

## Fase di Elaborazione – 1° Iterazione

### 2. Introduzione

Dopo aver terminato la fase di ideazione si procede alla fase di elaborazione in cui si ha lo scopo di affinare le idee precedentemente viste analizzando e implementando in modo iterativo lo sviluppo software. Scopo dell'elaborazione sarà inoltre correggere alcuni errori non considerati durante la fase di ideazione.

Durante la prima iterazione ci si focalizzerà sui casi d'uso UC1 relativo al modulo di Gestione dell'iscrizione.

#### 2.2 Analisi orientata agli oggetti

Un primo approccio che permette di verificare e fornire i dettagli sul dominio è la Modellazione di Business che comprende il completamento di un **Modello di Dominio** che contenga nello specifico i Grafici, i Concetti, gli Attributi e le associazioni più significative per il caso d'uso scelto.

Inoltre si avranno ulteriori elementi significativi per definire i tratti salienti del caso d'uso, ovvero il **Diagramma di sequenza** e i **Contratti delle operazioni**.

Nel nostro caso (relativo all'UC1), per poter garantire lo scenario di successo sarà necessario focalizzarsi sulle seguenti classi concettuali:

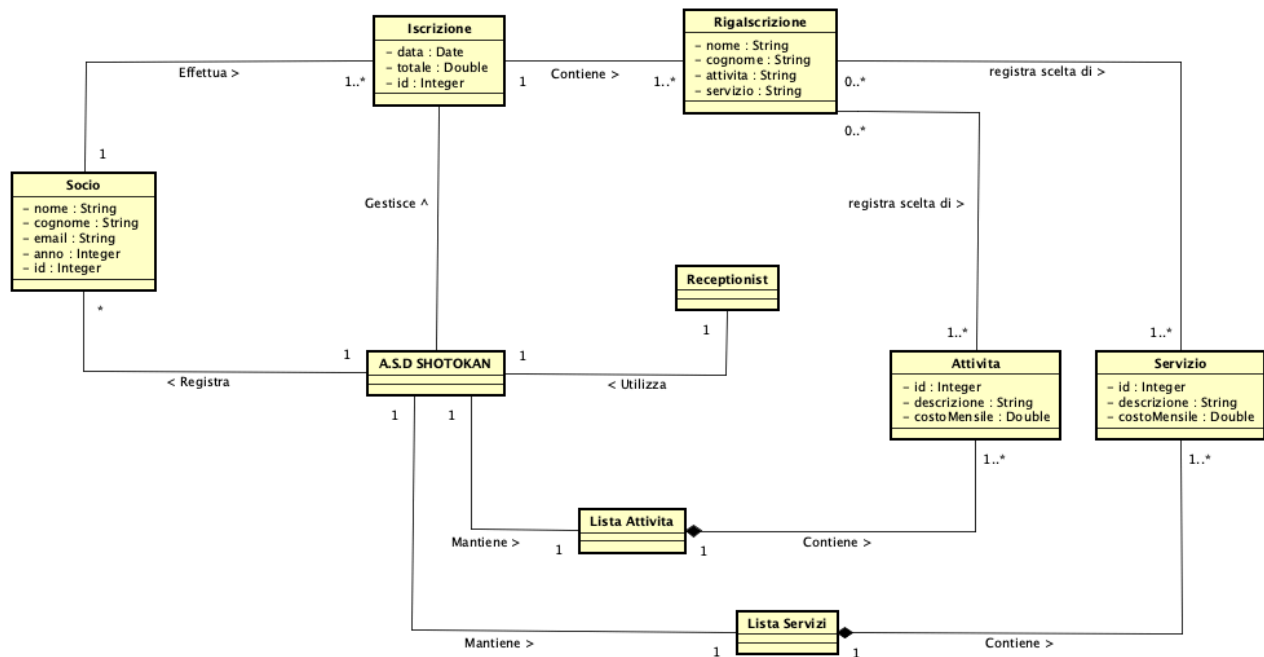
- **Iscrizione**  
Indica l'iscrizione effettuata da un cliente
- **Socio**  
Cliente della palestra
- **Receptionist**  
Attore primario, che interagisce col sistema
- **Attività**  
Rappresenta la prima scelta effettuata dall'utente
- **Servizio**  
Rappresenta la seconda scelta effettuata dall'utente
- **A.S.D.SHOTOKAN**  
Il sistema utilizzato per effettuare le operazioni
- **Lista delle attività**  
Catalogo che mostra tutte le attività disponibili
- **Lista dei servizi**  
Catalogo che mostra tutti i servizi disponibili

### 2.2.1 Applicazione Pattern (Prototype, Singleton e Command)

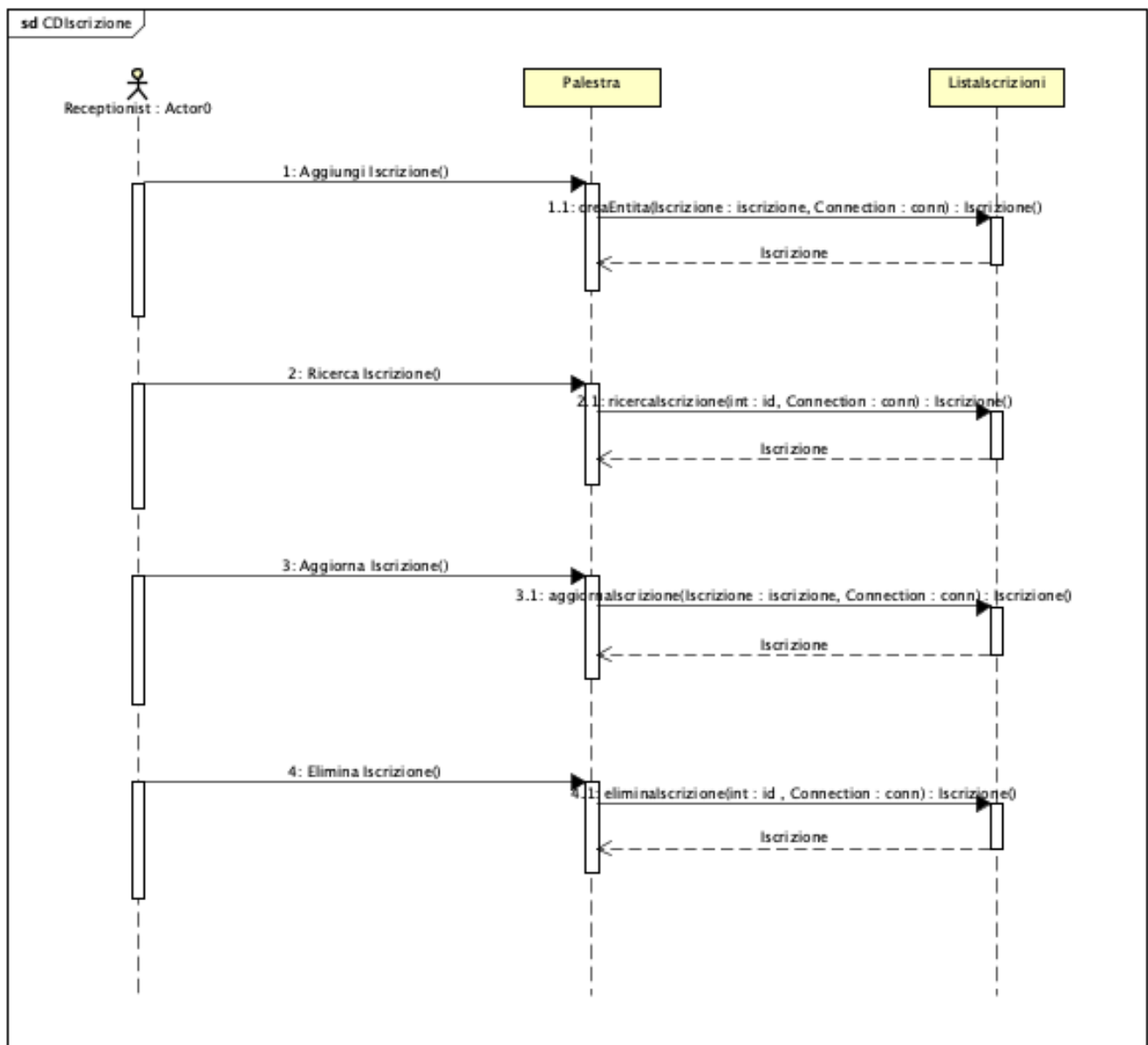
L'applicazione e la descrizione dei pattern utilizzati si trova nella sezione dedicata di questa documentazione ( Documentazione Generale > Pattern e Test.pdf).

### 2.2.2 Analisi orientata ad oggetti

Tenuto conto degli elementi sopra avremo il seguente **Modello di Dominio**:



Come detto in precedenza l'analisi orientata agli oggetti è definita nella sua completezza dalla presenza di 3 elementi. Il secondo da introdurre è il **Diagramma di Sequenza (SSD)**, che nel nostro caso sarà il seguente:



Il terzo elemento fondamentale per la definizione completa dell'analisi orientata agli oggetti è dato dai **Contratti delle operazioni**, che descrivono gli eventi di sistema elaborati nel Diagramma di Sequenza (SSD).

#### **Contratto CO1 : creaEntita(iscrizione: Iscrizione, conn : Connection)**

*Operazione:*

- **creaEntita(iscrizione :Iscrizione, conn : Connection) : Iscrizione**

*Riferimenti:*

- UC1: Gestisci Iscrizione

*Pre-Condizioni:*

- è in corso l'operazione di Iscrizione

*Post-condizioni:*

- l'attributi nome, cognome della suddetta istanza sono riempiti con il nome e cognome del socio.
- l'attributo attività e servizio sono compilati con il campo descrizione delle relative classi

**Contratto CO2 : ricercaEntita( id : int, conn: Connection)**

*Operazione:*

- **ricercaEntita( id : int, conn: Connection) : Iscrizione**

*Riferimenti:*

- UC1: Gestisci Iscrizione

*Pre-Condizioni:*

- è in corso l'operazione di Iscrizione

**Contratto CO3 : aggiornaEntita(iscrizione :Iscrizione, conn : Connection)**

*Operazione:*

- **modificaEntita(iscrizione :Iscrizione, conn : Connection) : Iscrizione**

*Riferimenti:*

- UC1: Gestisci Iscrizione

*Pre-Condizioni:*

- è in corso l'operazione di iscrizione

**Contratto CO4 : eliminaEntita(id : int , conn : Connection) : Iscrizione**

*Operazione:*

- **eliminaEntita(id : int , conn : Connection)**

*Riferimenti:*

- UC1: Gestisci Iscrizione

*Pre-Condizioni:*

- è in corso l'operazione di iscrizione

**Contratto CO5: calcolaTotale(conn : Connection)**

*Operazione:*

- **calcolaTotale(conn : Connection) : double**

*Riferimenti:*

- UC1: Gestisci Iscrizione

*Pre-Condizioni:*

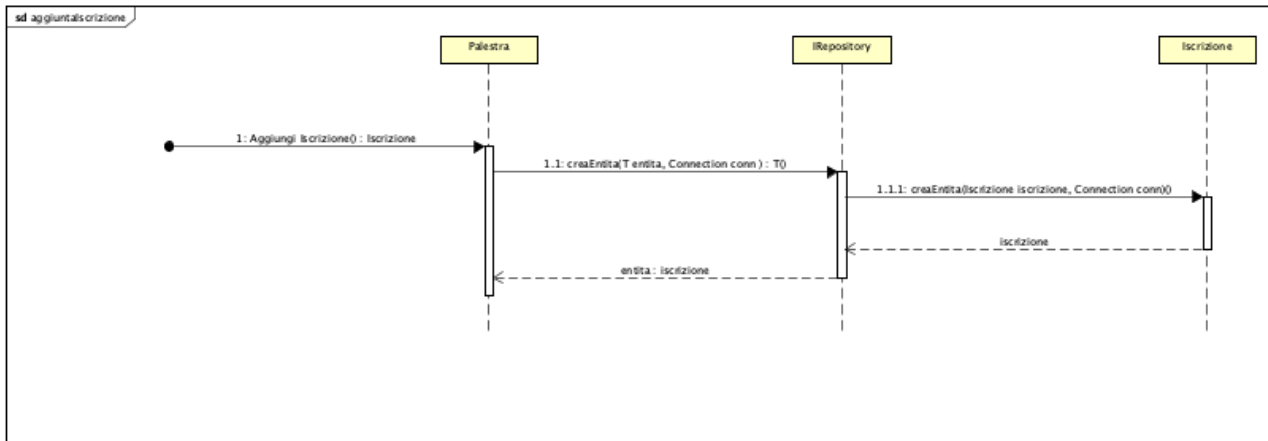
- è in corso l'operazione di Iscrizione

## 2.3 Progettazione

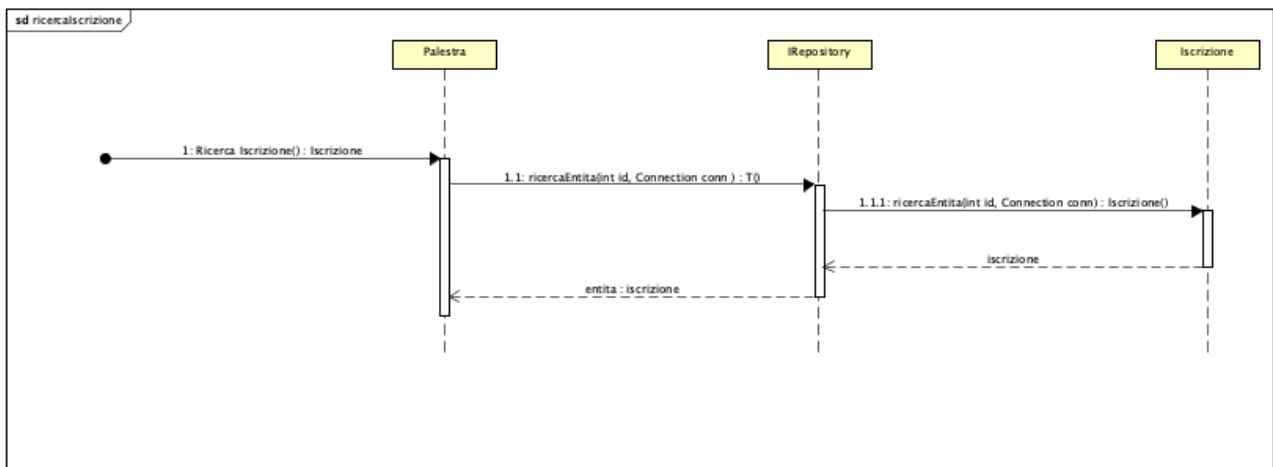
Al fine di poter definire gli oggetti software e le loro responsabilità, da un punto di vista statico o dinamico, si definisce un Modello di progetto, consistente in una serie di Diagrammi di Sequenza (SSD) e delle Classi.

### 2.3.1 Diagrammi di Sequenza relativi al caso d'uso UC1 : Gestione Iscrizione

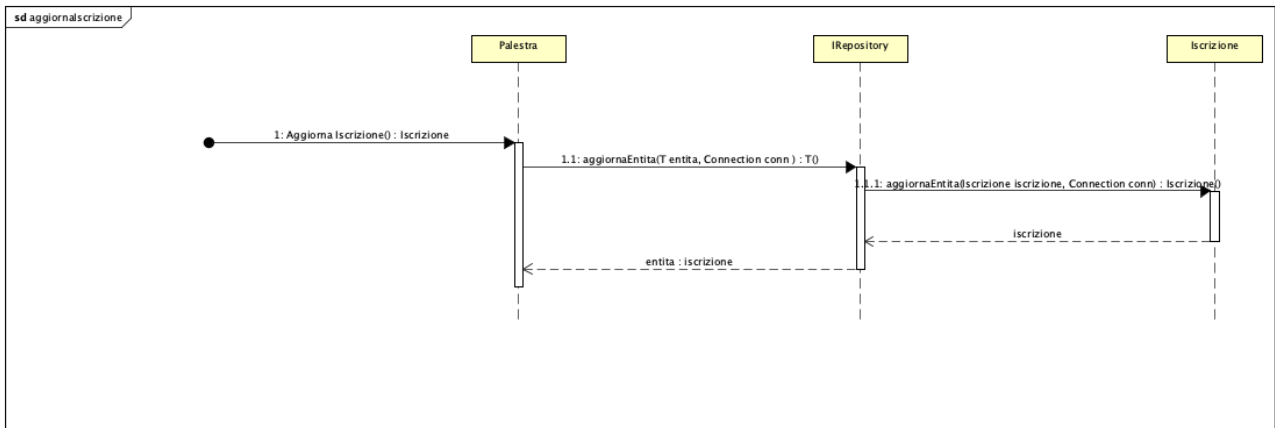
- **Aggiungi Iscrizione**



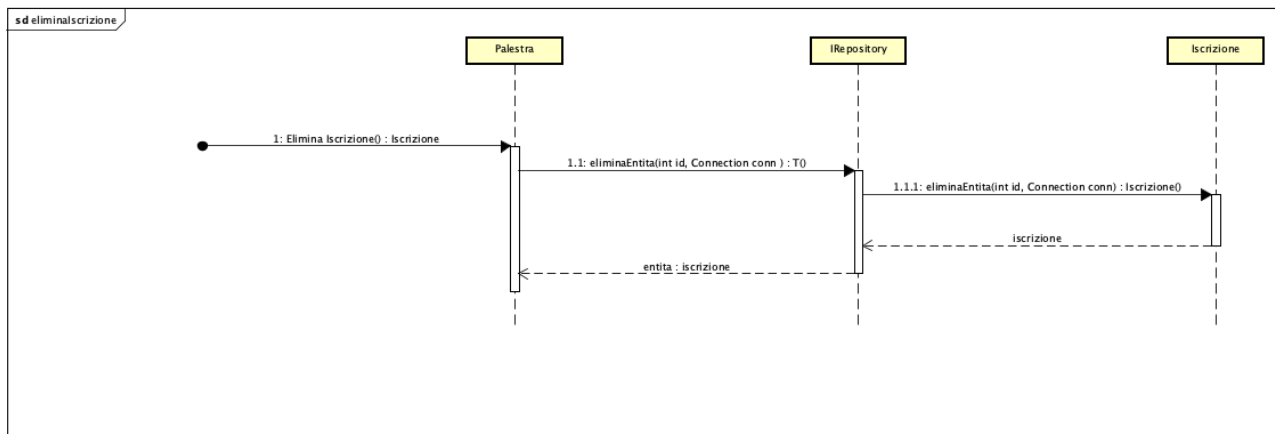
- **Ricerca Iscrizione**



### ○ Aggiorna Iscrizione



### ○ Elimina Iscrizione



## 2.3.1 Diagramma delle classi

Al fine di rendere più chiara l'immagine si dispone la stessa in verticale e nella pagina successiva (l'immagine e il file in formato asta si trovano nella cartella Elaborazione (Iter1) > Diagramma delle classi).

Definizione di Repository Pattern  
Il Repository Pattern è un pattern di design che definisce un'interfaccia per la gestione dei dati e gli elementi che la implementano. Questo pattern è utile per separare la logica di business dalla logica di accesso ai dati e per facilitare il testing e la manutenzione del codice.

