

# Documento di Analisi dei Requisiti (Iterazione 0)

Sistema di Predizione e Ottimizzazione Strategie F1

## Team di Sviluppo

Andrea Birolini (mat. 123456)

Ivan Caccamo (mat. 654321)

Luca Rossi (mat. 1086223)

23 novembre 2025

## Indice

<b>1</b>	<b>Introduzione e Visione</b>	<b>2</b>
<b>2</b>	<b>Tool Chain e Metodologia</b>	<b>2</b>
<b>3</b>	<b>Attori del Sistema</b>	<b>2</b>
<b>4</b>	<b>Requisiti Funzionali</b>	<b>2</b>
4.1	Gestione Dati e Previsione (Backend ML) . . . . .	2
4.2	Ottimizzazione Strategia (Core Java) . . . . .	3
<b>5</b>	<b>Requisiti Non Funzionali</b>	<b>3</b>
<b>6</b>	<b>Modellazione dei Requisiti</b>	<b>3</b>
<b>7</b>	<b>Architettura del Sistema</b>	<b>3</b>
<b>8</b>	<b>Pianificazione Iterazioni Future</b>	<b>3</b>

# 1 Introduzione e Visione

Il progetto **SPS-F1** mira a sviluppare un sistema software di supporto decisionale per le strategie di gara in Formula 1. Il sistema combina tecniche di *Machine Learning* per la predizione dei parametri di degrado degli pneumatici con algoritmi di ottimizzazione esatta per determinare la strategia di pit-stop ideale.

## 2 Tool Chain e Metodologia

In accordo con le linee guida del corso, il team adotterà il processo di sviluppo agile AMDD. La tool chain selezionata è la seguente:

- **Linguaggio Core (Backend/Algoritmi):** Java (JDK 17+).
- **Machine Learning Service:** Python (Librerie: Scikit-Learn, Pandas, Flask/FastAPI).
- **Gestione Versionamento:** GitHub.
- **Build Automation:** Maven.
- **Testing:** JUnit 5 (Unit Testing), Postman (API Testing).
- **Analisi Statica:** STAN4J.
- **Documentazione:** LaTeX.

## 3 Attori del Sistema

- **Ingegnere di Strategia (Stratega):** Utente principale che configura lo scenario di gara e analizza le strategie proposte.
- **Data Engineer (Admin):** Responsabile dell'aggiornamento dei dati storici e del riaddestramento dei modelli ML.

## 4 Requisiti Funzionali

I seguenti requisiti definiscono le funzionalità principali previste per il sistema.

### 4.1 Gestione Dati e Previsione (Backend ML)

- **RF-01 (Importazione Storico):** Il sistema deve permettere l'importazione di dati storici (tempi sul giro, usura) da fonti esterne (CSV).
- **RF-02 (Training Modello):** Il sistema deve poter addestrare un modello di regressione sui dati storici per apprendere il comportamento delle mescole (C1-C6).
- **RF-03 (Predizione Parametri):** Dato un circuito e una mescola, il sistema deve predire:
  - Tempo base sul giro ( $T_{base}$ ).
  - Fattore di degrado per giro ( $D_{rate}$ ).

## 4.2 Ottimizzazione Strategia (Core Java)

- **RF-04 (Configurazione Scenario):** Lo stratega deve poter inserire: numero di giri totali, mescole disponibili, tempo perso in pit-lane.
- **RF-05 (Calcolo Strategia Ottimale):** Il sistema deve calcolare la sequenza di soste e mescole che minimizza il tempo totale di gara.
- **RF-06 (Visualizzazione Risultati):** Il sistema deve presentare la strategia vincente (es. "Giro 1-20: Soft, Giro 21-53: Hard") e il tempo totale stimato.

## 5 Requisiti Non Funzionali

- **RNF-01 (Performance):** L'algoritmo di ottimizzazione deve restituire la strategia migliore in meno di 3 secondi per una gara standard (50-70 giri).
- **RNF-02 (Interoperabilità):** La comunicazione tra il modulo Java e il modulo Python deve avvenire tramite API REST standard (JSON).
- **RNF-03 (Affidabilità):** Il sistema deve gestire scenari in cui mancano dati storici per un circuito, utilizzando valori di default sicuri.

## 6 Modellazione dei Requisiti

Di seguito viene riportato il Diagramma dei Casi d'Uso che illustra le interazioni tra gli attori e il sistema SPS-F1.

## 7 Architettura del Sistema

La topologia del sistema prevede un'architettura a microservizi ibrida, come illustrato nel seguente Diagramma di Deployment.

## 8 Pianificazione Iterazioni Future

- **Iterazione 1:** Implementazione algoritmo di ottimizzazione (Java) con dati mock.
- **Iterazione 2:** Sviluppo servizio ML (Python) e integrazione API.
- **Iterazione 3:** Sviluppo Interfaccia Utente e finalizzazione documentazione.

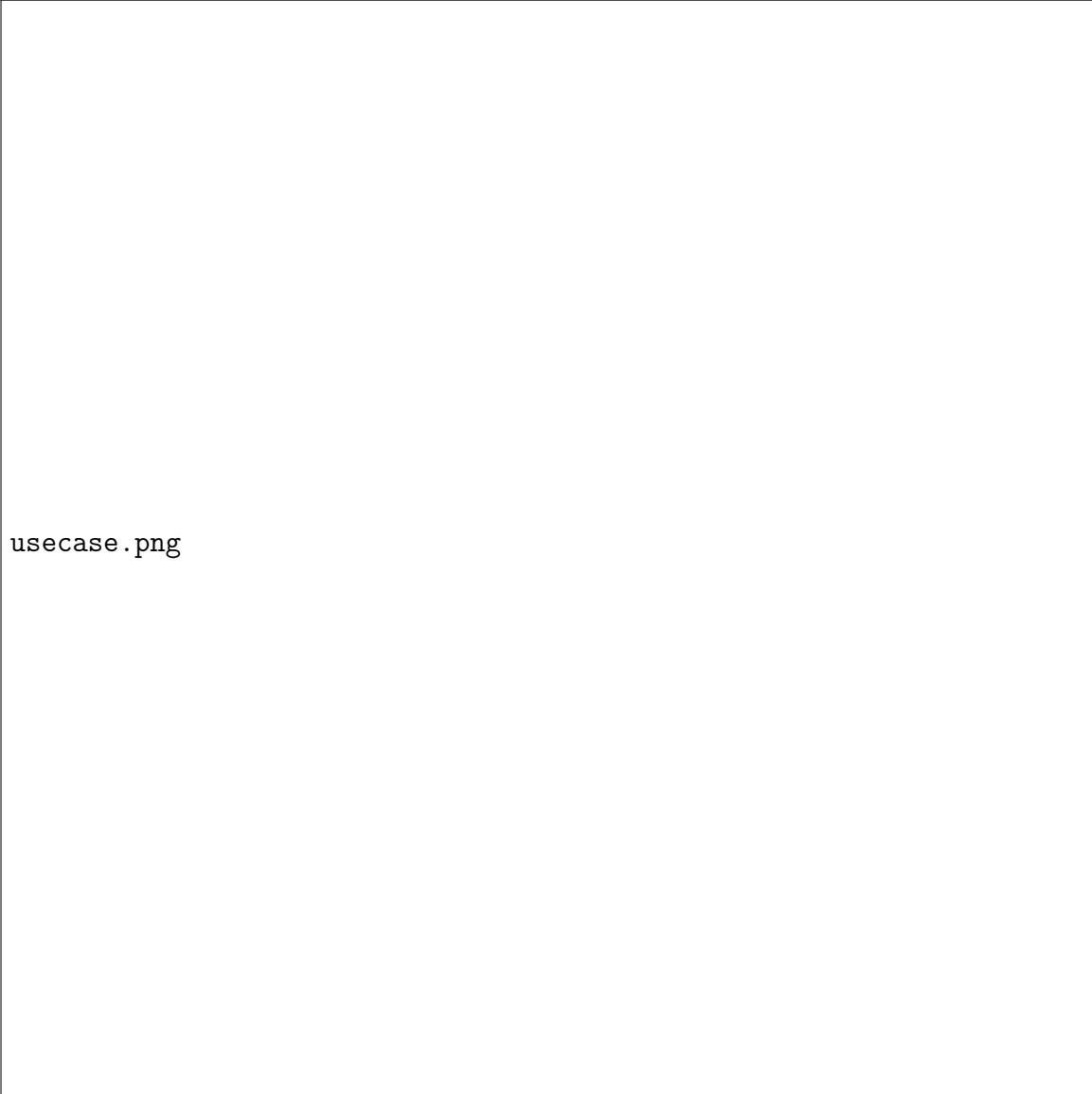


Figura 1: Diagramma dei Casi d’Uso - Iterazione 0

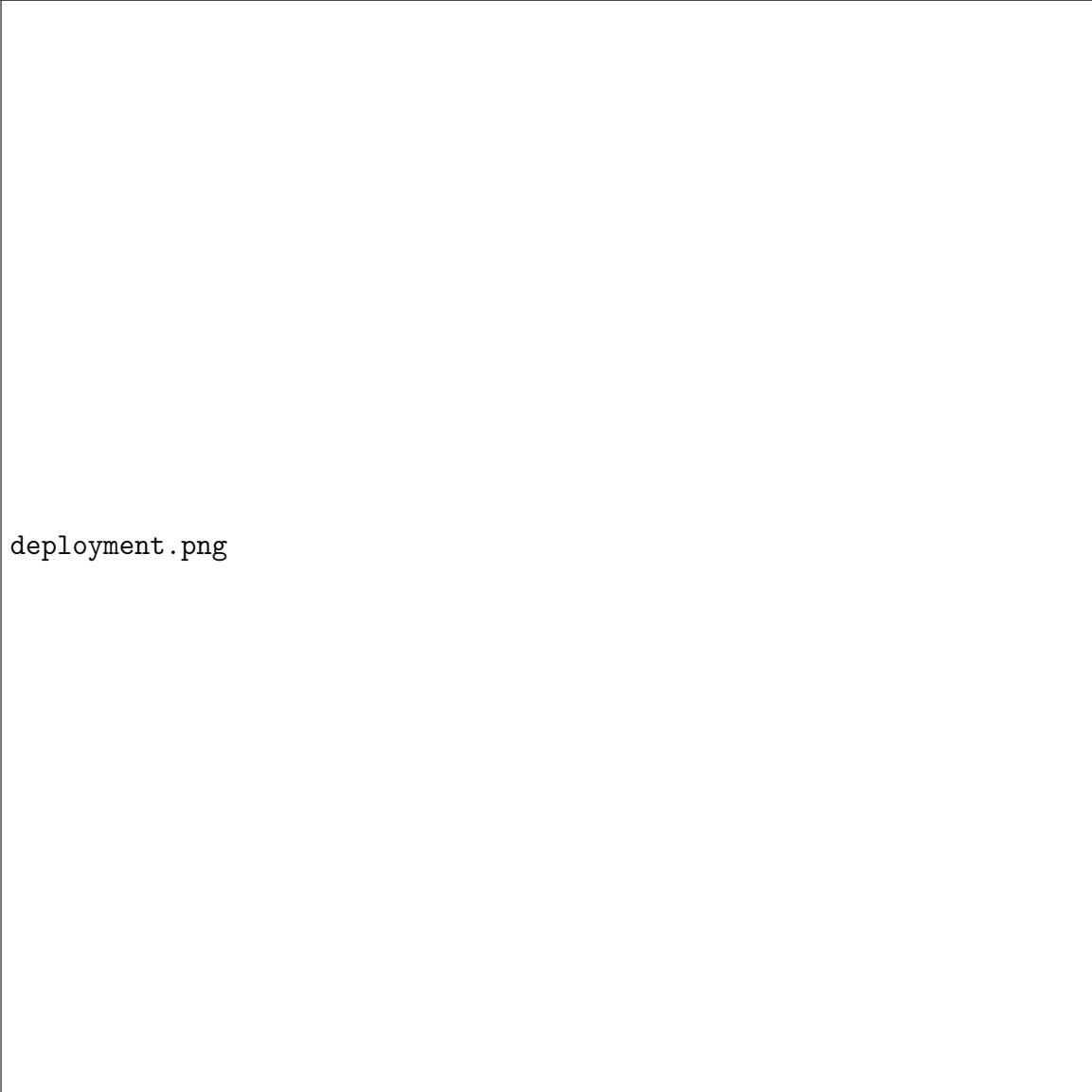


Figura 2: Diagramma di Deployment - Configurazione Iniziale