
REQUISITI NON FUNZIONALI

1. Efficienza delle Prestazioni

L'obiettivo è garantire tempi di risposta rapidi per l'utente finale anche sotto carico.

- **Requisito Originale:** Operazioni di lettura < 2 sec con dataset N=10000.
- **Espansione Tecnica:**
 - **Latenza:** Le operazioni di lettura critiche (es. visualizzazione classifica, calendario) devono mantenere un tempo di risposta inferiore a **2 secondi** per il 95° percentile delle richieste (p95).
 - **Carico Dati:** Il sistema deve garantire queste prestazioni con un volume dati iniziale fino a **10.000 record**, con degrado lineare controllato oltre tale soglia.
 - **Concorrenza:** Il sistema deve supportare fino a [Inserire numero, es. 500] utenti concorrenti senza degrado delle prestazioni superiore al 20%.
- **Priorità:** Alta

2. Scalabilità

Capacità del sistema di gestire carichi di lavoro crescenti.

- **Requisito Originale:** Crescita numero stagioni/dati; separazione livelli UI/Business/Persistenza.
- **Espansione Tecnica:**
 - **Architettura:** Adozione rigorosa di un'architettura a tre livelli (Three-Tier Architecture) per disaccoppiare Frontend, Logica di Business e Accesso ai Dati, facilitando la scalabilità orizzontale dei singoli nodi.
 - **Database:** Il database deve supportare l'indicizzazione efficiente per gestire la crescita storica dei dati (stagioni passate) senza rallentare le query attive.
- **Priorità:** Media

3. Disponibilità (Availability)

Garanzia di accesso al servizio.

- **Requisito Originale:** Disponibile al 100% per visualizzazione.
- **Espansione Tecnica:**
 - **SLA (Service Level Agreement):** L'obiettivo per le funzionalità di sola lettura è un uptime del **99.9%** (il 100% è teorico, si consiglia di prevedere finestre di manutenzione).
 - **Failover:** In caso di guasto del database principale, il sistema deve poter passare in modalità "Sola Lettura" utilizzando una replica o una cache per garantire la visualizzazione.
- **Priorità:** Media

4. Sicurezza

Protezione dei dati e degli accessi.

- **Requisito Originale:** Password memorizzate; autenticazione admin.
- **Espansione Tecnica:**
 - **Gestione Credenziali:** Le password non devono mai essere salvate in chiaro. Utilizzo obbligatorio di algoritmi di hashing robusti (es. BCrypt o Argon2) con *salt* univoco per utente.
 - **Autenticazione:** Implementazione di un sistema di autenticazione sicuro (es. OAuth2 o JWT) per le funzioni amministrative.
 - **Trasmissione Dati:** Tutto il traffico deve avvenire su protocollo crittografato (HTTPS/TLS 1.2+).
- **Priorità:** Alta

5. Manutenibilità

Facilità di modifica e aggiornamento del software.

- **Requisito Originale:** Codice modulare, documentazione API, pattern DAO/Service/Controller.

- **Espansione Tecnica:**
 - **Design Pattern:** Adozione rigorosa dei pattern **MVC** (Model-View-Controller) e **Dependency Injection**. Separazione netta tramite DAO (Data Access Object) e Service Layer.
 - **Qualità del Codice:** Adozione di standard di coding (es. Java Code Conventions) verificati tramite strumenti di analisi statica (es. SonarQube).
 - **Documentazione:** Le API REST devono essere documentate tramite standard OpenAPI (Swagger).
 - **Priorità:** Alta
-

6. Integrità dei Dati

Coerenza e affidabilità delle informazioni.

- **Requisito Originale:** Transazioni atomiche per inserimento/aggiornamento.
 - **Espansione Tecnica:**
 - **Transazionalità (ACID):** Le operazioni critiche che coinvolgono più tabelle (es. inserimento risultato partita + ricalcolo classifica) devono essere avvolte in una singola transazione atomica. In caso di errore in uno step, l'intera operazione deve effettuare il *rollback*.
 - **Vincoli:** Utilizzo di vincoli di integrità referenziale (Foreign Keys) a livello di database per prevenire orfani.
 - **Priorità:** Alta
-

7. Testabilità

Capacità di verificare il corretto funzionamento del software.

- **Requisito Originale:** Unit/integration tests, copertura 60-80%, CI pipeline.
- **Espansione Tecnica:**
 - **Copertura:** Target di *Code Coverage* minimo del **60%** (ottimale **80%**) per la business logic (Core).
 - **Strategia di Test:** Implementazione della piramide dei test:
 1. **Unit Test:** Isolati (uso di Mockito/Junit).
 2. **Integration Test:** Verifica interazione Database/API.
 3. **E2E Test:** Flussi critici utente.

- **CI/CD:** La pipeline di Continuous Integration deve bloccare il merge se i test falliscono o se la copertura scende sotto la soglia minima.
 - **Priorità:** Media
-

8. Usabilità

Facilità d'uso per l'utente finale.

- **Requisito Originale:** Interfaccia intuitiva, pochi click, accessibilità base.
 - **Espansione Tecnica:**
 - **Efficienza d'uso:** Le operazioni frequenti (es. inserimento risultato) devono essere completabili in massimo **3 click**.
 - **Accessibilità:** Conformità agli standard **WCAG 2.1 livello AA** (contrasto colori sufficiente, etichette per screen reader, navigabilità da tastiera).
 - **Responsività:** L'interfaccia deve adattarsi fluidamente a dispositivi desktop, tablet e mobile.
 - **Priorità:** Alta
-

9. Localizzazione e Internazionalizzazione (i18n)

Adattabilità a lingue e formati regionali.

- **Requisito Originale:** Default lingua IT.
- **Espansione Tecnica:**
 - **Resource Bundles:** Nessuna stringa di testo deve essere *hardcoded* nel codice sorgente. Tutti i testi devono risiedere in file di proprietà esterni per facilitare future traduzioni.
 - **Formati:** Gestione corretta di date, valute e numeri secondo il *locale* italiano (it-IT) di default.
- **Priorità:** Bassa

10. Logging e Monitoraggio

Osservabilità del sistema.

- **Requisito Originale:** Log centralizzati (INFO/WARN/ERROR), metriche base.
- **Espansione Tecnica:**
 - **Livelli di Log:** Utilizzo standardizzato dei livelli: ERROR (richiede intervento immediato), WARN (potenziale problema), INFO (audit trail).
 - **Privacy:** I log non devono mai contenere dati sensibili (PII) o password in chiaro.
 - **Retention:** I log devono essere conservati per almeno [es. 30 giorni] per scopi di debug e audit.
- **Priorità:** Media

11. Backup e Disaster Recovery

Sicurezza contro la perdita di dati.

- **Requisito Originale:** Backup periodico e ripristino documentato.
- **Espansione Tecnica:**
 - **RPO (Recovery Point Objective):** Perdita massima di dati accettabile fissata a **24 ore** (backup giornaliero notturno).
 - **RTO (Recovery Time Objective):** Tempo massimo per il ripristino del servizio fissato a **4 ore**.
 - **Verifica:** Procedura di test del ripristino (restore) da effettuare semestralmente.
- **Priorità:** Media

12. Conformità Legale e Privacy

Rispetto delle normative vigenti.

- **Requisito Originale:** GDPR, consensi, cancellazione.
- **Espansione Tecnica:**

- **Diritto all'Oblío:** Implementazione di una funzione tecnica che permetta l'anonimizzazione o la cancellazione irreversibile dei dati utente su richiesta (Art. 17 GDPR).
 - **Minimizzazione:** Raccolta limitata ai soli dati strettamente necessari per il servizio.
 - **Cookie Policy:** Banner per la gestione granulare dei consensi ai cookie.
 - **Priorità:** Alta
-

13. Portabilità

Capacità di eseguire il software in ambienti diversi.

- **Requisito Originale:** JVM standard (OpenJDK 11+).
- **Espansione Tecnica:**
 - **Ambiente di Runtime:** Piena compatibilità con **OpenJDK 11 LTS** o versioni superiori.
 - **Containerizzazione (Opzionale ma consigliato):** Predisposizione dell'applicazione per essere pacchettizzata in container Docker, garantendo l'indipendenza dal sistema operativo sottostante (Linux/Windows).
- **Priorità:** Media