Proyecto Integrador

Bootcamp Data & Analytics

Analitica Noviembre

31 de Mayo de 20223

BUILD YOUR SQUAD

Ernesto Davogustto

Índice

Aspectos Generales

Escuelitas S.A es una empresa especializada en brindar asesoría para la contratación de nuevos jugadores en la NBA. Han contratado nuestros servicios con el objetivo de analizar los datos relacionados y desarrollar una herramienta de visualización que les permita tomar decisiones informadas.

Con el fin de llevar a cabo este proyecto, Escuelitas S.A nos ha proporcionado una base de datos alojada en la Plataforma de Microsoft Azure. Esta base de datos contiene varias tablas con esquemas específicos que almacenan información relevante para la contratación de jugadores en la NBA.

Nuestra tarea principal es analizar los datos presentes en estas tablas, realizar las transformaciones necesarias y generar visualizaciones interactivas que permitan a Escuelitas S.A explorar y comprender fácilmente la información recopilada. Estas visualizaciones brindarán una visión clara de las estadísticas y características de los jugadores, lo que facilitará la toma de decisiones basadas en datos sólidos y objetivos.

A continuación, presentaremos las tablas junto con sus esquemas correspondientes, lo que nos permitirá tener una mejor comprensión de la estructura y los tipos de datos presentes en la base de datos proporcionada.

- Teams:
    - LEAGUE_ID:integer
    - TEAM_ID:integer
    - MIN_YEAR:integer
    - MAX_YEAR:integer
    - ABBREVIATION:string
    - NICKNAME:string
    - YEARFOUNDED:integer
    - CITY:string
    - ARENA:string
    - ARENACAPACITY:integer
    - OWNER:string
    - GENERALMANAGER:string

    - HEADCOACH:string
    - DLEAGUEAFFILIATION:string
- Ranking:
    - TEAM_ID:integer
    - LEAGUE_ID:integer
    - SEASON_ID:integer
    - STANDINGSDATE:date
    - CONFERENCE:string
    - TEAM:string
    - G:integer
    - W:integer
    - L:integer
    - W_PCT:double
    - HOME_RECORD:string
    - ROAD_RECORD:string
    - RETURNTOPLAY:double
- Players:
    - PLAYER_NAME:string
    - TEAM_ID:integer
    - PLAYER_ID:integer
    - SEASON:integer
- Games: GAME_DATE_EST:date
    - GAME_ID:integer
    - GAME_STATUS_TEXT:string
    - HOME_TEAM_ID:integer
    - VISITOR_TEAM_ID:integer
    - SEASON:integer
    - TEAM_ID_home:integer
    - PTS_home:double
    - FG_PCT_home:double
    - FT_PCT_home:double
    - FG3_PCT_home:double
    - AST_home:double
    - REB_home:double
    - TEAM_ID_away:integer
    - PTS_away:double

    - FG_PCT_away:double
    - FT_PCT_away:double
    - FG3_PCT_away:double
    - AST_away:double
    - REB_away:double
    - HOME_TEAM_WINS:integer
- Games_details:
    - GAME_ID:integer
    - TEAM_ID:integer
    - TEAM_ABBREVIATION:string
    - TEAM_CITY:string
    - PLAYER_ID:integer
    - PLAYER_NAME:string
    - NICKNAME:string
    - START_POSITION:string
    - COMMENT:string
    - MIN:string
    - FGM:double
    - FGA:double
    - FG_PCT:double
    - FG3M:double
    - FG3A:double
    - FG3_PCT:double
    - FTM:double
    - FTA:double
    - FT_PCT:double
    - OREB:double
    - DREB:double
    - REB:double

Para resolver la problemática de Escuelitas S.A., se va a implementar una arquitectura que veremos más adelante.

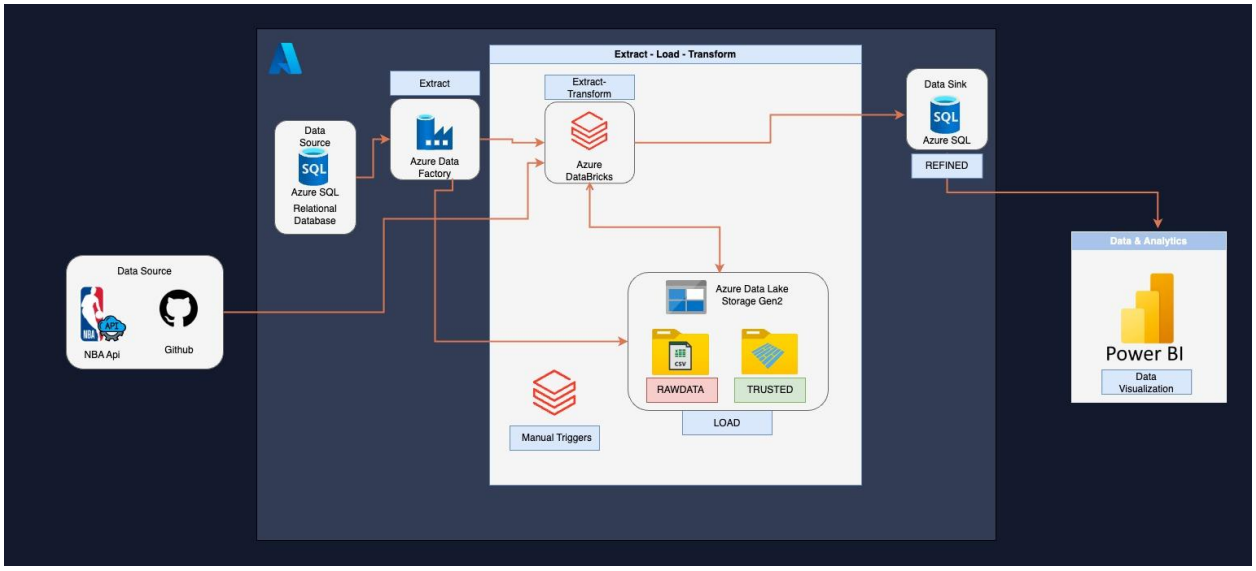
Glosario NBA

Con el fin de entender algunos términos presentes en las bases de datos ponemos a disposición la lista de términos utilizados en la NBA.

- **FGM (Field Goals Made):** Representa el número de tiros de campo convertidos exitosamente por un jugador.
- **FGA (Field Goals Attempted):** Indica la cantidad total de intentos de tiros de campo realizados por un jugador, independientemente de si fueron convertidos o no.
- **FG_PCT (Field Goal Percentage):** Es el porcentaje de tiros de campo convertidos con respecto a los intentados. Se calcula dividiendo los FGM entre los FGA y multiplicando el resultado por 100 para obtener un valor en porcentaje.
- **FG3M (Three-Point Field Goals Made):** Es el número de tiros de tres puntos anotados por un jugador.
- **FG3A (Three-Point Field Goals Attempted):** Representa la cantidad total de intentos de tiros de tres puntos realizados por un jugador.
- **FG3_PCT (Three-Point Field Goal Percentage):** Es el porcentaje de tiros de tres puntos convertidos con respecto a los intentados. Se calcula dividiendo los FG3M entre los FG3A y multiplicando el resultado por 100 para obtener un valor en porcentaje.
- **FTM (Free Throws Made):** Indica la cantidad de tiros libres exitosos realizados por un jugador.
- **FTA (Free Throws Attempted):** Representa el número total de intentos de tiros libres realizados por un jugador.
- **FT_PCT (Free Throw Percentage):** Es el porcentaje de tiros libres convertidos con respecto a los intentados. Se calcula dividiendo los FTM entre los FTA y multiplicando el resultado por 100 para obtener un valor en porcentaje.
- **OREB (Offensive Rebounds):** Representa el número de rebotes ofensivos capturados por un jugador. Estos son los rebotes obtenidos por el equipo que está atacando después de un tiro fallado.
- **DREB (Defensive Rebounds):** Indica el número de rebotes defensivos capturados por un jugador. Estos son los rebotes obtenidos por el equipo que está defendiendo después de un tiro fallado del equipo contrario.
- **REB (Total Rebounds):** Es la suma de los rebotes ofensivos y defensivos capturados por un jugador. Representa el número total de rebotes en los que el jugador participó, ya sea en el lado ofensivo o defensivo.

Arquitectura del Proyecto

Con el fin de llevar a cabo el proyecto de Análisis y creación de la herramienta para Escuelitas S.A. se planteó la siguiente arquitectura:



En nuestra arquitectura de datos, seguimos el proceso de ELT (Extract-Load-Transform) para integrar información de tres fuentes diferentes: Azure SQL, la API de NBA y GitHub.

Para la ingesta de datos desde Azure SQL, utilizamos Azure Data Factory, que nos permite extraer los datos directamente y cargarlos en el Datalake de Azure. Para las fuentes externas como la API de NBA y GitHub, empleamos una notebook en Databricks a traves de Azure Data Factory. En la etapa inicial, los datos se almacenan en el Datalake de Azure en formato CSV, en la carpeta correspondiente a la etapa "Raw".

Posteriormente, pasamos a la etapa "Trusted", donde se realizan las limpiezas y transformaciones necesarias en las tablas utilizando Databricks. Estos datos limpios y transformados se vuelven a almacenar en el Datalake de Azure, esta vez en formato Parquet, en la carpeta destinada a la etapa "Trusted".

En la última etapa, conocida como "Refined", los datos se toman del Datalake en su formato estandarizado y se aplican transformaciones adicionales utilizando Databricks. Esto incluye la

realización de joins de tablas para obtener la información precisa y necesaria. Los datos refinados se almacenan en una nueva instancia de base de datos.

A partir de esta instancia de base de datos, obtenemos los datos necesarios para generar visualizaciones en Power BI, permitiendo obtener productos de datos de alto valor.

En resumen, el proceso de integración de datos abarca la extracción de datos desde distintas fuentes, su carga en el Datalake en las etapas "Raw" y "Trusted", y finalmente, la transformación y refinamiento en la etapa "Refined" para obtener datos de calidad, que luego se utilizan para generar visualizaciones en Power BI.

Documentación de Procesos

El proyecto se realizó en la suscripción de Azure **UNDA - Formación 1** en el grupo de recursos **AN_Formacion2023.**

Se utilizan las siguientes Instancias:

Datalake: datalakean

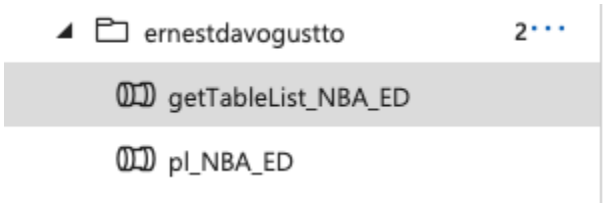
Azure Data Factory: ADF-Integrador

SQL: DWIntegrador, NBA

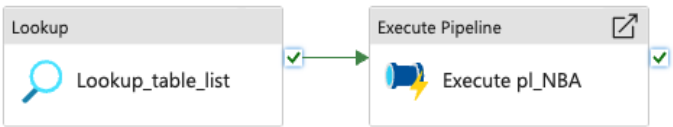
Databricks: databricks-formacion

Ingesta Data Factory

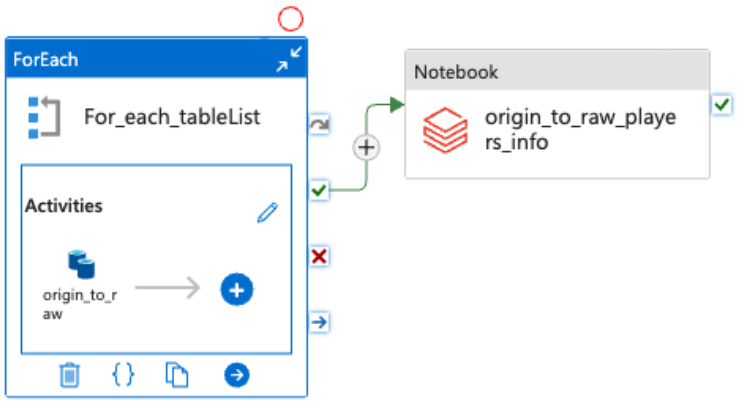
Para realizar la ingesta de datos, se utilizó Azure Data Factory en la instancia ADF-Integrador. En esta instancia, se crearon dos pipelines que desempeñaron roles específicos en el proceso de ingesta.



1. Pipeline "getTableList_NBA_ED": En este pipeline, se utilizó el método Lookup para obtener los nombres de las tablas que se incluirían en la ingesta. Una vez completada esta consulta, se ejecutó un Execute Pipeline para llamar al siguiente pipeline.



2. Pipeline "pl_NBA_ED": Este pipeline consta de los siguientes pasos:
    - ForEach: Se utilizó un bucle ForEach para iterar sobre los nombres de las tablas obtenidos en el paso anterior.
    - Extracción de datos: Para cada tabla, se realizó una extracción de datos desde la base de datos correspondiente. Los datos extraídos se almacenaron en el Datalake de Azure en el contenedor "ernestodavogustto", en la carpeta "integrador/raw_data" y en formato CSV.
    - Ejecución de la notebook "Origin_to_raw_players_info": Esta notebook se encargó de obtener la otra parte de la información requerida desde GitHub y la API de la NBA. Los datos adicionales también se almacenaron en la misma carpeta y formato en el Datalake de Azure.



Azure Databricks

Origin_to_raw_players_info

Nos conectamos a la API de la NBA para traernos la información de los jugadores activos y a github para obtener un dataset con los salarios de los jugadores

```
#instalamos API de NBA
#pip install nba_api
```

Cmd 6

```
from nba_api.stats.static import players
from nba_api.stats.endpoints import commonplayerinfo
import pandas as pd

# Obtener la lista de jugadores activos
active_players = players.get_active_players()

# Crear una lista para almacenar los DataFrames de información de cada jugador
player_info_list = []

# Recorrer los jugadores y obtener su información
for player in active_players:
    player_id = player['id']
    player_info_endpoint = commonplayerinfo.CommonPlayerInfo(player_id=player_id)
    player_info_df = player_info_endpoint.get_data_frames()[0]
    player_info_list.append(player_info_df)

# Concatenar los DataFrames en uno solo
players_info = pd.concat(player_info_list)

players_info = spark.createDataFrame(players_info)

```

Tenemos 3 tablas nuevas con la el siguiente esquema:

- players_info:
  - PERSON_ID:long
  - FIRST_NAME:string
  - LAST_NAME:string
  - DISPLAY_FIRST_LAST:string
  - DISPLAY_LAST_COMMA_FIRST:string
  - DISPLAY_FI_LAST:string
  - PLAYER_SLUG:string
  - BIRTHDATE:string
  - SCHOOL:string
  - COUNTRY:string
  - LAST_AFFILIATION:string

    - HEIGHT:string
    - WEIGHT:long
    - SEASON_EXP:long
    - JERSEY:double
    - POSITION:string
    - ROSTERSTATUS:string
    - GAMES_PLAYED_CURRENT_SEASON_FLAG:string
    - TEAM_ID:long
    - TEAM_NAME:string
    - TEAM_ABBREVIATION:string
    - TEAM_CODE:string
    - TEAM_CITY:string
    - PLAYERCODE:string
    - FROM_YEAR:long
    - TO_YEAR:long
    - DLEAGUE_FLAG:string
    - NBA_FLAG:string
    - GAMES_PLAYED_FLAG:string
    - DRAFT_YEAR:string
    - DRAFT_ROUND:string
    - DRAFT_NUMBER:string
    - GREATEST_75_FLAG:string
    -
- players_info:
    - rank:long
    - name:string
    - position:string
    - team:string
    - salary:long
    - season:long
- active_players:
    - first_name:string
    - full_name:string
    - id:long
    - is_active:boolean
    - last_name:string

Herramientas Utilizadas: Spark, Pandas

Lenguajes: Python

raw_to_trusted: Capa de Limpieza

Transformaciones

Para las transformaciones utilizamos python como lenguaje de programación con su libreria pyspark.

1. Cambiamos todos los nombres de las columnas de todas las tablas por lowercase, para hacer más sencillas sus consultas.
2. Tabla **Teams:**
   - 2.1. Eliminamos las columnas `"league_id", "max_year", "min_year", "dleagueaffiliation", "owner", "generalmanager".` No relevantes para el análisis.
   - 2.2. Cambiamos nombres de las columnas: `"abbreviation":"abbr", "yearfounded": "year_founded", "arenacapacity": "arena_capacity".`
   - 2.3. Asignamos imagen al equipo con formato "https://cdn.ssref.net/req/202305101/tlogo/bbr/{abbr}-2023.png" Para luego mostrar los logos de los equipos en PowerBI
3. Tabla **active_players:**
   - 3.1. Eliminamos columnas: `"first_name", "last_name".` Ya que tambien contiene una columna "full_name"
4. Tabla **players_info:**
   - 4.1. Eliminamos columnas `'first_name','last_name', 'display_last_comma_first','display_fi_last', 'player_slug','school','last_affiliation','weight','jersey','roster status','games_played_current_season_flag','team_id','team_name','team_abbreviation','team_code','team_city','playercode','from_year', 'to_year','dleague_flag','nba_flag','games_played_flag','greatest_75_flag'.` Algunas no son relevantes para el analisis y otras se repiten en otras tablas
   - 4.2. Cambiamos nombre de la columna "display_first_last" que contiene el nombre y apellido del jugador a "player_name"
   - 4.3. Casteamos la columna "birthdate" a tipo fecha.
   - 4.4. Generamos un slug para los jugadores para así asignar una imagen al jugador y utilizarla en el dashboard de powerBI. Formato: "https://www.basketball-reference.com/req/202106291/images/headshots/{player_slug}.jpg". Columna "player_headshot".
   - 4.5. Convertimos la estatura de pies-inches a CM
5. Tabla **player_salaries:**
   - 5.1. Eliminamos las columnas `"rank", "position", "team"`
6. Tabla **games:**

6.1. Eliminamos columnas `"game_status_text"`, `"visitor_team_id"`, `"team_id_home"`
6.2. Creamos columna con el id del equipo ganador. “winner_team_id”.
6.3. Cambiamos nombre de “team_id_away” a “away_team_id” para conservar misma estructura.

7. Tabla **game_details:**
   7.1. Elimino columnas `"team_abbreviation"`, `"team_city"`, `"player_name"`, `"nickname"`, `"start_position"`, `"comment"`. Algunas tienen datos que estan en otras tablas, con las cuales voy a hacer join.
   7.2. Rellenamos valores nulos con “0” para poder realizar promedios luego.
   7.3. Creamos columna con minutos y segundos jugado. “min_played”, “sec_played”, a partir de la columna “min”, que tenia minutos y segundos en formato string separado por “-”.
8. Tabla **ranking:**
   8.1. Cambiamos nombres de columnas: `"g": "games_played", "w": "wins", "l":"loses", "w_pct": "win_pct"`)
   8.2. Cambiamos la columna “season” como un año (el numero antes del año representa a la conferencia)
   8.3. La columna “standingsdate” contiene la fecha en que se registro la marca para el equipo. Entonces nos quedamos con el ultimo registro que se tomo para cada equipo por temporada.
9. Se realizó una limpieza general a todas las tablas, borrando duplicados y registros nulos
10. Se almacenan todas las tablas modificadas en el datalake, esta vez en la carpeta ”trusted_data” en formato parquet.

trusted_to_refined: Capa Analítica del proceso de los datos.

Elaboración de modelos

En la capa analítica del proceso de los datos, se lleva a cabo la elaboración de modelos para el análisis de información. A continuación, se describen las acciones realizadas en esta etapa:

1. Importación de tablas desde la carpeta "trusted_data" del datalake:

   En esta etapa, se importan las tablas necesarias desde la carpeta "trusted_data" del datalake para su procesamiento en la capa analítica.

2. Creación de la tabla "active_players_info":

   Se crea la tabla "active_players_info" a partir de la combinación de dos tablas: "active_players", que contiene la información de los jugadores activos de la NBA, y "players_info". Se realiza un INNER JOIN para obtener solo la información de los jugadores activos. Además, se eliminan las columnas con información repetida, como "full_name" e "id".

3. Creación de la tabla "players_salaries":

   A partir de las tablas "players" y "players_salaries", se crea la tabla "players_salaries" que contiene la información de los jugadores por temporada y sus respectivos salarios. Se realiza un JOIN entre estas dos tablas y se filtra la información para incluir solo los jugadores activos.

4. Procesamiento de la tabla "games":

   Se realiza un JOIN entre la tabla "games" y la tabla "games_details" utilizando la columna "game_id". Esto permite obtener todos los detalles de cada juego. Posteriormente, se realiza un JOIN entre la tabla resultante y la tabla "active_players" para filtrar únicamente los juegos en los que participan los jugadores activos.

Como resultado de este proceso, se obtienen tres tablas principales:

- "active_players_info": Contiene la información detallada de los jugadores activos.
- "players_salaries": Almacena los datos de los jugadores activos junto con sus respectivos salarios por temporada.

- "games": Incluye los detalles de los juegos en los que participan los jugadores activos.

Además, se mantienen las tablas existentes de "teams" y "ranking", las cuales no sufrieron modificaciones en esta etapa de la capa analítica, son almacenadas en una base de datos llamada "DWIntegrador" en la plataforma de Azure. Esta base de datos proporciona un entorno seguro y escalable para el almacenamiento y gestión de los datos analíticos.

Visualización de los Datos

Se utilizó PowerBI como herramienta para la visualización de los datos, aprovechando la base de datos creada en el proyecto. A través de un Dashboard interactivo, se logró ofrecer diversas opciones de búsqueda y análisis de jugadores en la NBA.

El Dashboard permite realizar búsquedas por equipo, lo que brinda la posibilidad de visualizar los mejores jugadores filtrados por estadísticas específicas. Además, se ofrece una lista completa de jugadores por temporada, facilitando la exploración de datos detallados.

Para un enfoque más específico, se implementó una función de búsqueda de jugador en particular. Esta función proporciona estadísticas detalladas por temporada, acompañadas de gráficos que representan visualmente dichas estadísticas. Además, se incluyó un análisis del salario de los jugadores, lo que permite evaluar si un jugador se encuentra dentro del presupuesto establecido.

Adicionalmente, se habilitó una búsqueda general que muestra los mejores jugadores, con la capacidad de filtrarlos por posición, rango de edad y rango de salario. Esto brinda flexibilidad al usuario para explorar y comparar jugadores según criterios específicos.

En resumen, la herramienta desarrollada en PowerBI proporciona un Dashboard interactivo que facilita la visualización y el análisis de datos de jugadores en la NBA. Ofrece opciones de búsqueda por equipo, jugador específico y una búsqueda general, brindando a los usuarios una experiencia

completa y personalizable para explorar y entender mejor las estadísticas y características de los jugadores.