# Pattern utilizzati

* **Creator**: è stato utilizzato questo “*Pattern GRASP di base*” nell’operazione registraCliente(…) dell’UC1 per indicare che lo stesso sistema Fitness Ready crea un’istanza di Cliente.
* **Information Expert**: questo “*Pattern GRASP di base*” è stato utilizzato in tutti i casi d’uso per individuare le classi aventi le informazioni necessarie a svolgere le differenti operazioni e di conseguenza assegnargli le dovute responsabilità, come ad esempio nell’operazione registraCliente(…) dell’UC1 in cui il sistema Fitness Ready è in possesso delle informazioni per registrare un nuovo cliente ed aggiungerlo alla lista degli utenti.
* **Controller**: questo “*Pattern GRASP di base*” è stato utilizzato per specificare che il sistema Fitness Ready, ovvero il *Facade Controller*, è l’entità avente l’incarico di coordinare in maniera opportuna tutte le informazioni provenienti dalla UI.
* **Polymorphism**: questo “*Pattern GRASP*” è stato utilizzato per ottenere l’accesso il corretto accesso alle funzioni del sistema Fitness Ready in base al tipo di utente (Cliente o Amministratore) che effettua il login tramite l’operazione effettuaLogin(…).
* **Singleton**: questo “*Pattern GoF Creazionale*” è stato utilizzato per creare un'unica istanza del sistema Fitness Ready e fornire un accesso globale a tale unica istanza.
* **Composite:** questo “*Pattern GoF Strutturale*” è stato utilizzato per creare la composizione di una dieta ed un piano esercizi da associare alle schede fitness in modo da consentire il trattamento dei suddetti come composti da: diversi pasti (colazione, pranzo, cena) avente ognuno le proprie portate per quanto riguarda la dieta; diversi tipi di esercizi (cardio, gambe, braccia, torso, ecc…) per quanto riguarda il piano esercizi.
* **Observer:** questo “*Pattern GoF Comportamentale*” è stato utilizzato per fare in modo che, alla creazione di una prenotazione (ovvero il nostro “Osservatore”), il sistema crei un nuovo acquisto ed un nuovo ordine, quando la disponibilità del prodotto relativo alla prenotazione effettuata (ovvero il nostro “Osservato”) venga aggiornato e sia pari o superiore alla quantità prenotata dal cliente, e contemporaneamente cancelli la prenotazione stessa.
* **Iterator**: questo “*Pattern GoF Comportamentale*” è stato utilizzato per ottenere un accesso sequenziale controllato alle varie liste utilizzate nell’implementazione dei casi d’uso UC9, UC10, UC11 ed UC12 per l’eliminazione sicura di uno specifico elemento dalle stesse, evitando così l’insorgere di errori riguardanti l’accesso multiplo alle suddette da parte di più entità che le coinvolgono.