# TalentFinder - Architecture Documentation

**Table of Contents**

1. Introduction

2. General System Architecture

3. System Modules

4. Dependencies and Tools

5. Communication between Modules

# 1. Introduction TalentFinder is a project aimed at analyzing football data and evaluating player talents based on collected information from various sources. This documentation describes the system architecture to understand the project structure.

# 2. General System Architecture The TalentFinder system consists of several modules, each responsible for a different aspect of data processing and analysis. These modules work together to collect, process, and analyze football data. These modules are *scrapers*, *edit\_data, database*, and *ml*.

# 3. System Modules

**1. Data Collection Module (Scrapers)**  
- Responsible for fetching data from external sources such as sofascore and transfermarkt, along with cleaning data from unnecessary information.  
- Scripts:  
 - scrappers/sofascore/get\_players\_ids.py  
 - scrappers/sofascore/get\_ratings\_from\_sofascore.py  
 - scrappers/transfermarkt/transfermarkt\_scrapper.py  
  
**2. Data Processing Module (Edit\_data)**  
- Responsible for processing the collected data and preparing it for analysis.  
- Scripts:  
 - database/add\_flags.py  
 - database/flags\_KM.py  
 - database/merge\_two\_csv\_with\_flags.ipynb

**3. Database (Database)**  
- The database where the collected data is stored.  
  
**4. Data Analysis Module (ML)**  
- sible for analyzing data and training predictive models.  
- Scripts:  
 - ml/random\_forest/random\_forest.ipynb   
 - *ml/* *clasification /* *Clasification\_transfer.ipynb*  
 - *ml/* *regresion /* *Regresion\_transfers.ipynb*

# 4. Dependencies and Tools

The TalentFinder project uses a variety of tools and libraries essential for its proper functioning. All dependencies are listed in the *requirements.txt* file.  
  
**Example dependencies:**  
- pandas  
- numpy  
- requests  
- beautifulsoup4  
- scikit-learn  
- jupyter  
- matplotlib  
  
To install all dependencies, run:  
*pip install -r requirements.txt*

# 5. Communication between Modules

Communication between modules occurs through the exchange of data in CSV format and through database connections. Each module saves its results to appropriate files, which are then used by other modules.