****

**Relatório TP1**

**Integração de Sistemas de Informação**

**Docente:** Célio Carvalho

**Alunos:** 18861 – André Felicidade

19256 – Yuri Lemos

19337 – João Figueiredo

# Índice

# 

[Índice 2](#_Toc89037122)

[Introdução 3](#_Toc89037123)

[Desenvolvimento da solução – Composição do ambiente de trabalho 4](#_Toc89037124)

[Desenvolvimento da Solução – Tópicos e Soluções Correspondentes 5](#_Toc89037125)

[ Tópico 1 5](#_Toc89037126)

[ Tópico 2 6](#_Toc89037127)

[ Tópico 3 8](#_Toc89037128)

[ Tópico 4 10](#_Toc89037129)

[ Tópicos 5, 6 e 7 10](#_Toc89037130)

# Introdução

Neste trabalho foi-nos proposto a realização de uma ou várias soluções, em que o foco principal era o consumo de e criação de microserviços, podendo estes depois serem e publicados por nós.

Inserido num contexto de obter soluções para alguns dos problemas que pudessem surgir em tempos de pandemia, o trabalho consistiu, brevemente referindo, na realização de “api’s restfull” e “soap”, numa pequena solução de “ETL” (Extract, Transform and Load), clientes simples feitos em “windows forms” e bases de dados em “mysql”.

# Desenvolvimento da solução – Composição do ambiente de trabalho

Para desenvolver as várias soluções deste trabalho prático, utilizamos como IDE o Visual Studio 2019, na linguagem de programação c#. Utilizamos ficheiros em formato JSON e XML também.

Para operar nas bases de dados usamos o *MySQL* *Workbench*.

Para testar as *requests* às *APIs* utilizamos o *insomnia* ou o próprio cliente em *Windows forms*.

# Desenvolvimento da Solução – Tópicos e Soluções Correspondentes

Todas as soluções foram realizadas em c# com base de dados em mySQL e clients com windows forms.

* Tópico 1 – API RESTFull com operações CRUD relacionadas com requisições de produtos. Realizada em c#. Base de dados em mySQL.

Diagrama UML 1 - ApiANEPC
 

Figura 1 - Diagrama UML ApiANEPC

* Tópico 2 – Client que consome a API anterior. Em windows forms e c#.

Exemplo: encomendar produtos.

|  |
| --- |
| Figura 2 - Painel inicial |
| Figura 3- Painel de fluxo, escolha de encomenda ou consulta de histórico |
| Figura 4 - Fluxo de nova encomenda, adicionar à encomenda |
| Figura 5 - Adicionar produto à encomenda designando o seu ID e a quantidade desejada |
| Figura 6 - Após adicionar todos os produtos desejados, requisitar |

* Tópico 3 – Serviço SOAP para simular um sistema da DGS e cliente para o consumir.

|  |
| --- |
| Figura 7 - Painel principal cliente soap  Figura 8 - Diagrama do SOAP |
| Figura 9 - consulta de infetados |
| Figura 10 - Consulta de suspeitos |

* Tópico 4 – Solução ETL que importa ficheiros em JSON e XML com estrutura definida para uma base de dados.

|  |
| --- |
| Figura 11 - Diagrama UML do FileImporter |

* Tópico 5 – Dashboard de consulta de dados relativos às equipas, encomendas de produtos, visitas e número de infetados.

|  |
| --- |
| Figura 12 - Consulta de dados internos na dashboard |
| Figura 13 - Consumo da API da database myisi |

* Tópico 6 - Implementar autenticação OAuth para a dashboard do tópico anterior.

|  |
| --- |
|  |

* Tópico 7 – Mostrar números atuais da pandemia na dashboard anterior, consumindo uma API externa.

|  |
| --- |
| Figura 14 - Diagrama UML do Consumo da API |

# Breve descrição dos processos:

**Tópicos 1 e 2**: Um cliente que consome uma *API*, chamando as suas rotas para cada método apropriado, designando também o tipo de pedido *http* feito sempre que se chama uma rota.

**Tópico 3:** Um conjunto *SOAP* e cliente que utiliza *SOAP*. Nesta arquitetura não há rotas, nem pedidos *http* tipados. Há apenas a chamada de métodos por parte do cliente que estão também definidos na origem.

**Tópico 4:** Uma solução de *ETL* que faz uso de librarias em c# para ler ficheiros *JSON* e *XML* e os transformar em objetos que podem ser carregados para uma base de dados.

**Tópico 5:** Cliente que exibe o resultado de *queries* a uma base de dados.

**Tópico 6:** No cliente supracitado haveria autenticação via *OAuth*, que consistiria primeiramente no registo de uma aplicação feita por nós, de seguida obteríamos *tokens* (em grosso modo) para cambiar com o sistema *OAuth* de forma a sempre que tivéssemos que fazer um pedido á API, utilizássemos o *token* de acesso em vez das credenciais do utilizador.

**Tópico 7:** Consumo de uma *API* externa. Utilizar as rotas desta para receber ficheiros *JSON* de formato desconhecido e des-serializar os mesmo para objetos, de seguida exibir os objetos/informação destes na *dashboard*.

# Conclusão:

Durante a realização deste surgiram diversos problemas que foram resolvidos com auto pesquisa. Embora não tenha sido implementada a autenticação Oauth, não deixamos de investigar sobre o assunto. Compreendemos a importância dos microserviços no mundo atual, e como a integração entre sistemas atraves de ficheiros Json ou Xml são realmente importantes e eficientes.

Aprendemos também a realizar “api´s restfull”, algo que é bastante pedido no mercado de trabalho, e que estas são independentes da tecnologia do lado do cliente.

Realizamos também um serviço “Soap”, que é algo com um overhead mais pesado que as “api´s” normais, mas que garante uma certa segurança/rigidez nos processos.

# Bibliografia/Webgrafia:

* <https://deanhume.com/a-simple-guide-to-using-oauth-with-c/>
* <https://firstinfinity.wordpress.com/modeling_rest_web_services/>
* <https://lucid.app/persona/lucidchart> ( utilizado para fazer os diagramas ).
* <https://stackoverflow.com/questions/32361031/using-list-between-forms>
* <https://stackoverflow.com/questions/38639481/winforms-c-sharp-pass-variable-between-forms>
* <https://stackoverflow.com/questions/16167924/c-sharp-with-mysql-insert-parameters>
* <https://stackoverflow.com/questions/65045746/warning-cs7022-the-entry-point-of-the-program-is-global-code-ignoring-progra>
* <https://stackoverflow.com/questions/20688922/the-entity-type-type-is-not-part-of-the-model-for-the-current-context>
* <https://stackoverflow.com/questions/58000123/visual-studio-cant-target-net-framework-4-8>
* <https://stackoverflow.com/questions/38494279/how-do-i-get-an-oauth-2-0-authentication-token-in-c-sharp>
* <https://stackoverflow.com/questions/18805735/deserialize-multiple-objects-from-stream-by-json-net>
* <https://stackoverflow.com/questions/38544094/how-to-pass-values-from-one-class-to-another-class-using-wpf>
* <https://stackoverflow.com/questions/13574442/trying-to-send-a-file-to-another-class-in-c-sharp>