

## ENGENHARIA INFORMÁTICA

### Enunciado do trabalho prático TP-3

Integração de Sistemas (de Informação)

Ano letivo 2025/2026

## Objetivo

O presente trabalho prático 3 (TP3) tem como objetivo desenvolver as capacidades dos estudantes na utilização de métodos de representação de dados (através da utilização de linguagens de anotação) para auxiliar a integração de diferentes sistemas e permitir assim a interoperabilidade dos mesmos.

Leia atentamente todo o enunciado e contacte os docentes da disciplina em caso de dúvidas.

- **Cada grupo de trabalho deverá ter 2 elementos ou de forma individual, podendo os alunos continuar com os grupos realizados no trabalho TP-B, ou juntarem-se 2 elementos que realizaram o trabalho TP2-B de forma individual.**

## Regras

- Para o desenvolvimento dos trabalhos propostos, deverá ser usada múltiplas linguagens de programação.
- Os pormenores de implementação que se devem à interpretação dos enunciados por parte dos grupos de alunos deverão ser descritos no relatório com detalhe e justificação das opções tomadas.
- A implementação de funcionalidades extra não presentes no enunciado será valorizada, desde que estas funcionalidades não modifiquem os requisitos obrigatórios e não reduzam a dificuldade do trabalho. As funcionalidades extra implementadas deverão ser documentadas no relatório.
- A apresentação de relatórios e/ou implementações **não originais e que constituam plágio**, conduzem à imediata atribuição de **nota zero** no trabalho de grupo e a **eventuais processos disciplinares**.

## ENGENHARIA INFORMÁTICA

### Enunciado do trabalho prático TP-3

Integração de Sistemas (de Informação)

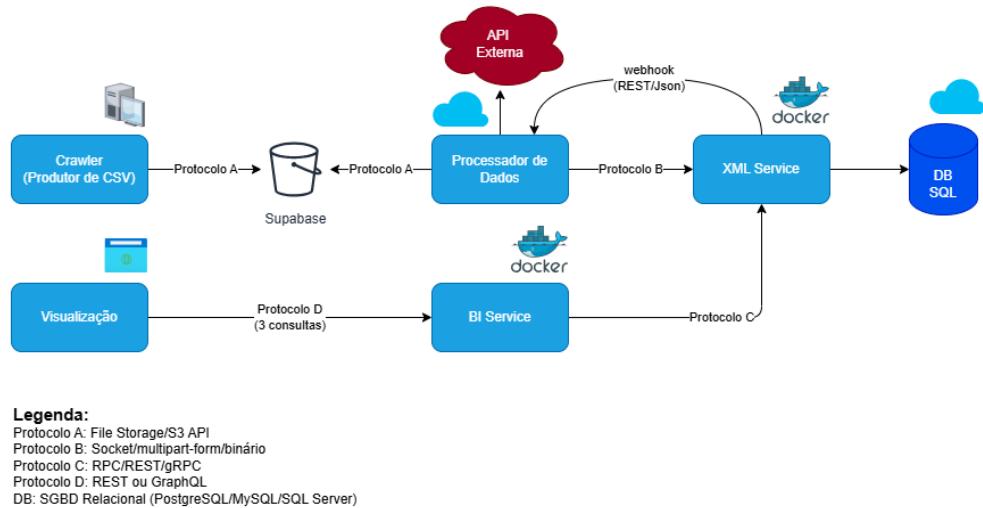
Ano letivo 2025/2026

## Avaliação e Entrega

- O trabalho prático TP3 vêm no seguimento dos trabalhos práticos realizados durante o semestre, parte da avaliação da Componente Prática da disciplina de **Integração de Sistemas**, nomeadamente, o TP1, TP2-A e TP2-B e será integrado com a componente TP3 e mais algumas funcionalidades, que serão descritas mais adiante neste documento.
- A nota é atribuída individualmente aos elementos do grupo segundo a apresentação, visualização e discussão dos elementos entregues e as impressões obtidas pelos docentes acerca do aluno durante o decorrer das aulas de acompanhamento.
- Para aprovação à disciplina, a nota da Componente Prática deverá ter a classificação mínima obrigatória de **10.0 valores**.
- As discussões orais deste trabalho serão realizadas no dia indicado no calendário e no horário previamente definido para o efeito. Os grupos de trabalho devem reservar o dia e hora marcados, na sua agenda. Os trabalhadores-estudantes devem, ao abrigo do respetivo estatuto, solicitar a folha justificativa de exame para a empresa.
- O trabalho prático deverá ser submetido através do Moodle seguindo as instruções lá indicadas. A entrega deverá conter os seguintes elementos:
  - Código fonte;
  - Slides em PowerPoint / PDF (segundo estrutura a indicar);
  - Vídeo com duração máxima de 5 minutos com instruções de utilização e demonstração de funcionalidades;

## Descrição do trabalho

Esquemático Geral:



Fluxo de Trabalho:

1. Origem de dados: O **Crawler (Produtor de CSV)** periodicamente (simule a periodicidade) gera um arquivo CSV contendo dados de mercado. O CSV é salvo no **Storage (Bucket) do Supabase**. (Protocolo: File Storage/S3 API).
2. Processamento assíncrono: O **Processador de Dados** monitoriza o bucket (via *polling*, notificação, etc), baixa o novo CSV, e lê seus dados em *stream*. Consome uma **API Externa** (REST/SOAP) para obter dados complementares. Combina os dados CSV + API Externa e gera um novo CSV com atributos desacoplados da origem (nomes que fazem sentido no seu domínio). Envia um Id da requisição, um mapper, uma url do webhook e o novo CSV para o **XML Service** (Socket, *Multipart-Form*, binário, etc). Após confirmação que o XML foi salvo corretamente, apaga os CSVs.
3. O **XML Service** recebe a requisição. Cria o XML, valida o XML, salva na base de dados, e envia uma mensagem para o Webhook\_URL do **Processador de Dados**, informando o ID\_Requisicao, o Status (OK, ERRO\_VALIDACAO, ERRO\_PERSISTENCIA) e a ID do documento no DB. O XML é persistido em uma coluna do tipo XML no DB. O **XML Service** usa as funções **XPath/XMLTable** para consultar o XML persistido no DB e retorna os dados brutos ou sumarizados ao **BI Service**.
4. A função do **BI Service** é definir a interface de consumo de dados (GraphQL ou REST) e **transformar** o resultado do XML Service no formato final para a **Visualização**. O BI Service expõe *endpoints* que aceitam parâmetros de consulta. (Protocolo: GraphQL ou REST/JSON). Para responder às consultas, o BI Service chama o **XML Service** (Protocolo: RPC/REST/gRPC) solicitando dados, passando os critérios de filtro. O **BI Service** agrupa, calcula e formata a resposta antes de enviá-la para a **Visualização**.
5. A **Visualização** pode ser uma web página simples ou apenas o postman, que consome o **BI Service**.

# ENGENHARIA INFORMÁTICA

## Enunciado do trabalho prático TP-3

Integração de Sistemas (de Informação)

Ano letivo 2025/2026

### Requisitos:

1. A solução final deve ser implementada com **pelo menos** 4 serviços distintos.
2. Deve ser utilizado **pelo menos** 3 linguagens de programação distintas, para o desenvolvimento dos serviços.
3. **Deve ser** utilizado o recurso de storage (buckets) do Supabase.
4. As imagens Docker devem estar versionadas no Docker Hub.
5. Todos os códigos devem estar versionados no GitHub/GitLab/Bitbucket (monorepo ou multirepo).
6. A entidade criada, em XML, **deve ser** desacoplada (no que toca aos nomes dos atributos) da origem (csv). E a entidade **deve ser** enriquecida com dados de outra fonte (API externa), para além da origem (csv).
7. A entidade criada, em XML, deve ter **pelo menos** 1 nível hierárquico.
8. Deve ser utilizado pelo menos quatro (4) protocolos de comunicação distintos para as interfaces de serviço:
  - a. RESTful (entre dois serviços distintos).
  - b. GraphQL (para a interface do BI Service ou XML Service).
  - c. RPC ou XML-RPC (entre dois serviços distintos).
  - d. gRPC (entre dois serviços distintos).
9. A comunicação deve utilizar **pelo menos dois formatos de dados** diferentes: **XML** e **JSON**.
10. Deve ser implementado **pelo menos um endpoint de Webhook** para comunicação assíncrona (como o *callback* do **XML Service** para o **Processador de Dados**).
11. O conjunto de serviços deve ser distribuído por pelo menos três ambientes de hospedagem diferentes:
  - a. Máquina local.
  - b. Contêiner(es) Docker, orquestrado por Docker Compose.
  - c. Um ambiente de Cloud (Azure, AWS ou Google Cloud).
12. A base de dados deve ser um **SGBD Relacional** (PostgreSQL/MySQL/SQL Server) configurado para suportar o tipo de dado XML.
13. **Tabela de Dados:** A tabela de persistência no DB deve ter, no mínimo, os seguintes campos: ID, XML\_DOCUMENTO (tipo XML), DATA\_CRIACAO, MAPPER\_VERSION.
14. **BI Service:** O **BI Service** deve expor **pelo menos três consultas** complexas (filtros, agregações ou transformações) que demonstrem a capacidade de **consultar o XML usando XPath** na base de dados. Essas consultas devem ser feitas pelo BI Service chamando um *endpoint* dedicado no XML Service.

# ENGENHARIA INFORMÁTICA

## Enunciado do trabalho prático TP-3

Integração de Sistemas (de Informação)

Ano letivo 2025/2026

Exemplo de XML Final:

```
<RelatorioConformidade DataGeracao="2025-12-04" Versao="1.0">
  <Configuracao ValidadoPor="XML_Service_ID_789" Requisitante="Processador_ID_456">
    <Regulador Nome="SEC" DataUltimaAtualizacao="2025-11-01"/>
  </Configuracao>
  <Ativos>
    <Ativo IDInterno="CSV_A101" Ticker="GOGL" Tipo="Tecnologia">
      <DetalheNegociacao>
        <PrecoAtual Fonte="CSV" Moeda="USD">150.25</PrecoAtual>
        <Volume Negociado="1200000" Unidade="Ações"/>
      </DetalheNegociacao>
      <HistoricoAPI>
        <Media30d>145.00</Media30d>
        <Maximo6m>160.00</Maximo6m>
      </HistoricoAPI>
    </Ativo>
    <Ativo IDInterno="CSV_B202" Ticker="TSLA" Tipo="Automotivo">
      </Ativo>
    </Ativos>
  </RelatorioConformidade>
```