Instituto Politécnico do Cávado e do Ave

Licenciatura em Engenharia de Sistemas Informáticos
Arquitetura de Computadores

Projeto F1 ETL

João Faria – 26425

26 de maio de 2025

Índice

1. Enquadramento

Este projeto foi desenvolvido no âmbito da Unidade Curricular de Integração de Sistemas de Informação, do curso de Engenharia Informática. O objetivo central foi a implementação de um processo ETL (Extract, Transform, Load) com dados provenientes da API pública de Fórmula 1.

A integração de sistemas é uma componente essencial nas organizações modernas, permitindo consolidar informação dispersa em diferentes fontes e sistemas. Através dos processos de ETL, é possível extrair dados brutos, transformá-los de acordo com regras de negócio definidas e carregá-los num repositório central de dados para posterior análise.

Neste contexto, o projeto desenvolvido — denominado F1 ETL Dashboard — tem como base a integração e visualização de dados reais de corridas de Fórmula 1. Os dados foram obtidos a partir de ficheiros externos (em formato JSON e CSV), processados e normalizados com recurso à ferramenta n8n, e armazenados numa base de dados PostgreSQL.

O n8n, uma plataforma de automação open-source baseada em workflows, desempenhou um papel central no processo. Foi utilizada para:

- Orquestrar o pipeline de dados, desde a extração à carga na base de dados;
- Aplicar transformações automáticas, incluindo junções, normalização e validação de campos;
- Gerar logs automáticos de execução em cada job, registando o número de registos inseridos, o estado do processo e eventuais erros;
- Automatizar o controlo de versões e execução periódica dos jobs.

Adicionalmente, foi desenvolvido um dashboard interativo em Python (Streamlit) que permite visualizar os resultados do processo ETL e explorar dados agregados, como rankings de pilotos por circuito, melhores voltas e médias de tempos.

2. Problema

No contexto deste projeto, o problema identificado consiste na falta de uma visão unificada e estruturada dos dados da Fórmula 1, nomeadamente os relativos a pilotos, corridas, circuitos e voltas. Estes dados, apesar de ricos em informação, estão frequentemente armazenados de forma separada e desorganizada, o que dificulta a sua interpretação e análise.

3. Estratégia Utilizada

A estratégia seguida neste projeto baseou-se na criação de um pipeline ETL (Extract, Transform, Load) totalmente funcional, concebido para recolher, transformar, armazenar e visualizar dados reais da Fórmula 1.
O foco principal foi garantir consistência, ausência de redundância e fácil acesso aos dados, tirando partido de ferramentas modernas, escaláveis e integráveis.

Ferramentas e Abordagem

- **n8n — Plataforma de Integração e Automação**

O n8n foi a ferramenta principal utilizada para a construção dos processos ETL. Trata-se de uma solução *open-source* e *low-code*, amplamente utilizada em contextos de integração de dados e automação de processos.

Através da criação de *workflows visuais*, foi possível orquestrar todas as etapas do pipeline:

Extração de dados da API pública da Fórmula 1 (Ergast API), que disponibiliza dados históricos e em tempo real sobre corridas, pilotos, voltas e circuitos;

Transformação e limpeza dos dados (ex.: normalização de tempos, unificação de nomes, formatação de chaves de referência);

Verificação de duplicados e inserção apenas de novos registos, evitando redundância e mantendo a integridade dos dados;Registo automático de execuções e erros no *log* (`etl_log`), permitindo monitorização e rastreabilidade do processo.

A escolha do n8n deveu-se à sua versatilidade e facilidade de integração com serviços externos, sem exigir desenvolvimento manual de código extenso. Além disso, permite incorporar scripts em JavaScript, o que oferece controlo total nas transformações.

- **Supabase — Base de Dados em Cloud Moderna**

Para o armazenamento dos dados, optou-se por utilizar o Supabase, uma plataforma *Backend-as-a-Service* baseada em PostgreSQL.
O Supabase oferece um ambiente cloud moderno, escalável e colaborativo, permitindo aceder à base de dados de forma segura e centralizada a partir de qualquer aplicação cliente.

As principais vantagens da utilização do Supabase neste contexto foram:

- Facilidade de comunicação entre o n8n e o dashboard (via API REST ou Python client);

- Gestão automática de autenticação, permissões e logs;

- Persistência em cloud, eliminando a necessidade de gerir infraestrutura local;
- Compatibilidade total com PostgreSQL, garantindo acesso a todas as suas funcionalidades (joins, chaves estrangeiras, constraints, etc.);
- Disponibilidade constante, ideal para pipelines ETL que necessitam de fiabilidade e acessos simultâneos.

A modelação da base de dados foi cuidadosamente estruturada, contendo tabelas como:

- `f1_drivers` — dados de pilotos (nome, código, nacionalidade, data de nascimento);
- `f1_circuits` — dados de circuitos (nome, localização, coordenadas);
- `f1_races` — informações sobre corridas (temporada, ronda, circuito, data);
- `f1_laps` — tempos de volta e posições;
- `etl_log` — registo de execuções do pipeline.

Cada tabela está interligada através de chaves primárias e estrangeiras, assegurando integridade referencial e não duplicação de dados.
Antes de inserir novos registos, o n8n valida se o piloto, corrida ou circuito já existem, garantindo que não há redundância na base de dados.

- **Python + Streamlit — Visualização Interativa**

A camada de visualização foi desenvolvida em Python, com recurso à biblioteca Streamlit, permitindo construir um dashboard interativo e dinâmico, diretamente ligado ao Supabase.
Este dashboard permite:

- Explorar leaderboards por circuito, temporada e ronda;
- Filtrar resultados em tempo real, com campos opcionais para visualizar dados de todas as temporadas;
- Exibir imagens dos circuitos automaticamente, obtidas via pesquisa online, complementando a análise visual dos resultados.

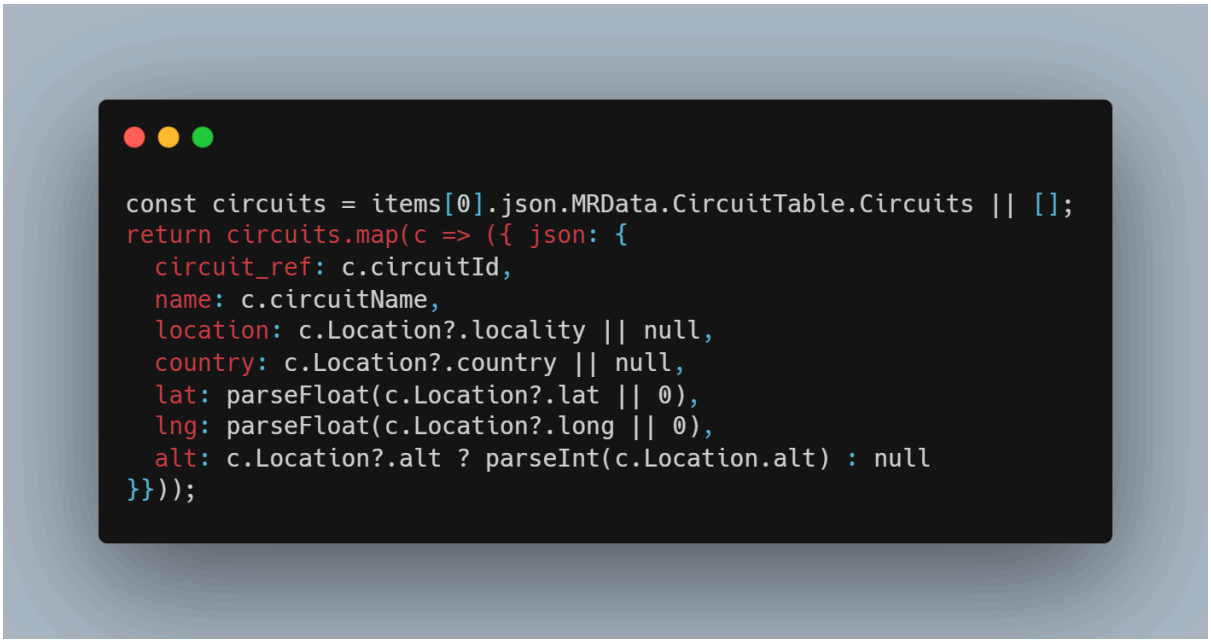
Esta integração direta entre o Streamlit e o Supabase demonstra como uma abordagem *cloud-native* simplifica a comunicação entre componentes, permitindo uma experiência fluida tanto na recolha de dados como na sua exploração analítica.

4. Transformações

As transformações representam a fase intermédia e crítica do processo ETL, onde os dados extraídos da API pública da Fórmula 1 (Ergast API) são normalizados, limpos e convertidos para um formato adequado ao modelo relacional implementado no Supabase.
Esta fase é totalmente automatizada através de *workflows* no n8n, permitindo que os dados brutos sejam processados antes da sua inserção nas tabelas finais (`f1_circuits`, `f1_drivers`, `f1_races`, `f1_laps`).

A seguir são descritas as principais transformações realizadas, com base nos nós (nodes) utilizados no n8n.

- **Circuitos**



```
const circuits = items[0].json.MRData.CircuitTable.Circuits || [];
return circuits.map(c => ({ json: {
  circuit_ref: c.circuitId,
  name: c.circuitName,
  location: c.Location?.locality || null,
  country: c.Location?.country || null,
  lat: parseFloat(c.Location?.lat || 0),
  lng: parseFloat(c.Location?.long || 0),
  alt: c.Location?.alt ? parseInt(c.Location.alt) : null
}}));
```

- **Condutores**

```
const drivers = items[0].json.MRData.DriverTable.Drivers || [];
return drivers.map(d => ({ json: {
  driver_ref: d.driverId,
  code: d.code || null,
  forename: d.givenName,
  surname: d.familyName,
  nationality: d.nationality,
  dob: d.dateOfBirth
}}));
```

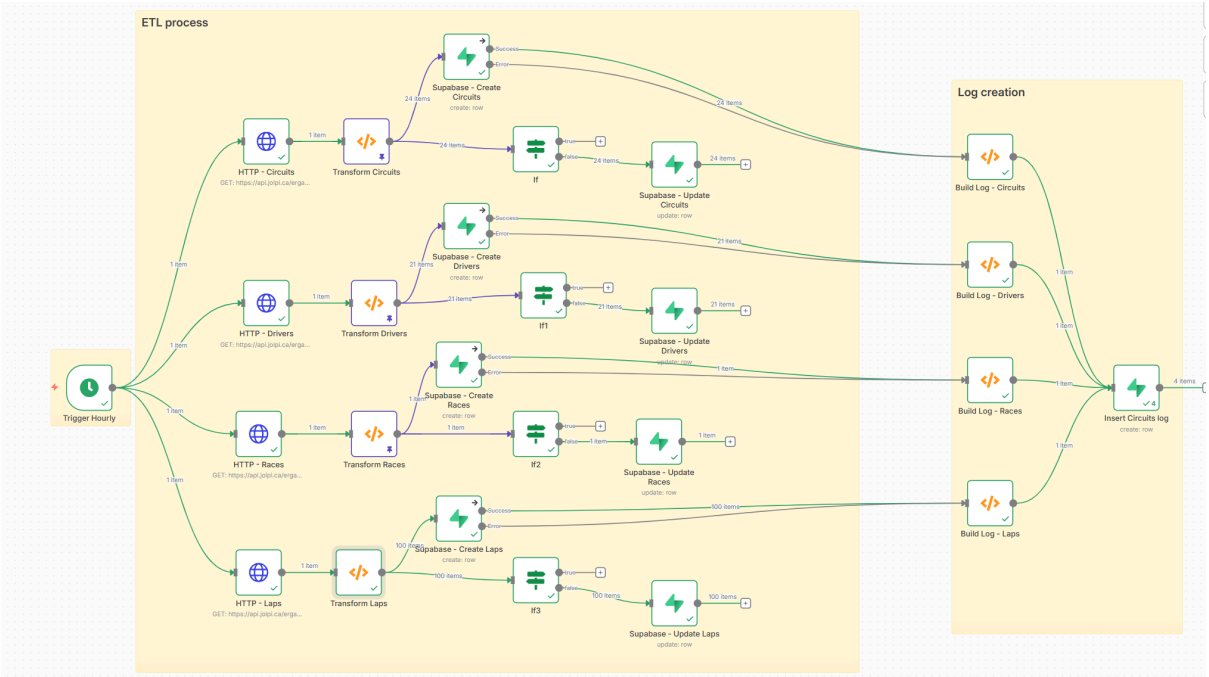
- **Corridas**

```
const races = items[0].json.MRData.RaceTable.Races || [];
return races.map(r => ({ json: {
  year: parseInt(r.season),
  round: parseInt(r.round),
  name: r.raceName,
  date: r.date,
  time: r.time,
  circuit_ref: r.Circuit?.circuitId || null
}}));
```

- **Voltas**

```
// 🏁 Converter "1:23.456" -> milissegundos
const convertToMs = (timeStr) => {
  if (!timeStr) return null;
  const parts = timeStr.split(':');
  if (parts.length === 2) {
    const [min, rest] = parts;
    const [sec, ms] = rest.split('.');
    return (parseInt(min) * 60 + parseInt(sec)) * 1000 + parseInt(ms.padEnd(3, '0'));
  } else if (parts.length === 3) {
    const [hour, min, rest] = parts;
    const [sec, ms] = rest.split('.');
    return ((parseInt(hour) * 3600) + (parseInt(min) * 60) + parseInt(sec)) * 1000 +
parseInt(ms.padEnd(3, '0'));
  }
  return null;
};

// 🔹 Extrai os dados principais
const races = items[0].json.MRData?.RaceTable?.Races || [];
const laps = [];

for (const race of races) {
  const raceId = race.round;
  const raceName = race.raceName;
  const season = race.season;
  const circuitId = race.Circuit?.circuitId;
  const circuitName = race.Circuit?.circuitName;
  const date = race.date;

  if (!race.Laps) continue;

  for (const lap of race.Laps) {
    const lapNumber = parseInt(lap.number);

    for (const timing of lap.Timings) {
      laps.push({
        json: {
          round: raceId,
          season,
          race_name: raceName,
          circuit_ref: circuitId,
          circuit_name: circuitName,
          date,
          driver_ref: timing.driverId,
          lap_number: lapNumber,
          position: parseInt(timing.position),
          lap_time: timing.time,
          milliseconds: convertToMs(timing.time)
        }
      });
    }
  }
}

return laps;
```

5. Jobs



O processo de Jobs representa a orquestração completa do pipeline ETL no n8n, responsável por coordenar todas as etapas de extração, transformação e carga de dados na base de dados Supabase.

O *workflow* é acionado automaticamente através de um gatilho horário (*Trigger Hourly*), garantindo que os dados se mantêm atualizados de forma periódica.
A partir deste ponto, o fluxo divide-se em quatro ramos paralelos, cada um responsável por um tipo de entidade:

1. Circuits – Recolhe dados dos circuitos através de um nó HTTP Request, aplica a transformação JavaScript (`Transform Circuits`) e insere os resultados no Supabase.
    - Caso o registo já exista, é atualizado em vez de duplicado, garantindo a não redundância dos dados.
2. Drivers – Executa a mesma lógica para os pilotos, transformando os dados e garantindo a atualização dos campos sem duplicações.
3. Races – Responsável por importar e transformar as corridas, relacionando-as com o circuito correspondente e a temporada associada.
4. Laps – Importa os tempos de volta, converte o tempo em milissegundos e associa cada volta ao respetivo piloto e corrida.

Após a execução de cada ramo, são criados logs automáticos através do bloco Log creation, que agrega a informação sobre:

- o nome do processo executado,
- o número de linhas inseridas ou atualizadas,
- o estado (sucesso ou erro),
- e a hora da execução.

Esses logs são armazenados na tabela `etl_log` do Supabase, permitindo rastreabilidade total e fácil monitorização de falhas.

6. Vídeo com Demonstração

Para complementar o relatório, foi gravado um vídeo demonstrativo mostrando a execução completa do pipeline ETL e a utilização do dashboard de análise no Streamlit. O vídeo pode ser acedido através do QR Code abaixo:

7. Conclusão e Trabalhos Futuros

O desenvolvimento deste projeto permitiu consolidar conhecimentos sobre Integração de Sistemas de Informação e, em particular, sobre a implementação prática de processos ETL (Extract, Transform, Load) com recurso a ferramentas modernas e escaláveis.

A utilização do n8n revelou-se uma solução extremamente eficiente para a automação e orquestração de fluxos de dados, permitindo construir um pipeline visual, modular e facilmente ajustável. O facto de o sistema estar totalmente integrado com o Supabase, uma plataforma cloud moderna baseada em PostgreSQL, trouxe vantagens significativas como:

- acesso remoto e seguro aos dados,
- sincronização automática entre processos,
- e facilidade de comunicação entre sistemas e utilizadores.

O dashboard final, desenvolvido em Streamlit, fornece uma visualização intuitiva e dinâmica dos dados da Fórmula 1, permitindo filtrar por temporada, corrida e circuito. A inclusão de gráficos e tabelas interativas facilita a análise comparativa de desempenho entre pilotos, oferecendo uma perspetiva clara sobre os melhores tempos e médias por pista.

Como evolução natural deste projeto, seria possível integrar novas fontes de dados e ampliar o contexto de análise. Algumas propostas incluem:

- Comparação entre dados reais da F1 e tempos obtidos em simuladores como o Assetto Corsa, permitindo avaliar o realismo das simulações;
- Expansão do dashboard para incluir estatísticas históricas, previsões e tendências;
- Publicação do dashboard como uma aplicação web pública, acessível a qualquer utilizador, com autenticação via Supabase Auth.

Estas melhorias iriam reforçar o caráter dinâmico e analítico do sistema, transformando-o numa ferramenta ainda mais poderosa para exploração e monitorização de dados desportivos. Em particular, a integração de dados provenientes de simuladores como o Assetto Corsa seria uma adição especialmente interessante e divertida, permitindo comparar os tempos reais da Fórmula 1 com os tempos obtidos em simulação. Esta funcionalidade não só acrescentaria um elemento lúdico ao projeto, como também possibilitaria uma análise curiosa sobre o realismo e precisão dos simuladores face ao desempenho real dos pilotos.

8. Referências

- Ergast API - Fórmula 1 Data: https://ergast.com/mrd/
- n8n Documentation: https://docs.n8n.io/
- Supabase Documentation: https://supabase.com/docs
- Streamlit Documentation: https://docs.streamlit.io/