### **Challenge 03: Configurazione del personaggio**

L'obiettivo è

* **Strutturare** il processo di configurazione e validazione del personaggio Eroe in modo robusto.
* **Modularizzare** la logica per facilitare l'espansione del codice in diverse fasi del gioco.

**Ciclo Globale del Gioco (Main Loop):**

* **Stato del Gioco:** Inizializzare lo stato del personaggio con un **Dizionario (dict)**.
* **Gestione Fasi Principali:** Utilizzare lo **structural pattern matching (match/case)** sul comando principale per indirizzare il programma alla fase corretta (es. 1. Storia, 2. Configurazione, 3. Termina).

**Requisiti Strutturali e di Flusso (Fase di Configurazione):**

* **Ciclo di Configurazione:** All'interno della fase di configurazione, implementare un **Ciclo Indefinito (while True)** per mantenere il menu di creazione attivo e gestibile.
* **Gestione Logica/Comandi Interni:** Utilizzare **match/case** per gestire le opzioni specifiche di configurazione all'interno del ciclo (1. Scegli Nome, 2. Scegli Classe, 3. Completa).
* **Terminazione del Ciclo:** Usare l'istruzione **break** per uscire dal ciclo di configurazione *solo* quando tutti i dati obbligatori del personaggio (Nome e Classe) sono stati validati e assegnati.

**Requisiti di Validazione e Dati:**

* **Definizione Classi:** Le statistiche base delle classi (es. Nome e HP) devono essere definite in un contenitore **Immutabile** come una **Lista di Tuple (list[tuple])**.
* **Validazione Nome:** Verificare la lunghezza del nome (tra 3 e 15 caratteri).
* **Assegnazione Classe:** Utilizzare un **Ciclo for** per iterare sulle classi disponibili e accettare la selezione tramite indice.
* **Aggiornamento Dati:** Assegnare gli HP iniziali corretti al Dizionario del personaggio utilizzando il valore estratto dalla Tupla selezionata.