

# ITERAZIONE 3

## 1.0 Introduzione

La terza iterazione si concentrerà sui seguenti aspetti:

- Implementazione di nuovi casi d'uso
- Refactoring del codice e del testing.

In particolare, in questa seconda iterazione andremo ad implementare:

- UC5: ISCRIZIONE CLIENTE A CORSO
- UC9: VALUTAZIONE PIENEZZA DI UN CORSO

Si tratta degli ultimi due casi d'uso del sistema.

Il refactoring del codice ha portato ad avere delle modifiche in alcuni metodi e in particolare al tipo dell'attributo tipologiaCliente in Descrizione Corso. Si rimanda al diagramma delle classi nella sua versione aggiornata.

## 1.1 Aggiornamento dei casi d'uso

Per visionare le modifiche apportate ai suddetti, si consulti il documento “Modello dei casi d'uso” riportato nella cartella in versione aggiornata per questa iterazione.

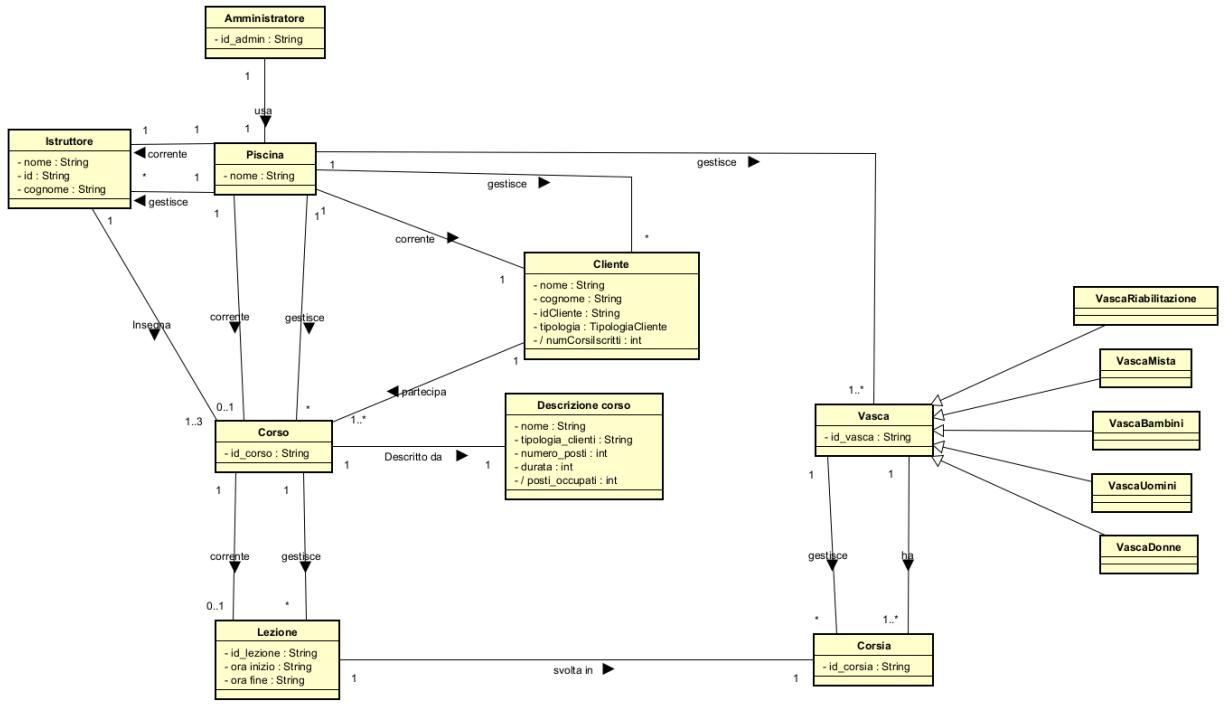
Eventuali ulteriori modifiche ai documenti saranno visionabili tramite la cronologia.

## Fase di Analisi

Non sono presenti nuove classi concettuali identificate in questa iterazione, però è stato aggiornato:

- Il tipo dell'attributo tipologiaCliente in Descrizione Corso è passato dall'essere String al tipo “TipologiaVasca” già presente in vasca e definito opportunamente nel codice come enum.

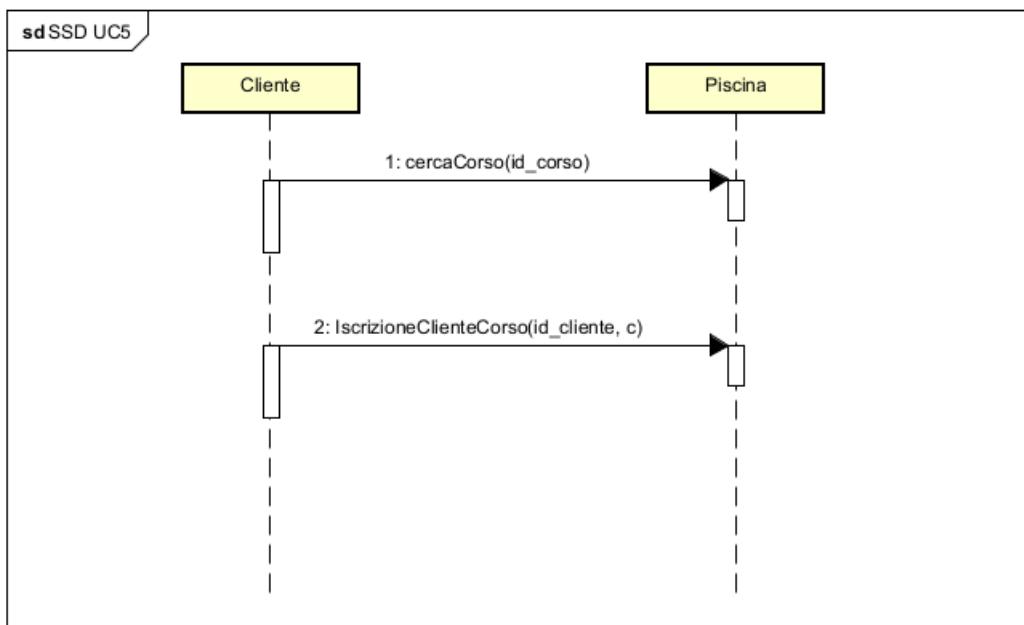
Si è arrivati al seguente Modello di Dominio:



## 2.1 Diagrammi di sequenza di sistema UC5

- UC5 ISCRIZIONE CLIENTE A CORSO

Il seguente Diagramma di Sequenza di Sistema (SSD) illustra il corso di eventi I/O tra il cliente e il sistema Piscina.



### 2.1.1 Contratti delle operazioni

Riferimento:	UC5.1
--------------	-------

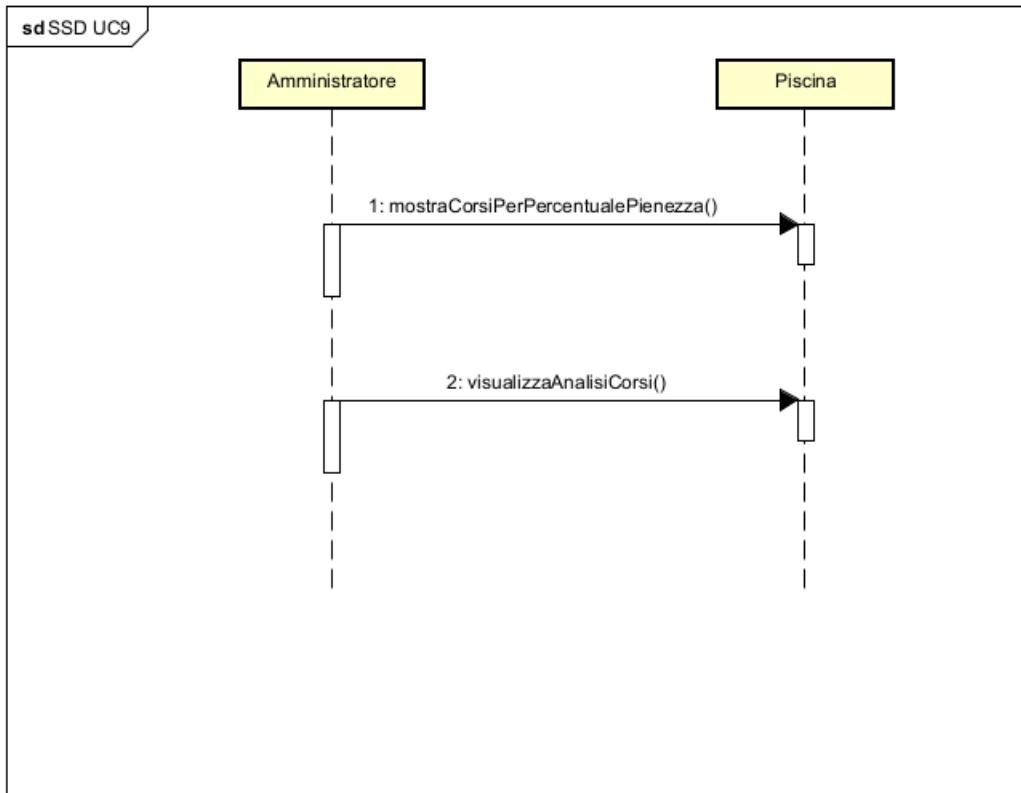
<b>Operazione:</b>	cercaCorso(id_corso)
<b>Pre-condizioni:</b>	<ul style="list-style-type: none"> <li>Nessuna</li> </ul>
<b>Post-condizioni:</b>	<ul style="list-style-type: none"> <li>Se esiste viene recuperata l'istanza c di Corso</li> </ul>

<b>Riferimento:</b>	UC5.2
<b>Operazione:</b>	iscrizioneClienteCorso(id_cliente, c)
<b>Pre-condizioni:</b>	<ul style="list-style-type: none"> <li>Esiste l'istanza c di Corso</li> </ul>
<b>Post-condizioni:</b>	<ul style="list-style-type: none"> <li>Viene recuperata l'istanza cl di Cliente</li> <li>Viene creata un'associazione tra l'istanza c di Corso e l'istanza cl di Cliente</li> </ul>

## 2.2 Diagrammi di sequenza di sistema UC9

- UC9: VALUTAZIONE PIENEZZA CORSO

Il seguente Diagramma di Sequenza di Sistema (SSD) illustra il corso di eventi I/O tra l'amministratore e il sistema Piscina.



## 2.2.1 Contratti delle operazioni

<b>Riferimento:</b>	UC9.1
<b>Operazione:</b>	mostraCorsiPerPercentualePienezza()
<b>Pre-condizioni:</b>	<ul style="list-style-type: none"><li>• Nessuna</li></ul>
<b>Post-condizioni:</b>	<ul style="list-style-type: none"><li>• Nessuna</li></ul>

<b>Riferimento:</b>	UC9.2
<b>Operazione:</b>	visualizzaAnalisiCorsi()
<b>Pre-condizioni:</b>	<ul style="list-style-type: none"><li>• Nessuna</li></ul>
<b>Post-condizioni:</b>	<ul style="list-style-type: none"><li>• Viene creata la lista List&lt;Corso&gt; ListaDaEliminare</li><li>• Viene creata la lista List&lt;Corso&gt; ListaDaAmpliare</li></ul>

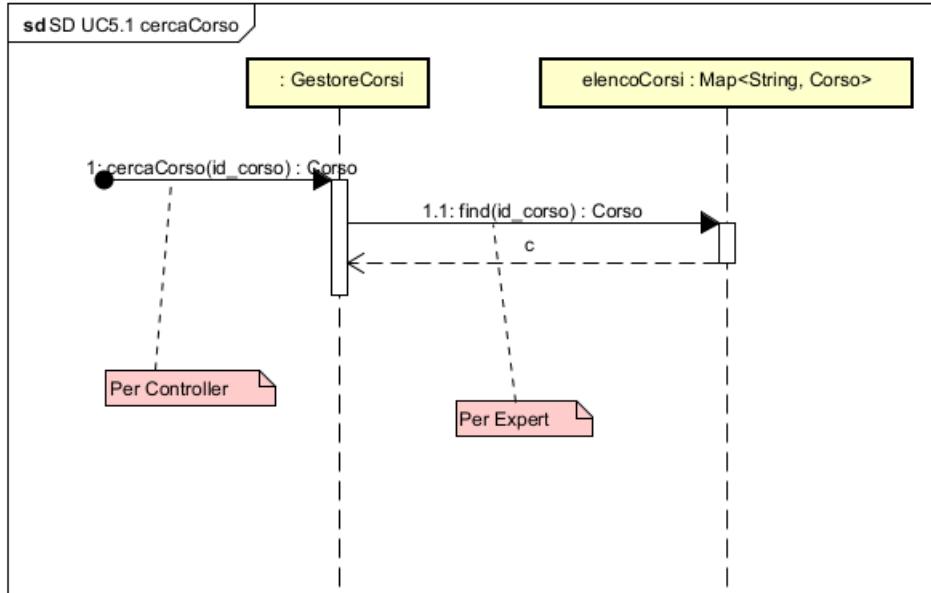
## 3.0 Fase di Progettazione

Gli elaborati principali della fase di progettazione sono i diagrammi di interazione, articolati in diagrammi di sequenza e diagrammi di interazione, con lo scopo di descrivere il comportamento del sistema da un punto di vista dinamico durante i casi d'uso presi in considerazione per questa seconda iterazione. Insieme a questi, il diagramma delle classi rappresenta il sistema da un punto di vista statico. Di seguito vengono riportati:

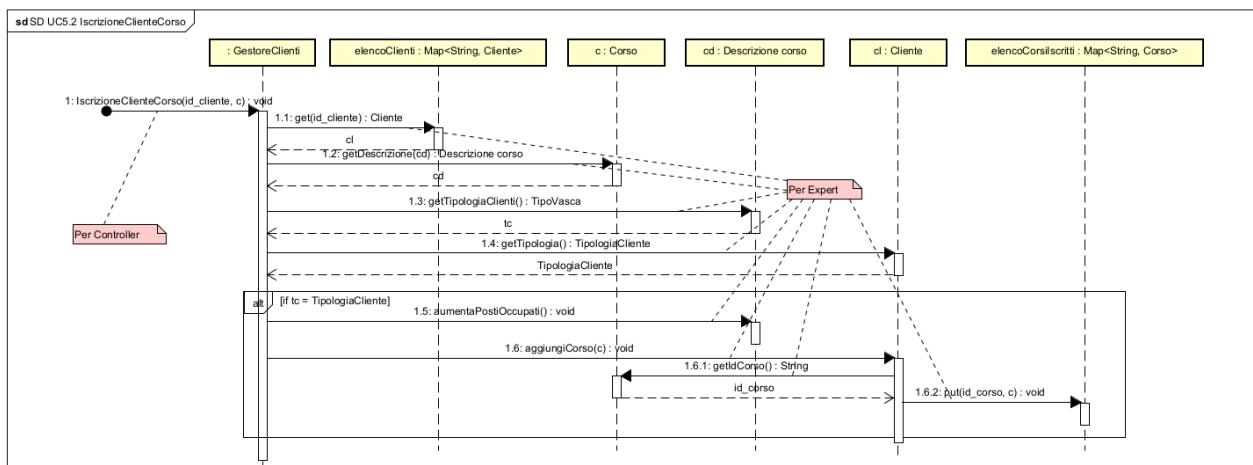
### 3.1 Diagrammi di Sequenza UC5

Si è utilizzato un Controller GestoreClienti per lavorare come interfaccia del sistema. Questo avrà il compito di creare l'associazione tra l'istanza del Cliente e l'istanza del corso a cui desidera iscriversi.

cercaCorso



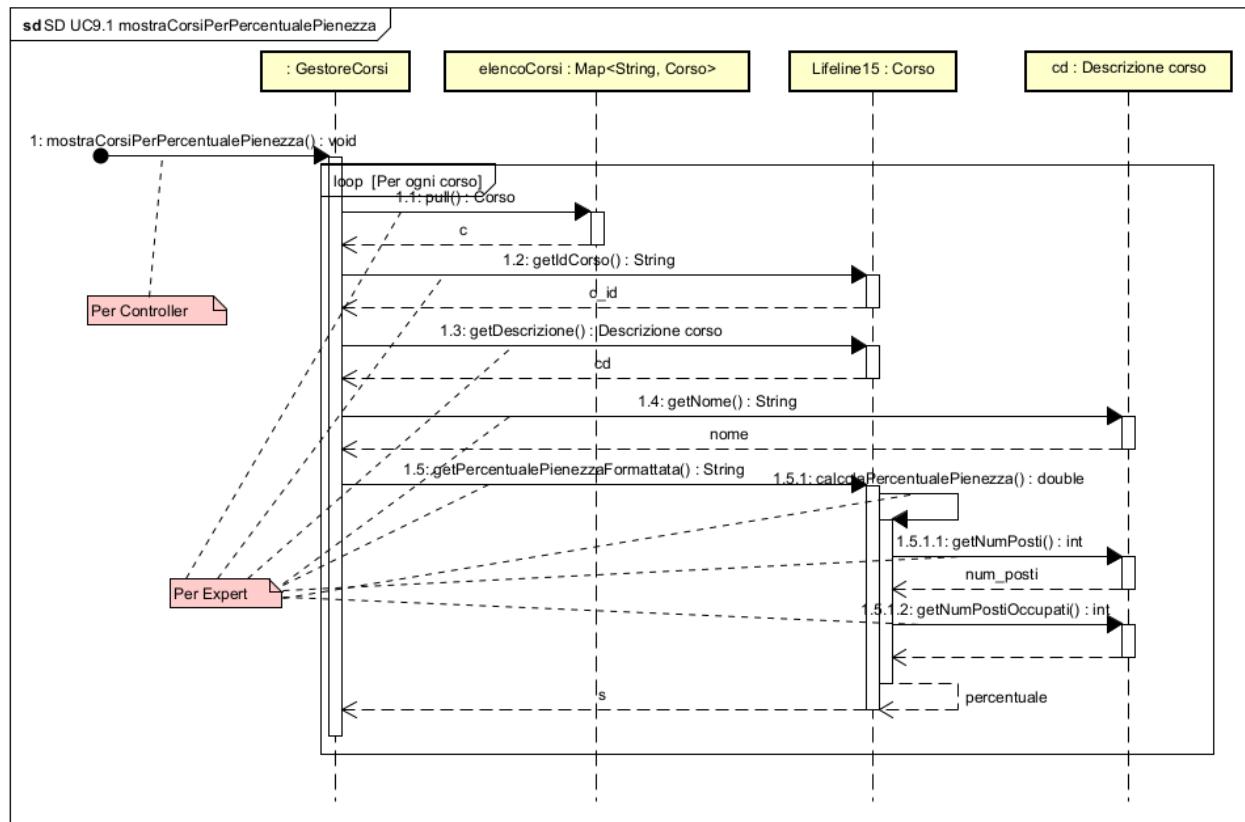
### iscrizioneClienteCorso



### 3.2 Diagrammi di Sequenza UC9

Si è utilizzato un Controller GestoreClienti per lavorare come interfaccia del sistema. Avrà il compito di visualizzare il report e creare le due nuove liste di corsi, una che consiglia quali eliminare, una che consiglia quali ampliare secondo le regole di dominio.

### mostraCorsiPerPercentalePienezza

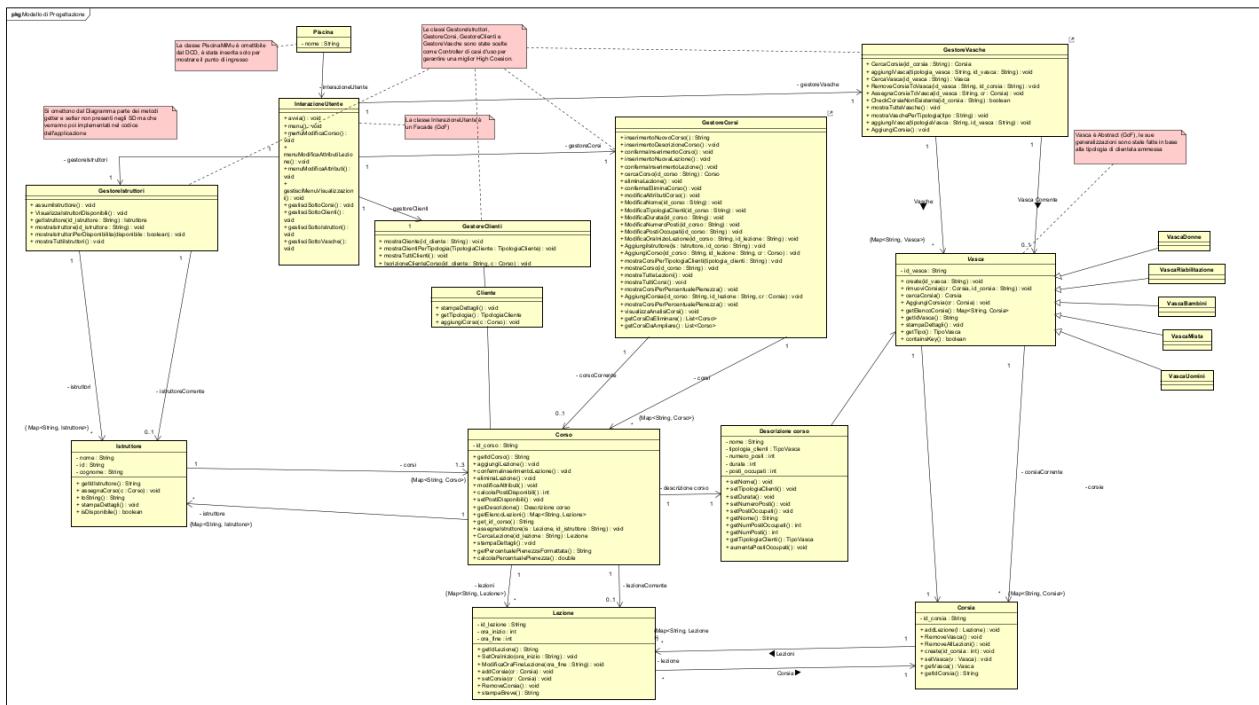


## **visualizzaAnalisiCorsi**



## 3.4 Diagramma delle Classi

Di seguito l'evoluzione del DCD:



## 5.0 Testing

## 5.1 Individuazione dei casi di test

In una fase preliminare al testing, si è svolta un'analisi del codice per individuare le classi e i metodi da testare. La strategia è stata quella di dare priorità ai metodi cruciali per il funzionamento del sistema relativamente ai casi d'uso implementati e ai fixing di test precedenti per il già citato refactoring.

Di seguito sono elencati i nuovi metodi di test:

CorsoTest

- ### **1. testCalcolaPercentualePienezza\_standard**

Verifica che la percentuale di posti occupati sia calcolata correttamente in condizioni standard.

- ## 2. testCalcolaPercentualePienezza\_zeroOccupati

Controlla che la percentuale sia 0 quando non ci sono posti occupati.

- ### 3. testCalcolaPercentualePienezza\_zeroPosti

Verifica che la percentuale gestisca correttamente il caso con zero posti totali evitando divisione per zero.

- #### 4. **testGetPercentualePienezzaFormattata**

Controlla che la percentuale di pienezza sia restituita correttamente come stringa formattata con due decimali.

## GestoreClientiTest

- ## 1. testIscrizioneClienteCorso\_ok

Verifica che un cliente possa iscriversi correttamente a un corso compatibile con la sua tipologia.

2. **testIscrizioneClienteCorso\_TipologiaNonCorrispondente**

Controlla che l'iscrizione fallisca se la tipologia del cliente non corrisponde a quella del corso, lanciando TipologiaNonCorrispondente.

3. **testIscrizioneClienteCorso\_ClienteGiAlScrittoException**

Verifica che l'iscrizione di un cliente già iscritto a un corso generi ClienteGiAlScrittoException.

4. **testIscrizioneClienteCorso\_PostiPieniException**

Controlla che l'iscrizione fallisca quando il corso ha raggiunto il numero massimo di posti disponibili, lanciando PostiPieniException.

## GestoreCorsiTest

1. **testGetCorsiDaEliminare**

Verifica che i corsi con pochi partecipanti vengano identificati correttamente come da eliminare.

2. **testGetCorsiDaAmpliare**

Controlla che i corsi quasi pieni vengano identificati correttamente come da ampliare.