# Elaborazione – Iterazione 1

## 1.1 Introduzione

Conclusa la fase di ideazione, si passa alla fase di elaborazione. Obiettivo dell’Iterazione 1 sarà quello di implementare lo scenario principale di successo e tutte le estensioni individuate nel caso d’uso UC1: “Inserimento nuovo Padeleur” nella fase di ideazione.

## 1.2 Aggiornamento dei casi d’uso

## Nel file "Modello dei casi d'uso Iterazione 1" è stato apportato un aggiornamento al titolo dell'UC1, che ora è stato modificato in "Registrazione nuovo Padeleur". Inoltre, si è scelto di includere l'attributo "e-mail" nell'entità Utente (Padeleur), in quanto è pratica comune in molti altri siti web verificare l'identità dell'utente attraverso questa credenziale e successivamente richiedere ulteriori informazioni. Rispetto alla fase di ideazione dell'UC1, lo scenario principale di successo è stato modificato.

## 1.3 Analisi Orientata agli Oggetti

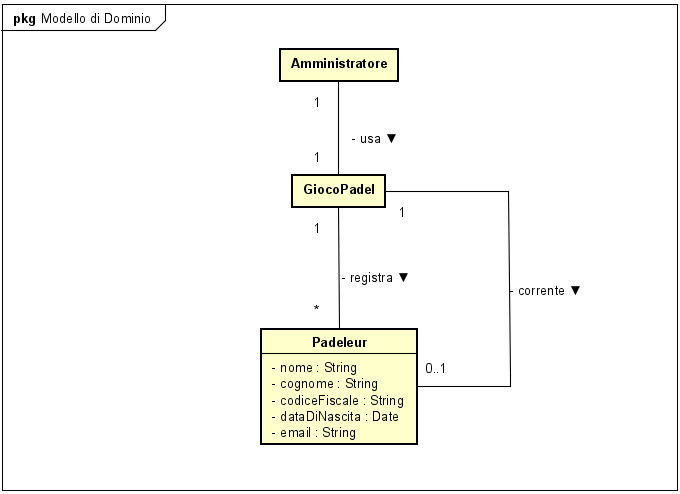
L'analisi orientata agli oggetti si basa sulla creazione di una descrizione del dominio da un punto di vista ad oggetti. Vengono utilizzati diversi strumenti per fornire tale descrizione: Modello di Dominio, SSD, e Contratti delle operazioni.

### 1.3.1 Modello di Dominio

Per il caso d’uso scelto UC1 sono stati identificati le seguenti classi concettuali:

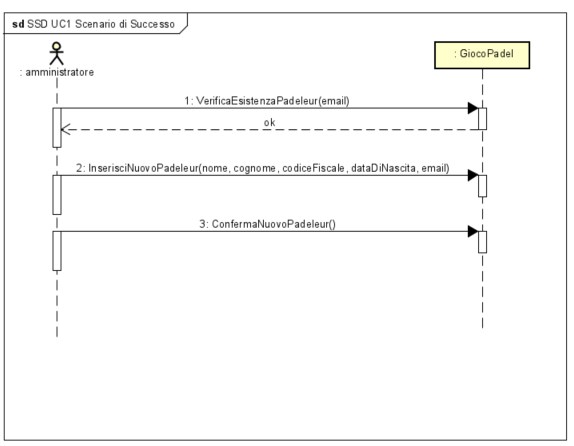
* GiocoPadel: rappresenta il Sistema;
* Amministratore: rappresenta l’Amministratore di Sistema, l’attore primario di questo caso d’uso;
* Utente (Padeleur): rappresenta un giocatore che vuole utilizzare il Sistema per prenotare una partita di Padel.

È stato ricavato il seguente Modello di Dominio:



### 1.3.2 Diagramma di sequenza di sistema

Procedendo con l’analisi Orientata agli Oggetti, andiamo a creare il Diagramma di Sequenza di Sistema (SSD) per lo scenario del caso d’uso scelto UC1. Avremo:



In questo diagramma, l'Amministratore richiede l’e-mail del Padeleur al sistema GiocoPadel, che a sua volta verifica se il Padeleur esiste già nel sistema o meno. Se il Padeleur non esiste ancora su GiocoPadel, l'amministratore inserisce i dati anagrafici del Padeleur nel Sistema e conferma la registrazione. In caso contrario, il Sistema segnala all'amministratore che il Padeleur è già stato registrato in precedenza e il processo di registrazione si interrompe.

### 1.3.3 Contratto delle operazioni

Di seguito viene indicato il Contratto delle operazioni per l’UC1:

|  |  |
| --- | --- |
| **Operazione:** | VerificaEsistenzaPadeleur(email). |
| **Riferimenti:** | Caso d’uso UC1: Registrazione nuovo Padeleur. |
| **Pre-condizioni:** | - |
| **Post-condizioni:** | è stato restituito un Messaggio di Verifica dal Sistema. |

|  |  |
| --- | --- |
| **Operazione:** | InserisciNuovoPadeleur (nome, cognome, codiceFiscale, dataDiNascita, email). |
| **Riferimenti:** | Caso d’uso UC1: Registrazione nuovo Padeleur. |
| **Pre-condizioni:** | Il Messaggio di Verifica ha dato esito positivo, cioè nel Sistema non esiste nessun Utente (Padeleur) con l’e-mail inserita. |
| **Post-condizioni:** | 1. È stata creata l’istanza nuovoPadelur di Padeleur. 2. Gli attributi nome, cognome, codiceFiscale, dataDiNascita, email di nuovoPadeleur sono stati inizializzati correttamente. 3. nuovoPadeleur è stato associato a GiocoPadel tramite l’associazione “corrente”. |

|  |  |
| --- | --- |
| **Operazione:** | ConfermaNuovoPadeleur() |
| **Riferimenti:** | Caso d’uso UC1: Registrazione nuovo Padeleur. |
| **Pre-condizioni:** | È in corso l’inserimento del nuovo Padeleur. |
| **Post-condizioni:** | nuovoPadeleur è stata aggiunto a GiocoPadel tramite l’associazione “registra”. |

## 1.4 Progettazione

Di seguito sono presenti i Diagrammi di Sequenza:

VerificaEsistenzaPadeleur(email : String)

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

InserisciNuovoPadeleur (nome : String, cognome : String, codiceFiscale : String, dataDiNascita : Date, email : String)

Immagine che contiene testo, linea, Diagramma, schermata

Descrizione generata automaticamente

ConfermaPadeleur()

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente

## 1.4.1 Modello delle classi di Progetto

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

## 1.5 Test

Sono stati creati dei test automatizzati per verificare che i metodi e le classi da noi implementate siano funzionanti:

* verificaElencoPadeleur(): Utilizziamo il metodo getElencoPadeleur per ottenere la mappa elencoPadeleur al di fuori della classe. Se il test va a buon fine, abbiamo verificato che l’elenco dei Padeleur non sia vuoto e che sia stato selezionato correttamente.
* verificaDettagliPersona1(): Verifichiamo che i dati della persona di cui si vuole effettuare il controllo siano tutti presenti in modo da poterne verificare la sua presenza all’interno dell’elenco analizzato precedentemente. Se il test va a buon fine significa che tutto è stato inserito correttamente.
* verificaEsistenzaPadeleur(): Utilizziamo il metodo verificaEsistenzaPadeleur di GiocoPadel per verificare, tramite l’indirizzo e-mail univoco del Padeleur, se esso sia presente all’interno dell’elenco. Se il test va a buon fine l’utente esiste già all’interno dell’elenco.
* verificaNonEsistenzaPadeleur(): Utilizziamo il metodo verificaEsistenzaPadeleur di GiocoPadel per verificare, tramite l’indirizzo e-mail univoco del Padeleur, se esso non sia presente all’interno dell’elenco. Se il test va a buon fine l’utente non esiste ancora all’interno dell’elenco.
* inserisciNuovoPadeleur(): Utilizziamo il metodo inserisciNuovoPadeleur di GiocoPadel per inserire nell’elenco un Padeleur che non ha già fatto la registrazione in precedenza. Se il test va a buon fine, il nuovo Padeleur viene correttamente inserito nell’elenco.
* salvaPadeleurSuFile(): Utilizziamo il metodo salvaPadeleurSuFile di GiocoPadel per scrivere all’interno del file di testo selezionato in modo da salvare i dati relativi al Padeleur. Se il test va a buon fine, il Padeleur viene salvato su file con successo.