seleziona il corso e, se ha un costo, viene eseguito un pagamento, altrimenti viene direttamente iscritto.</li></ol>

	4. Il sistema registra lo studente inserito, sia lato "Studente" che lato "Corso".
<b>Estensioni</b>	<ol style="list-style-type: none"> <li>1. In qualsiasi momento, il sistema fallisce: il sistema ricostruisce lo stato precedente, rileva le anomalie e mostra l'errore allo studente. Dopo la risoluzione, lo studente può ritentare a iscriversi.</li> <li>2. Lo studente non ha dei dati di pagamento validi, quindi annulla l'iscrizione.</li> </ol>
<b>Requisiti speciali</b>	Interfaccia intuitiva per la selezione e l'iscrizione. Corsi già presenti nel sistema.
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Poche volte al giorno.
<b>Varie</b>	

### **UC5: caricamento contenuti in un corso**

<b>Nome</b>	UC5: caricamento contenuti di un corso
<b>Portata</b>	Gestione sistema di StudyHub
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Studente
<b>Parti interessate e interessi</b>	Studente docente: vuole inserire i suoi contenuti in un corso per condividerli con gli altri studenti del corso.
<b>Pre-condizioni</b>	Lo studente è autenticato nel sistema. Il corso deve esistere. Lo studente è il creatore del corso.
<b>Garanzia di successo</b>	Lo studente torna al menù principale. Il sistema registra il contenuto nel corso.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"> <li>1. Lo studente accede a StudyHub e seleziona "Carica contenuto".</li> <li>2. Lo studente seleziona il corso al quale caricare il contenuto tra quelli che ha creato.</li> <li>3. Lo studente inserisce i dati del contenuto.</li> <li>4. Il sistema registra il contenuto all'interno del corso e riporta lo studente al menù principale con un messaggio di conferma.</li> </ol>
<b>Estensioni</b>	<ol style="list-style-type: none"> <li>1. In qualsiasi momento, il sistema fallisce: il sistema ricostruisce lo stato precedente, rileva le anomalie e mostra l'errore allo studente. Dopo la risoluzione, lo studente può ritentare a caricare i contenuti.</li> <li>2. Lo studente non ha dei corsi creati, quindi annulla il pagamento.</li> </ol>
<b>Requisiti speciali</b>	Interfaccia intuitiva per il caricamento del contenuto.
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Poche volte al giorno.
<b>Varie</b>	

### **UC6: caricamento appunti**

<b>Nome</b>	UC6: caricamento appunti
<b>Portata</b>	Gestione sistema di StudyHub
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Studente
<b>Parti interessate e interessi</b>	Studente: vuole inserire gli appunti personali nel sistema per un'eventuale condivisione.
<b>Pre-condizioni</b>	Lo studente è autenticato nel sistema.
<b>Garanzia di successo</b>	Il sistema registra l'appunto tra quelli dello studente e nella lista di tutti gli appunti.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"> <li>1. Lo studente accede a StudyHub e seleziona "Carica appunto".</li> <li>2. Lo studente inserisce i dati dell'appunto.</li> <li>3. Il sistema registra l'appunto all'interno della lista degli appunti dello studente.</li> </ol>
<b>Estensioni</b>	<ol style="list-style-type: none"> <li>1. In qualsiasi momento, il sistema fallisce: il sistema ricostruisce lo stato precedente, rileva le anomalie e mostra l'errore allo studente. Dopo la risoluzione, lo studente può ritentare a caricare l'appunto.</li> </ol>
<b>Requisiti speciali</b>	Interfaccia intuitiva per il caricamento degli appunti.
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Poche volte al giorno.
<b>Varie</b>	

## 1.2 Analisi orientata agli oggetti

L'analisi orientata agli oggetti si basa sulla creazione di una descrizione del dominio da un punto di vista ad oggetti. Vengono utilizzati diversi diagrammi e modelli per fornire questa descrizione: il modello di dominio, gli SSD (Sequence System Diagram) e i contratti delle operazioni.

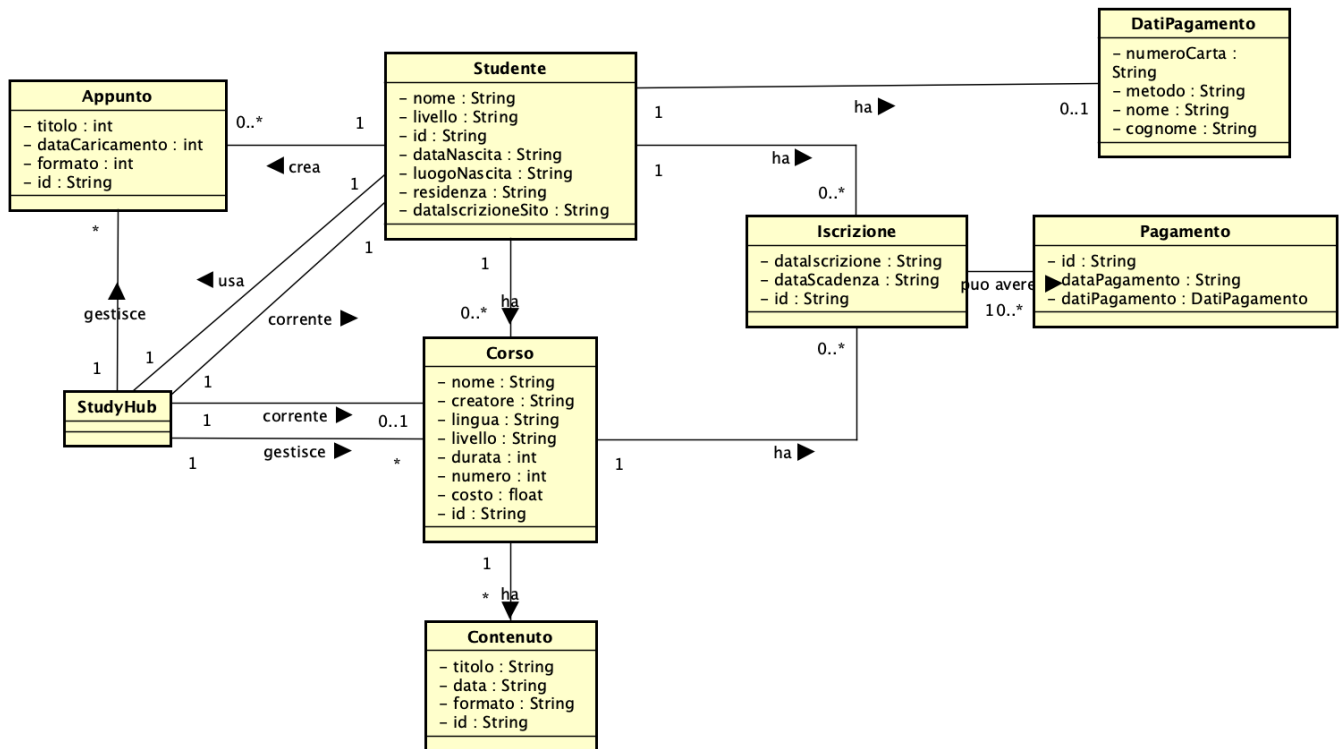
### 1.2.1 Modello di dominio

Il modello di dominio è l'elaborato nel quale vengono identificati i concetti (gli oggetti), i relativi attributi e le associazioni considerati significativi nel dominio preso in esame. Sono state identificate le seguenti classi concettuali:

- StudyHub: il Singleton che rappresenta l'intera applicazione, il sistema complessivo che gestisce ed esegue materialmente tutte le funzioni;
- Studente: il cliente della nostra applicazione, gestisce il suo profilo, i suoi appunti e i suoi corsi, che ha creato o a cui si è iscritto;
- Appunto: documento di diversi formati appartenente ad un singolo studente;
- Corso: rappresenta un canale attraverso cui condividere contenuti (PDF, lezioni) ad una platea di studenti iscritti, è creato a sua volta da uno studente, senza differenziazione di ruoli e privilegi;
- Contenuto: documento di diversi formati caricato dal creatore del corso nel corso stesso, disponibile per tutti gli studenti iscritti;

- Iscrizione: collegamento tra un corso ed uno studente che può iscriversi a più corsi, concretizzato in un oggetto per tenere traccia delle date e perché ad esso sono collegati i pagamenti;
- Pagamento: richiesto da alcuni corsi per completare l'iscrizione di uno studente;
- DatiPagamento: dati del metodo scelto per completare il pagamento;
- Contenuto: documento di diversi formati (PDF, lezione) caricato dal creatore del corso per condividerlo con tutti gli studenti iscritti.

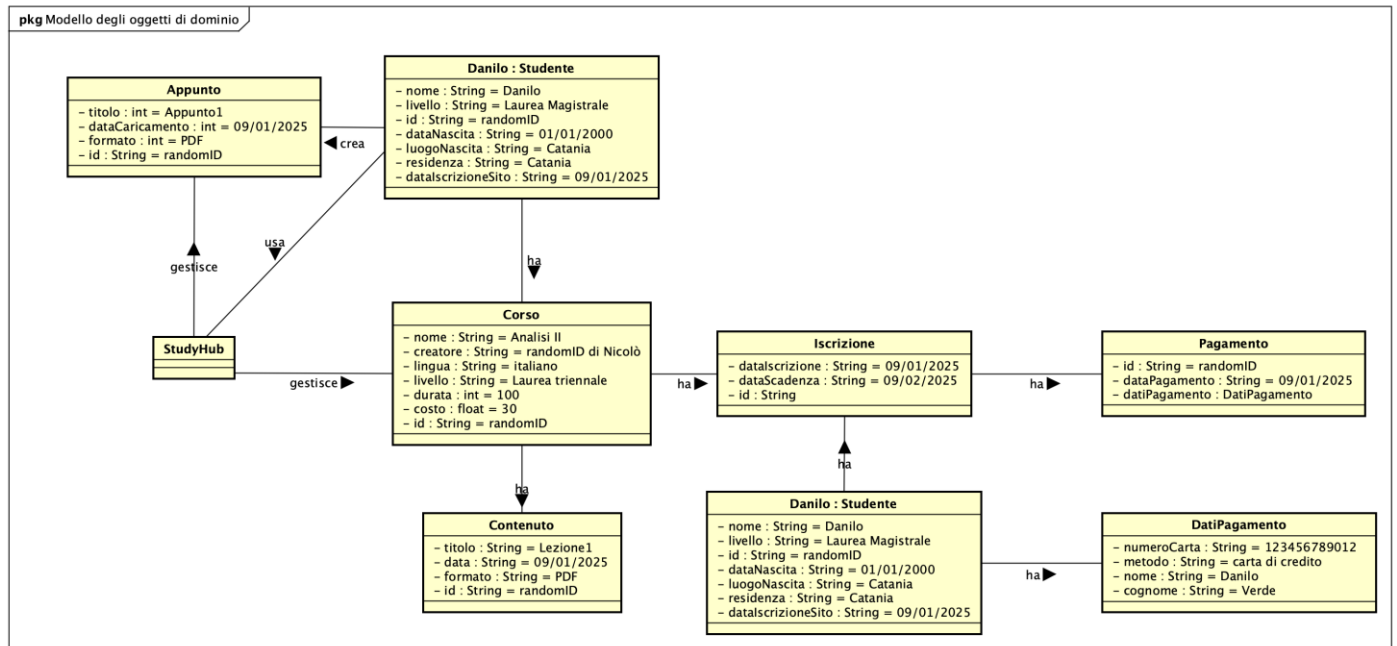
Questi oggetti, con attributi e associazioni, formano il seguente modello di dominio:



### 1.2.2 Modello degli oggetti di dominio

Il modello degli oggetti di dominio dà una visione schematica degli oggetti del dominio e aiuta a visualizzare la struttura dei dati, dando una panoramica delle entità coinvolte.

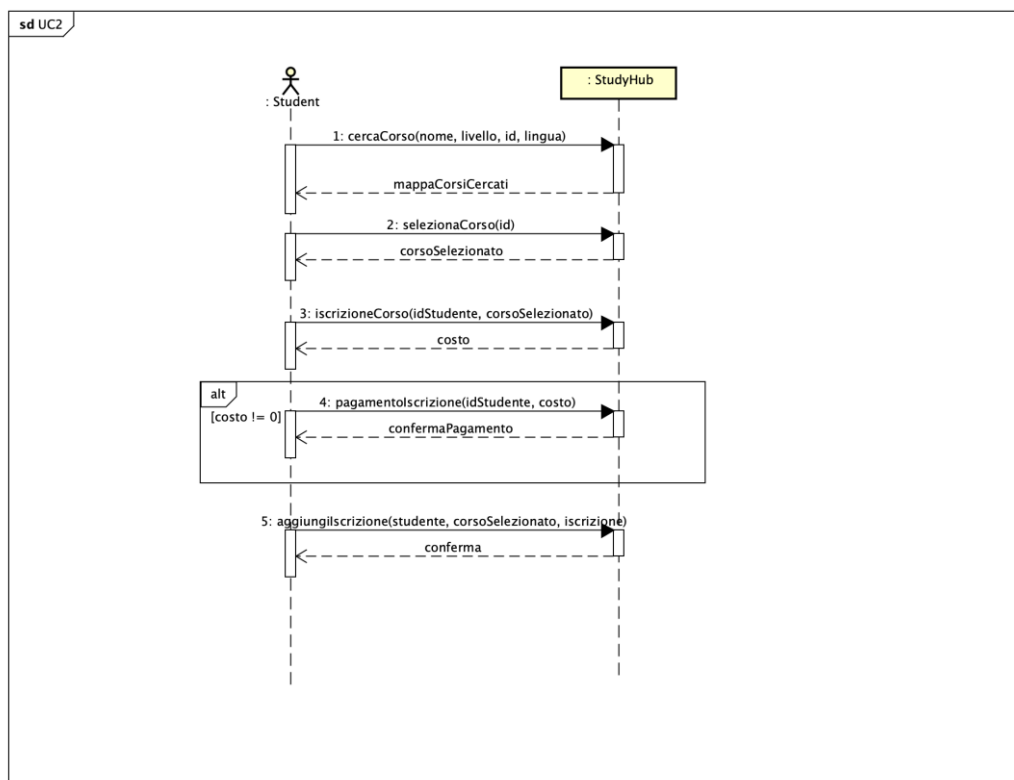
Questo è un esempio in cui ci sono due studenti, uno che crea un corso e uno che è iscritto:



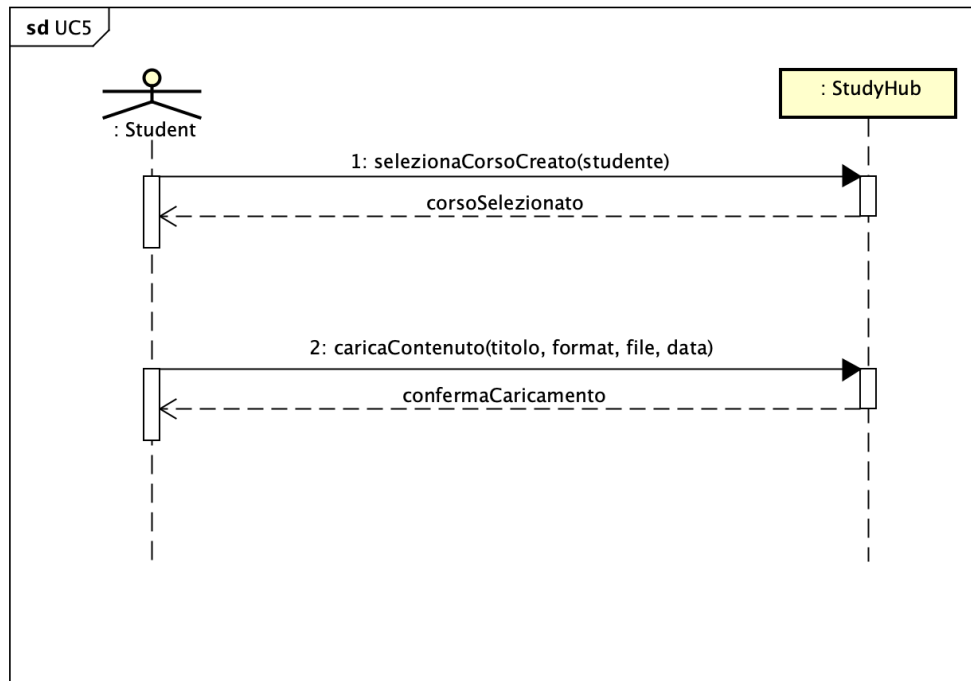
### 1.2.3 Diagrammi di sequenza di sistema

Oltre ai due modelli sopra l'analisi orientata agli oggetti prevede anche i diagrammi di sequenza di sistema (SSD) che illustrano il susseguirsi degli eventi nei casi d'uso.

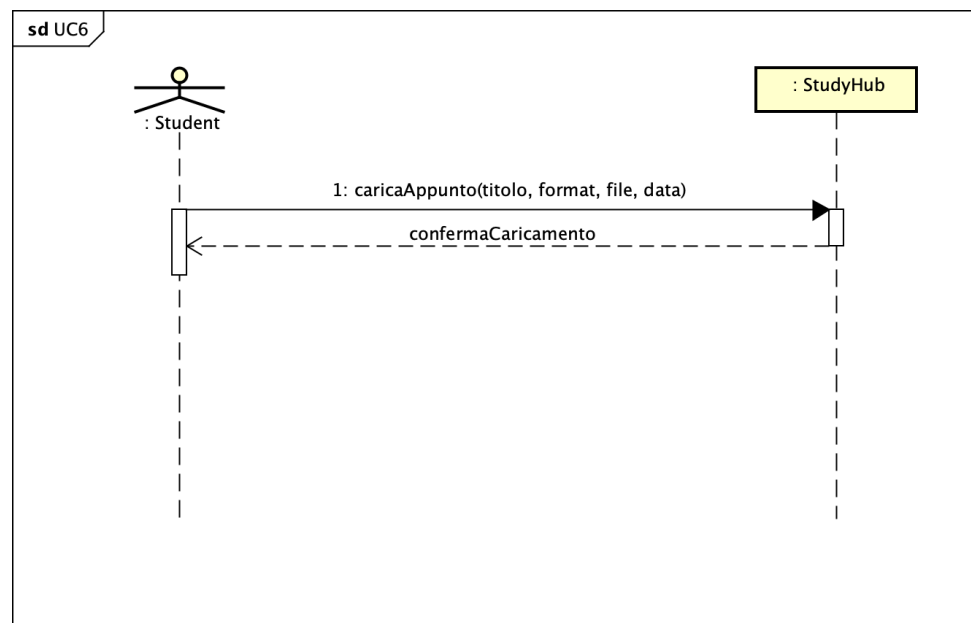
#### 1.2.3.1 SSD Caso d'uso UC2



#### 1.2.3.2 SSD Caso d'uso UC5



### 1.2.3.3 SSD Caso d'uso UC6



### 1.2.4 Contratti delle operazioni

Infine, i contratti descrivono le principali operazioni di sistema che gestiscono gli eventi.

#### **1.2.4.1 Contratti Operazioni UC2**

Contratto CO1: selezionaCorso

<b>Operazioni</b>	selezionaCorso(id : string)
<b>Riferimenti</b>	UC2: Ricerca ed iscrizione ad un corso.
<b>Pre-condizioni</b>	Esiste una lista di corsi e lo studente è iscritto al sistema.
<b>Post-condizioni</b>	È stata selezionata un'istanza c di un corso.

Contratto CO2: iscrizioneCorso

<b>Operazioni</b>	iscrizioneCorso(id_studente: string, id_corso: string)
<b>Riferimenti</b>	UC2: Ricerca ed iscrizione ad un corso.
<b>Pre-condizioni</b>	È stato selezionato un corso ed è stata creata un'istanza di un corso.
<b>Post-condizioni</b>	È stata creata un'istanza di iscrizione i ed è stata ricevuta una conferma di iscrizione.

Contratto CO3: pagamentoIscrizione

<b>Operazioni</b>	pagamentoIscrizione()
<b>Riferimenti</b>	UC2: Ricerca ed iscrizione ad un corso.
<b>Pre-condizioni</b>	Il corso prevede un costo ed è stata creata un'istanza di iscrizione i.
<b>Post-condizioni</b>	È stata ricevuta una conferma di pagamento e di avvenuta iscrizione.

#### **1.2.4.3 Contratti Operazioni UC5**

#### Contratto CO1: selezionaCorsoCreato

<b>Operazioni</b>	selezionaCorsoCreato(studente: studente)
<b>Riferimenti</b>	UC5: Caricamento contenuti.
<b>Pre-condizioni</b>	Lo studente ha una lista dei corsi creati.
<b>Post-condizioni</b>	Viene selezionato uno tra i corsi creati e viene settato come corsoSelezionato.

#### Contratto CO2: caricaContenuto

<b>Operazioni</b>	caricaContenuto(contenuto: contenuto)
<b>Riferimenti</b>	UC5: Caricamento contenuti.
<b>Pre-condizioni</b>	Lo studente è il creatore del corso.
<b>Post-condizioni</b>	È stata aggiunta un'istanza di contenuti c al corso selezionato.

#### 1.2.4.3 Contratti Operazioni UC6

##### Contratto CO1: caricaAppunto

<b>Operazioni</b>	caricaAppunto(appunto: appunto)
<b>Riferimenti</b>	UC6: Caricamento appunti.
<b>Pre-condizioni</b>	Lo studente è iscritto.
<b>Post-condizioni</b>	È stata aggiunta un'istanza di appunti a e viene aggiunta ad una lista di appunti.

##### Contratto CO2: confermaCaricamento

<b>Operazioni</b>	confermaCaricamento()
<b>Riferimenti</b>	UC6: Caricamento appunti.
<b>Pre-condizioni</b>	Esiste una lista di appunti da caricare.
<b>Post-condizioni</b>	Le istanze sono caricate nella lista degli appunti dello studente.

## 1.3 Progettazione

La progettazione orientata agli oggetti è interessata alla definizione degli oggetti software, delle loro responsabilità e a come questi collaborano per soddisfare i

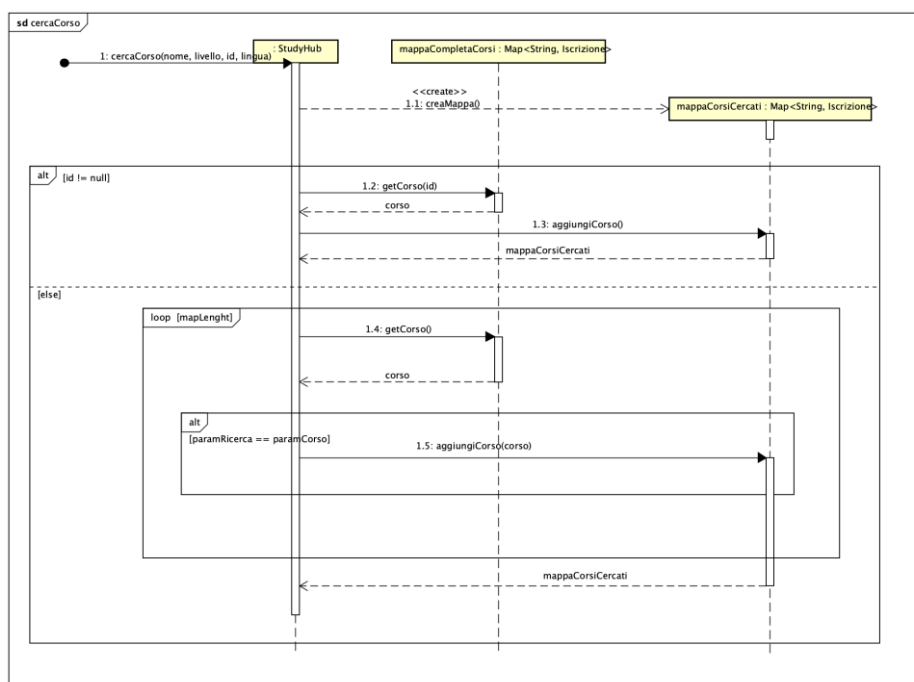


requisiti individuati nei passi precedenti. L'elaborato principale di questa fase che è stato preso in considerazione è il modello di progetto, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (diagrammi di interazione) che da un punto di vista statico (diagramma delle classi). Seguono dunque i diagrammi di Interazione più significativi e il diagramma delle classi relativi ai casi d'uso determinati a seguito di un attento studio degli elaborati scritti in precedenza.

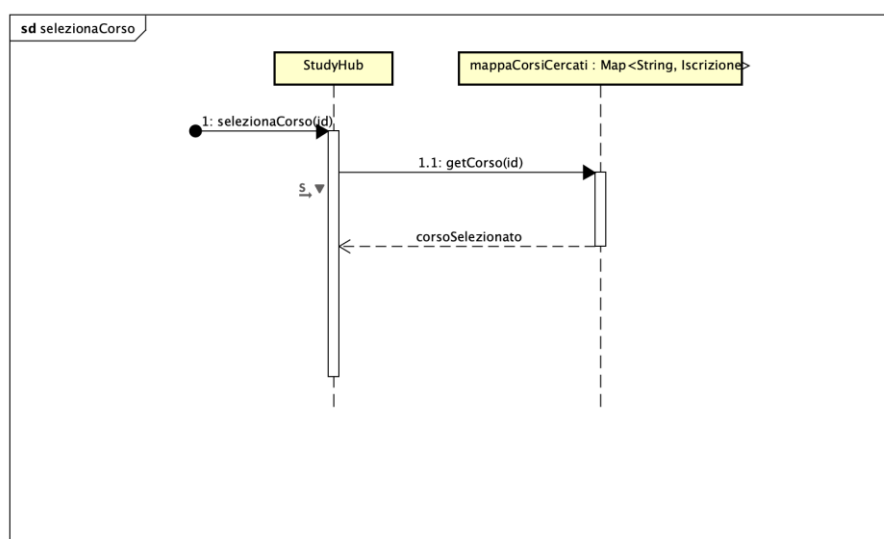
### 1.3.1 Diagrammi di sequenza

#### 1.3.1.1 – UC2

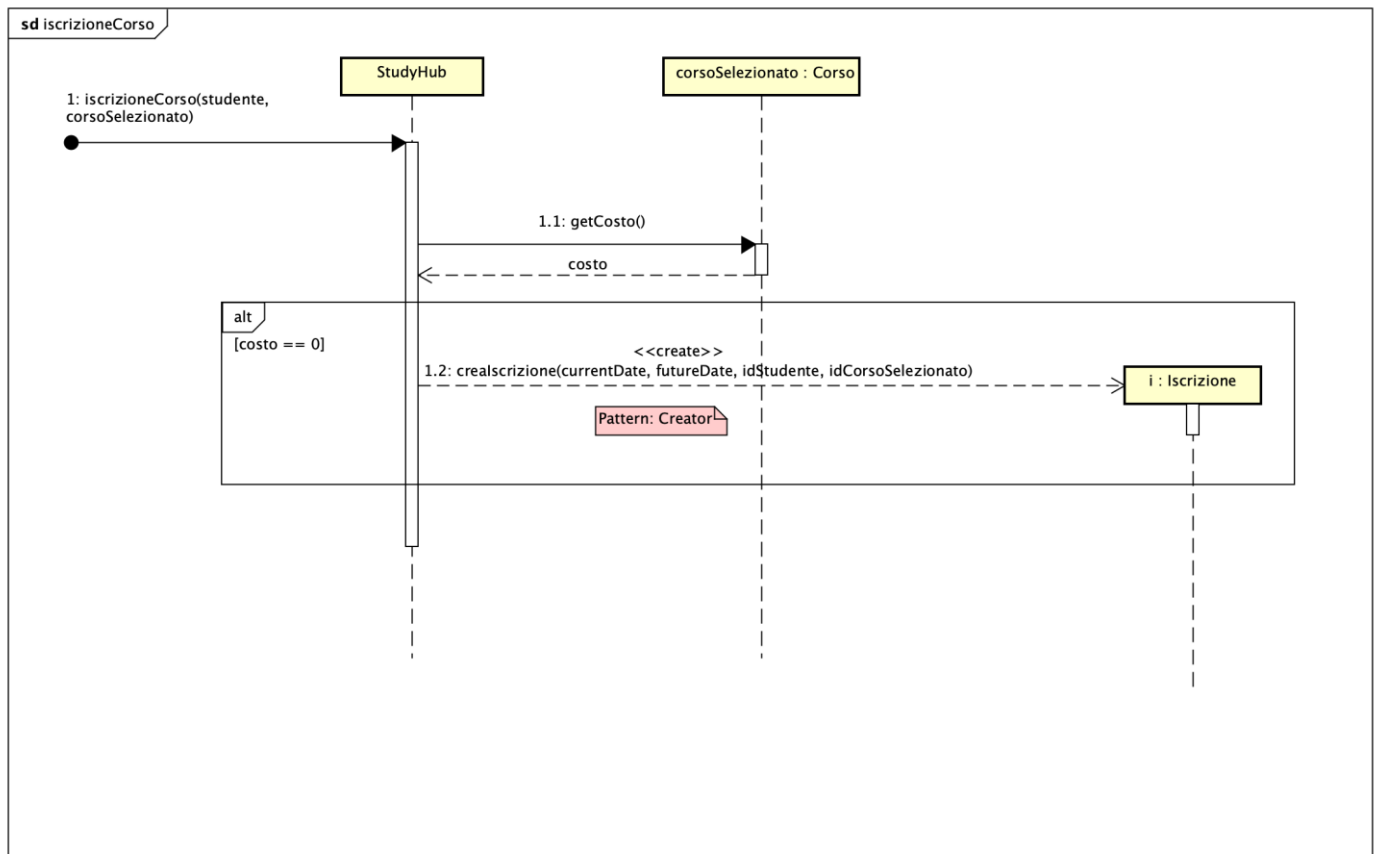
- cercaCorso



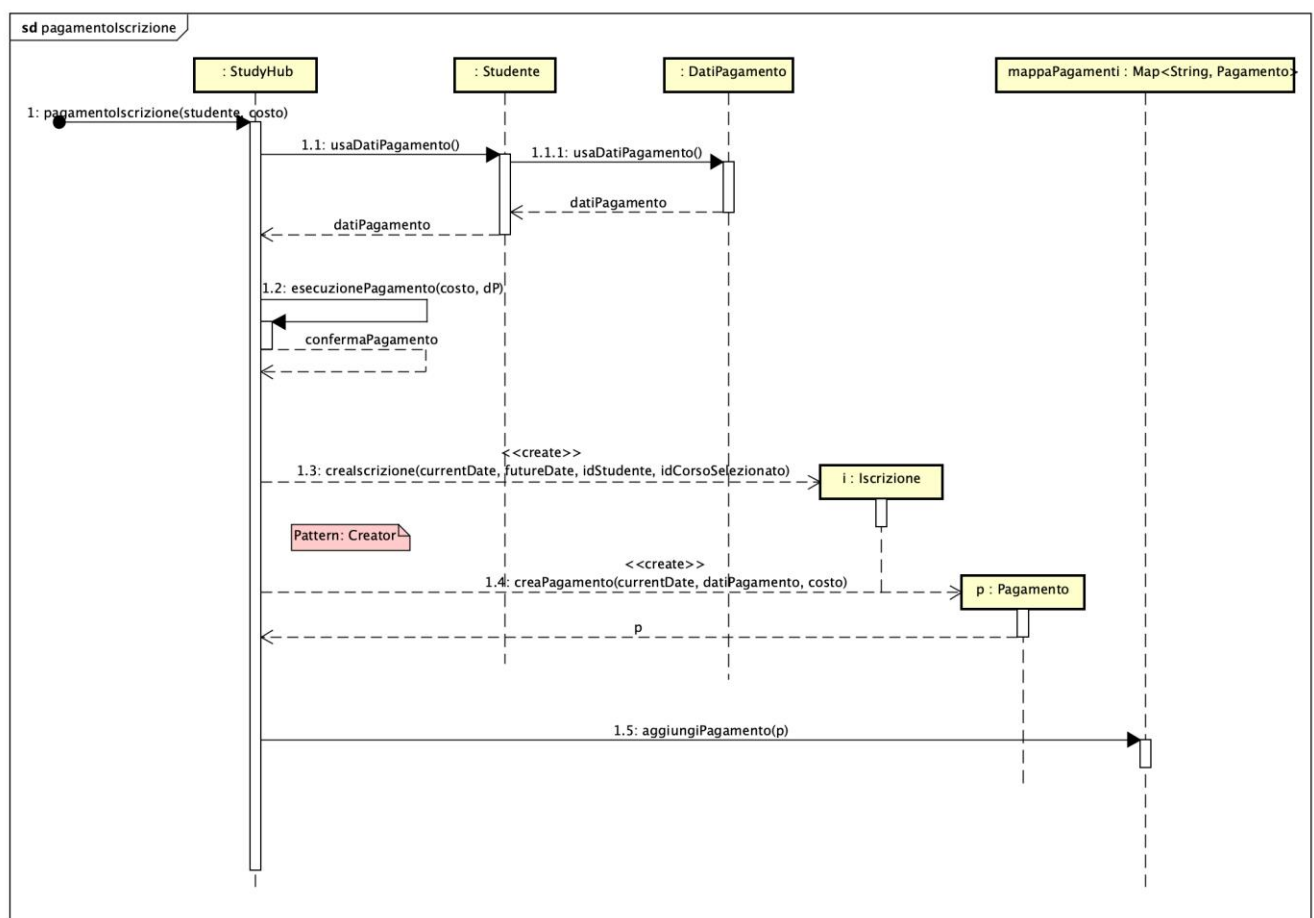
- selezionaCorso



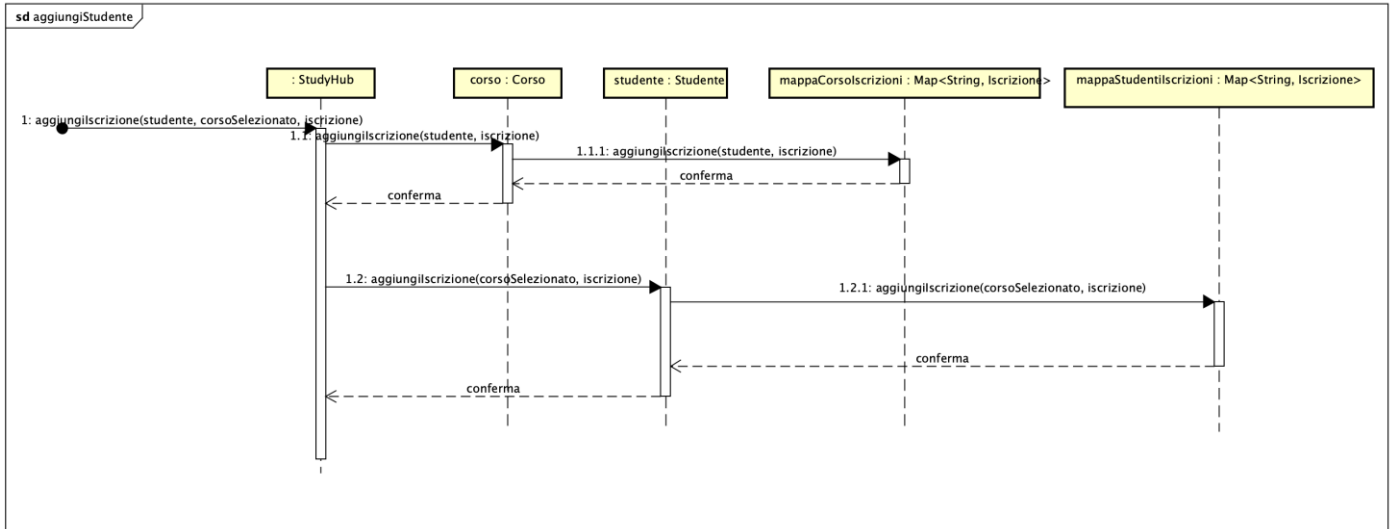
- iscrizioneCorso



- pagamentoIscrizione

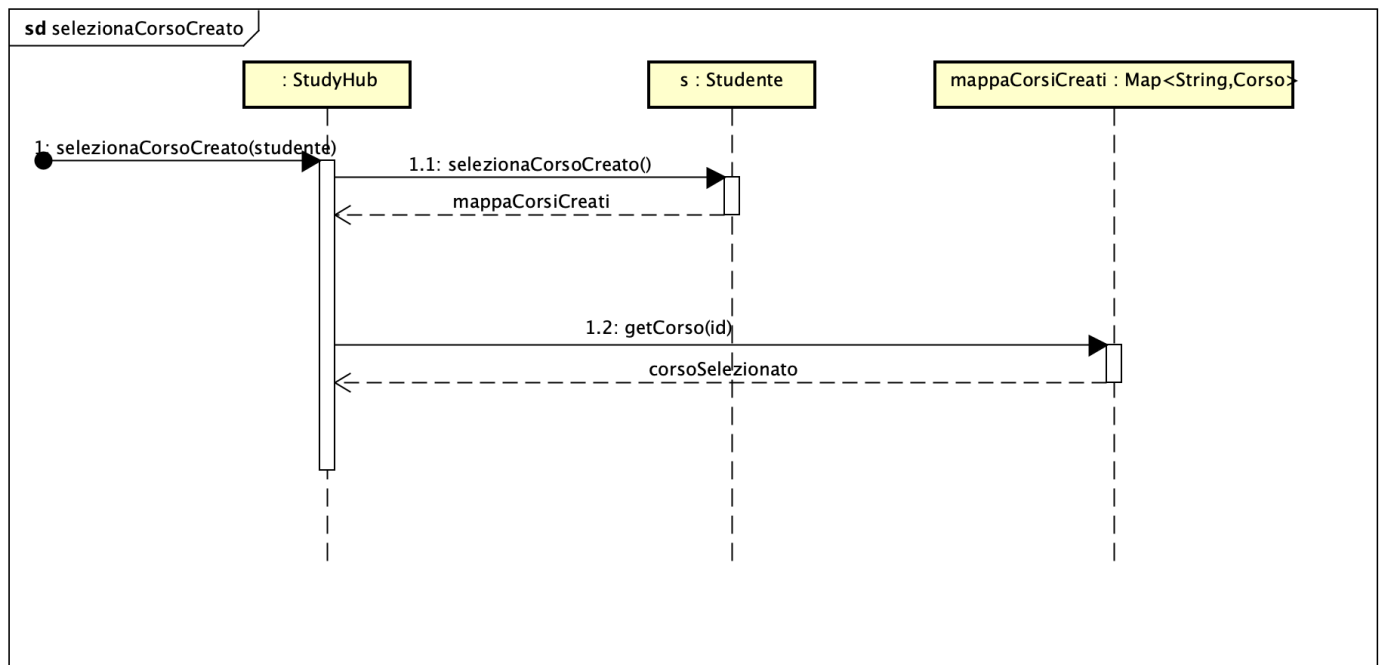


- aggiungiStudente

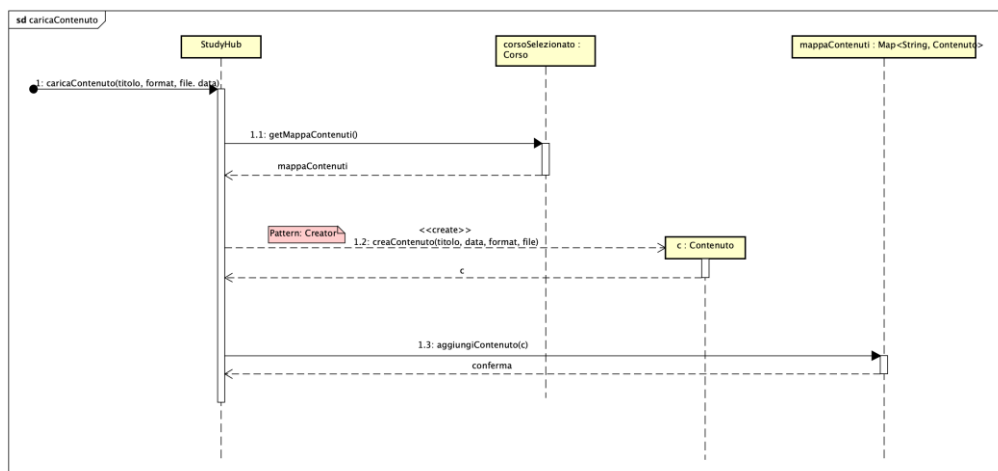


### 1.3.1.2 – UC5

- selezionaCorsoCreato

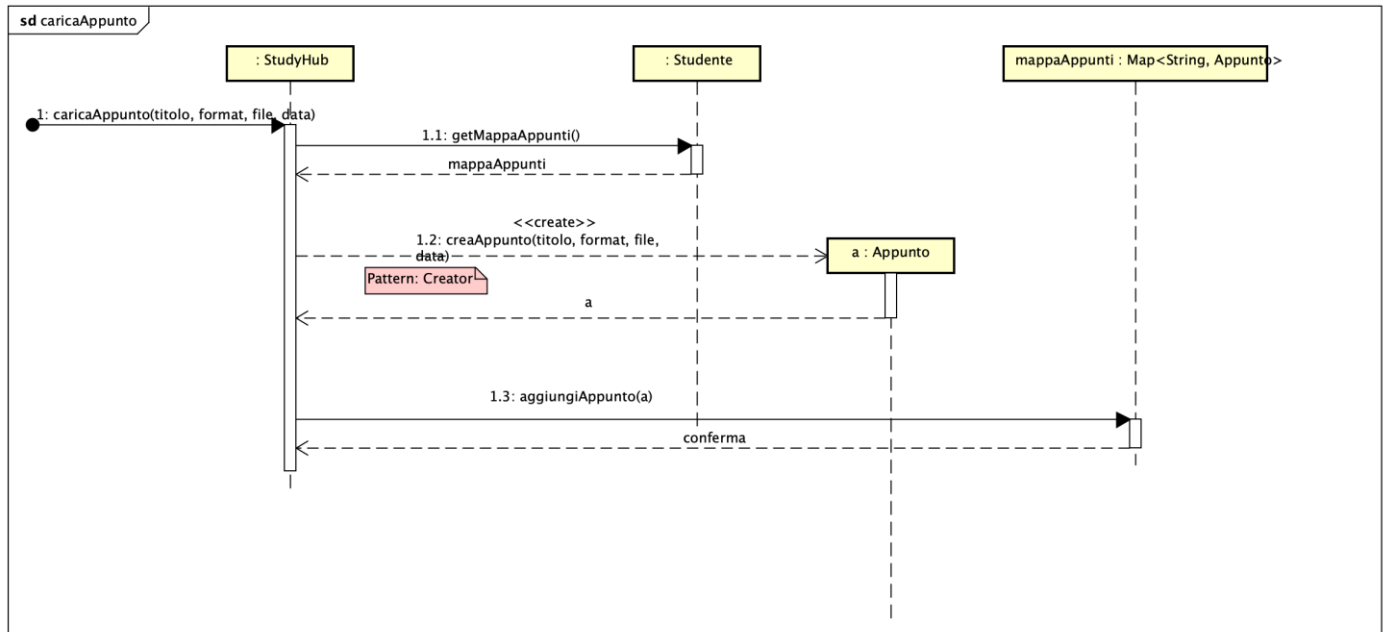


- caricaContenuto



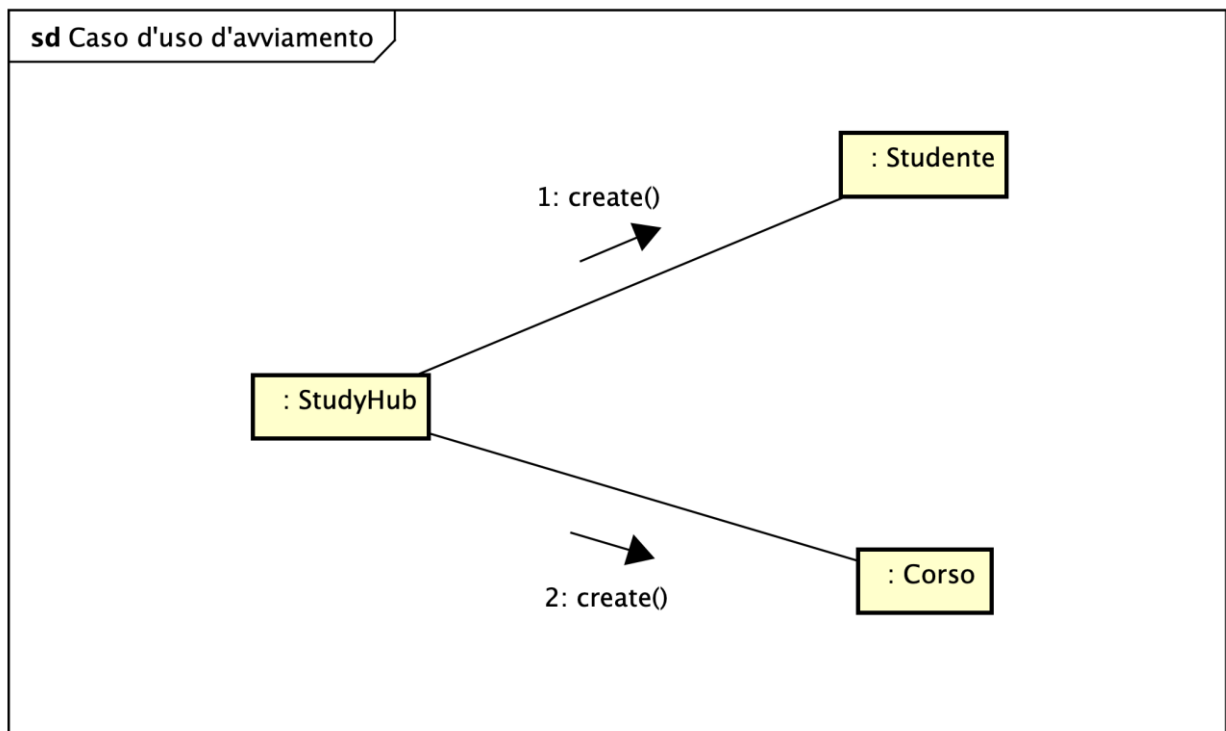
### 1.3.1.3 – UC6

- caricaAppunto



### 1.3.1.4 – Caso d'uso di avviamento

Per fare partire il sistema all'inizio del suo ciclo vitale è stato ideato un caso d'uso d'avviamento per creare degli studenti e dei corsi



### 1.3.2 Diagramma delle classi

