

Licenciatura em Sistemas Informáticos
Integração de Sistemas Informáticos
Relatório do Trabalho Prático

André Cardoso & Leonel Fernandes
18848 & 18850

Barcelos, 4 de dezembro de 2021

Contents

1	Introdução	5
2	Modelo de Base de Dados	7
2.1	Serviço RESTfull API	7
2.2	Serviço SOAP	8
2.3	Importação de Ficheiros XML & JSON	9
3	Requisições de Equipas	11
3.1	Estrutura	11
3.2	Serviços e Modelos de Dados	12
3.2.1	Formato Serviços	12
3.2.2	Formato Modelos	14
3.3	‘Protocolo’ de Ação	17
3.4	Cliente	17
4	Aplicação Casos CoViD-19	19
4.1	Estrutura	19
4.2	Interfaces e Modelos de Dados	20
4.2.1	Formato Serviços	20
4.2.2	Formato Modelos	21
4.3	‘Protocolo’ de Ação	21

4.4	Cliente	21
5	Importação de XML/JSON	23
5.1	Formato XML	23
5.2	Formato JSON	24
5.3	Carregamento dos Ficheiros	24
6	Dashboard	25
6.1	Equipas	25
6.2	Visitas Diárias	26
6.3	Número Médio de Infetados	27
6.4	DGS	28
7	Conteúdos Extra	29
7.1	OAuth	29
7.2	Documentação	29
8	Conclusão	31

1 Introdução

Para este trabalho prático temos de desenvolver 3 serviços, assim como um cliente para cada um e ainda um dashboard para apresentar alguns dados sobre todas as API que foram desenvolvidas.

Os clientes devem ser capazes de comunicar com as API de forma a efetuarem as operações necessárias para resolução do problema.

O dashboard deve conseguir aceder a cada API e, assim como os clientes, conseguir efetuar as operações necessárias.

2 Modelo de Base de Dados

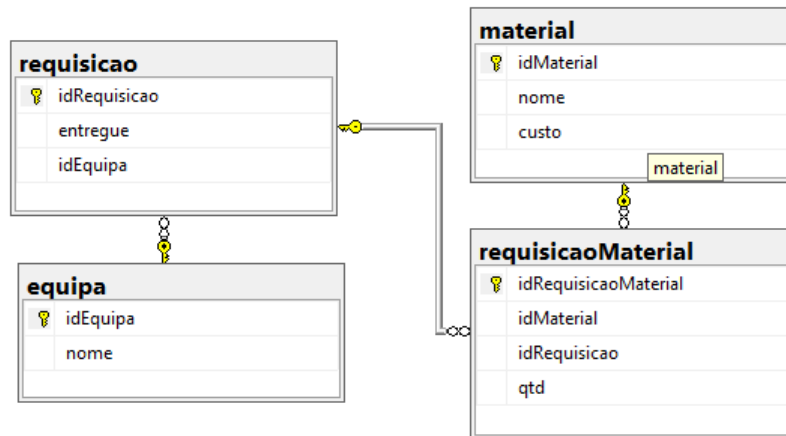
Para desenvolver uma solução ao problema proposto é necessária uma base de dados para armazenamento e acesso aos dados necessários. O modelo varia para cada problema apresentado:

2.1 Serviço RESTfull API

O registo de novas requisições de produtos para as visitas por parte das equipas multidisciplinares envolve quatro tabelas: "equipa", "material", "requisicao" e "requisicaoMaterial". Ao ser adicionada uma nova requisição são também adicionados os materiais presentes na mesma através da tabela "requisicaoMaterial" que guarda a quantidade de cada material. Todas as tabelas têm uma identificação "id" que se auto-incrementa.

Cada tabela guarda diferentes informações:

- Equipa: "idEquipa", "nome".
- Requisição: "idRequisicao", "entregue", (fk)"idEquipa".
- RequisiçãoMaterial: "idRequisicaoMaterial", (fk)"idRequisicao", (fk)"idMaterial", "qtd".
- Material: "idMaterial", "nome", "custo".



2.2 Serviço SOAP

O registo de casos de novos casos CoViD assim como os respetivos contactos envolvem um ou vários utentes. Para tal é criada uma tabela *Utente* com chave primária `idutente` simbolizadora de um valor representativo de um cidadão como o NIF (ou até o Número de Cidadão) e ainda um nome.

A adição de um caso requer uma data de (deteção da infeção) e o id do utente a que este se refere. Cada caso tem o seu id individual e auto incrementado.

Caso este tenha entrado em contacto com outros cidadãos, os respetivos devem ser adicionados como membros da tabela *Contactos*, que estabelece relação entre o caso atribuído ao primeiro infetado e os cidadãos que estejam em risco de infeção.

2.3 Importação de Ficheiros XML & JSON

A fiscalização do cumprimento das normas de isolamento envolve os utentes que são alvo da visita assim como irregularidades que possam ser encontradas durante esta última.

Para armazenamento destes dados adiciona-se as tabelas fiscalização e irregularidades, reutilizando a tabela já existente para o registo de casos no ponto anterior.

A tabela fiscalização mantém uma data de visita e o id do cidadão visitado. Tendo em conta a possibilidade de deteção de várias irregularidades numa visita esta tabela mantém uma relação de um para muitos, guardando o id da fiscalização em causa e uma breve descrição da irregularidade detetada.

3 Requisições de Equipas

Para atender à necessidade de organização das requisições efetuadas pelas equipas multidisciplinares foi desenhada uma API RestFull, assim como um cliente capaz de fazer uso da mesma.

3.1 Estrutura

Cada tabela tem as suas operações básicas:

- Equipa
 - Adição de Novas Equipas
 - Leitura das Equipas Existentes
 - Leitura das Equipas com mais valor gasto em requisições
- Requisição
 - Adição de Novas Requisições
 - Leitura das Requisições Existentes
 - Alteração do valor "entregue"
- RequisiçãoMaterial
 - Adição de Novas RequisiçõesMaterial
 - Leitura das RequisiçõesMaterial Existentes
- Material
 - Adição de Novos Materiais
 - Leitura dos Materiais Existentes

- Leitura dos Materiais mais usados em Requisições

Numa tentativa de simplificação acreditamos ser desnecessária as implementações de:

- Equipa
 - Atualização de Equipa
 - Remoção de Equipa
- Requisição
 - Remoção de Requisições Existentes
- RequisiçãoMaterial
 - Atualização de RequisiçãoMaterial
 - Remoção de RequisiçõesMaterial Existentes
- Material
 - Atualização de Materiais
 - Remoção de Materiais Existentes

3.2 Serviços e Modelos de Dados

De modo a criar serviços é necessário definir as rotas para cada um deles, assim como o tipo de modelos de dados que cada uma recebe ou envia.

3.2.1 Formato Serviços

Aqui serão descritas todas as rotas definidas na API Rest. De referir que todas as rotas aqui presentes estão disponíveis no "Swagger".

Equipa

Rotas para o controller de Equipa

- [Route("api/Equipa/getall ")]
public string GetEquipa()
- [Route("api/Equipa/getEquipaMaisCara ")]
public string GetEquipaMaisCara()
- [Route("api/Equipa/addequipa ")]
public string AddEquipa([FromBody] string equipa)

Requisição

Rotas para o controller de Requisição

- [Route("api/Requisicao/getall ")]
public string GetRequisicoes()
- [Route("api/Requisicao/addRequisicao ")]
public string AddRequisicao(string requisicao)
- [Route("api/requisicao/getrequisicaobyequipa/{idEquipa} ")]
public string GetRequisicaoByEquipca(int idEquipa)
- [Route("api/requisicao/updateEntregue/{idRequisicao} ")]
public string UpdateRequisicaoEntregue(int idRequisicao)

Material

Rotas para o controller de Material

- `[Route("api/Material/getall")]`
`public string GetMateriais()+`
- `[Route("api/Material/getMaterialMaisUsado")]`
`public string GetMaterialMaisUsado()`
- `[HttpPost("api/Material/addmaterial")]`
`public string AddMaterial([FromBody] string material)`

RequisiçãoMaterial

- `[Route("api/RequisicaoMaterial/getall")]`
`public string GetRequisicoesMateriais()`
- `[Route("getmaterialbyrequisicao/{idRequisicao}")]`
`public string GetRequisicaoMaterialbyRequisicaoEquipa +`
`+ (int idRequisicao)`
- `[Route("postRequisicaoMateriais/{idEquipa}")]`
`public string AddRequisicaoMaterial +`
`+ (int idEquipa,[FromBody] JsonElement body)`

3.2.2 Formato Modelos

Aqui estão descritos os modelos para cada uma das tabelas necessárias para a API Rest

Equipa

```
public class ModeloEquipa
{
    private int idEquipa;
    private string nome;

    [Required]
    public int Id{get=>idEquipa;set=>idEquipa=value;}
    [Required]
    public string Nome{get=>nome;set=>nome=value;}
}
```

Requisicao

```
public class ModeloRequisicao
{
    private int idRequisicao;
    private bool entregue;
    private int idEquipa;

    [Required]
    public int Id{get=>idRequisicao;set=>idRequisicao=value;}
    public bool Entregue{get=>entregue;set=>entregue=value;}
    public int IdEquipa{get=>idEquipa;set=>idEquipa=value;}
}
```

Material

```
public class ModeloMaterial
{
    private int idMaterial;
    private string nome;
    private float custo;

    [Required]
    public int Id{get=>idMaterial;set=>idMaterial=value;}
    public string Nome{get=>nome;set=>nome=value;}
    public float Custo{get=>custo;set=>custo=value;}
}
```

RequisicaoMaterial

```
public class ModeloRequisicaoMaterial
{
    private int idRequisicaoMaterial;
    private int idMaterial;
    private int idRequisicao;
    private int qtd;

    [Required]
    public int Id{get=>idRequisicaoMaterial;set=> +
+ idRequisicaoMaterial=value;}
    [Required]
    public int IdMaterial{get=>idMaterial;set=>idMaterial=value;}
    [Required]
    public int IdRequisicao{get=>idRequisicao;set=>idRequisicao=valu
    public int Qtd{get=>qtd;set=>qtd=value;}
}
```

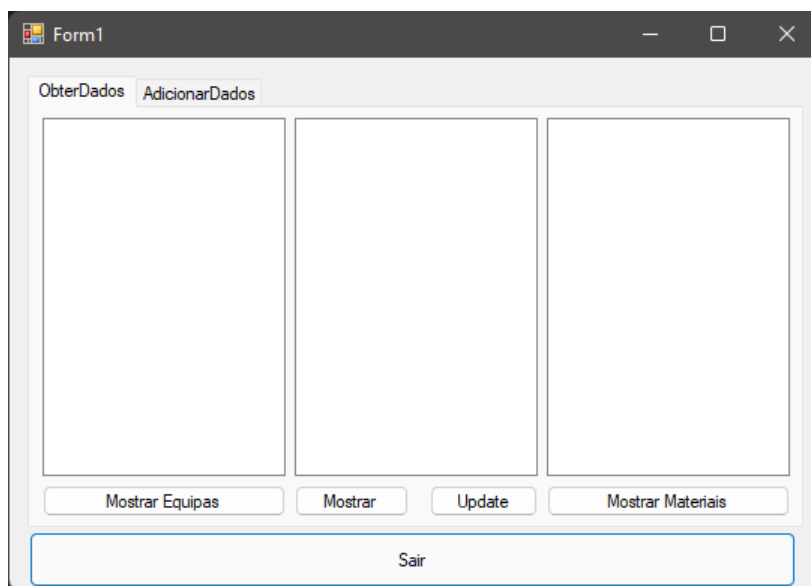

3.3 ‘Protocolo’ de Ação

- Abrir Conexão Servidor Base de Dados
- Definição de Query
- Execução de Query
- Apresentação de Dados/Resultado
- Encerrar Conexão Servidor Base de Dados

3.4 Cliente

Para fazer utilização dos serviços rest foi criado um cliente simples com "Windows Forms" de forma a conseguir utilizar as rotas definidas.

Tab do Cliente para Obter Dados



Tab do Cliente para Enviar Dados

Form1

ObterDados AdicionarDados

Obter Equipas

Criar Equipa

Obter Materiais

0 Adicionar Material X

Criar Material

Enviar Requisição

Sair

4 Aplicação Casos CoViD-19

É esperado que equipas que detetem casos de positivos de CoViD tenham a capacidade de:

- Identificar a primeira infeção.
- Identificar todas as pessoas com quem o infetado esteve em contacto.

Para responder a estas necessidades foi criado um Serviço Web SOAP e uma aplicação cliente para comunicação com este. Todos os dados inseridos serão armazenados numa base de dados já descrita.

4.1 Estrutura

Cada tabela tem as suas operações básicas:

- Utentes
 - Adição de Novos Cidadãos
 - Leitura dos Cidadãos Existentes
- Casos
 - Adição de Novos Casos
 - Leitura dos Casos Existentes
- Contactos
 - Adição de Novos Contactos
 - Leitura dos Contactos Existentes

Numa tentativa de simplificação acreditamos ser desnecessária as implementações de:

- Utentes
 - Atualização de Cidadãos
 - Remoção dos Cidadãos Existentes
- Casos
 - Atualização de Casos
 - Remoção dos Casos Existentes
- Contactos
 - Atualização de Contactos
 - Remoção dos Contactos Existentes

4.2 Interfaces e Modelos de Dados

De modo a criar os serviços (foi criado um serviço por cada um - utentes, casos, contactos) é necessário definir interfaces para cada um de modo a definir os contractos existentes. Abaixo é apresentado o formato de cada um.

4.2.1 Formato Serviços

```
[ServiceContract]
public interface IService
{
    [OperationContract(Name = "FunctionName")]
    DataSet Function1();

    [OperationContract]
    string Function2();
}
```

4.2.2 Formato Modelos

```
[DataContract]
public class ModeloDados
{
    // Constructor
    [DataMember]
    public string x { get; set; }
    [DataMember]
    public int y { get; set; }
}
```

Para facilitar na criação de queries ‘INSERT’ foi implementado o seguinte método de override a ToString():

```
public override string ToString()
{
    return "\"" + this.x.ToString() + "\", "
        + this.y.ToString() + "\"";
}
```

4.3 ‘Protocolo’ de Ação

- Abrir Conexão Servidor Base de Dados
- Definição de Query
- Execução de Query
- Apresentação de Dados/Resultado
- Encerrar Conexão Servidor Base de Dados

4.4 Cliente

Para consumir estes serviços foi implementada uma simples aplicação Forms com a capacidade de inserção e apresentação de dados.

5 Importação de XML/JSON

É esperado que as equipas de verificação do cumprimento das medidas de isolamento digitem relatórios com as informações dos Fiscalizados e das irregularidades registadas.

Os ficheiros exemplos podem ser encontrados ena pasta ‘/files’.

Os relatórios podem seguir sob os seguintes formatos:

5.1 Formato XML

Formato relatório PSP:

```
<?xml version="1.0" encoding="UTF-8"?>

<Fiscalizacao>
  <Fiscalizado nif="123" data="2021-11-01">
    <Irregularidade>
      primeira no xml
    </Irregularidade>
    <Irregularidade>...</Irregularidade>
  </Fiscalizado>
  <Fiscalizado nif="123" data="2021-11-01">
    </Fiscalizado>
  </Fiscalizacao>
```

5.2 Formato JSON

Formato relatório GNR:

```
{
  "Fiscalizados":
  [
    {
      "nif": 123,
      "data": "2003-03-03",
      "irregularidades":
      [
        {
          "descricao": "uma json"
        },
        {
          ...
        }
      ]
    },
    {
      ...
    }
  ]
}
```

5.3 Carregamento dos Ficheiros

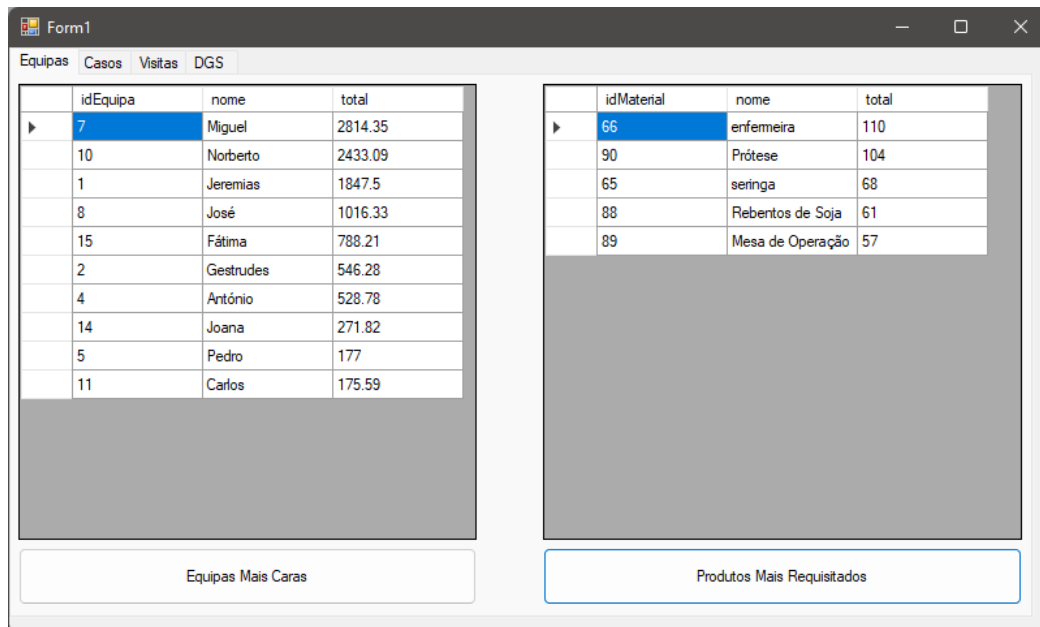
Cada ficheiro deve ser aberto na aba correspondente, na aplicação cliente de modo a ser tratado corretamente.

Cada um é carregado para memória aquando da sua leitura para as estruturas de dados já estudadas. No momento do seu carregamento são verificados quanto à sua validade.

6 Dashboard

6.1 Equipas

É enviado para o servidor Rest uma request para enviar os dados das Equipas mais caras e dos produtos mais requisitados. O servidor faz uma query à base de dados e envia os dados para o cliente, dados estes que são formatados para serem apresentados da seguinte forma:



The screenshot shows a web application window titled 'Form1' with a navigation bar containing 'Equipas', 'Casos', 'Visitas', and 'DGS'. The main content area displays two tables side-by-side. The left table, titled 'Equipas Mais Caras', lists team details with columns 'idEquipa', 'nome', and 'total'. The right table, titled 'Produtos Mais Requistados', lists product details with columns 'idMaterial', 'nome', and 'total'. Both tables have a grey shaded area below the data rows.

	idEquipa	nome	total
▶	7	Miguel	2814.35
	10	Norberto	2433.09
	1	Jeremias	1847.5
	8	José	1016.33
	15	Fátima	788.21
	2	Gestrudes	546.28
	4	António	528.78
	14	Joana	271.82
	5	Pedro	177
	11	Carlos	175.59

	idMaterial	nome	total
▶	66	enfermeira	110
	90	Prótese	104
	65	seringa	68
	88	Rebentos de Soja	61
	89	Mesa de Operação	57

6.2 Visitas Diárias

É necessário selecionar a tabela de fiscalização, separando as entradas inseridas no último mês. De seguida é feito o calculo diário de visitas, e formatado de modo a apresentar numa tabela.



6.3 Número Médio de Infetados

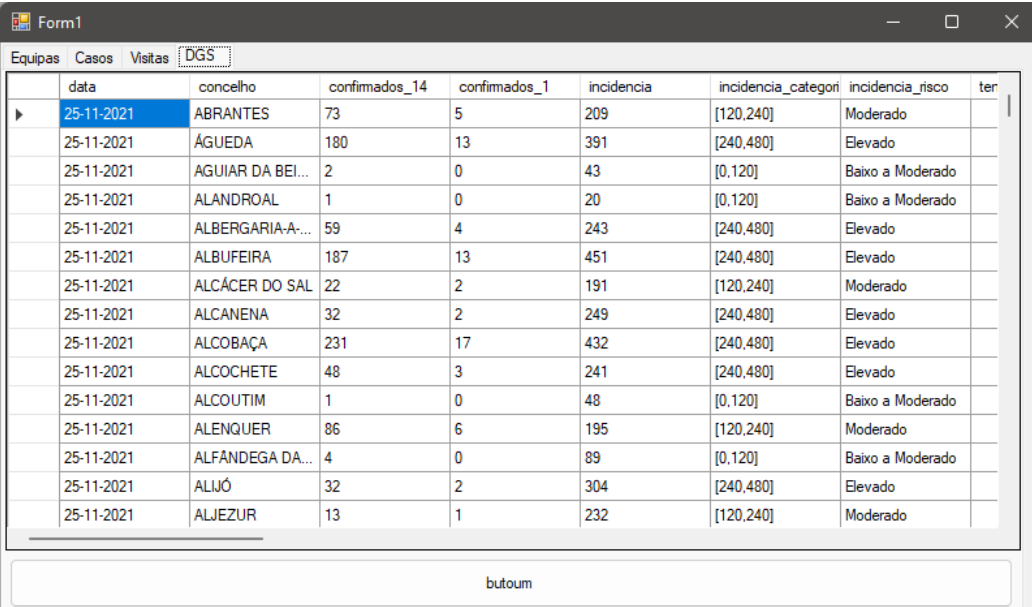
Selecionando as tabelas de Casos, pesquisamos e escolhemos as entradas inseridas nos últimos meses. De seguida é feito o calculo mensal de infeções e de seguida a média.

As tabelas são formatadas de modo a apresentar duas tabelas - Tabela de Casos Mensais e Tabela de Casos nos últimos 6 Meses.



6.4 DGS

Nesta secção são representados dados provenientes de uma API Rest cujo objetivo é mostrar dados sobre COVID em Portugal (<https://covid19-api.vost.pt> para documentação de rotas). Os dados provenientes da request são formatados e apresentados no dashboard:



	data	concelho	confirmados_14	confirmados_1	incidencia	incidencia_categoria	incidencia_risco	ter
▶	25-11-2021	ABRANTES	73	5	209	[120,240]	Moderado	
	25-11-2021	ÁGUEDA	180	13	391	[240,480]	Elevado	
	25-11-2021	AGUIAR DA BEI...	2	0	43	[0,120]	Baixo a Moderado	
	25-11-2021	ALANDROAL	1	0	20	[0,120]	Baixo a Moderado	
	25-11-2021	ALBERGARIA-A...	59	4	243	[240,480]	Elevado	
	25-11-2021	ALBUFEIRA	187	13	451	[240,480]	Elevado	
	25-11-2021	ALCÁCER DO SAL	22	2	191	[120,240]	Moderado	
	25-11-2021	ALCANENA	32	2	249	[240,480]	Elevado	
	25-11-2021	ALCOBAÇA	231	17	432	[240,480]	Elevado	
	25-11-2021	ALCOCHETE	48	3	241	[240,480]	Elevado	
	25-11-2021	ALCOUTIM	1	0	48	[0,120]	Baixo a Moderado	
	25-11-2021	ALENQUER	86	6	195	[120,240]	Moderado	
	25-11-2021	ALFÂNDEGA DA...	4	0	89	[0,120]	Baixo a Moderado	
	25-11-2021	ALJÓ	32	2	304	[240,480]	Elevado	
	25-11-2021	ALJEZUR	13	1	232	[120,240]	Moderado	

butoum

7 Conteúdos Extra

Este último capítulo do trabalho prático abrange os elementos desenvolvidos após a data de entrega original.

7.1 OAuth

Para a implementação de OAuth é utilizada uma API que será utilizada através de uma aplicação cliente em Forms. Esta aplicação cliente foi adaptada para comunicar com a API, pois à data da entrega original foi implementada como uma aplicação offline.

Utilizando JSON Web Tokens, é possível manter uma sessão com autenticação e autorizações aberta.

Para fazer uso do JWT são encriptados os dados de autenticação assim como informações adicionais (data de criação e tempo de vida, do token).

7.2 Documentação

Foi, para além da secção anterior, adicionada documentação para (algum) código, as funções têm uma breve descrição do seu funcionamento. A documentação pode ser acedida através do ficheiro "documentação" na pasta raiz do projeto.

8 Conclusão

Para concluir, neste trabalho prático foi necessário aplicar os conceitos aprendidos em aula, nomeadamente o funcionamento das API "Rest" e "SOAP", como comunicar com elas e como fazer o tratamento de dados.

A utilização ainda de "Windows Forms" é uma mais valia pois, sendo algo que nunca usamos, fortalece o nosso conhecimento sobre linguagens como o "C#".

Ainda foi utilizada uma API Rest Externa (<https://covid19-api.vost.pt>) onde podemos testar a nossa capacidade de tratamento de dados provenientes de outras API.

Achamos que este trabalho foi sem dúvida uma mais valia para o nosso conhecimento.