Perfetto! Iniziamo con la creazione del documento di implementazione che riassume il ragionamento, lo scopo e le modalità d'uso delle funzionalità che abbiamo discusso per l'importazione avanzata dello storico giocatori e la successiva "traduzione" delle statistiche per il ProjectionEngineService.

Poi, analizzeremo i tuoi ultimi log e definiremo i passi successivi.

**Documento di Implementazione: Import Avanzato Storico Giocatori e Conversione Statistiche per Lega**

Versione: 1.0 (di questa specifica funzionalità)

Data: 31 maggio 2025

**1. Scopo e Obiettivo dell'Implementazione**

L'obiettivo principale di questa implementazione è migliorare l'accuratezza delle proiezioni generate dal ProjectionEngineService, in particolare per i giocatori che hanno militato in leghe di diverso livello competitivo (es. Serie B vs Serie A) e per i quali si desidera utilizzare dati storici più ricchi (potenzialmente da fonti come FBRef.com, raccolti manualmente).

Questo si ottiene introducendo:

* Un meccanismo per importare dati storici dei giocatori che specificano la **lega di militanza** per ogni stagione.
* Un sistema di **fattori di conversione** per "tradurre" le performance statistiche di un giocatore da una lega di origine (es. Serie B) al livello atteso della lega target principale (Serie A) prima che queste statistiche vengano usate per calcolare le medie ponderate.
* Un nuovo importer e comando Artisan dedicati per gestire questi dati storici arricchiti, mantenendo la retrocompatibilità con i metodi di importazione esistenti.

**2. Componenti e Modifiche Implementate**

**2.1. Modifiche al Database e Modello**

* **Tabella historical\_player\_stats:**
  + Aggiunta della colonna league\_name (VARCHAR, nullable, commento: 'Lega in cui sono state registrate queste statistiche (es. Serie A, Serie B)').
  + *Migrazione Eseguita:* add\_league\_name\_to\_historical\_player\_stats\_table.
* **Modello App\Models\HistoricalPlayerStat:**
  + Il campo league\_name è stato aggiunto all'array $fillable.

**2.2. Nuovo Importer: App\Imports\PlayerSeasonStatsImport.php**

* **Scopo:** Importare dati storici dei giocatori da file CSV/XLSX che contengono la colonna league\_name (e potenzialmente altre statistiche dettagliate).
* **Funzionalità Chiave:**
  + Utilizza WithHeadingRow per accedere alle colonne tramite il loro nome (normalizzato a lowercase senza spazi o simboli per flessibilità).
  + Si aspetta PlayerFantaPlatformID (o alias come idfanta, id) nel file sorgente per collegare lo storico al giocatore corretto nella tabella players. **Non crea nuovi giocatori** se l'ID non viene trovato nel DB players.
  + Legge la colonna NomeLega (o alias come lega) per popolare historical\_player\_stats.league\_name. Accetta un parametro --league nel comando Artisan come fallback se la colonna manca o è vuota in una riga.
  + Popola/aggiorna i record in historical\_player\_stats usando updateOrCreate basato su player\_fanta\_platform\_id, season\_year, league\_name, e team\_name\_for\_season (per gestire correttamente i trasferimenti infra-stagionali o giocatori che giocano in più leghe nella stessa stagione per squadre diverse, anche se quest'ultimo scenario è raro).
  + Gestisce la conversione dei dati e fornisce conteggi di record processati, creati e aggiornati.
* **File:** app/Imports/PlayerSeasonStatsImport.php (codice fornito e discusso).

**2.3. Nuovo Comando Artisan: players:import-advanced-stats**

* **Signature:** players:import-advanced-stats {filepath} {--league="Serie B"}
* **Scopo:** Interfaccia da riga di comando per utilizzare l'importer PlayerSeasonStatsImport.php.
* **Argomenti e Opzioni:**
  + filepath: Percorso al file CSV/XLSX contenente i dati storici arricchiti.
  + --league: Lega di default da assegnare ai record se la colonna "Lega" manca nel file o in una riga specifica.
* **Esempio d'Uso:**  
  Bash  
  php artisan players:import-advanced-stats storage/app/import/dati\_fbref\_serie\_b\_23-24.csv --league="Serie B"
* **File:** app/Console/Commands/PlayersImportAdvancedStats.php (codice fornito e discusso).

**2.4. Modifiche alla Configurazione: config/projection\_settings.php**

* **Nuova Sezione:** player\_stats\_league\_conversion\_factors
* **Struttura:** Un array associativo dove la chiave è il league\_name (es. "Serie B") e il valore è un altro array associativo con i fattori di conversione per statistica (es. ['goals\_scored' => 0.60, 'assists' => 0.65, 'avg\_rating' => 0.95]). La "Serie A" dovrebbe avere fattori 1.0.
* **Scopo:** Fornire al ProjectionEngineService i moltiplicatori per "tradurre" le performance da una lega di origine al livello atteso della Serie A.

**2.5. Modifiche al Servizio: App\Services\ProjectionEngineService.php**

* **Metodo calculateWeightedAverageStats(Player $player) Modificato:**
  + **Lettura league\_name:** Per ogni record HistoricalPlayerStat, legge il valore del nuovo campo league\_name.
  + **Retrocompatibilità:** Se league\_name è NULL (per i dati storici importati prima di questa modifica, tramite TuttiHistoricalStatsImport.php), il sistema assume un default (es. "Serie A" o applica un fattore di conversione neutro di 1.0).
  + **Applicazione Fattori:** Recupera i player\_stats\_league\_conversion\_factors dalla configurazione. Per ogni statistica per partita calcolata da un record storico (es. gol/partita, assist/partita, media voto grezza), moltiplica il valore per il fattore di conversione corrispondente alla league\_name di quel record *prima* di includerla nella media ponderata stagionale.
  + **Log Migliorato:** Per tracciare l'applicazione dei fattori.

**3. Flusso Operativo Consigliato per l'Utente**

1. **Preparazione Dati Storici Arricchiti (Manuale, es. da FBRef.com):**
   * L'utente consulta fonti esterne per statistiche dettagliate, specialmente per giocatori di squadre neopromosse o con storico in leghe diverse dalla Serie A.
   * Viene preparato un file CSV/XLSX. **Colonne Chiave Obbligatorie per l'Importer PlayerSeasonStatsImport:**
     + PlayerFantaPlatformID (o un alias come idfanta): L'ID Fantacalcio del giocatore, che DEVE esistere nella tabella players.
     + StagioneAnnoInizio (es. 2023 per la stagione 2023-24).
     + NomeLega (o Lega): Es. "Serie A", "Serie B".
   * **Altre Colonne Consigliate/Supportate dall'Importer:** NomeGiocatoreDB, NomeSquadraDB, RuoloFantacalcio, PG (Partite Giocate), MinutiGiocati, Reti, Assist, RigoriTentati, RigoriSegnati, Ammonizioni, Espulsioni, Autogol, GolSubitiPortiere, RigoriParatiPortiere, MediaVotoOriginale. (L'importer è flessibile ai nomi delle intestazioni grazie alla normalizzazione).
2. **Importazione Dati Storici Arricchiti:**
   * Utilizzare il comando:  
     Bash  
     php artisan players:import-advanced-stats storage/app/import/mio\_file\_dati\_fbref.csv --league="Serie B"  
     (L'opzione --league serve come fallback se la colonna "Lega" manca in alcune righe del CSV).
3. **Configurazione Fattori di Conversione:**
   * L'utente definisce e calibra i player\_stats\_league\_conversion\_factors in config/projection\_settings.php. L'analisi dei dati reali (come quelli di FBRef) è una guida fondamentale per questa calibrazione.
4. **Aggiornamento Tier Squadre:**
   * Assicurarsi che la tabella team\_historical\_standings sia aggiornata (tramite API o import CSV con teams:import-standings-file).
   * Eseguire php artisan teams:set-active-league --target-season-start-year=YYYY ... per definire le squadre partecipanti alla Serie A per la stagione di proiezione.
   * Eseguire php artisan teams:update-tiers YYYY-YY per calcolare i tier dinamici.
5. **Generazione e Analisi Proiezioni Giocatori:**
   * Utilizzare php artisan test:projection {playerId} per giocatori chiave.
   * Il ProjectionEngineService ora utilizzerà lo storico "tradotto" per lega.
   * Analizzare gli output, confrontarli con aspettative, CRd, e dati qualitativi.
   * Iterare sulla calibrazione dei parametri (fattori di conversione, moltiplicatori di tier, default stats) per affinare le proiezioni.

**4. Benefici**

* **Proiezioni più Accurate:** Specialmente per giocatori che cambiano lega (es. da Serie B a Serie A) o con storico in leghe di diverso livello.
* **Maggiore Flessibilità:** Capacità di incorporare dati storici più dettagliati da fonti diverse.
* **Migliore Calibrazione:** L'utente può usare dati esterni (come FBRef) per informare la configurazione dei fattori di conversione.
* **Retrocompatibilità Mantenuta:** L'importer esistente per i file XLSX standard (TuttiHistoricalStatsImport.php) non viene modificato.

Ora, analizziamo i log che hai fornito per players:import-advanced-stats.

I log mostrano:

[PlayerSeasonStatsImport] Giocatore con FantaPlatformID IDADRIANSEMPER\_PLACEHOLDER non trovato nel DB. Riga saltata. (e così via per tutti i giocatori).

Questo conferma che il problema principale nell'esecuzione del comando players:import-advanced-stats è che i valori che hai usato nella colonna PlayerFantaPlatformID del tuo file fbref\_pisa\_2024-25.csv (es. "IDADRIANSEMPER\_PLACEHOLDER") non sono ID numerici validi che esistono nella colonna fanta\_platform\_id della tua tabella players.

**Azione Correttiva Immediata per Te:**

1. **Prepara il File CSV Correttamente:** Per ogni giocatore nel tuo file fbref\_pisa\_2024-25.csv (e altri file simili che preparerai):
   * **Trova il fanta\_platform\_id ufficiale** di quel giocatore. Questo ID proviene dal listone ufficiale del Fantacalcio che hai importato usando MainRosterImport.php. Se un giocatore è nuovo e non era nel listone precedente, dovrai attendere il nuovo listone per ottenere il suo ID ufficiale, oppure (meno ideale) assegnargli un ID fittizio univoco e assicurarti che esista un record corrispondente nella tabella players con quell'ID fittizio.
   * **Sostituisci i placeholder** (es. "IDADRIANSEMPER\_PLACEHOLDER") nella colonna PlayerFantaPlatformID del tuo CSV con questi ID numerici corretti.
2. **Assicurati che le Intestazioni del CSV corrispondano (dopo normalizzazione) a quelle attese da PlayerSeasonStatsImport.php**:
   * Le chiavi che l'importer cerca (dopo averle convertite in lowercase senza spazi) sono ad esempio: playerfantaplatformid, nomegiocatore, nomesquadradb, stagioneannoinizio, nomelega, ruolofantacalcio, pg o partitegiocate, min o minutigiocati, reti o golsegnati, assist, etc. Verifica che le intestazioni nel tuo CSV siano chiare e che l'importer le possa mappare correttamente.

Una volta che il CSV è corretto con i PlayerFantaPlatformID validi, riesegui il comando:

php artisan players:import-advanced-stats storage/app/import/fbref\_pisa\_2024-25.csv --league="Serie B"

Dovresti vedere un numero di Processed diverso da zero.