

# Trabalho de Integração de Sistemas

## Parte 1

Dimensão do grupo: 2 alunos<sup>1</sup>

Ferramentas e tecnologias sugeridas: Sikulix, Visual Studio, C#, ASP.NET, Microsoft SQL Server, Git

Acompanhamento e registo: Git (com commits regulares)<sup>2</sup>

Submissão de Trabalhos: Nonio/Inforestudante.

## Contexto

Uma empresa do setor industrial necessita transferir dados de um sistema de desktop legado localizado numa linha de produção para um sistema de análise de dados baseado na web. Atualmente, não existe integração direta entre os dois sistemas, o que resulta em processos manuais demorados e suscetíveis a erros.

O objetivo deste trabalho é desenvolver uma solução de integração robusta que permita a comunicação automática entre estes dois sistemas heterogéneos. A solução deve garantir a transferência eficiente e fiável dos dados da produção, assegurando a integridade dos mesmos e evitando a duplicação de informação. Pretende-se que esta integração seja realizada utilizando tecnologias modernas de middleware, possibilitando a sincronização em tempo real ou quase real dos dados de produção para análise e tomada de decisão.

---

<sup>1</sup> Os grupos devem ser registados em: <https://forms.microsoft.com/e/g94Zm1pdqu>

<sup>2</sup> Obrigatória a inclusão da equipa docente no projeto ([jpsousa@utad.pt](mailto:jpsousa@utad.pt); [hparedes@utad.pt](mailto:hparedes@utad.pt))

## Dados a Sincronizar

Os dados a sincronizar entre as duas aplicações referem-se à produção de itens numa linha de produção e incluem:

- Data: Data de produção do artigo no formato: yyyy-mm-dd
- Hora: Hora de produção do artigo no formato: hh:mm:ss
- Código da peça: Composto por 8 caracteres
  - Os primeiros dois caracteres representam o código do tipo de produto e podem ser: aa, ab, ba ou bb
  - Os últimos seis caracteres representam o identificador único da peça
- Tempo de produção: Duração entre 10 e 50 segundos
- Resultados do teste:
  - 01 - Ok
  - 02 - Falha na inspeção visual
  - 03 - Falha na inspeção de resistência
  - 04 - Falha na inspeção de dimensões
  - 05 - Falha na inspeção de estanqueidade
  - 06 - Desconhecido

## Requisitos do Sistema Desktop

- A aplicação Desktop deverá gerar todos os dados anteriormente descritos de forma aleatória em espaços de tempo entre 5 a 20 segundos.

## Arquitetura do Sistema Web

O sistema web possui uma base de dados denominada "Producao" com duas tabelas:

- Produto: Armazena informações básicas da peça
  - ID\_Produto (Chave Primária) - Identificador único autoincrementado
  - Codigo\_Peca - Código completo da peça (8 caracteres)
  - Data\_Producao - Data de produção
  - Hora\_Producao - Hora de produção
  - Tempo\_Producao - Tempo de produção em segundos
- Testes: Armazena os resultados dos testes realizados
  - ID\_Testes (Chave Primária) - Identificador único autoincrementado

- ID\_Produto (Chave Estrangeira) - Referência ao ID\_Produto na tabela Produto
- Codigo\_Resultado - Código do resultado do teste (01-06)
- Data\_Teste - Data de realização do teste

## Funcionalidades do Sistema Web

O sistema web deverá permitir:

- Inserção
- Atualização
- Remoção

Estas operações serão realizadas através de uma API Web Services REST.

## Acesso à Base de Dados

O acesso à base de dados para inserção, alteração e consulta de dados é realizado através de Stored Procedures, com o objetivo de:

- Suportar a validação de dados
- Garantir a segurança no acesso à informação

## Regras de Validação

- Verificar se a peça existe (não podem existir duas peças com o mesmo código)
- Verificar se o código do teste de uma peça já existe no sistema
  - Caso não exista, deverá inserir o resultado do teste como "06 - Desconhecido"

## Sistema de Contabilidade

Existe ainda uma base de dados relacionada com a parte contabilística da empresa que calcula o custo de cada peça.

A base de dados "Contabilidade" contém uma tabela "Custos\_Peca" com os seguintes campos:

- ID\_Custo (Chave Primária) - Identificador único autoincrementado
- ID\_Produto (Chave Estrangeira) - Referência ao ID\_Produto na tabela Produto da base de dados "Producao"

- `Codigo_Peca` - Código completo da peça
- `Tempo_Producao` - Tempo de produção em segundos
- `Custo_Producao` - Custo total de produção em euros
- `Lucro` - Valor do lucro em euros (venda - custo de produção - prejuízo)
- `Prejuizo` - Valor do prejuízo em euros

## Integração com a base de dados do sistema de informação da contabilidade

Sempre que um novo registo for inserido na tabela "Produto" da base de dados "Producao", deverá ser inserido, através de um Trigger, um novo registo na tabela "Custos\_Peca" com a seguinte lógica:

### Custo de Produção (associado ao tempo de produção)

- aa: €1,9/s
- ab: €1,3/s
- ba: €1,7/s
- bb: €1,2/s

### Prejuízo (associado ao tempo de produção e tipo de falha)

Para todos os tipos de produto (aa, ab, ba, bb), o prejuízo é calculado da seguinte forma:

- aa: €0,9/s + custo do tipo de falha
- ab: €1,1/s + custo do tipo de falha
- ba: €1,2/s + custo do tipo de falha
- bb: €1,3/s + custo do tipo de falha

### Custo do Tipo de Falha

- 01 (Ok): €0
- 02 (Falha na inspeção visual): €3
- 03 (Falha na inspeção de resistência): €2
- 04 (Falha na inspeção de dimensões): €2,4
- 05 (Falha na inspeção de estanqueidade): €1,7
- 06 (Desconhecido): €4,5

### Cálculo do Lucro

O lucro é calculado considerando um valor de venda fixo por tipo de produto, menos o custo de produção e prejuízo:

- aa: Valor de venda €120 - (Custo de Produção + Prejuízo)
- ab: Valor de venda €100 - (Custo de Produção + Prejuízo)
- ba: Valor de venda €110 - (Custo de Produção + Prejuízo)
- bb: Valor de venda €90 - (Custo de Produção + Prejuízo)

# Parte 2: Expansão do Sistema de Integração

Para melhorar o sistema de integração desenvolvido na Parte 1, esta segunda parte do trabalho propõe a incorporação de novas tecnologias e desafios, visando a criação de uma solução de integração mais completa e robusta.

## API com Web Services SOAP Financeiros

Foi solicitado ao IT o desenvolvimento de uma API com serviços financeiros para que outros sistemas possam consumir dados relevantes para a análise financeira da produção. Os seguintes Web Services SOAP devem ser implementados:

- Peça com Maior Prejuízo:
  - Descrição: Retorna o código da peça que resultou no maior prejuízo.
- Obter Custos Totais de Produção num período:
  - Descrição: Retorna o custo total de produção para todas as peças produzidas num período especificado.
- Obter Lucro total num Período:
  - Descrição: Retorna o lucro total obtido com a produção de todas as peças num período especificado.
- Obter Prejuízo total de cada uma das peças num período:
  - Descrição: Retorna o prejuízo total associado à produção de cada uma das peças dentro do período especificado.
- Obter Dados Financeiros Detalhados por Peça:
  - Descrição: Retorna todos os dados financeiros relevantes para uma peça específica.

## Desacoplamento de sistemas utilizando o Rabbit MQ

Foi pedido ao departamento de IT, para criar uma aplicação que substitua o sistema desktop legado apresentado na Parte 1. Assim, a nova aplicação deverá reunir da linha de produção os mesmos dados da aplicação legada: Data e hora de produção do artigo, código da peça, tempo de produção, resultados do teste.

Com o objetivo de desacoplar a aplicação desktop do sistema web e permitir uma comunicação mais flexível, implemente uma comunicação assíncrona utilizando RabbitMQ.

- A aplicação desktop, deverá publicar os dados de produção num tópico/exchange do RabbitMQ.
- Deverá ser desenvolvida uma aplicação que:
  - consuma os dados do Rabbit MQ e insira esses dados na Plataforma Web.
  - A aplicação tem uma GUI que deverá apresentar os dados da produção da peça, apenas se está se esta falhou no teste.

## Utilização de Rabbit MQ Streams para Streaming de Dados em Tempo Real

Pretende-se que a empresa implemente uma solução inovadora de transparência e gestão estratégica, fornecendo acesso instantâneo e detalhado aos dados de produção a todos os níveis de gestão. Para isso pretende-se adicionar à aplicação desktop a funcionalidade de publicar os dados numa estrutura de dados que sejam não destrutivas. Os dados deverão ser guardados de forma persistente em backlogs para poderem ser consumidos no futuro por aplicações data analytics e deverá existir o isolamento no consumo de dados entre consumidores.

Pretende-se que através de um painel de visualização de dados gestores possam acompanhar métricas críticas em tempo real, permitindo uma análise precisa e ágil do desempenho operacional. Considere a possibilidade de utilizar o Rabbit MQ Streams ou Kafka para o streaming de dados de produção em tempo real. Para isso crie uma nova aplicação desktop com interface GUI que faça a leitura de dados do stream de dados e apresente essas informações em

tempo real ao gestor. Esta aplicação deverá implementar algumas funcionalidades de Data Analytics como: número total de peças produzidas, número total de peças produzidas sem falhas, número total de peças OK, tempo médio de produção de cada peça.

### Requisitos Adicionais

- Documente resumidamente as decisões de design e implementação para cada uma das integrações (SOAP, RabbitMQ, RabbitMQ Stream ou Kafka).
- Apresente testes que demonstrem o funcionamento correto das todas as soluções desenvolvidas.