

FitJournal Analytics - Sistema Inteligente de Diario Deportivo y Nutricional

Descripción del Proyecto

Un sistema completo que combina el registro de actividades deportivas y alimentación con análisis estadístico avanzado, machine learning y visualizaciones interactivas para predecir mejoras en la salud y optimizar rutinas personales.

Características Principales

1. Sistema de Registro Manual Inteligente

- **Diario deportivo:** Formularios intuitivos para ejercicios, duración, intensidad, sensaciones
- **Diario nutricional:** Base de datos de alimentos local, calculadora de macronutrientes
- **Interfaz adaptativa:** Formularios que se adaptan según el tipo de actividad/comida
- **Análisis de patrones:** El sistema aprende de tus entradas para sugerir valores comunes

2. Análisis Estadístico y Predictivo

- **Modelos ML:** Predicción de pérdida de peso, ganancia muscular, resistencia
- **Correlaciones:** Relación entre dieta, ejercicio y métricas de salud
- **Tendencias temporales:** Análisis de progreso a corto y largo plazo
- **Recomendaciones personalizadas:** IA que sugiere mejoras basadas en patrones

3. Visualizaciones Interactivas

- **Dashboards dinámicos:** Gráficos en tiempo real con Plotly/Dash
- **Mapas de calor:** Distribución de entrenamientos y patrones alimentarios
- **Análisis comparativo:** Benchmarking personal vs. objetivos
- **Reportes automáticos:** PDFs con resúmenes semanales/mensuales

Librerías y Tecnologías Innovadoras

Core del Sistema

```
python
```

```
# Procesamiento de datos y análisis
```

```
import pandas as pd
```

```
import numpy as np
```

```
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
```

```
from sklearn.cluster import KMeans
```

```
from sklearn.preprocessing import StandardScaler
```

```
from scipy import stats
```

```
import joblib
```

```
# Visualización avanzada
```

```
import plotly.graph_objects as go
```

```
import plotly.express as px
```

```
from plotly.subplots import make_subplots
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
# Web framework y dashboard
```

```
import streamlit as st
```

```
import dash
```

```
from dash import dcc, html, Input, Output
```

Características Avanzadas

```
python
```

```
# Análisis nutricional y deportivo
```

```
from nutrition_data import SpanishFoodDatabase # Base de datos española
```

```
import exercise_database # Base de datos de ejercicios MET
```

```
# Base de datos y almacenamiento
```

```
import sqlite3
```

```
from sqlalchemy import create_engine
```

```
import pickle
```

```
# Análisis temporal y series
```

```
from statsmodels.tsa.arima.model import ARIMA
```

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
from prophet import Prophet # Para predicciones temporales
```

```
# Análisis estadístico avanzado
```

```
from scipy.optimize import minimize
```

```
from sklearn.metrics import mean_absolute_error, r2_score
```

Librerías Únicas y Originales

```
python
```

```
# Análisis biomédico y nutricional
```

```
from nutrition_analysis import MacroCalculator, MicronutrientTracker
```

```
from exercise_science import METCalculator, CalorieEstimator
```

```
from health_metrics import BMRCalculator, BodyCompositionEstimator
```

```
# Procesamiento de lenguaje natural para análisis de textos
```

```
import spacy
```

```
from textblob import TextBlob
```

```
import nltk
```

```
# Análisis de patrones temporales
```

```
import holidays # Para considerar días festivos en análisis
```

```
from datetime import datetime, timedelta
```

```
import calendar
```

```
# Bases de datos nutricionales locales
```

```
from food_composition_spain import BEDCA_Database # Base española oficial
```

```
import nutrient_requirements # Requerimientos nutricionales por demografía
```

Arquitectura del Sistema

1. Módulo de Captura de Datos

- **SmartForms:** Formularios inteligentes que se adaptan al contexto
- **FoodDatabase:** Base de datos nutricional española integrada
- **ActivityTracker:** Sistema de seguimiento manual con autocompletado
- **DataValidation:** Validación inteligente y detección de inconsistencias

2. Motor de Análisis

- **HealthMetricsEngine:** Cálculo de métricas de salud (BMI, BFP, VO2 max estimado)
- **CorrelationAnalyzer:** Análisis de correlaciones entre variables
- **PredictiveModels:** Modelos ML para predicciones de progreso
- **PatternRecognition:** Identificación de patrones y anomalías

3. Sistema de Recomendaciones

- **NutritionOptimizer:** Optimización de dietas basada en objetivos
- **WorkoutPlanner:** Planificación inteligente de entrenamientos
- **ProgressTracker:** Seguimiento y ajuste de objetivos
- **HealthAlerts:** Alertas proactivas sobre salud

4. Interfaz y Visualización

- **InteractiveDashboard:** Dashboard principal con Streamlit/Dash
- **ReportGenerator:** Generación automática de reportes
- **DataExporter:** Exportación a diferentes formatos
- **MobileView:** Versión optimizada para móviles

Características Técnicas Innovadoras

1. Análisis Predictivo Avanzado

```
python

class HealthPredictor:
    def predict_weight_loss(self, current_data, target_date):
        # Modelo que considera metabolismo basal, ejercicio, dieta
        pass

    def estimate_performance_improvement(self, exercise_history):
        # Predicción de mejoras en rendimiento deportivo
        pass

    def nutrition_impact_analysis(self, meal_plans):
        # Análisis del impacto nutricional en la salud
        pass
```

2. Procesamiento de Imágenes Inteligente

```
python

class FoodRecognition:
    def identify_food(self, image):
        # Identificación automática de alimentos
        pass

    def estimate_portions(self, image, reference_object):
        # Estimación de porciones usando referencias
        pass

    def nutrition_extraction(self, identified_foods):
        # Extracción automática de información nutricional
        pass
```

3. Análisis de Señales Biológicas

```
python
```

```
class BiometricAnalyzer:
    def heart_rate_variability(self, hr_data):
        # Análisis de variabilidad de frecuencia cardíaca
        pass

    def sleep_quality_assessment(self, sleep_data):
        # Evaluación de calidad del sueño
        pass

    def stress_level_detection(self, biometric_data):
        # Detección de niveles de estrés
        pass
```

Funcionalidades Destacadas

1. Registro Manual Inteligente

- Formularios que aprenden de tus patrones de entrada
- Autocompletado contextual basado en historial personal
- Detección automática de inconsistencias en datos

2. Base de Datos Nutricional Española Integrada

- Integración completa con BEDCA (Base oficial española)
- Búsqueda inteligente por ingredientes y platos regionales
- Cálculo automático de macros/micros por porción personalizada

3. Formularios Adaptativos Inteligentes

- Formularios que "aprenden" de tus patrones de entrada
- Autocompletado basado en historial personal
- Validación en tiempo real de datos inconsistentes

4. Sistema de Objetivos Dinámicos

- Establecimiento de metas SMART personalizadas
- Ajuste automático de objetivos basado en progreso
- Alertas cuando te desvías de tu plan

5. Análisis Comparativo Temporal

- Comparación con períodos anteriores (mismo mes del año pasado)
- Análisis de estacionalidad en hábitos
- Identificación de patrones cíclicos personales

Métricas y KPIs del Sistema

Salud Física

- Composición corporal (peso, grasa, músculo)
- Capacidad cardiovascular (VO2 max, frecuencia cardíaca en reposo)
- Fuerza y resistencia (1RM estimado, tiempo de fatiga)
- Flexibilidad y movilidad

Salud Nutricional

- Balance calórico
- Distribución de macronutrientes
- Micronutrientes esenciales
- Índice de inflamación dietética

Bienestar General

- Calidad del sueño
- Niveles de estrés
- Estado de ánimo (mediante cuestionarios)
- Energía percibida

Implementación Técnica

Base de Datos

```
sql

-- Estructura principal de tablas
CREATE TABLE users (id, name, birth_date, gender, height, goals);
CREATE TABLE workouts (id, user_id, date, type, duration, intensity, calories);
CREATE TABLE nutrition (id, user_id, date, meal_type, food_items, calories, macros);
CREATE TABLE biometrics (id, user_id, date, weight, body_fat, heart_rate);
CREATE TABLE predictions (id, user_id, date, metric, predicted_value, confidence);
```

API Design

```
python
```

RESTful API endpoints

/api/v1/users/{user_id}/workouts

/api/v1/users/{user_id}/nutrition

/api/v1/users/{user_id}/predictions

/api/v1/users/{user_id}/reports

/api/v1/food/recognize (POST con imagen)

/api/v1/voice/process (POST con audio)

Deployment

- **Backend:** FastAPI + PostgreSQL + Redis
- **Frontend:** Streamlit/Dash para prototipo, React para versión final
- **ML Pipeline:** MLflow para versionado de modelos
- **Containerización:** Docker + Docker Compose
- **Cloud:** AWS/GCP con auto-scaling

Entregables del TFC

1. **Aplicación Funcional** - Sistema completo desplegado
2. **Modelos ML Entrenados** - Algoritmos de predicción validados
3. **Dashboard Interactivo** - Visualizaciones en tiempo real
4. **Documentación Técnica** - Arquitectura y APIs documentadas
5. **Dataset Sintético** - Datos de prueba para demostración
6. **Análisis de Resultados** - Validación estadística de predicciones
7. **Manual de Usuario** - Guía completa de uso
8. **Video Demo** - Demostración de todas las funcionalidades

Valor Diferencial

- **Enfoque Holístico:** Combina deporte, nutrición y bienestar
- **IA Personalizada:** Modelos adaptados a cada usuario
- **Tecnología Cutting-edge:** Uso de librerías especializadas poco comunes
- **Aplicabilidad Real:** Solución práctica para un problema cotidiano
- **Escalabilidad:** Arquitectura preparada para crecimiento
- **Innovación:** Características únicas como análisis por voz e imagen

Este proyecto demuestra dominio técnico avanzado, aplicación práctica de ML/IA, y capacidad de integrar múltiples tecnologías en una solución coherente y útil.