



INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática



## Bases de Dados 2

Ivan Viegas

Viviane Candeias

Beja

27 de janeiro 2021

INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

## Bases de Dados 2

Relatório sobre o Trabalho Prático de Bases de Dados 2 na Escola Superior de  
Tecnologia e Gestão do Instituto Politécnico de Beja

Elaborado por:

Ivan Viegas

Viviane Candeias

Orientado por:

Isabel Sofia Sousa Brito

Gonçalo Fontes

# ÍNDICE

---

1	Introdução.....	4
2	Descrição do processo de criação da BD .....	6
2.1	Modelo conceptual .....	6
2.2	Modelo físico .....	8
2.3	Modelo Lógico .....	9
2.4	Implementação no SSMS .....	9
3	Descrição do Método de preenchimento da BD .....	10
3.1	Scripts python.....	10
3.2	Scripts feitos em Excel.....	11
4	Descrição dos Stored Procedures e Triggers .....	12
4.1	Criação de Stored Procedures.....	12
4.1.1	Scripts feitos na Base de dados .....	12
4.2	Criação de Triggers.....	13
4.2.1	Scripts feitos na Base de dados .....	13
5	Descrição da Segurança .....	15
5.1	Login/User .....	15
5.2	Permissões .....	16
5.3	Roles/Schemas .....	16
5.4	Máscara e Criptografia .....	17
5.5	Auditoria.....	21
6	Descrição das Cópias de segurança (Backups) .....	23
7	Descrição do Desempenho .....	26
7.1	Índices Clusterizados e não clusterizados .....	26
7.1.1	Clusterizado .....	26
7.1.2	Não clusterizado .....	26
7.2	Filegroups .....	27
7.2.1	Scripts do filegroup feitos no trabalho .....	29
8	Descrição da Manutenção e automatização do servidor .....	30
9	Descrição de outras funcionalidades .....	32

9.1	Criação de um database mail .....	32
9.1.1	Criação do DataMail:.....	32
9.2	Reportbuilder .....	34
9.2.1	Instalação e configuração do software .....	34
10	Conclusões .....	39
10.1	Sumário do que foi atingido .....	39
10.2	Sumário do que não foi atingido .....	39
10.3	Trabalho futuro.....	39
11	Bibliografia .....	41
12	Anexos.....	42
12.1	Anexo 1 – Criação das tabelas .....	42
12.2	Anexo 2 – Código python para inserir dados .....	44
12.3	Anexo 3 – Excel .....	76
12.4	Anexo 4 – Stored procedure.....	77
12.5	Anexo 5 – Segurança .....	88
12.6	Anexo 6– Triggers .....	95
12.7	Anexo 7 – Backups.....	96
12.8	Anexo 8 – Filegroups .....	97
12.9	Anexo 9 – Report builder.....	102

## Índices de figuras

Figura 1 - Modelo conceitual do projeto .....	7
Figura 2 - Modelo físico do projeto.....	8
Figura 3 - Modelo logico da base de dados .....	9
Figura 4 - Exemplo como são construídos os scripts em python .....	10
Figura 5 - Exemplo da função random com probabilidade.....	11
Figura 6 - Exemplo da função random para data .....	11
Figura 7 - Permissões .....	18
Figura 8 - Parâmetro para visualizar .....	18
Figura 9- Aviso.....	19
Figura 10 - Coluna encriptada.....	19
Figura 11- Erro de inserção .....	20
Figura 12 - Código para mascarar a tabela .....	20
Figura 13- Visualização de dados encriptados.....	21
Figura 14 - Código T-sql da auditoria .....	22
Figura 15 - Planejamento dos filegroups .....	28
Figura 16 - select que permite ver a chave e a que filegroup esta inserida. ....	29
Figura 17 - Select mais detalhado sobre os filegroups .....	29
Figura 18 - Diretoria dos backups .....	31
Figura 19 - Plano de manutenção construído.....	31
Figura 20 - Aceder a função Database Mail.....	32
Figura 21 - Wizard da criação do database mail.....	32
Figura 22 - Criação do perfil.....	33
Figura 23 - Configuração do mail.....	33
Figura 24 - Conexão com a base de dados .....	35
Figura 25 - Inclusão dos comandos.....	35
Figura 26 - Inclusão de tabela .....	36
Figura 27 - Relatório pronto.....	37
Figura 28 - Inclusão de gráfico .....	37
Figura 29 - Relatório por concelho .....	38
Figura 30 - Relatório Geral .....	38

# 1 INTRODUÇÃO

---

No âmbito da Unidade Curricular de **Bases de Dados 2** do 2º ano da **Licenciatura em Engenharia Informática**, do **Instituto Politécnico de Beja** foi-nos proposto o desenvolvimento de uma base de dados com o objetivo de gerir os dados/informação de uma plataforma de monitorização da COVID-19, em que a entidade responsável pela saúde, em Portugal, no caso a DGS, poderá além emitir relatórios/boletins diários sobre a situação epidemiológica, alimentar o banco de dados com informações atualizadas sobre a situação.

Sabemos que há uma preocupação geral com a situação delicada que se passa no país em se tratando da COVID-19, e percebe-se a necessidade de informação fiel e atualizada para ajudar no combate a esta pandemia. Com isso em mente pesquisamos exemplos de sistemas compatíveis, inclusive o SINAVE, tentando assim atender o que nos foi solicitado neste projeto, de maneira eficiente e otimizada.

Este Relatório está dividido por **7 etapas** principais:

- **Criação da BD e dos seus elementos;**
- **Método de preenchimento da BD;**
- **Stored Procedures e Triggers;**
- **Segurança;**
- **Cópias de segurança (Backups);**
- **Desempenho;**
- **Manutenção e automatização do servidor;**

## 2 DESCRIÇÃO DO PROCESSO DE CRIAÇÃO DA BD

---

Neste ponto pretende-se mostrar a configuração e manutenção de um servidor de bases de dados (*SQL Server 2019*). Este servidor tem pelo menos 5 discos, os quais são utilizados para a aumentar o desempenho e garantir a gestão de memória.

Também mostrar como foi desenvolvido a base de dados utilizando os recursos obtidos na unidade curricular de Base de Dados I, como também pesquisas feitas na internet.

### 2.1 MODELO CONCEPTUAL

O nosso modelo conceptual foi pensado com base num sistema atual em Portugal que cria a notificação obrigatória de doenças transmissíveis e outros riscos em saúde pública, permitindo ao médico notificar em tempo real, à autoridade de saúde local, a ocorrência de uma doença transmissível para assim começarem as implementações de medidas de prevenção e controle da epidemia, tentando com isso limitar a disseminação da doença e a ocorrência de casos adicionais.

No nosso caso em específico, modulamos uma Base de dados adequada para a doença COVID-19. Uma doença provocada pelo novo coronavírus SARS-COV-2, altamente contagiosa que alcançou todo o mundo se transformando numa pandemia mundial.

A **Figura 1** mostra o nosso modelo conceitual da base de dados, onde atribuímos 8 entidades cada uma com os seus atributos. Foi criado levando em consideração a 3 FN.

A maneira que idealizamos a construção desta base de dados foi para otimizar a inserção e processamento dos dados, para isso nossas tabelas tem a cardinalidades entre si bem definidas e bem estruturadas nos permitindo acesso total as tabelas e seus atributos. Descrevendo melhor a utilização do nosso modelo, o usuário começa sempre por inserir uma notificação, essa tem um motivo ou vários para ser criada, ao mesmo temos que apura os sintomas que deram origem ou não a mesma. Depois de criada temos uma cardinalidade múltipla que nos permite fazer a ligação da notificação com a entidade pessoa, que por sua vez já está contida as informações pessoais

do utente, o seu endereço e também se teve contacto com outra pessoa. Também tem relação com a entidade resultado que quando é criado a notificação também é criado um resultado (através de um *Trigger*) porém com o campo do mesmo a dizer “Aguardando”. A partir do momento que temos a informação do resultado fazemos um *update* do campo criado atualizando-o para Positivo, Negativo ou Inconclusivo. A entidade resultado também está associado a entidade tipos de positivo que nos permiti saber se o mesmo está ativo ou se já se recuperou ou morreu da doença.

Durante a implementação do modelo conceitual tiramos dúvidas e consultamos os orientadores do trabalho, e surgiu uma pergunta da parte de um deles sobre o índice R(t), então pensamos em diversas maneiras de como conseguiríamos que nossa base de dados respondesse a este questionamento e para isso criamos a entidade contacto que permite saber se a pessoa esteve em contacto com outra ou até com várias.

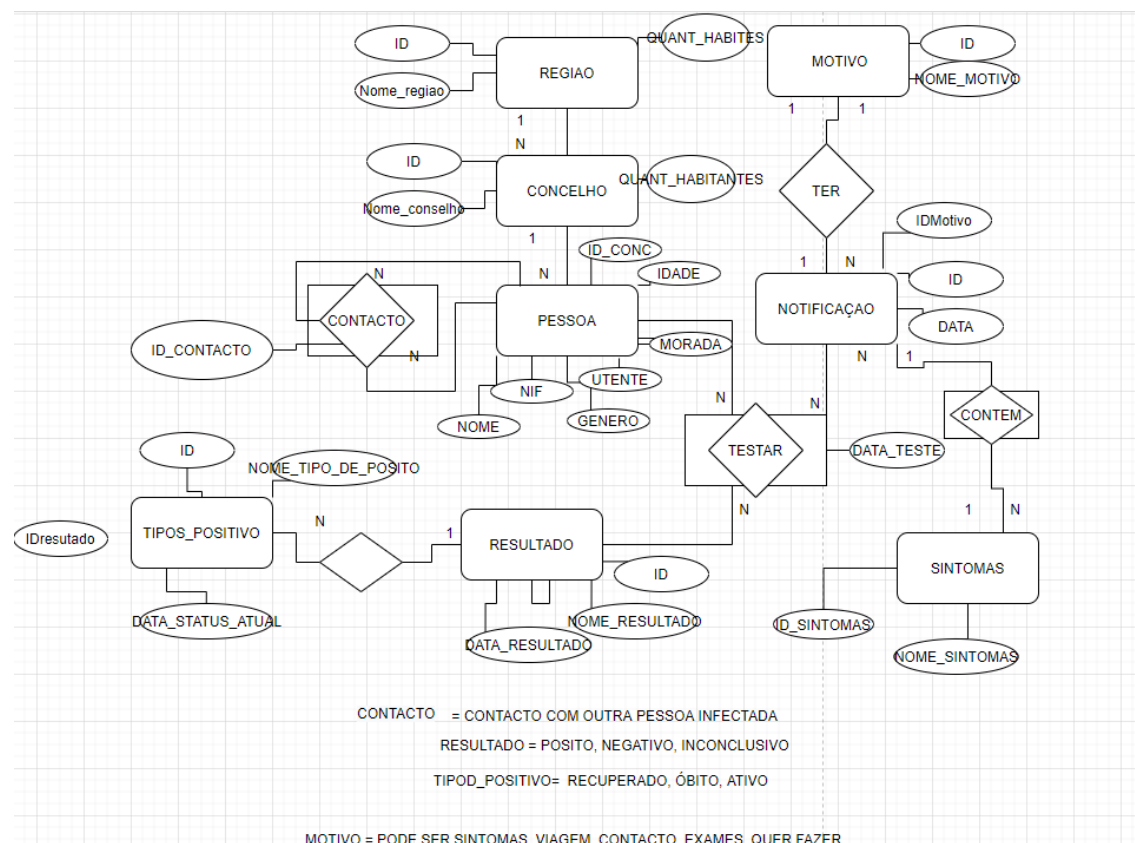


Figura 1 - Modelo conceitual do projeto



## 2.2 MODELO FÍSICO

Depois de fazermos o modelo conceitual e revermos se estava completo, começamos por construir o modelo físico.

A **figura 2** mostra as entidades referidas no tópico anterior e também mostra novas entidades originadas pelas cardinalidades de **N** para **N** entre as entidades:

```
REGIAO(ID_REGIÃO, NOME_REGIÃO, QUANT_HABITANTES)
CONSELHO(ID_CONSELHO, NOME_CONSELHO, QUANT_HABITANTES, ID_REGIÃO)
PESSOA(ID_PESSOA, NIF, , NOME, IDADE, GENERO, MORADA, ID_CONSELHO)
CONTACTO(ID_CONTACTO, ID_PESSOA)
MOTIVO(ID_MOTIVO, NOME_MOTIVO)
NOTIFICAÇÃO(ID_NOTIFICAÇÃO, DATA, ID_MOTIVO)
NOTIFICAÇÃO_SINTOMAS(ID_NOTIFICAÇÃO, ID_SINTOMAS)
RESULTADO (ID_RESULTADO, NOME_RESULTADO, DATA_RESULTADO)
TIPOS_POSITIVO(ID_TIPOS_DE_POSITIVOS, NOME_TIPO_POSITIVO, ID_RESULTADO,
DATA_TIPO_POSITIVO)
PESSOA_NOTIFICAÇÃO_RESULTADO(ID_PESSOA, ID_NOTIFICAÇÃO, ID_RESULTADO,)
SINTOMAS(ID_SINTOMAS, NOME_SINTOMAS)
```

*Figura 2 - Modelo físico do projeto*

## 2.3 MODELO LÓGICO

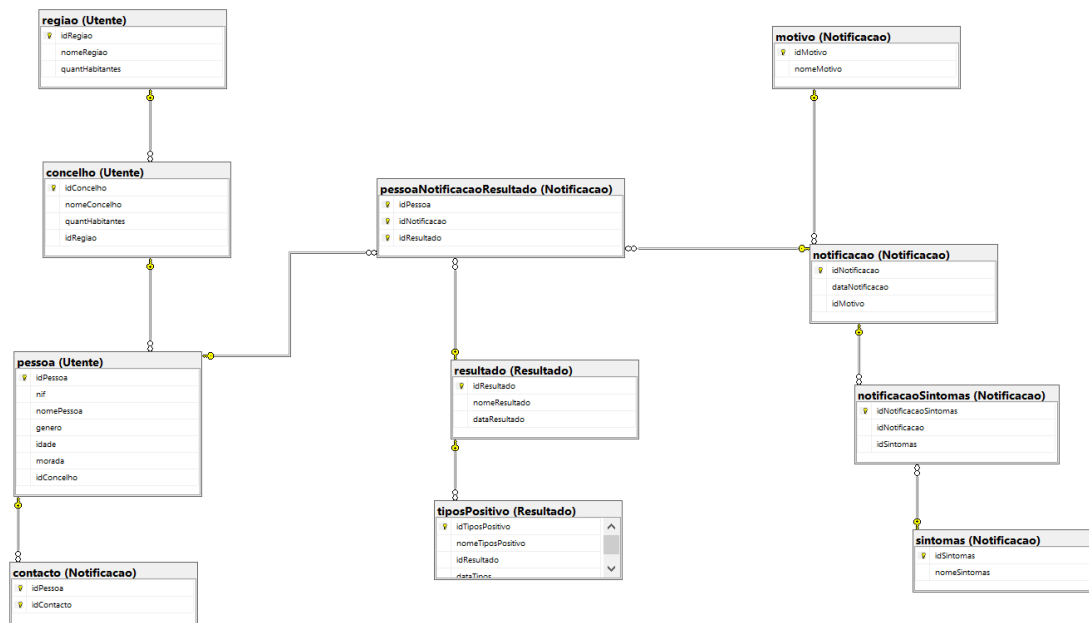


Figura 3 - Modelo logico da base de dados

## 2.4 IMPLEMENTAÇÃO NO SSMS

Como já tínhamos feito o modelo físico, começamos por construir a base de dados no SSMS utilizando a linguagem *T-SQL* que se encontra no **anexo 1**, devido a um problema com o disco interno de um dos alunos que realizou este projeto, onde o mesmo tinha os scripts originais feitos dentro do disco que deixou de funcionar, tivemos que criar os scripts com a ferramenta do SSMS que cria os scripts das tabelas que estão na base de dados.

Começamos por inserir as tabelas sem levar em consideração as chaves estrangeiras, ou os índices, ou seja, inserimos as tabelas vazias depois inserimos os seus dados para só a seguir adicionarmos as chaves estrangeiras e índices, para tal utilizamos os comandos *ALTER TABLE* e *WITH CHECK*. Tal procedimento só tivemos conhecimento após assistirmos as aulas da cadeira de Base de Dados 2 e sem dúvidas foi a melhor maneira de começar a nossa BD, pois ficou mais fácil de identificar erros minimizando assim retrabalho.

### 3 DESCRIÇÃO DO MÉTODO DE PREENCHIMENTO DA BD

---

Neste ponto iremos demonstrar como foram gerados e inseridos os dados que permitiram fazer os testes na base de dados construída no tópico anterior, inserimos pelo menos 1500 dados para testar a base de dados, com exceção em algumas das tabelas que foram +1500 dados e na tabela Pessoa foi inserido 10002 dados, onde o utente 10002 corresponde a ninguém (campos zerados).

#### 3.1 SCRIPTS PYTHON

No **anexo 2** contêm todos os scripts gerados para podermos inserir os dados na base de dados.

Basicamente estes scripts em *Python* vão abrir um ficheiro *SQL (Query)* e caso não existir vão criar esse ficheiro, e vão ter várias linhas preenchidas como podem ver na **figura 4**.

Como foi mencionada o tópico Implementação no SSMS, devido a uma perda de dados (*Querys*) não temos as *querys* que supostamente foram introduzidas na base de dados, mas podemos gerar outras executando os scripts feitos em *python*.

```
INSERT INTO pessoa VALUES (2000000000,'Kenzo Valido', 'Masculino', 31, 'Avenida de Moreno n10 ', 303);
INSERT INTO pessoa VALUES (2000000001,'Prince Gois Parafita', 'Masculino', 45, 'Praça Madureira n18 ',209);
INSERT INTO pessoa VALUES (2000000002,'Wilson Djayson Martinho Mendonça', 'Masculino', 99, 'Avenida de Jardim n22 ', 24);
INSERT INTO pessoa VALUES (2000000003,'Suri Morgado', 'Femenino', 84, 'Rua de Borba n5 ', 196);
INSERT INTO pessoa VALUES (2000000004,'Nídia Beltrão Lancastre', 'Femenino', 87, 'Rua de Vilalobos n41 ',33);
INSERT INTO pessoa VALUES (2000000005,'Elena Olga Quesado Maior', 'Femenino', 25, 'Rua de Janes n38 ', 83);
INSERT INTO pessoa VALUES (2000000006,'Yanni Avelar', 'Masculino', 35, 'Avenida de Valentim n35 ', 93);
INSERT INTO pessoa VALUES (2000000007,'Alcino Xisto Coronel', 'Masculino', 95, 'Praça Orricha n47 ',230);
INSERT INTO pessoa VALUES (2000000008,'Giorgi Dêlcio Bonito Domingues', 'Masculino', 73, 'Avenida de Girão n42 ', 246);
INSERT INTO pessoa VALUES (2000000009,'Vanessa Quinta', 'Femenino', 77, 'Avenida de Moita n47 ', 17);
INSERT INTO pessoa VALUES (2000000010,'Catarina Bacelar Abelho', 'Femenino', 48, 'Travessa Viana n30 ',18);
INSERT INTO pessoa VALUES (2000000011,'Maria Cadija Ornelas Negreiros', 'Femenino', 69, 'Praça Raposo n46 ', 103);
INSERT INTO pessoa VALUES (2000000012,'Willian Godinho', 'Masculino', 10, 'Rua de Teles n44 ', 26);
INSERT INTO pessoa VALUES (2000000013,'Alcino Imperial Macena', 'Masculino', 32, 'Avenida de Severiano n27 ',111);
INSERT INTO pessoa VALUES (2000000014,'Ioan Jonathan Mateus Lobato', 'Masculino', 78, 'Avenida de Vasques n17 ', 123);
INSERT INTO pessoa VALUES (2000000015,'Mercês Murtinho', 'Femenino', 20, 'Rua de Gameiro n32 ', 168);
INSERT INTO pessoa VALUES (2000000016,'Aaliyah Queiroga Muniz', 'Femenino', 38, 'Rua de Valverde n42 ',55);
INSERT INTO pessoa VALUES (2000000017,'Dulce Nicole Mota Miguel', 'Femenino', 71, 'Avenida de Fazendeiro n25 ', 286);
INSERT INTO pessoa VALUES (2000000018,'Yannick Quinterno', 'Masculino', 72, 'Travessa Bezerril n31 ', 86);
INSERT INTO pessoa VALUES (2000000019,'Vasile Manso Almeida', 'Masculino', 44, 'Avenida de Palos n18 ',146);
INSERT INTO pessoa VALUES (2000000020,'Samuel Jonathan Proença Marins', 'Masculino', 62, 'Avenida de Paião n14 ', 161);
INSERT INTO pessoa VALUES (2000000021,'Marília Pestana', 'Femenino', 26, 'Avenida de Argolo n40 ', 78);
INSERT INTO pessoa VALUES (2000000022,'Diva Barateiro Póvoas', 'Femenino', 82, 'Praça Santarém n27 ',118);
INSERT INTO pessoa VALUES (2000000023,'Heloisa Luenna Botica Vargas', 'Femenino', 95, 'Travessa Marins n39 ', 276);
```

Figura 4 - Exemplo como são construídos os scripts em python

## 3.2 SCRIPTS FEITOS EM EXCEL

Como nossos scripts em *Python* eram limitados e a lógica que pretendíamos aplicar para as próximas inserções eram mais complexas, decidimos usar o Excel para isso. Queríamos fazer na realidade um *front-end* para que no futuro o utilizador pudesse fazer tais inserções de maneira mais dinâmica, porém infelizmente não tínhamos o tempo necessário para desenvolvê-lo pois seria um assunto que não abordamos em sala de aula.

Usamos algumas funções *Random* com índice para tentarmos manipularmos a diversificação nas tabelas resultados, tiposPositivos, sintomas e motivo, inclusive aquelas que tinham como campo a data. Tal função disponibilizava os dados de acordo como percentual de probabilidade que queríamos, por exemplo, se quiséssemos que aparecessem mais casos Ativos do que óbito na tabela tiposPositivo aumentávamos o percentual de probabilidade para isso acontecer. Depois de feito, concatenamos as células de acordo com a necessidade para inserção no Sql server.

No **anexo 3** temos alguns exemplos de *insert's* e disponibilizaremos na documentação deste projeto o arquivo .xlsx para demonstração.



```
=SE.ERRO(ÍNDICE($A$2:$A$16;ALEATÓRIOENTRE(1;LINS($A$2:$A$16)/$E$1));"")
```

Figura 5 - Exemplo da função random com probabilidade



```
=ÍNDICE($L$3:$L$33;ALEATÓRIOENTRE(1;LINS($L$3:$L$33)))
```

Figura 6 - Exemplo da função random para data

## 4 DESCRIÇÃO DOS STORED PROCEDURES E TRIGGERS

---

Neste ponto pretende-se que explicar o que são *Stored Procedures* e *Triggers*, demonstrando a nossa perceção sobre o assunto e como o aplicamos na nossa base de dados.

### 4.1 CRIAÇÃO DE STORED PROCEDURES

*Stored Procedure*, que traduzido significa Procedimento Armazenado, é um conjunto de comandos em SQL que podem ser executados de uma vez só, como vemos em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual, não é um procedimento atualizado, mas é de grande valia para o cotidiano de um Administrador de BD.

Um *Stored Procedure* pode reduzir o tráfego na rede, melhorar a performance da base de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc.

#### 4.1.1 Scripts feitos na Base de dados

Basicamente criamos 30 *Stored Procedures* na nossa base de dados, pois imaginamos um cenário onde em qualquer momento a informação que alguém precisasse estaria mais fácil de ser alcançada executando apenas o SP. Alguns são o conjunto de soluções referente ao *dashboard* que existe da **DGES** e outros são situações referidas no enunciado como também outras sugeridas pelos professores e outras criamos a partir da necessidade de nossa BD.

Estes scripts estão no **Anexo 4**, onde contém por todos os SP que criamos e o comentário explicando o cada um faz.

Alguns em especial como o [dbo].[Utentepessoa], [dbo].[Valordeincidencia], [dbo].[IndiceRT], [dbo].[RiscoporConselho] foram criados para dinamizar a inserção de dados e emissão de relatórios respetivamente.

## 4.2 CRIAÇÃO DE TRIGGERS

O termo *trigger* (gatilho em inglês) define uma estrutura do base de dados que funciona, como o nome sugere, como uma função que é disparada consoante alguma ação do utilizador. Geralmente essas ações que disparam os *triggers* são alterações nas tabelas por meio de operações de inserção, exclusão e atualização dos dados (*insert*, *delete* e *update*).

Um gatilho está relacionado a uma tabela, sempre que uma dessas ações é efetuada sobre essa tabela, é possível dispará-lo para executar alguma tarefa, ou para servir de alerta se algo está mudando ou está alarmante, dependendo do modo de sua configuração.

### 4.2.1 Scripts feitos na Base de dados

Basicamente criamos 3 *Triggers* na nossa base de dados, pois achamos que eram aqueles que supostamente eram necessário para automatizar e nos alertar de alguma maneira sobre a base de dados.

Estes scripts estão no **Anexo 6**, lá estão os detalhamentos de como funciona os scripts dos triggers: [createresul], [checkpoo] e [resultado\_tipospositivos] respetivamente.

O trigger [checkpoo] verifica se passou 1% da população infetada e quando isso acontece disparam, fazendo um print e consequentemente manda um mail (iremos falar mais sobre isso no tópico Descrição de outras funcionalidades) para um dos criadores a partir de um e-mail criado para o projeto, ele utiliza o SP porcpoo para fazer a comparação com os percentuais.

Já o trigger [createresul] surgiu da necessidade de criarmos mais um tipo de positivo para o domínio da tabela resultado, pois ao inserirmos uma notificação a mesma ainda não tem resultado definido e com isso geramos um resultado com o nome “AGUARDANDO” (momento em que o trigger dispara) para assim que obtivermos o resultado definitivo faremos um update através de um SP para o valor desejado: POSITIVO, NEGATIVO OU INCONCLUSIVO.

Por fim o trigger [resultado\_tipospositivos] foi criado através da necessidade que tínhamos de criar um histórico para nossa BD, ou seja, quando inserimos ou fazemos um update de um resultado POSITIVO em nossa tabela resultado, esse trigger cria automaticamente um tipo de positivo chamado ATIVO na tabela tipoPositivo, pois subtemde-se que todo POSITIVO é ATIVO até se tornar um RECUPERADO ou ÓBITO. Quando passarem 10 dias (prazo para a recuperação do COVID-19) ou se o utente vier a falecer, faremos um insert com a data e o novo “status” na tabela tipoPositivo, criando assim um histórico para esse paciente.

## 5 DESCRIÇÃO DA SEGURANÇA

---

Um dos pontos de discussão que tivemos durante a construção de nossa base de dados, foi a maneira de como deixaríamos a nossa BD segura, no sentido de proteger seus dados, porém ainda sim de fácil manipulação e acesso para os usuários em específico.

Primeiro aspeto que lembramos quando falamos de segurança em um sistema de banco de dados foi que esse sistema é um sistema como outro qualquer, a segurança dela esta atrelada a segurança do ambiente que ele está inserido. Nesse sentido todos os aspectos da segurança como por exemplo o da informação, de hardware e de a rede, são pertinentes a segurança do banco de dados, inclusive também temos de levar em consideração questões éticas e legais e a política de uso da informação que são estabelecidas pelas empresas.

A nossa principal preocupação seria evitar a perda da integridade (proteger as informações contra modificações inapropriadas como criação ou exclusão de dados ou modificação nos seus valores), a perda de disponibilidade (indisponibilidade de funcionalidades e dados) e a perda da confidencialidade (disponibilização de dados ou informação de forma não autorizada).

### 5.1 LOGIN/USER

Criamos quatro Logins e quatro usuários, [UserAdm], [UserCrud], [UserJrDb], [UserRead] fizemos através Transact-SQL como irão verificar no **anexo 5**.

As características de nossos usuários deixamos de uma maneira generalizada para que possa ser adaptada a realidade que qualquer empresa:

[UserAdm] – é o super usuário tem acesso total a BD, pode criar contas, conceder e revogar privilegios e estabelecer níveis de segurança;

[UserCrud] – é o usuário que só faz coisas básicas a nível da BD, foi-lhe dado algumas permissões para aceder a base de dados. Pode ser um funcionário que recebe relatórios diários com notificações a emitir e resultados a inserir;



[UserJrDb]-faz parte do time do adm da base de dados, porém não tem acesso a tudo a nível do servidor;

[UserRead] – pode ser o funcionário que emite os relatórios diários ou os funcionários de outros setores que usam esses dados apenas para consulta.

## 5.2 PERMISSÕES

Quando pensamos em quais permissões disponibilizar aos nossos usuários chegamos a conclusão que o menor nível de permissão possível para o usuário fazer o trabalho seria a melhor solução, ou seja, para não termos indisponibilidade de dados e nem acesso a dados ou informações não autorizadas, o usuário deve ser capaz de ter acesso aos dados necessários com nível de acesso suficiente (e não mais do que suficiente), para que o usuário realize seu trabalho.

Utilizamos os comandos Grant e Revok para dar e retirar permissões conforme está no [anexo 5](#). Cada usuário tem permissões específicas, como por exemplo o usuário UserCrud foi-lhe dado permissões de inserir dados, alterar dados, pesquisar dados e excluir dados de uma tabela além de permissões para executar alguns stored procedure, já o user JrDB só tem acesso a fazer consultas, inclusive poder fazer view em SP, porém não pode alterá-las nem as criar. A nível de base de dados não tem permissões para manipular nenhum dado, já a nível de servidor pode fazer a manutenção(algumas), por isso criamos um Role específico para ele. O user Ready, a nível de base de dados só pode fazer select (consultar dados), inclusive tem acesso a algumas SP com essa finalidade.

## 5.3 ROLES/SCHEMAS

Os schemas agrupam logicamente tabelas, procedimentos e visualizações, são uma coleção de objetos dentro de um determinado database, no nosso entendimento podemos olhá-la como divisões departamentais. São de fundamental importância para a performance e segurança da

BD principalmente porque você também pode conceder permissões para apenas um esquema, para que os usuários possam ver apenas o esquema ao qual têm acesso e nada mais.

Para nosso caso subdividimos o nosso conjunto de tabelas em 3 Schemas, foram criadas através do Management e exportados para o T-sql, conforme pode ser visto no **anexo 5**:

**Notificação** - onde estão as tabelas contacto, motivo,notificação, notificacaoSintomas, pessoaNotificacaoResultado e sintomas;

**Resultado** - onde estão as tabelas resultado e tiposPositivos;

**Utente** - onde estão as tabelas concelho, pessoa e região.

## 5.4 MÁSCARA E CRIPTOGRAFIA

Para deixar ainda mais segura a nossa base de dados tentamos implementar uma Criptografia, porém infelizmente aconteceu perda de disponibilidade, ou seja, houve indisponibilidade de funcionalidades e dados. Esse problema ocorreu quando encriptamos algumas colunas e o usuário crud não conseguia inserir dados nessa coluna, com isso desistimos da encriptação, resolvemos esse problema mascarando dados para alguns utilizadores para assim garantirmos a confidencialidade dos dados.

Tivemos dificuldade com a criptografia pois como as colunas das tabelas criptografadas não eram de fácil acesso e não conseguíamos dar permissões CRUD para nossos usuários sem habilitar a visualização dos dados, pois só quem conseguiria ver tudo seria o ADM. Com isso criptografamos apenas uma coluna da tabela sintomas a nível de demonstração. Para isso fizemos o always encrped que é um recurso criado para proteger dados confidenciais fornecendo uma separação entre aqueles que possuem os dados (e podem exibi-lo) e aqueles que gerenciam os dados (mas que não devem ter acesso).

Percebemos que podemos dar permissão ao UserAdm para visualizar os dados encriptados dessa tabela e assim o fizemos:

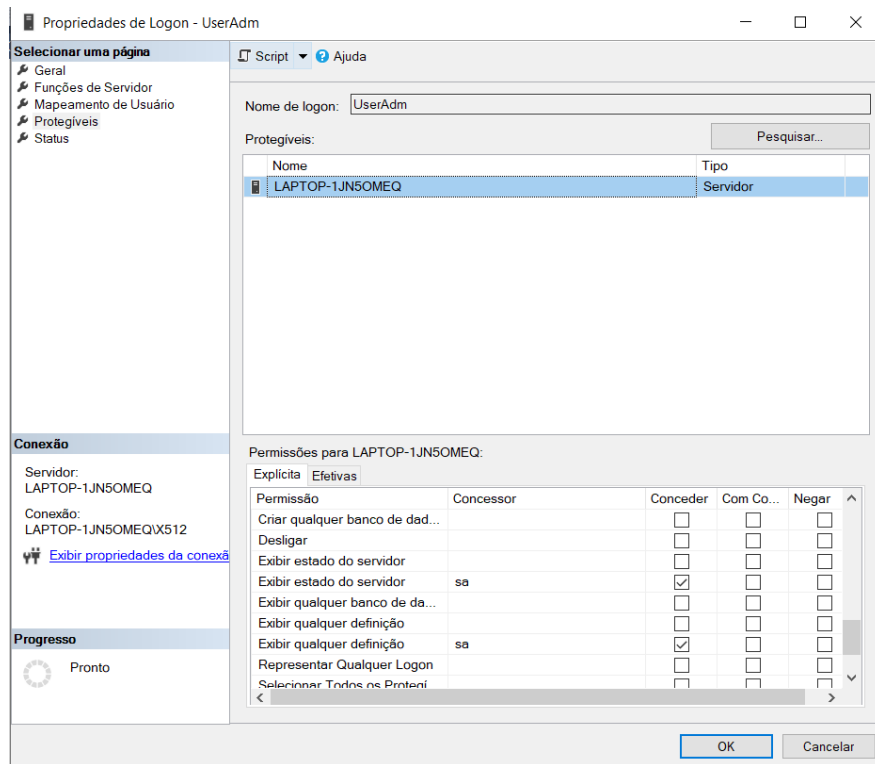


Figura 7 - Permissões

E para ara visualizar novamente os dados originais no SSMS, informamos o parâmetro:

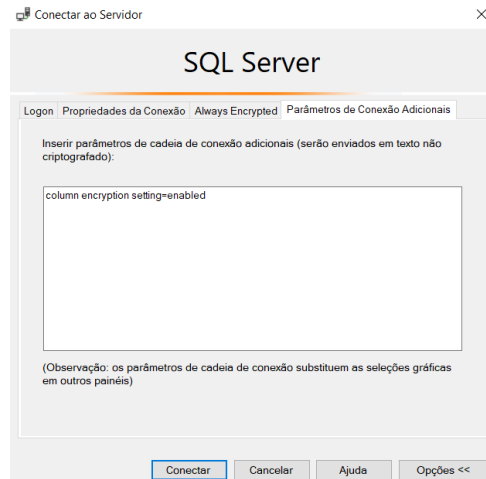


Figura 8 - Parâmetro para visualizar

A primeira vez que entramos deu o seguinte aviso:

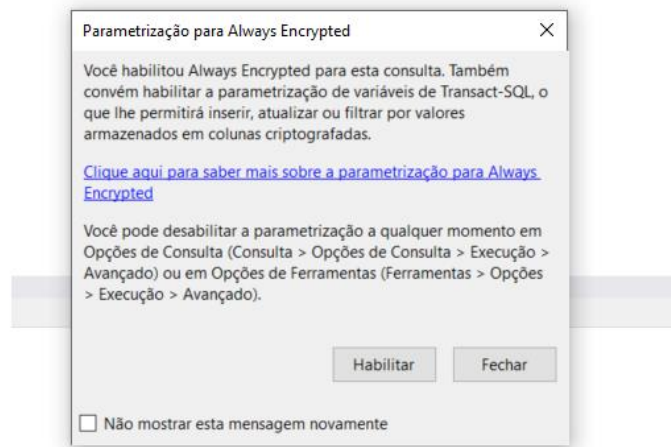


Figura 9- Aviso

Para encriptar fomos na tabela sintomas e clicamos com o botão direito, seleccionamos a coluna que queríamos encriptar:

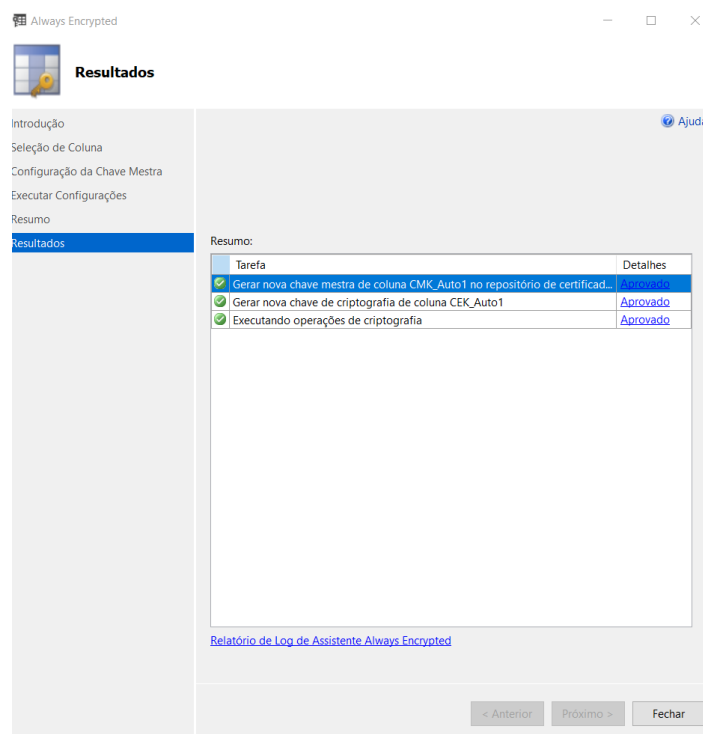
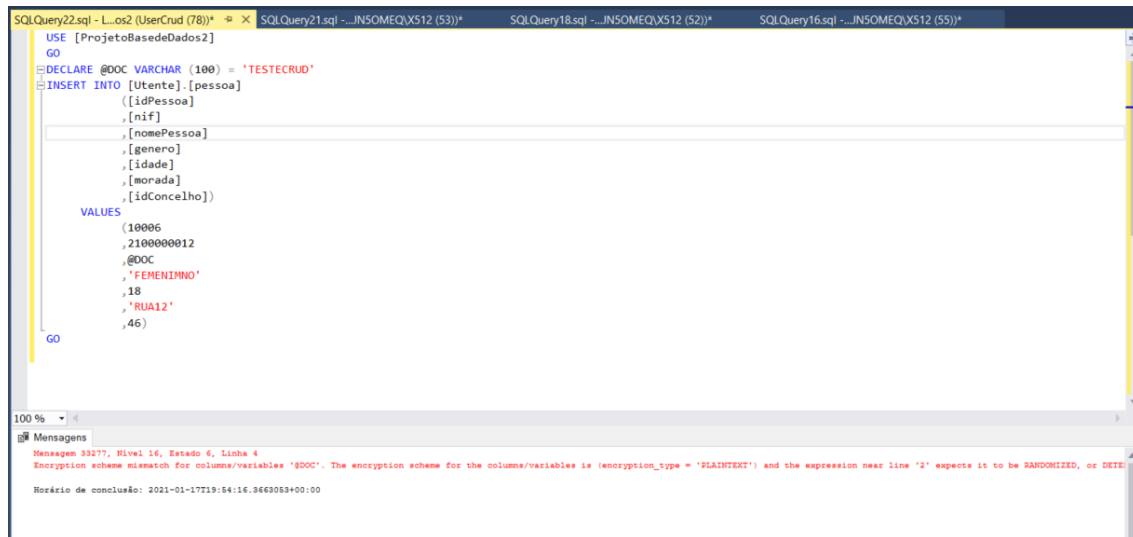


Figura 10 - Coluna encriptada

A nossa primeira escolha foi a tabela Pessoa, por conta de todos os dados pessoais que há nela, porém dava sempre um erro quando nosso usuário Crud fazia um insert:



```
USE [ProjetoBaseDados2]
GO
DECLARE @DOC VARCHAR (100) = 'TESTECRUD'
INSERT INTO [Utente].[pessoa]
([idPessoa]
,[nif]
,[nomePessoa]
,[genero]
,[idade]
,[morada]
,[idConcelho])
VALUES
(10006
,2100000012
,@DOC
,'FEMENINO'
,18
,'RUA12'
,46)
GO
```

100 %

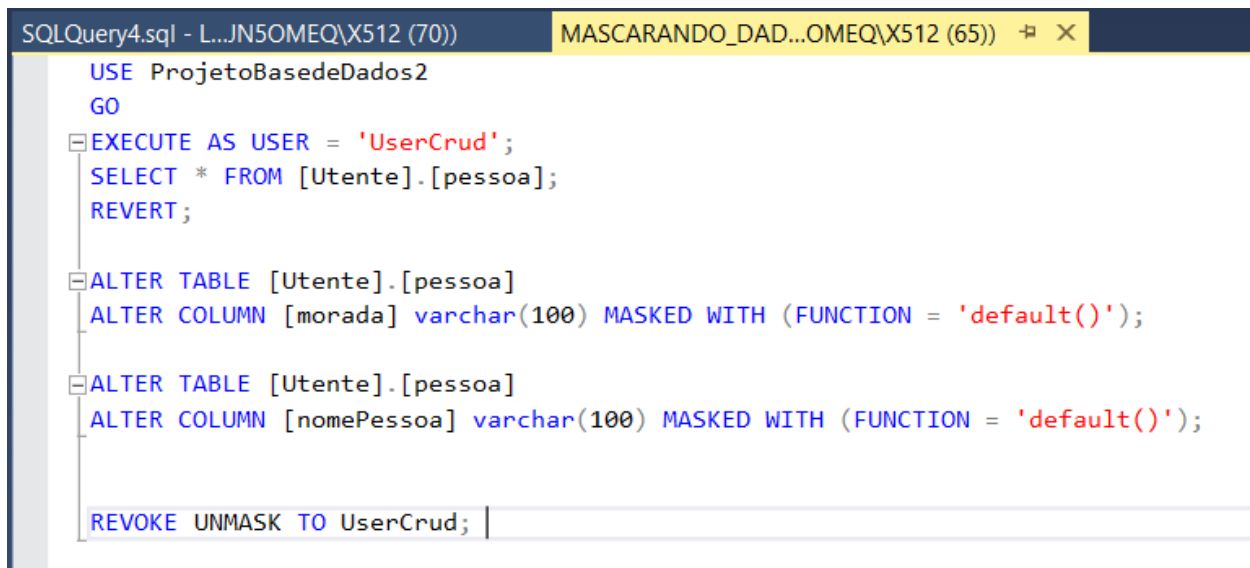
Mensagens

Msg 33277, Nível 16, Estado 6, Linha 4  
Encryption scheme mismatch for column/variables 'DOC'. The encryption scheme for the column/variables is (encryption\_type = 'PLAINTEXT') and the expression near line '2' expects it to be RANDOMIZED, or DETE...

Horário de conclusão: 2021-01-17T19:04:16.8663053+00:00

Figura 11- Erro de inserção

Diante deste impasse resolvemos usar a máscara para ocultar esses dados pessoais de nossos usuários para isso tivemos de dar permissão com o comando ALTER ANY MASK



```
USE ProjetoBaseDados2
GO
EXECUTE AS USER = 'UserCrud';
SELECT * FROM [Utente].[pessoa];
REVERT;

ALTER TABLE [Utente].[pessoa]
ALTER COLUMN [morada] varchar(100) MASKED WITH (FUNCTION = 'default()');

ALTER TABLE [Utente].[pessoa]
ALTER COLUMN [nomePessoa] varchar(100) MASKED WITH (FUNCTION = 'default()');

REVOKE UNMASK TO UserCrud;
```

Figura 12 - Código para mascarar a tabela

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Pesquisador de Objetos' (Object Explorer) pane displays the server hierarchy for 'LAPTOP-1JN5OMEQ (SQL Server 15.0.2080.9 - UserCrud)'. The central pane shows a query window with the SQL statement: `SELECT * FROM [Utente].[pessoa];`. Below the query window, the 'Resultados' (Results) pane displays a table with 7 columns: `idPessoa`, `nif`, `nomePessoa`, `genero`, `idade`, `morada`, and `idConcelho`. The data is presented in 17 rows, with the first row highlighted. All data values are encrypted, represented by 'xxxx' or 'xxxxx'.

	idPessoa	nif	nomePessoa	genero	idade	morada	idConcelho
1	1	2000000000	xxxx	Masculino	89	xxxx	117
2	2	2000000001	xxxx	Masculino	73	xxxx	177
3	3	2000000002	xxxx	Masculino	65	xxxx	171
4	4	2000000003	xxxx	Feminino	27	xxxx	293
5	5	2000000004	xxxx	Feminino	55	xxxx	285
6	6	2000000005	xxxx	Feminino	67	xxxx	160
7	7	2000000006	xxxx	Masculino	67	xxxx	294
8	8	2000000007	xxxx	Masculino	44	xxxx	187
9	9	2000000008	xxxx	Masculino	75	xxxx	161
10	10	2000000009	xxxx	Feminino	21	xxxx	264
11	11	2000000010	xxxx	Feminino	87	xxxx	213
12	12	2000000011	xxxx	Feminino	38	xxxx	233
13	13	2000000012	xxxx	Masculino	33	xxxx	207
14	14	2000000013	xxxx	Masculino	72	xxxx	219
15	15	2000000014	xxxx	Masculino	95	xxxx	2
16	16	2000000015	xxxx	Feminino	99	xxxx	302
17	17	2000000016	xxxx	Feminino	83	xxxx	118

Figura 13- Visualização de dados encriptados

## 5.5 AUDITORIA

A auditoria é um histórico de ações feitos na base de dados ou servidores que nos permite verificar qual comando ou função comprometeu o nosso servidor, e o onde houve fuga de informação.

Nós iremos guardar os ficheiros destas auditorias noutro disco à parte para não comprometer o espaço de outro disco.

Abaixo um print do T-sql que utilizamos para criar nossa auditoria:

```
01. USE master
02. GO
03. /***** Object:  Audit [Auditoria_server]    Script Date: 18/01/2021 22:56:02 *****/
04. CREATE SERVER AUDIT [Auditoria_server]
05. TO FILE
06. (
07.     FILEPATH = N'D:\Audit\
08.     ,MAXSIZE = 0 MB
09.     ,MAX_ROLLOVER_FILES = 2147483647
10.     ,RESERVE_DISK_SPACE = OFF
11. ) WITH (QUEUE_DELAY = 1000, ON_FAILURE = FAIL_OPERATION, AUDIT_GUID = 'f149f166-bde6-4694-a440-30f7cad2d787')
12. ALTER SERVER AUDIT [Auditoria_server] WITH (STATE = ON)
13. GO
14. USE [master]
15. GO
16. CREATE SERVER AUDIT SPECIFICATION [ServerAuditServer]
17. FOR SERVER AUDIT [Auditoria_server]
18. ADD (BACKUP_RESTORE_GROUP),
19. ADD (AUDIT_CHANGE_GROUP),
20. ADD (FAILED_LOGIN_GROUP),
21. ADD (SUCCESSFUL_LOGIN_GROUP),
22. ADD (APPLICATION_ROLE_CHANGE_PASSWORD_GROUP),
23. ADD (USER_CHANGE_PASSWORD_GROUP)
24. WITH (STATE = ON)
25. GO
26. USE [ProjetoBaseDados2]
27. GO
28.
29. CREATE DATABASE AUDIT SPECIFICATION [AuditDB]
30. FOR SERVER AUDIT [Auditoria_server]
31. ADD (EXECUTE ON DATABASE::[ProjetoBaseDados2] BY [UserCrud]),
32. ADD (DELETE ON DATABASE::[ProjetoBaseDados2] BY [UserCrud]),
33. ADD (INSERT ON DATABASE::[ProjetoBaseDados2] BY [UserCrud]),
34. ADD (SELECT ON DATABASE::[ProjetoBaseDados2] BY [UserCrud]),
35. ADD (EXECUTE ON DATABASE::[ProjetoBaseDados2] BY [UserJrDb]),
36. ADD (DELETE ON DATABASE::[ProjetoBaseDados2] BY [UserJrDb]),
37. ADD (INSERT ON DATABASE::[ProjetoBaseDados2] BY [UserJrDb]),
38. ADD (SELECT ON DATABASE::[ProjetoBaseDados2] BY [UserJrDb]),
39. ADD (SELECT ON DATABASE::[ProjetoBaseDados2] BY [UserRead]),
40. ADD (EXECUTE ON DATABASE::[ProjetoBaseDados2] BY [UserAdm]),
41. ADD (DELETE ON DATABASE::[ProjetoBaseDados2] BY [UserAdm]),
42. ADD (INSERT ON DATABASE::[ProjetoBaseDados2] BY [UserAdm]),
43. ADD (SELECT ON DATABASE::[ProjetoBaseDados2] BY [UserAdm])
```

Figura 14 - Código T-sql da auditoria

## 6 DESCRIÇÃO DAS CÓPIAS DE SEGURANÇA (BACKUPS)

---

Existem 3 modelos diferentes de recuperação do SQL Server, devemos selecionar o modelo de recuperação do SQL Server para gerir os ficheiros de log e nos preparar para uma recuperação do SQL em caso de desastre. Neste ponto iremos mostrar modelo que foi escolhido e o porque do mesmo ser escolhido.

Como foi mencionado existem 3 tipos de modelo de recuperação:

- Modelo de Recuperação *Simple*
- Modelo de Recuperação *Full*
- Modelo de Recuperação *Bulk-logged*

### **Modelo de Recuperação Simple:**

**Recuperação Simple** requer menos administração. É mais fácil de gerir do que os modelos *Full* ou *Bulk-Logged*, mas tem maior exposição à perda de dados se um ficheiro de dados estiver danificado. A Recuperação *Simple* não é a escolha mais apropriada para os sistemas de produção em que a perda de alterações recentes é inaceitável. Ao usar Recuperação *Simple*, o intervalo de backup deve ser longo o suficiente para impedir que a sobrecarga de backup afete o trabalho de produção, ainda que seja curto o suficiente para impedir a perda de quantidades significativas de dados.

**Vantagem:** permite operações de cópia em grande escala de alto desempenho. Recupera o espaço de log para manter os requisitos de espaço pequenos.

**Desvantagem:** alterações desde o banco de dados mais recente ou backup diferencial deverão ser refeitas.



### **Modelo de Recuperação full:**

A Recuperação *full* permite o *SQL Server* preservar o log de transações até que faça o *backup*. Isso permite que crie um plano de recuperação em caso de desastre, que inclua uma combinação de *backups* de base de dados completos e diferenciais em conjunto com *backups* de log de transação.

Este plano tem a maior flexibilidade para restaurar base de dados ao usar o modelo de recuperação *full* quando ocorre uma falha no banco de dados. Além de preservar as modificações de dados armazenadas no log de transações, o modelo de recuperação *full* permite restaurar uma base de dados para um ponto específico no tempo.

**Vantagem:** nenhum trabalho é perdido devido a um ficheiro de dados perdido ou danificado. Pode recuperar a um ponto arbitrário no tempo.

**Desvantagem:** se o log estiver danificado, alterações desde o backup de log mais recente deverão ser refeitas.

### **Modelo de Recuperação *bulk-logged*:**

O modelo de recuperação *bulk-logged* é um modelo que tem a finalidade especial, funciona de maneira semelhante ao modelo de recuperação *full*. A única diferença está na maneira como ele lida com as operações de modificação de dados em massa. O modelo *bulk-logged* regista essas operações no log de transações ao usar um registo técnico conhecido como registo mínimo. Isso economiza significativamente no tempo de processamento, mas impede que use a opção de restauração ponto-no-tempo.

**Vantagem:** permite operações de cópia em alta escala de alto desempenho, espaço mínimo de log é usado por operações em alta escala.

**Desvantagem:** se o log estiver danificado ou operações em alta escala ocorreram desde o backup de log mais recente, as alterações desde o último backup deverão ser refeitas.

Nós no nosso trabalho só podíamos usar 2 dos 3 disponíveis, o full ou bulk-logged, mas analisando melhor a nossa base de dados decidimos utilizar o full, pois este oferecia mais segurança de dados e como se trata de uma base de dados que a informação a cada segundo é importante.

No **anexo 7** está os códigos exemplos que nos permite fazer *Backups* e *Restores*.

## 7 DESCRIÇÃO DO DESEMPENHO

---

### 7.1 ÍNDICES CLUSTERIZADOS E NÃO CLUSTERIZADOS

Um índice é uma estrutura em disco associada a uma tabela ou exibição, que agiliza a recuperação das linhas de uma tabela ou exibição. Um índice contém chaves criadas de uma ou mais colunas da tabela ou exibição. Essas chaves são armazenadas em uma estrutura que habilita o SQL Server a localizar a linha ou as linhas associadas aos valores de chave de forma rápida e eficaz.

#### 7.1.1 Clusterizado

Os índices clusterizados classificam e armazenam as linhas de dados da tabela ou exibição com base em seus valores de chave. Essas são as colunas incluídas na definição do índice. Pode haver apenas um índice clusterizado por tabela, pois as linhas de dados podem ser classificadas somente em uma única ordem, por exemplo, algumas tabelas tem duas chaves primárias, conseqüentemente dois índices clusterizados.

O único momento em que as linhas de dados de uma tabela são armazenadas na ordem de classificação é quando a tabela contém um índice clusterizado. Se a tabela não possuir nenhum índice clusterizado, suas linhas de dados ficarão armazenadas em uma estrutura não ordenada denominada *heap*.

#### 7.1.2 Não clusterizado

Os índices não clusterizados têm uma maneira distinta de ser estruturados. Cada entrada de valor de uma chave tem um ponteiro para a linha de dados que contém o valor de chave.

O ponteiro de um índice, em um índice não clusterizado, é denominado de localizador de linhas. A estrutura do localizador de linhas depende das páginas de dados serem armazenadas em um

*heap* ou em uma tabela clusterizada. Para o *heap*, o localizador de linhas é um ponteiro para a linha. Para a tabela clusterizada, o localizador de linhas é a chave de índice clusterizado.

Podemos adicionar colunas que não são chave ao nível da folha do índice não clusterizado, para assim ignorar os limites de chave de um índice existente e executar consultas completamente abrangidas e indexadas.

## 7.2 FILEGROUPS

*FileGroups* são estruturas lógicas que sustentam os ficheiros de dados em uma base de dados. Um Base de dados padrão possui um ficheiro de dados e um ficheiro de log; o ficheiro de dados está associado a um *filegroup* chamado *PRIMARY*.

Além dos ficheiros .MDF e .LDF, é possível criar outros ficheiros para armazenamento de dados. Estes ficheiros secundários possuem extensão .NDF, de *Secondary Data Files* e podem ser criados no mesmo *filegroup* do ficheiro .MDF (*PRIMARY*) ou em outro *filegroup*. A decisão de utilizar o mesmo *filegroup* ou criar um novo depende da finalidade do ficheiro secundário.

Analisando a nossa base de dados construímos os nossos filegroups da seguinte maneira:

Filegroups PRIMARY: Onde armazenamos os ficheiros do sistema e dois índices clusterizamos distribuir os vários índices clusterizados das tabelas por *filegroups* e criar um *filegroup* para os índices não clusterizados, na **figura 15** mostra o planeamento da distribuição de tabelas pelos *filegroups* e também onde ficavam armazenados esses filegroups, distribuímos por vários discos porque assim é mais fácil o processador processar mais dados pois estão distribuídos por mais discos.

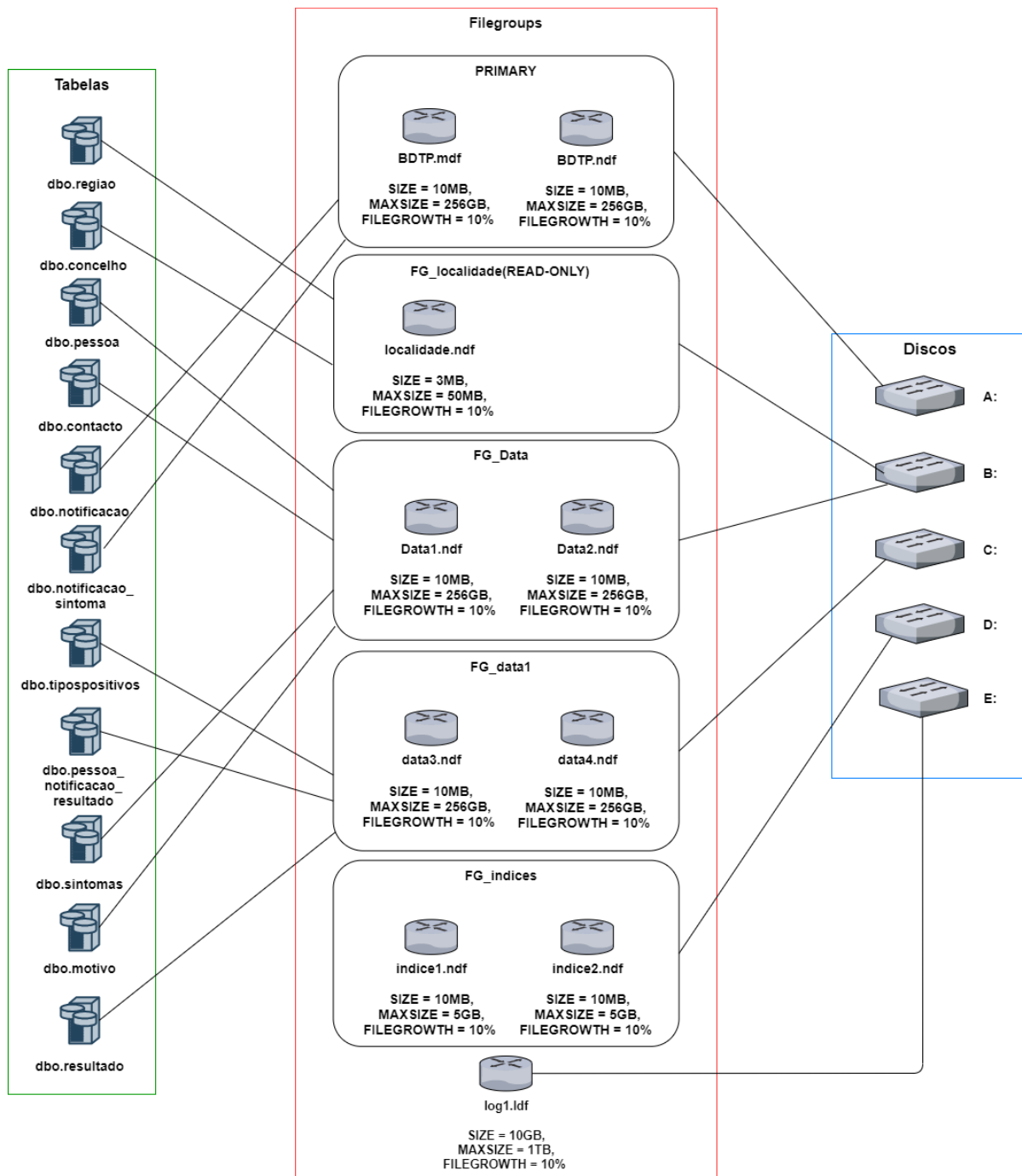


Figura 15 - Planejamento dos filegroups

## 7.2.1 Scripts do filegroup feitos no trabalho

Para implementar os *filegroups* na nossa base de dados começamos por criar os mesmos, mas como já tínhamos criado as chaves estrangeiras na etapa de inserção de dados tivemos de as retirar para mover as chaves clusterizadas para o seu *filegroup* pretendido, depois de inseridas voltamos a colocar as chaves estrangeiras.

Os códigos usados para a execução das tarefas acima referidas encontram-se no **anexo 8**

Os resultados obtidos foram os seguintes:

	name	type	name	index_id	name
1	notificacao	U	PK_Notificacao	1	PRIMARY
2	notificacaoSintomas	U	PK_notificacaoSintomas	1	PRIMARY
3	sysdiagrams	U	PK_sysdiagr_C2B05B61A14EEFB6	1	PRIMARY
4	sysdiagrams	U	UK_principal_name	2	PRIMARY
5	contacto	U	PK_contacto	1	FG_data
6	peessoa	U	PK_peessoa	1	FG_data
7	motivo	U	PK_motivo	1	FG_data
8	sintomas	U	PK_sintomas	1	FG_data
9	resultado	U	PK_resultado	1	FG_data1
10	tiposPositivo	U	PK_tiposPositivo	1	FG_data1
11	peessoaNotificacaoResultado	U	PK_peessoaNotificacaoResultado	1	FG_data1
12	notificacao	U	Notificacao_idmotivo_index	3	FG_Indices
13	resultado	U	resultado_index	2	FG_Indices
14	concelho	U	PK_concelho	1	FG_locali...
15	regiao	U	PK_regiao	1	FG_locali...

Figura 16 - select que permite ver a chave e a que filegroup esta inserida.

	Name_of_Object	Index_Name	Index_Type	Name_of_Filegroup	Object_Type	type	type_desc
1	contacto	PK_contacto	CLUSTERED	FG_data	U	FG	ROWS_FILEGROUP
2	peessoa	PK_peessoa	CLUSTERED	FG_data	U	FG	ROWS_FILEGROUP
3	notificacao	PK_Notificacao	CLUSTERED	PRIMARY	U	FG	ROWS_FILEGROUP
4	notificacao	Notificacao_idmotivo_index	NONCLUSTERED	FG_Indices	U	FG	ROWS_FILEGROUP
5	notificacaoSintomas	PK_notificacaoSintomas	CLUSTERED	PRIMARY	U	FG	ROWS_FILEGROUP
6	concelho	PK_concelho	CLUSTERED	FG_localidade	U	FG	ROWS_FILEGROUP
7	resultado	PK_resultado	CLUSTERED	FG_data1	U	FG	ROWS_FILEGROUP
8	resultado	resultado_index	NONCLUSTERED	FG_Indices	U	FG	ROWS_FILEGROUP
9	tiposPositivo	PK_tiposPositivo	CLUSTERED	FG_data1	U	FG	ROWS_FILEGROUP
10	sysdiagrams	PK_sysdiagr_C2B05B61A14EEFB6	CLUSTERED	PRIMARY	U	FG	ROWS_FILEGROUP
11	sysdiagrams	UK_principal_name	NONCLUSTERED	PRIMARY	U	FG	ROWS_FILEGROUP
12	regiao	PK_regiao	CLUSTERED	FG_localidade	U	FG	ROWS_FILEGROUP
13	peessoaNotificacao...	PK_peessoaNotificacaoResultado	CLUSTERED	FG_data1	U	FG	ROWS_FILEGROUP
14	motivo	PK_motivo	CLUSTERED	FG_data	U	FG	ROWS_FILEGROUP
15	sintomas	PK_sintomas	CLUSTERED	FG_data	U	FG	ROWS_FILEGROUP

Figura 17 - Select mais detalhado sobre os filegroups

## 8 DESCRIÇÃO DA MANUTENÇÃO E AUTOMATIZAÇÃO DO SERVIDOR

---

Neste ponto vamos fazer a manutenção e a automação do servidor, que basicamente nos permite agendar tarefas sem o administrador da base de dados ou outra pessoa tenha de executar na hora.

Passando algum tempo a tentar encontrar uma solução para resolver quando devia ser feito os backups decidimos que seria necessário implementarmos um plano de manutenção verificando assim a saúde de nossa base dados. Essa é uma tarefa muito comum para todos os DBAs e não é a mesma para todos, porém é diferente para cada servidor. Como muitas outras coisas, isso depende do que você precisa.

Com isso em mente desenvolvemos o nosso plano de manutenção da seguinte maneira (utilizamos a ferramenta assistente de plano de manutenção, e infelizmente perdemos a exportação que fizemos para o T-SQL), levando em consideração que esse plano servirá para qualquer base de dados nova inserida:

**Verificação da Integridade do Banco de Dados** – Ocorre todo mês no dia 1 a partir das 03:00, garante que qualquer problema de integridade seja reportado.

**Backup Full** - Ocorre todo mês no dia 1 as 04:00, garante o resguardo dos dados se por acaso houver falha no sistema ou hardware.

**Backup Diferencial** – Ocorre diariamente a cada 3 horas,

**Backup Log** – Ocorre diariamente a cada 5 minutos, decidimos esse período pois a perda de dados em nossa situação poderia ter implicações legais.

**Task Limpeza de Manutenção** – Ocorre diariamente as 02:00, tal tarefa deleta os arquivos de backups diretamente do diretório selecionado onde é salvo o arquivo, levando em consideração o período estabelecido, que no nosso caso foram de 7 dias (essas informações foram configuradas quando criamos o plano de manutenção através do assistente de plano de manutenção).

**\*Utilizamos o horário da madrugada pois supostamente é o horário que teria menos pessoas usando o servidor.**

Todos backups são salvos em subpastas separadas pelo nome da base de dados:

Nome	Data de modificação	Tipo	Tam
AdventureWorks2019	27/01/2021 18:00	Pasta de ficheiros	
DWConfiguration	27/01/2021 19:10	Pasta de ficheiros	
DWDiagnostics	27/01/2021 18:00	Pasta de ficheiros	
DWQueue	27/01/2021 18:00	Pasta de ficheiros	
ProjetoBaseDados2	27/01/2021 19:10	Pasta de ficheiros	
ReportServer	27/01/2021 19:10	Pasta de ficheiros	
ReportServerTempDB	27/01/2021 18:00	Pasta de ficheiros	
Teste	26/01/2021 12:52	Pasta de ficheiros	

Figura 18 - Diretoria dos backups

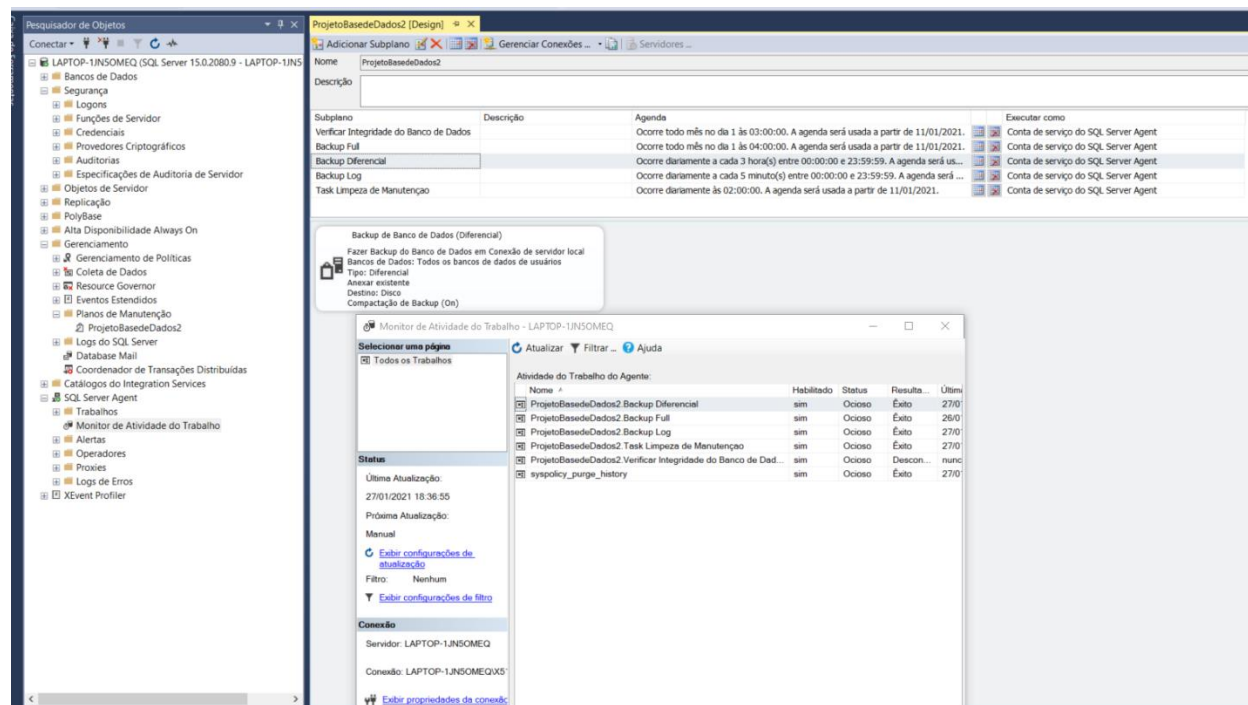


Figura 19 - Plano de manutenção construído



## 9 DESCRIÇÃO DE OUTRAS FUNCIONALIDADES

### 9.1 CRIAÇÃO DE UM DATABASE MAIL

Neste ponto decidimos criar um e-mail próprio que pudesse enviar um e-mail a um dos criadores a avisar que foi atingido 1,10,50 ou 75% da população infetada, utilizando o *stored procedure* mencionado no tópico *Stored Procedures*, assim permite destacar mais esse ponto de notificação.

#### 9.1.1 Criação do DataMail:

**Passo 1:** Ir ao *database mail* e carregar como lado direito do rato e escolher a opção a negrito.

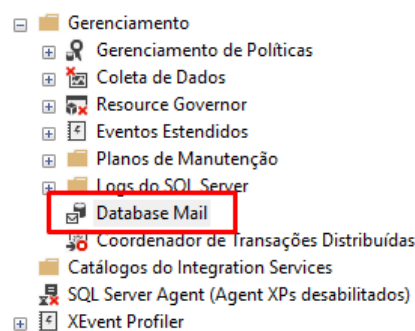


Figura 20 - Aceder a função Database Mail

**Passo2:** Escolher a primeira opção e carregar no botão avançar.

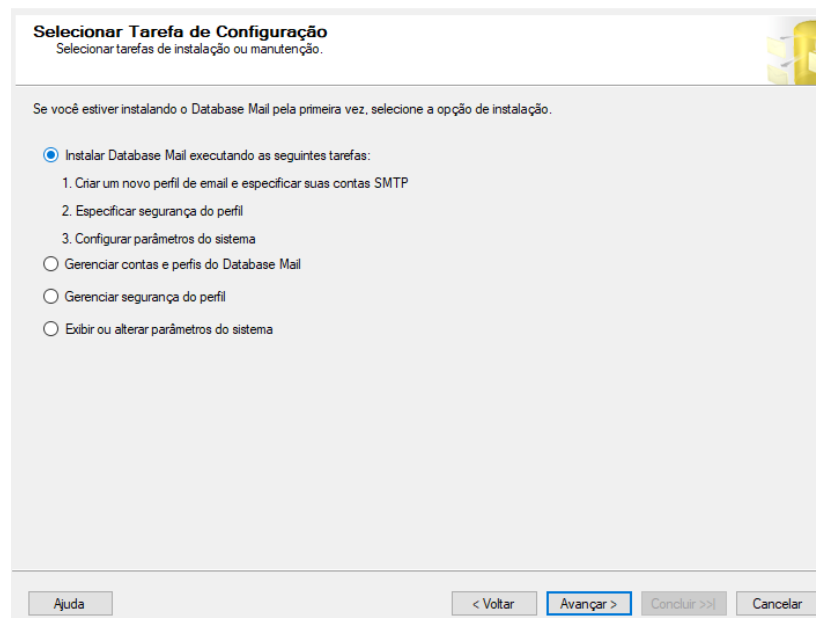


Figura 21 - Wizard da criação do database mail

**Passo3:** Dar o nome ao perfil e carregar em adicionar

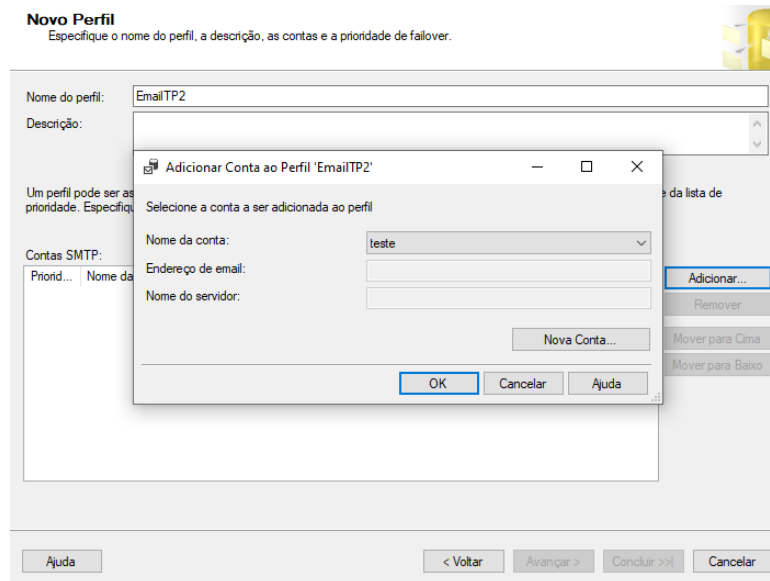


Figura 22 - Criação do perfil

**Passo4:** Carregar em Nova conta e inserir o nome da conta e inserir os campos como estão na imagem

Servidor de saída de emails (SMTP)

Endereço de email:	trabalhotp2bd2@gmail.com		
Nome para exibição:	From MasterDB Avisos		
Email de resposta:			
Nome do servidor:	smtp.gmail.com	Número da porta:	587

☒ Este servidor requer uma conexão segura (SSL)

Autenticação SMTP

☐ Autenticação do Windows usando credenciais do serviço Mecanismo de Banco de Dados

☒ Autenticação básica

Nome de usuário:	trabalhotp2bd2@gmail.com		
Senha:	*****		
Confirmar senha:	*****		

☐ Autenticação anônima

Figura 23 - Configuração do mail

Criamos um mail próprio que é o `trabalhotp2bd2@gmail.com`

**Passo5:** Basta carregar em avançar e criar o *datamail*.

## 9.2 REPORTBUILDER

Para atender uma das solicitações sobre a funcionalidade do nosso sistema de base de dados, criamos dois Relatórios/Informativos diários sobre a situação do COVID-19 no país e seus concelhos. Tais informativos poderão ser emitidos pelos usuários a partir da ferramenta Report Builder. Baixamos o software e fizemos a ligação ao sql, tornando a emissão do informativo fiel e atualizada aos dados que estão sql server. Fizemos dois Informativos um no âmbito do concelho e outro em âmbito nacional onde no último incluímos um gráfico.

### 9.2.1 Instalação e configuração do software

Como trata-se de um assunto que não foi abordado em sala de aula, buscamos ajuda na internet e em vídeos no Youtube para fazer a conexão com o SQL e para fazermos os designers dos relatórios.

**Passo 1:** Começamos por baixar e logo em seguida fazendo a ligação ao Banco de Dados:

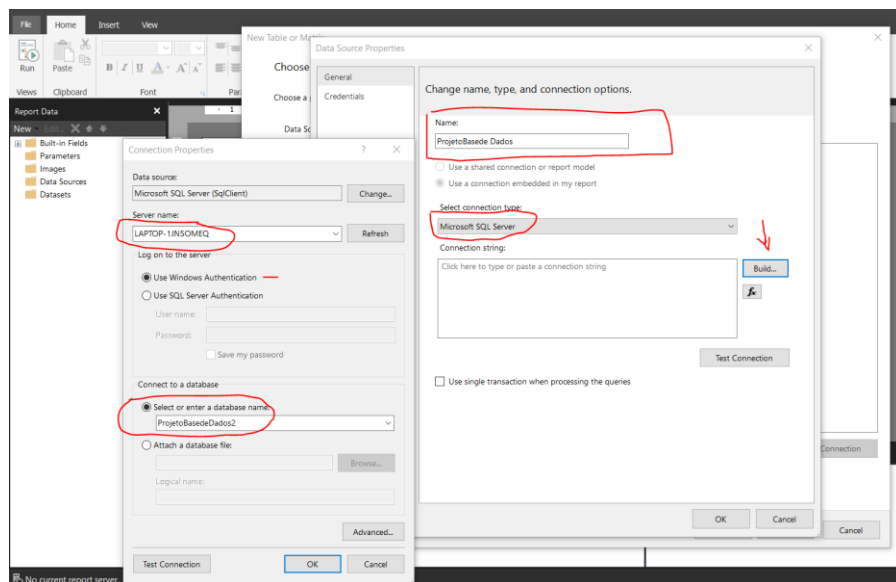


Figura 24 - Conexão com a base de dados

**Passo 2:** A seguir clicar em Next, depois em Edit as Text, colar os comandos do sql, nesse caso é um stored procedure, onde filtra por género os casos positivos:

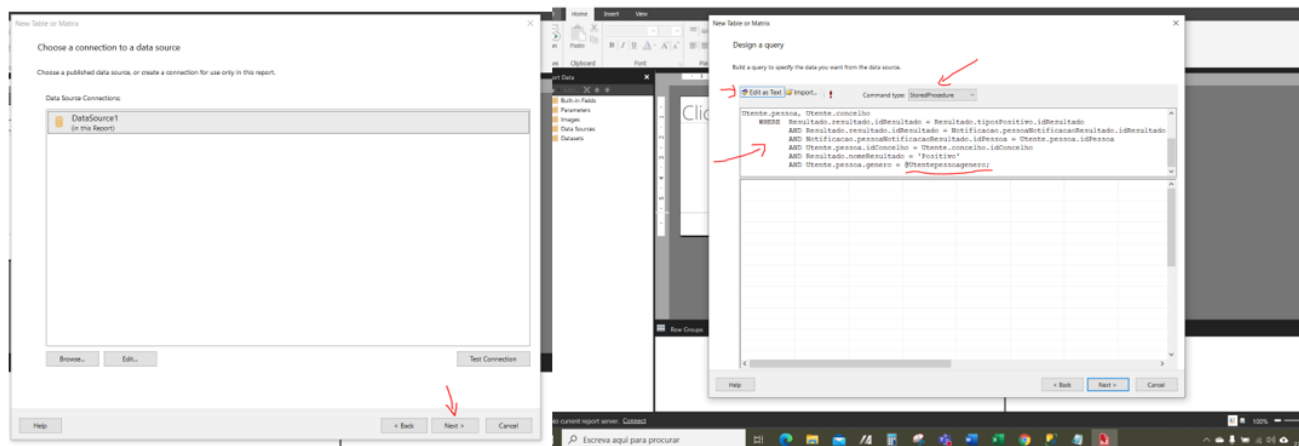


Figura 25 - Inclusão dos comandos

**Passo 3:** Depois clicar em Next e Finish e seguir criar a uma tabela para incluir os dados, e é nessa hora que pode-se incluir um designer apropriado para o informativo:

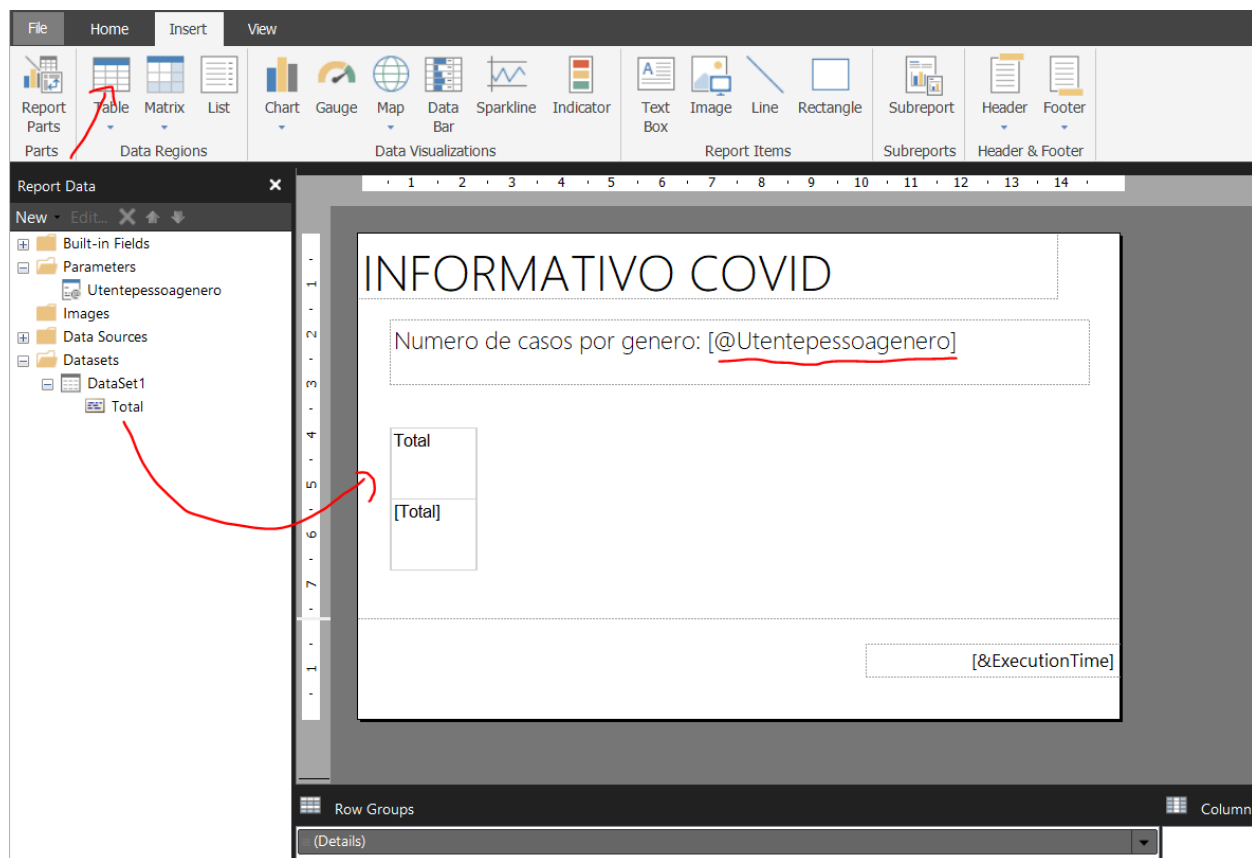


Figura 26 - Inclusão de tabela

**Passo 4:** Agora é correr e verificar como ficou o relatório, nesse momento você pode escolher qual o género deseja saber:

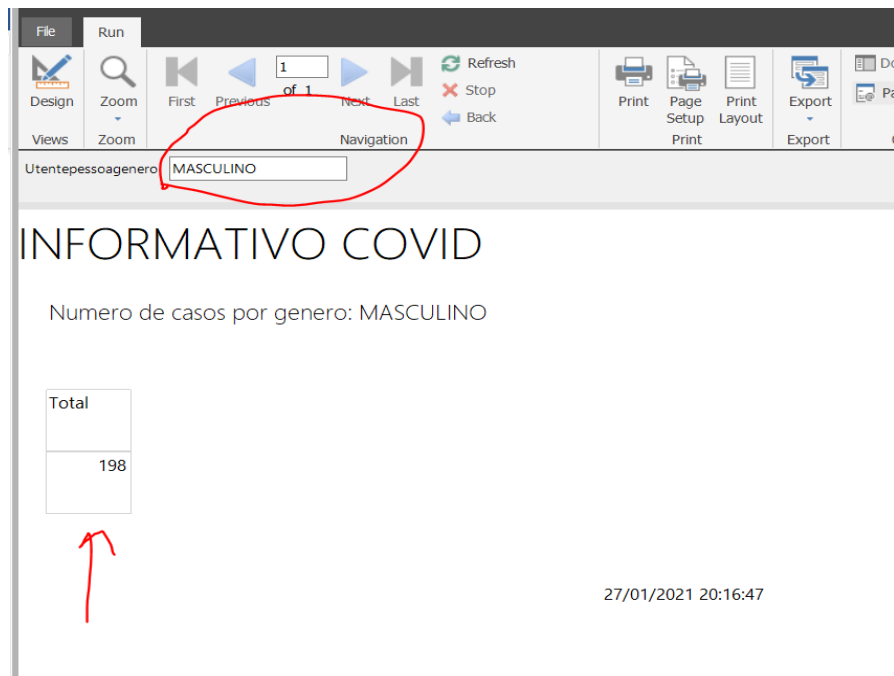


Figura 27 - Relatório pronto

**Passo 5:** Se quiser incluir um gráfico é só fazer o passo 3 e selecionar a planilha quer usar e a seguir escolher e configurar o gráfico quer:

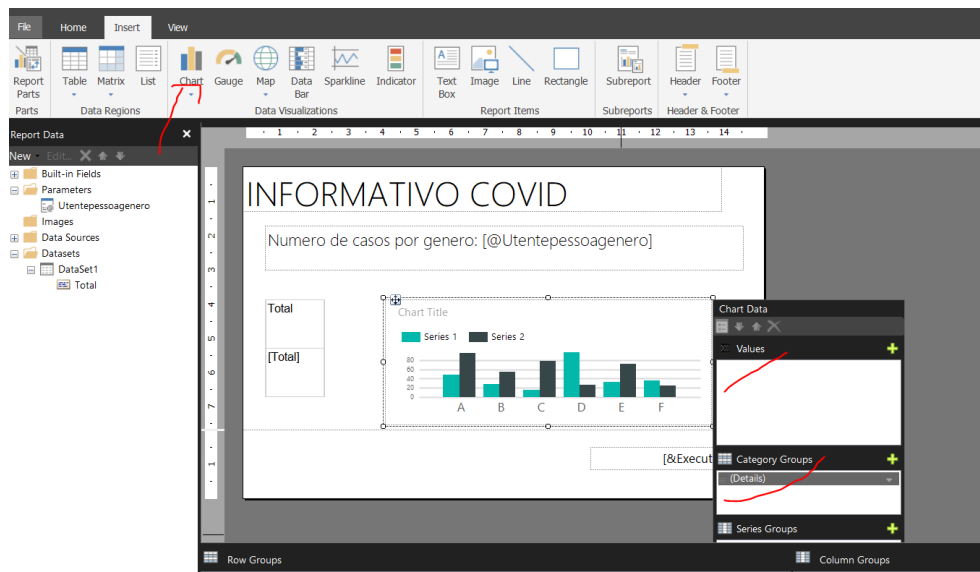


Figura 28 - Inclusão de gráfico

Os nossos informativos constam no **anexo 9**, como também iremos incluir os dois projetos feitos no ReportBuilder junto com a documentação desse trabalho. Abaixo um print:

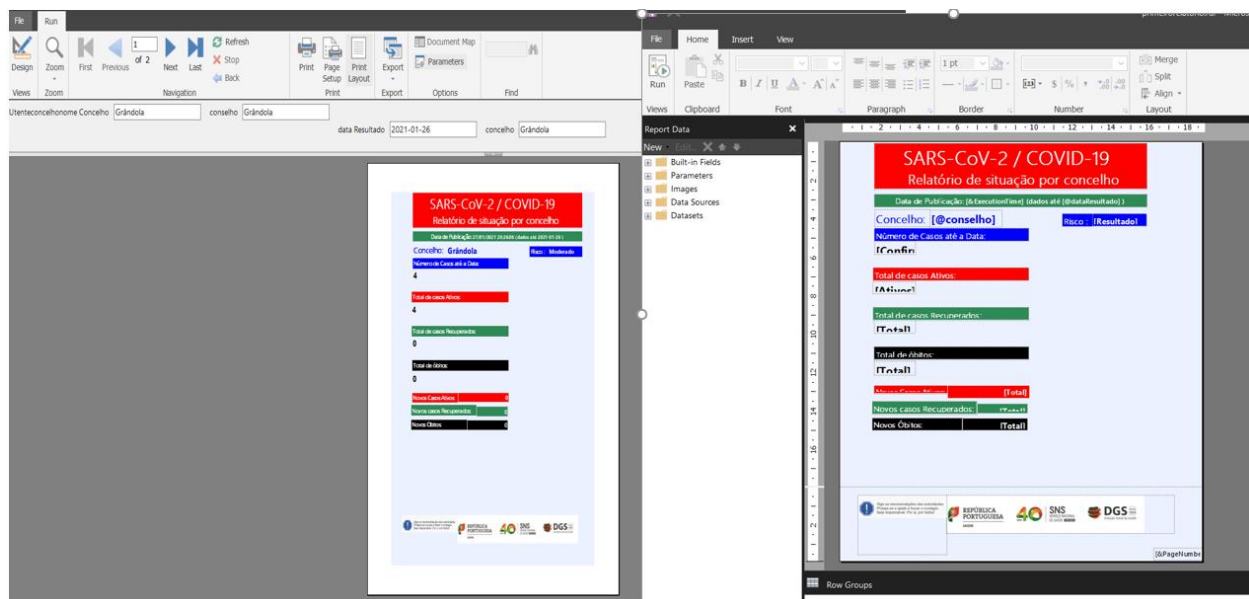


Figura 29 - Relatório por concelho

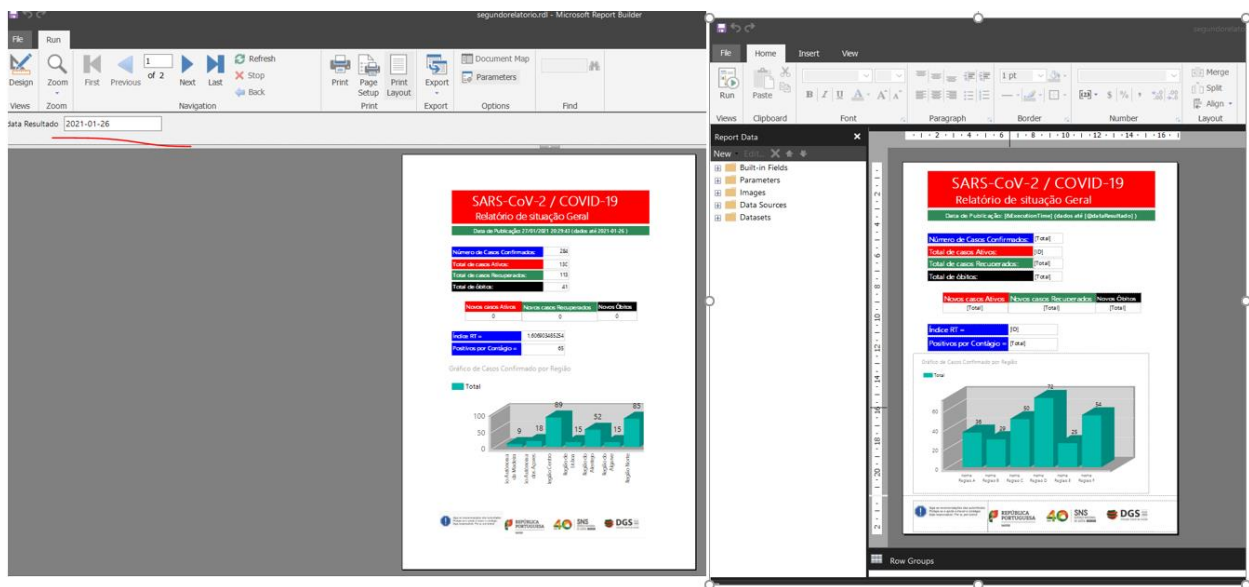


Figura 30 - Relatório Geral

## 10 CONCLUSÕES

---

### 10.1 SUMÁRIO DO QUE FOI ATINGIDO

Levando em consideração o conhecimento adquirido através de sala de aula, material disponibilizado no moodle, e pesquisas feitas na internet, foi possível realizar o trabalho proposto. Conseguimos atingir todos os objetivos propostos no enunciado, como também fizemos o planejamento do trabalho utilizando os tópicos mencionados no enunciado.

Este trabalho permitiu ter a sensação real de como seria criar um sistema de base de dados do zero, nos colocando no lugar de um DBA tomando as decisões que caberiam a esse profissional de tecnologia. Encaramos os pontos que tínhamos pouco conhecimentos como desafios e nos aprofundamos nas pesquisas, sendo muito válido, principalmente para as próximas etapas da cadeira em questão, no caso a realização de exames.

### 10.2 SUMÁRIO DO QUE NÃO FOI ATINGIDO

Alguns pontos queríamos realmente ter feito, para assim termos a experiencia completa, como por exemplo o desenvolvimento de um Front-end para a manipulação da base de dados com características CRUD (Creat, Read, Update, Delete) e emissão de relatórios, todos conectados e alimentados pela nossa BD. Outro ponto que infelizmente não conseguimos atingir foi a criação de um *dashboard* em tempo real. Não tivemos tempo para tal e acreditamos que um dos motivos foi a pausa forçada de 5 dias que tivemos que fazer, pois um dos alunos (por problemas técnicos), ficou sem computador. E para finalizar, foi frustrante não conseguirmos resolver o problema que tivemos com a Criptografia, pois acreditamos que com um pouco mais de pesquisas conseguiríamos sim criptografar e utilizar os dados criptografados.

### 10.3 TRABALHO FUTURO

Conseguimos perceber durante o processo de desenvolvimento de nossa BD, que seria mais eficiente começarmos a criar filegroups antes da criação de tabelas e etc. pois tivemos retrabalho ao inserir os nossos filegroups depois de ter incluído as tabelas e suas características.



Sem dúvida faríamos diferente da forma que fizemos anteriormente, pois no lugar de agrupar alguns filegrupos tínhamos feito uma filegroup para cada índice, para que com isso o desempenho da base de dados aumentasse ainda mais. Daí a importância de um filegroups bem estruturado.

Com certeza todo o conteúdo assimilado será fundamental para nosso desenvolvimento profissional e temos a certeza de que todo esforço dedicado neste trabalho de facto será de grande ajuda para futuros projetos, inclusive nesta área de administração de base dados, como também em outras áreas envolvidas com esta.

## 11 BIBLIOGRAFIA

---

Modelos de recuperação (SQL Server) - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/recovery-models-sql-server?view=sql-server-ver15>

Return Data from a Stored Procedure - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/return-data-from-a-stored-procedure?redirectedfrom=MSDN&view=sql-server-ver15>

EXECUTE AS Clause (Transact-SQL) - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/en-us/sql/t-sql/statements/execute-as-clause-transact-sql?redirectedfrom=MSDN&view=sql-server-ver15>

(6) How to create a backup Maintenance Plan in SQL Server - YouTube. (n.d.). Retrieved January 27, 2021, from [https://www.youtube.com/watch?v=oLnZttZWWtk&ab\\_channel=VoluntaryDBA](https://www.youtube.com/watch?v=oLnZttZWWtk&ab_channel=VoluntaryDBA)

Usar o Assistente de Plano de Manutenção | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from [https://docs.microsoft.com/pt-br/previous-versions/sql/sql-server-2012/ms191002\(v=sql.110\)?redirectedfrom=MSDN](https://docs.microsoft.com/pt-br/previous-versions/sql/sql-server-2012/ms191002(v=sql.110)?redirectedfrom=MSDN)

Modelos de recuperação no SQL Server: simple, full e bulk-logged - EaseUS. (n.d.). Retrieved January 27, 2021, from <https://br.easeus.com/backup-recovery/tres-modelos-de-recuperacao-do-sql-server.html>

Aplicar backups do log de transações (SQL Server) - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/apply-transaction-log-backups-sql-server?redirectedfrom=MSDN&view=sql-server-ver15>

Fazer backup de um log de transações - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/back-up-a-transaction-log-sql-server?redirectedfrom=MSDN&view=sql-server-ver15>

BACKUP (Transact-SQL) - SQL Server | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from <https://docs.microsoft.com/pt-br/sql/t-sql/statements/backup-transact-sql?redirectedfrom=MSDN&view=sql-server-ver15>

Using Files and Filegroups | Microsoft Docs. (n.d.). Retrieved January 27, 2021, from [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms187087\(v=sql.105\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms187087(v=sql.105)?redirectedfrom=MSDN)

## 12 ANEXOS

### 12.1 ANEXO 1 – CRIAÇÃO DAS TABELAS

```
1. --Tabela regiao
2. CREATE TABLE [regiao](
3.     [idRegiao] [smallint] NOT NULL,
4.     [nomeRegiao] [varchar](50) NOT NULL,
5.     [quantHabitantes] [int] NOT NULL,
6.     CONSTRAINT [PK_regiao] PRIMARY KEY CLUSTERED
7. (
8.     [idRegiao] ASC
9. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_localidade]
10. ) ON [FG_localidade]
11. GO
12.
13. --Tabela pessoa
14. CREATE TABLE [pessoa](
15.     [idPessoa] [int] NOT NULL,
16.     [nif] [int] NOT NULL,
17.     [nomePessoa] [varchar](100) NULL,
18.     [genero] [varchar](10) NULL,
19.     [idade] [int] NULL,
20.     [morada] [varchar](100) NULL,
21.     [idConcelho] [smallint] NOT NULL,
22.     CONSTRAINT [PK_pessoa] PRIMARY KEY CLUSTERED
23. (
24.     [idPessoa] ASC
25. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data]
26. ) ON [FG_data]
27. GO
28.
29. --Tabela concelho
30.
31. CREATE TABLE [concelho](
32.     [idConcelho] [smallint] NOT NULL,
33.     [nomeConcelho] [varchar](50) NULL,
34.     [quantHabitantes] [int] NULL,
35.     [idRegiao] [smallint] NOT NULL,
36.     CONSTRAINT [PK_concelho] PRIMARY KEY CLUSTERED
37. (
38.     [idConcelho] ASC
39. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_localidade]
40. ) ON [FG_localidade]
41. GO
42.
43. --Tabela tiposPositivo
44. CREATE TABLE [tiposPositivo](
45.     [idTiposPositivo] [int] NOT NULL,
46.     [nomeTiposPositivo] [varchar](50) NULL,
```

```

47.     [idResultado] [int] NULL,
48.     [dataTipos] [date] NULL,
49.     CONSTRAINT [PK_tiposPositivo] PRIMARY KEY CLUSTERED
50. (
51.     [idTiposPositivo] ASC
52. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data1]
53. ) ON [FG_data1]
54. GO
55.
56. --Tabela resultado
57. CREATE TABLE [resultado](
58.     [idResultado] [int] NOT NULL,
59.     [nomeResultado] [varchar](50) NULL,
60.     [dataResultado] [date] NOT NULL,
61.     CONSTRAINT [PK_resultado] PRIMARY KEY CLUSTERED
62. (
63.     [idResultado] ASC
64. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data1]
65. ) ON [FG_data1]
66. GO
67.
68. --Tabela sintomas
69. CREATE TABLE [sintomas](
70.     [idSintomas] [int] NOT NULL,
71.     [nomeSintomas] [varchar](50) NULL,
72.     CONSTRAINT [PK_sintomas] PRIMARY KEY CLUSTERED
73. (
74.     [idSintomas] ASC
75. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data]
76. ) ON [FG_data]
77. GO
78.
79. --Tabela Pessoa Notificacao Resultdo
80. CREATE TABLE [pessoaNotificacaoResultado](
81.     [idPessoa] [int] NOT NULL,
82.     [idNotificacao] [int] NOT NULL,
83.     [idResultado] [int] NOT NULL,
84.     CONSTRAINT [PK_pessoaNotificacaoResultado] PRIMARY KEY CLUSTERED
85. (
86.     [idPessoa] ASC,
87.     [idNotificacao] ASC,
88.     [idResultado] ASC
89. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_L
    OCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data1]
90. ) ON [FG_data1]
91.
92. --Tabela Notificacao Sintomas
93. CREATE TABLE [notificacaoSintomas](
94.     [idNotificacaoSintomas] [int] NOT NULL,
95.     [idNotificacao] [int] NOT NULL,
96.     [idSintomas] [int] NOT NULL,
97.     CONSTRAINT [PK_notificacaoSintomas] PRIMARY KEY CLUSTERED
98. (
99.     [idNotificacaoSintomas] ASC
100. )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLO
    W_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY
    ]

```

```

101.     ) ON [PRIMARY]
102.     GO
103.
104.     ---Tabela Notificacao
105.     CREATE TABLE [notificacao](
106.         [idNotificacao] [int] NOT NULL,
107.         [dataNotificacao] [date] NULL,
108.         [idMotivo] [smallint] NOT NULL,
109.         CONSTRAINT [PK_Notificacao] PRIMARY KEY CLUSTERED
110.     (
111.         [idNotificacao] ASC
112.     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
113.     ) ON [PRIMARY]
114.     GO
115.
116.     --Tabela Motivo
117.     CREATE TABLE [motivo](
118.         [idMotivo] [smallint] NOT NULL,
119.         [nomeMotivo] [varchar](15) NOT NULL,
120.         CONSTRAINT [PK_motivo] PRIMARY KEY CLUSTERED
121.     (
122.         [idMotivo] ASC
123.     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data]
124.     ) ON [FG_data]
125.
126.     --Tabela contacto
127.     CREATE TABLE [contacto](
128.         [idPessoa] [int] NOT NULL,
129.         [idContacto] [int] NOT NULL,
130.         CONSTRAINT [PK_contacto] PRIMARY KEY CLUSTERED
131.     (
132.         [idPessoa] ASC,
133.         [idContacto] ASC
134.     )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [FG_data]
135.     ) ON [FG_data]
136.     GO
137.
138.     GO

```

## 12.2 ANEXO 2 – CÓDIGO PYTHON PARA INSERIR DADOS

```

1. # cria um script sql que vai conter todos os concelhos de Portugal e o id que pertence
   da região
2. if __name__ == '__main__':
3.
4.     concelhos = ['Arcos de Valdevez'
5.                 , 'Caminha'
6.                 , 'Melgaço'

```

7.	,	'Monção'
8.	,	'Paredes de Coura'
9.	,	'Ponte da Barca'
10.	,	'Ponte de Lima'
11.	,	'Valença'
12.	,	'Viana do Castelo'
13.	,	'Vila Nova de Cerveira'
14.	,	'Amares'
15.	,	'Barcelos'
16.	,	'Braga'
17.	,	'Esposende'
18.	,	'Terras de Bouro'
19.	,	'Vila Verde'
20.	,	'Cabeceiras de Basto'
21.	,	'Fafe'
22.	,	'Guimarães'
23.	,	'Mondim de Basto[2]'
24.	,	'Póvoa de Lanhoso'
25.	,	'Vieira do Minho'
26.	,	'Vila Nova de Famalicão'
27.	,	'Vizela'
28.	,	'Arouca'
29.	,	'Espinho'
30.	,	'Gondomar'
31.	,	'Maia'
32.	,	'Matosinhos'
33.	,	'Oliveira de Azeméis'
34.	,	'Paredes'
35.	,	'Porto'
36.	,	'Póvoa de Varzim'
37.	,	'Santa Maria da Feira'
38.	,	'Santo Tirso[2]'
39.	,	'São João da Madeira'
40.	,	'Trofa[2]'
41.	,	'Vale de Cambra'
42.	,	'Valongo'
43.	,	'Vila do Conde'
44.	,	'Vila Nova de Gaia'
45.	,	'Amarante'
46.	,	'Baião'
47.	,	'Castelo de Paiva'
48.	,	'Celorico de Basto'
49.	,	'Cinfães'
50.	,	'Felgueiras'
51.	,	'Lousada'
52.	,	'Marco de Canaveses'
53.	,	'Paços de Ferreira'
54.	,	'Penafiel'
55.	,	'Resende'
56.	,	'Boticas'
57.	,	'Chaves'
58.	,	'Montalegre'
59.	,	'Ribeira de Pena'
60.	,	'Valpaços'
61.	,	'Vila Pouca de Aguiar'
62.	,	'Alijó'
63.	,	'Armamar'
64.	,	'Carraceda de Ansiães'
65.	,	'Freixo de Espada à Cinta'
66.	,	'Lamego'

67.	,	'Mesão Frio'
68.	,	'Moimenta da Beira'
69.	,	'Murça'
70.	,	'Penedono'
71.	,	'Peso da Régua'
72.	,	'Sabrosa'
73.	,	'Santa Marta de Penaguião'
74.	,	'São João da Pesqueira'
75.	,	'Sernancelhe'
76.	,	'Tabuaço'
77.	,	'Tarouca'
78.	,	'Torre de Moncorvo'
79.	,	'Vila Nova de Foz Côa'
80.	,	'Vila Real'
81.	,	'Alfândega da Fé'
82.	,	'Bragança'
83.	,	'Macedo de Cavaleiros'
84.	,	'Miranda do Douro'
85.	,	'Mirandela'
86.	,	'Mogadouro'
87.	,	'Vila Flor[2]'
88.	,	'Vimioso'
89.	,	'Vinhais'
90.	,	'Águeda'
91.	,	'Albergaria-a-Velha'
92.	,	'Anadia'
93.	,	'Aveiro'
94.	,	'Estarreja'
95.	,	'Ílhavo'
96.	,	'Murtosa'
97.	,	'Oliveira do Bairro'
98.	,	'Ovar'
99.	,	'Sever do Vouga'
100.	,	'Vagos'
101.	,	'Arganil'
102.	,	'Cantanhede'
103.	,	'Coimbra'
104.	,	'Condeixa-a-Nova'
105.	,	'Figueira da Foz'
106.	,	'Góis'
107.	,	'Lousã'
108.	,	'Mealhada[2]'
109.	,	'Mira'
110.	,	'Miranda do Corvo'
111.	,	'Montemor-o-Velho'
112.	,	'Mortágua[2]'
113.	,	'Oliveira do Hospital'
114.	,	'Pampilhosa da Serra'
115.	,	'Penacova'
116.	,	'Penela'
117.	,	'Soure'
118.	,	'Tábua'
119.	,	'Vila Nova de Poiares'
120.	,	'Batalha'
121.	,	'Leiria'
122.	,	'Marinha Grande'
123.	,	'Pombal'
124.	,	'Porto de Mós'
125.	,	'Alvaiázere'
126.	,	'Ansião'

127.	,	'Castanheira de Pera'
128.	,	'Figueiró dos Vinhos'
129.	,	'Pedrógão Grande'
130.	,	'Aguiar da Beira'
131.	,	'Carregal do Sal'
132.	,	'Castro Daire'
133.	,	'Mangualde'
134.	,	'Nelas'
135.	,	'Oliveira de Frades'
136.	,	'Penalva do Castelo'
137.	,	'Santa Comba Dão'
138.	,	'São Pedro do Sul'
139.	,	'Sátão'
140.	,	'Tondela'
141.	,	'Vila Nova de Paiva'
142.	,	'Viseu'
143.	,	'Vouzela'
144.	,	'Almeida'
145.	,	'Belmonte'
146.	,	'Celorico da Beira'
147.	,	'Covilhã'
148.	,	'Figueira de Castelo Rodrigo'
149.	,	'Fornos de Algodres'
150.	,	'Fundão'
151.	,	'Gouveia'
152.	,	'Guarda'
153.	,	'Manteigas'
154.	,	'Mêda'
155.	,	'Pinhel'
156.	,	'Sabugal'
157.	,	'Seia'
158.	,	'Trancoso'
159.	,	'Castelo Branco'
160.	,	'Idanha-a-Nova'
161.	,	'Oleiros'
162.	,	'Penamacor'
163.	,	'Proença-a-Nova'
164.	,	'Vila Velha de Ródão'
165.	,	'Alcobaça'
166.	,	'Alenquer'
167.	,	'Arruda dos Vinhos'
168.	,	'Bombarral'
169.	,	'Cadaval'
170.	,	'Caldas da Rainha'
171.	,	'Lourinhã'
172.	,	'Nazaré'
173.	,	'Óbidos'
174.	,	'Peniche'
175.	,	'Sobral de Monte Agraço'
176.	,	'Torres Vedras'
177.	,	'Abrantes'
178.	,	'Alcanena'
179.	,	'Constância'
180.	,	'Entroncamento'
181.	,	'Ferreira do Zêzere'
182.	,	'Mação [5]'
183.	,	'Ourém'
184.	,	'Sardoal'
185.	,	'Sertão'
186.	,	'Tomar'



187.	,	'Torres Novas'
188.	,	'Vila de Rei'
189.	,	'Vila Nova da Barquinha'
190.	,	'Alcochete'
191.	,	'Almada'
192.	,	'Amadora'
193.	,	'Barreiro'
194.	,	'Cascais'
195.	,	'Lisboa'
196.	,	'Loures'
197.	,	'Mafra'
198.	,	'Moita'
199.	,	'Montijo'
200.	,	'Odivelas'
201.	,	'Oeiras'
202.	,	'Palmela'
203.	,	'Seixal'
204.	,	'Sesimbra'
205.	,	'Setúbal'
206.	,	'Sintra'
207.	,	'Vila Franca de Xira'
208.	,	'Almeirim'
209.	,	'Alpiarça'
210.	,	'Azambuja'
211.	,	'Benavente'
212.	,	'Cartaxo'
213.	,	'Chamusca'
214.	,	'Coruche'
215.	,	'Golegã'
216.	,	'Rio Maior'
217.	,	'Salvaterra de Magos'
218.	,	'Santarém'
219.	,	'Alcácer do Sal'
220.	,	'Grândola'
221.	,	'Odemira'
222.	,	'Santiago do Cacém'
223.	,	'Sines'
224.	,	'Alter do Chão'
225.	,	'Arronches'
226.	,	'Avis'
227.	,	'Campo Maior'
228.	,	'Castelo de Vide'
229.	,	'Crato'
230.	,	'Elvas'
231.	,	'Fronteira'
232.	,	'Gavião'
233.	,	'Marvão'
234.	,	'Monforte'
235.	,	'Nisa'
236.	,	'Ponte de Sor'
237.	,	'Portalegre'
238.	,	'Sousel[4]'
239.	,	'Alandroal'
240.	,	'Arraiolos'
241.	,	'Borba'
242.	,	'Estremoz'
243.	,	'Évora'
244.	,	'Montemor-o-Novo'
245.	,	'Mora[4]'
246.	,	'Mourão'

247.	,	'Portel'
248.	,	'Redondo'
249.	,	'Reguengos de Monsaraz'
250.	,	'Vendas Novas'
251.	,	'Viana do Alentejo'
252.	,	'Vila Viçosa'
253.	,	'Aljustrel'
254.	,	'Almodôvar'
255.	,	'Alvito'
256.	,	'Barrancos'
257.	,	'Beja'
258.	,	'Castro Verde'
259.	,	'Cuba'
260.	,	'Ferreira do Alentejo'
261.	,	'Mértola'
262.	,	'Moura'
263.	,	'Ourique'
264.	,	'Serpa'
265.	,	'Vidigueira'
266.	,	'Albufeira'
267.	,	'Alcoutim'
268.	,	'Aljezur'
269.	,	'Castro Marim'
270.	,	'Faro'
271.	,	'Lagoa'
272.	,	'Lagos'
273.	,	'Loulé'
274.	,	'Monchique'
275.	,	'Olhão'
276.	,	'Portimão'
277.	,	'São Brás de Alportel'
278.	,	'Silves'
279.	,	'Tavira'
280.	,	'Vila do Bispo'
281.	,	'Vila Real de Santo António'
282.	,	'Angra do Heroísmo'
283.	,	'Calheta'
284.	,	'Corvo'
285.	,	'Horta'
286.	,	'Lagoa'
287.	,	'Lajes das Flores'
288.	,	'Lajes do Pico'
289.	,	'Madalena'
290.	,	'Nordeste'
291.	,	'Ponta Delgada'
292.	,	'Povoação'
293.	,	'Praia da Vitória'
294.	,	'Ribeira Grande'
295.	,	'Santa Cruz da Graciosa'
296.	,	'Santa Cruz das Flores'
297.	,	'São Roque do Pico'
298.	,	'Velas'
299.	,	'Vila do Porto'
300.	,	'Vila Franca do Campo'
301.	,	'Calheta'
302.	,	'Câmara de Lobos'
303.	,	'Funchal'
304.	,	'Machico'
305.	,	'Ponta do Sol'
306.	,	'Porto Moniz'

```

307.         , 'Porto Santo'
308.         , 'Ribeira Brava'
309.         , 'Santa Cruz'
310.         , 'Santana'
311.         , 'São Vicente']
312.
313.     populacao = ['20970',
314.                  '15873',
315.                  '8144',
316.                  '17902',
317.                  '8560',
318.                  '11210',
319.                  '41499',
320.                  '13283',
321.                  '84636',
322.                  '8877',
323.                  '18114',
324.                  '116531',
325.                  '181919',
326.                  '34057',
327.                  '6405',
328.                  '46865',
329.                  '15699',
330.                  '48271',
331.                  '152792',
332.                  '6985',
333.                  '21446',
334.                  '11898',
335.                  '131738',
336.                  '23840',
337.                  '20861',
338.                  '29484',
339.                  '165631',
340.                  '137727',
341.                  '174382',
342.                  '66113',
343.                  '86072',
344.                  '215284',
345.                  '62510',
346.                  '138525',
347.                  '68221',
348.                  '21761',
349.                  '38317',
350.                  '21399',
351.                  '96570',
352.                  '79579',
353.                  '299938',
354.                  '53366',
355.                  '18891',
356.                  '15567',
357.                  '19075',
358.                  '18470',
359.                  '56576',
360.                  '46790',
361.                  '51661',
362.                  '56709',
363.                  '69922',
364.                  '10241',
365.                  '5059',
366.                  '39345',

```

367.	'9090'	,
368.	'6031'	,
369.	'14932'	,
370.	'12009'	,
371.	'10703'	,
372.	'5792'	,
373.	'5683'	,
374.	'3312'	,
375.	'24959'	,
376.	'3996'	,
377.	'9729'	,
378.	'5480'	,
379.	'2610'	,
380.	'15830'	,
381.	'5917'	,
382.	'6649'	,
383.	'7154'	,
384.	'5384'	,
385.	'6017'	,
386.	'7761'	,
387.	'7716'	,
388.	'6541'	,
389.	'49868'	,
390.	'4568'	,
391.	'33586'	,
392.	'14550'	,
393.	'6877'	,
394.	'21808'	,
395.	'8481'	,
396.	'6073'	,
397.	'4070'	,
398.	'7847'	,
399.	'45992'	,
400.	'24128'	,
401.	'27298'	,
402.	'77916'	,
403.	'25965'	,
404.	'38405'	,
405.	'10244'	,
406.	'23944'	,
407.	'54120'	,
408.	'11403'	,
409.	'22685'	,
410.	'11068'	,
411.	'35068'	,
412.	'133724'	,
413.	'17597'	,
414.	'58866'	,
415.	'3825'	,
416.	'17128'	,
417.	'19892'	,
418.	'11831'	,
419.	'12687'	,
420.	'25230'	,
421.	'8856'	,
422.	'19331'	,
423.	'4052'	,
424.	'13812'	,
425.	'5439'	,
426.	'17277'	,

427.	'11403'	,
428.	'6929'	,
429.	'15840'	,
430.	'124857'	,
431.	'38404'	,
432.	'51684'	,
433.	'23288'	,
434.	'6626'	,
435.	'12106'	,
436.	'2650'	,
437.	'5608'	,
438.	'3429'	,
439.	'4740'	,
440.	'9290'	,
441.	'13928'	,
442.	'18618'	,
443.	'13030'	,
444.	'9920'	,
445.	'7175'	,
446.	'10506'	,
447.	'15488'	,
448.	'11602'	,
449.	'26548'	,
450.	'4723'	,
451.	'96991'	,
452.	'9661'	,
453.	'5926'	,
454.	'6407'	,
455.	'6978'	,
456.	'47127'	,
457.	'5652'	,
458.	'4561'	,
459.	'26719'	,
460.	'12486'	,
461.	'39103'	,
462.	'3037'	,
463.	'4617'	,
464.	'8607'	,
465.	'10748'	,
466.	'22412'	,
467.	'8946'	,
468.	'52192'	,
469.	'8157'	,
470.	'5045'	,
471.	'4831'	,
472.	'7390'	,
473.	'3167'	,
474.	'53641'	,
475.	'43596'	,
476.	'15082'	,
477.	'12533'	,
478.	'13627'	,
479.	'51540'	,
480.	'25670'	,
481.	'14180'	,
482.	'11719'	,
483.	'26487'	,
484.	'10490'	,
485.	'78220'	,
486.	'35377'	,

487.	'12860'
488.	'4002'
489.	'21214'
490.	'7989'
491.	'6323'
492.	'44068'
493.	'3739'
494.	'14682'
495.	'36902'
496.	'34970'
497.	'3321'
498.	'7402'
499.	'19505'
500.	'168987'
501.	'181724'
502.	'75419'
503.	'212474'
504.	'507220'
505.	'211359'
506.	'84008'
507.	'64526'
508.	'56887'
509.	'159602'
510.	'176218'
511.	'64214'
512.	'166835'
513.	'51559'
514.	'115758'
515.	'388434'
516.	'141603'
517.	'22569'
518.	'7087'
519.	'22445'
520.	'30214'
521.	'23740'
522.	'9253'
523.	'17629'
524.	'5375'
525.	'20340'
526.	'21268'
527.	'57398'
528.	'11712'
529.	'14570'
530.	'24621'
531.	'28725'
532.	'13631'
533.	'3191'
534.	'2860'
535.	'4249'
536.	'7907'
537.	'2951'
538.	'3185'
539.	'20706'
540.	'2986'
541.	'3347'
542.	'3054'
543.	'2989'
544.	'6149'
545.	'15092'
546.	'22359'

547.	'4454'	,
548.	'5064'	,
549.	'6944'	,
550.	'6790'	,
551.	'12816'	,
552.	'52454'	,
553.	'15740'	,
554.	'4188'	,
555.	'2456'	,
556.	'5870'	,
557.	'6387'	,
558.	'10036'	,
559.	'11259'	,
560.	'5142'	,
561.	'7719'	,
562.	'8285'	,
563.	'6746'	,
564.	'2462'	,
565.	'1645'	,
566.	'33550'	,
567.	'6946'	,
568.	'4599'	,
569.	'7848'	,
570.	'6202'	,
571.	'13749'	,
572.	'4653'	,
573.	'14374'	,
574.	'5498'	,
575.	'41123'	,
576.	'2244'	,
577.	'5599'	,
578.	'6274'	,
579.	'60974'	,
580.	'14681'	,
581.	'30442'	,
582.	'68873'	,
583.	'5182'	,
584.	'44607'	,
585.	'55416'	,
586.	'10416'	,
587.	'36174'	,
588.	'24750'	,
589.	'5154'	,
590.	'18888'	,
591.	'33903'	,
592.	'3205'	,
593.	'465'	,
594.	'14542'	,
595.	'22748'	,
596.	'1464'	,
597.	'4498'	,
598.	'5875'	,
599.	'4875'	,
600.	'67864'	,
601.	'5954'	,
602.	'21331'	,
603.	'32698'	,
604.	'4225'	,
605.	'2164'	,
606.	'3264'	,

```

607.         '5137',
608.         '5623',
609.         '11078',
610.         '10865',
611.         '33732',
612.         '104129',
613.         '20094',
614.         '8544',
615.         '2350',
616.         '5176',
617.         '12411',
618.         '44744',
619.         '6750',
620.         '5150',
621.     ]
622.
623.     nuts2 = ['1',
624.             '1',
625.             '1',
626.             '1',
627.             '1',
628.             '1',
629.             '1',
630.             '1',
631.             '1',
632.             '1',
633.             '1',
634.             '1',
635.             '1',
636.             '1',
637.             '1',
638.             '1',
639.             '1',
640.             '1',
641.             '1',
642.             '1',
643.             '1',
644.             '1',
645.             '1',
646.             '1',
647.             '1',
648.             '1',
649.             '1',
650.             '1',
651.             '1',
652.             '1',
653.             '1',
654.             '1',
655.             '1',
656.             '1',
657.             '1',
658.             '1',
659.             '1',
660.             '1',
661.             '1',
662.             '1',
663.             '1',
664.             '1',
665.             '1',
666.             '1',

```



667.	'1',
668.	'1',
669.	'1',
670.	'1',
671.	'1',
672.	'1',
673.	'1',
674.	'1',
675.	'1',
676.	'1',
677.	'1',
678.	'1',
679.	'1',
680.	'1',
681.	'1',
682.	'1',
683.	'1',
684.	'1',
685.	'1',
686.	'1',
687.	'1',
688.	'1',
689.	'1',
690.	'1',
691.	'1',
692.	'1',
693.	'1',
694.	'1',
695.	'1',
696.	'1',
697.	'1',
698.	'1',
699.	'1',
700.	'1',
701.	'1',
702.	'1',
703.	'1',
704.	'1',
705.	'1',
706.	'1',
707.	'1',
708.	'1',
709.	'3',
710.	'3',
711.	'3',
712.	'3',
713.	'3',
714.	'3',
715.	'3',
716.	'3',
717.	'3',
718.	'3',
719.	'3',
720.	'3',
721.	'3',
722.	'3',
723.	'3',
724.	'3',
725.	'3',
726.	'3',

727.	'3',
728.	'3',
729.	'3',
730.	'3',
731.	'3',
732.	'3',
733.	'3',
734.	'3',
735.	'3',
736.	'3',
737.	'3',
738.	'3',
739.	'3',
740.	'3',
741.	'3',
742.	'3',
743.	'3',
744.	'3',
745.	'3',
746.	'3',
747.	'3',
748.	'3',
749.	'3',
750.	'3',
751.	'3',
752.	'3',
753.	'3',
754.	'3',
755.	'3',
756.	'3',
757.	'3',
758.	'3',
759.	'3',
760.	'3',
761.	'3',
762.	'3',
763.	'3',
764.	'3',
765.	'3',
766.	'3',
767.	'3',
768.	'3',
769.	'3',
770.	'3',
771.	'3',
772.	'3',
773.	'3',
774.	'3',
775.	'3',
776.	'3',
777.	'3',
778.	'3',
779.	'3',
780.	'3',
781.	'3',
782.	'3',
783.	'3',
784.	'3',
785.	'3',
786.	'3',

787.	'3'
788.	'3'
789.	'3'
790.	'3'
791.	'3'
792.	'3'
793.	'3'
794.	'3'
795.	'3'
796.	'3'
797.	'3'
798.	'3'
799.	'3'
800.	'3'
801.	'3'
802.	'3'
803.	'3'
804.	'3'
805.	'3'
806.	'3'
807.	'3'
808.	'3'
809.	'2'
810.	'2'
811.	'2'
812.	'2'
813.	'2'
814.	'2'
815.	'2'
816.	'2'
817.	'2'
818.	'2'
819.	'2'
820.	'2'
821.	'2'
822.	'2'
823.	'2'
824.	'2'
825.	'2'
826.	'2'
827.	'4'
828.	'4'
829.	'4'
830.	'4'
831.	'4'
832.	'4'
833.	'4'
834.	'4'
835.	'4'
836.	'4'
837.	'4'
838.	'4'
839.	'4'
840.	'4'
841.	'4'
842.	'4'
843.	'4'
844.	'4'
845.	'4'
846.	'4'

847.	'4'	,
848.	'4'	,
849.	'4'	,
850.	'4'	,
851.	'4'	,
852.	'4'	,
853.	'4'	,
854.	'4'	,
855.	'4'	,
856.	'4'	,
857.	'4'	,
858.	'4'	,
859.	'4'	,
860.	'4'	,
861.	'4'	,
862.	'4'	,
863.	'4'	,
864.	'4'	,
865.	'4'	,
866.	'4'	,
867.	'4'	,
868.	'4'	,
869.	'4'	,
870.	'4'	,
871.	'4'	,
872.	'4'	,
873.	'4'	,
874.	'4'	,
875.	'4'	,
876.	'4'	,
877.	'4'	,
878.	'4'	,
879.	'4'	,
880.	'4'	,
881.	'4'	,
882.	'4'	,
883.	'4'	,
884.	'4'	,
885.	'5'	,
886.	'5'	,
887.	'5'	,
888.	'5'	,
889.	'5'	,
890.	'5'	,
891.	'5'	,
892.	'5'	,
893.	'5'	,
894.	'5'	,
895.	'5'	,
896.	'5'	,
897.	'5'	,
898.	'5'	,
899.	'5'	,
900.	'5'	,
901.	'7'	,
902.	'7'	,
903.	'7'	,
904.	'7'	,
905.	'7'	,
906.	'7'	,

```

907.         '7',
908.         '7',
909.         '7',
910.         '7',
911.         '7',
912.         '7',
913.         '7',
914.         '7',
915.         '7',
916.         '7',
917.         '7',
918.         '7',
919.         '7',
920.         '6',
921.         '6',
922.         '6',
923.         '6',
924.         '6',
925.         '6',
926.         '6',
927.         '6',
928.         '6',
929.         '6',
930.         '6',
931.     ]
932.     k = 1
933.     with open('concelho.sql', 'w') as f:
934.         for i in range(len(concelhos)):
935.             f.write("INSERT INTO CONCELHO VALUES ({}, '{}', {}, {});\n".format(k,
concelhos[i], populacao[i], nuts2[i]))
936.             k = k + 1
937.         f.close()

```

```

1. #Cria 2 arrays sem que o segundo repita o do segundo
2. import random
3.
4. import numpy as np
5.
6. if __name__ == '__main__':
7.     lista = []
8.     lista1 = []
9.     for i in range (1500):
10.         k = random.randint(1,100000)
11.         lista.append(k)
12.     indexes = np.unique(lista, return_index=True)[1]
13.     p = [lista[index] for index in sorted(indexes)]
14.
15.     for i in range (len(p)):
16.         k1 = random.randint(1,100000)
17.         lista1.append(k1)
18.
19.     with open('contacto.sql', 'w') as f:
20.         for i in range (len(p)-1):
21.             f.write('INSERT INTO contacto VALUES ({}, {})\n'.format(p[i], lista1[i]))
22.     f.close()

```

```

1. # cria um ficheiro sql que vai conter todos os motivos e basta executar no sql
2. motivos = ['Sintomas', 'Viagem', 'Contacto com pessoas com covid-
   19', 'Efectuar Operacao', 'Passar fronteira',
3.           'Outros']
4.
5. if __name__ == '__main__':
6.     with open('motivo.sql', 'w') as f:
7.         k = 1
8.         for i in range (len(motivos) - 1):
9.             f.write('INSERT INTO motivo VALUES ({}, "{}")\n'.format(k,motivos[i]))
10.            k += 1
11.        f.close()

```

```

1. # cria um ficheiro sql que vai criar 1500 notificacoes com motivos random e basta
   executar no sql
2. import random
3.
4. if __name__ == '__main__':
5.     with open('notificacao.sql', 'w') as f:
6.         k = 1
7.         i = 0
8.         p = 2
9.         while i <= 1500:
10.            for w in range(random.randint(60, 70)):
11.                f.write('INSERT INTO notificacao VALUES ({}, 2020/12/{}, {})\n'.format(
   k, p, random.randint(1, 5)))
12.                k += 1
13.                i += 1
14.                p += 1
15.
16.        f.close()

```

```

1. # cria um ficheiro sql que vai as criar as inserções excluindo id notificações que não
   podiam aparecer por não terem sintomas e basta executar no sql
2. import random
3.
4. import numpy as np
5. def num(n):
6.     array = []
7.     for s in range (1,n):
8.         array.append(s)
9.     return array
10. def num2(n):
11.     array = []
12.     for s in range (1,n,5):
13.         array.append(s)
14.     return array
15. def rep (a,b):
16.     for i in b:
17.         if i in a:
18.             b.remove(i)

```

```

19.
20.
21. with open('notificacao_sintoma.sql', 'w') as f:
22.     if __name__ == '__main__':
23.         array = [3, 6, 13, 41, 61, 65, 66, 67, 68, 70, 71, 73, 83, 96, 98, 110, 114,
24.                 120, 125, 130, 139, 142, 148, 149, 157, 167, 172, 180, 182, 192, 195,
25.                 198, 202, 215, 219, 224, 250, 254, 259, 261, 268, 270, 277, 278, 283,
26.                 288, 298, 300, 317, 319, 327, 333, 334, 335, 340, 341, 344, 345, 349,
27.                 354, 355, 358, 359, 360, 367, 368, 369, 371, 372, 374, 375, 379, 380,
28.                 382, 390, 402, 411, 416, 422, 423, 427, 440, 445, 456, 457, 458, 459,
29.                 467, 473, 475, 482, 491, 493, 497, 498, 510, 513, 515, 520, 525, 526,
30.                 533, 536, 540, 543, 554, 556, 560, 561, 583, 586, 594, 598, 603, 608,
31.                 617, 623, 628, 632, 633, 634, 639, 642, 643, 646, 649, 654, 656, 659,
32.                 662, 667, 678, 679, 682, 686, 690, 696, 705, 713, 718, 722, 723, 729,
33.                 753, 756, 761, 766, 767, 769, 788, 789, 794, 795, 797, 803, 804, 807,
34.                 813, 817, 823, 824, 827, 828, 832, 838, 855, 860, 863, 866, 870, 871,
35.                 876, 880, 881, 884, 888, 892, 894, 902, 906, 907, 908, 914, 917, 920,
36.                 922, 928, 930, 931, 947, 949, 950, 961, 964, 965, 967, 969, 973, 976,
37.                 988, 993, 997, 1000, 1001, 1003, 1006, 1007, 1008, 1011, 1021, 1023, 1028,
38.                 1037, 1041, 1062, 1065, 1070, 1081, 1087, 1099, 1100, 1102, 1106, 1119,
39.                 1125, 1139, 1157, 1163, 1173, 1175, 1178, 1183, 1185, 1195, 1196, 1201,
40.                 1202, 1209, 1213, 1214, 1219, 1227, 1228, 1231, 1233, 1239, 1244, 1248,
41.                 1249, 1250, 1252, 1258, 1271, 1275, 1280, 1295, 1297, 1303, 1306, 1313,
42.                 1321, 1323, 1326, 1329, 1333, 1335, 1339, 1349, 1354, 1357, 1362, 1367,
43.                 1370, 1378, 1384, 1386, 1388, 1392, 1405, 1412, 1414, 1417, 1419, 1420,
44.                 1421, 1431, 1436, 1439, 1444, 1459, 1464, 1468, 1477, 1478, 1479, 1480,
45.                 1482]
46.         total = []
47.         for i in range(1500):
48.             lista = []
49.             for l in range(random.randint(1, 5)): # No minimo 1 sintoma e no max 4 sito
mas
50.                 r = random.randint(1, 13)
51.                 lista.append(r)
52.                 indexes = np.unique(lista)
53.                 total.append(indexes)
54.         all = num(1500)
55.         all2 = num2(1500)
56.         rep(all, all2)
57.         rep(array, all2)
58.         rep(array, all)
59.         rep(all2, all)
60.         rep(array, all2)
61.         print(all)
62.         print(all2)
63.         print(array)
64.         k = 0
65.         for o in array:
66.             c = total[o]
67.             for j in range(len(c)):
68.                 k += 1
69.                 f.write('INSERT INTO notificacao_sintoma VALUES ({} , {}, {});\n'.format(k, o
, c[j]))
70.
71.         for e in all:
72.             c = total[e]
73.             for j in range(len(c)):
74.                 k += 1
75.                 f.write('INSERT INTO notificacao_sintoma VALUES ({} , {}, {});\n'.format(k,
e, c[j]))

```

```

76.
77.     for q in all2:
78.         k += 1
79.
80.         f.write('INSERT INTO notificacao_sintoma VALUES ({}, {}, {});\n'.format(k, q, 1
81.         4))
82. f.close()

```

```

1. # vai gerar 10000 pessoas fazendo random de nomes comuns e apelidos comuns em Portugal
   e tbm um random de idades
2. import random
3.
4.
5. def gerarnomep():
6.     nomes_p = ["Santiago", "João", "Francisco", "Rodrigo", "Martim", "Afonso", "Tomás",
7.     "Miguel", "Duarte", "Gabriel",
8.     "Lourenço", "Gonçalo", "Pedro", "Guilherme", "Tiago", "Dinis", "Rafael",
9.     "Diogo", "Lucas", "Salvador",
10.    "Gustavo", "David", "Vicente", "Simão", "José", "Mateus", "Diego", "Manu
11.    el", "Henrique", "António",
12.    "Daniel", "Vasco", "Leonardo", "Bernardo", "André", "Luís", "Enzo", "Edu
13.    ardo", "Alexandre", "Isaac",
14.    "Kevin", "Rúben", "Filipe", "Matias", "Leandro", "Xavier", "Ricardo", "S
15.    amuel", "Artur", "Nuno", "Bruno",
16.    "Carlos", "Davi", "Sebastião", "Hugo", "Valentim", "Rui", "Frederico", "
17.    Renato", "Bryan", "Jorge",
18.    "Lorenzo", "Benjamim", "Joaquim", "Paulo", "Marco", "Noah", "Mário", "Fá
19.    bio", "Jaime", "Manel", "Tomé",
20.    "William", "Cristiano", "Ângelo", "Micael", "Gil", "Ivo", "Edgar", "Raúl
21.    ", "Fernando", "Brian", "Ivan",
22.    "César", "Emanuel", "Gaspar", "Luca", "Vítor", "Romeu", "Sandro", "Dilan
23.    ", "Sérgio", "Alex", "Igor",
24.    "Eric", "Marcelo", "Elias", "Denis", "Caio", "Isac", "Ismael", "Liam", "
25.    Lisandro", "Jonathan", "Moisés",
26.    "Yuri", "Muhammad", "Josué", "Cláudio", "Mauro", "Adrien", "Danilo", "Jo
27.    el", "Erik", "Álvaro", "Nelson",
28.    "Dário", "Márcio", "Martinho", "Wesley", "Matheus", "Levi", "Telmo", "Ma
29.    rcos", "Júlio", "Wilson",
30.    "Martin", "Victor", "Leonel", "Arthur", "Leo", "Oliver", "Abel", "Thiago
31.    ", "Cristian", "Benjamin",
32.    "Dylan", "Misael", "Edson", "Ian", "Fausto", "Christian", "Ezequiel", "J
33.    onas", "Andrei", "Raul",
34.    "Gerson", "Giovani", "Caetano", "Matteo", "Luan", "Ryan", "Adam", "Hélde
35.    r", "Nicolau", "Sebastian",
36.    "Cauã", "Patrick", "Estêvão", "Flávio", "Adriano", "James", "Óscar", "Ni
37.    colas", "Alexandru", "Kelvin",
38.    "Joshua", "Roberto", "Kendrick", "Augusto", "Cristóvão", "Alberto", "Jes
39.    us", "Natanael", "Domingos",
40.    "Vitor", "Adriel", "Erick", "Mark", "Jason", "Davide", "Ruben", "Félix",
41.    "Marcus", "Luciano", "Denzel",
42.    "Heitor", "Jonatã", "Gael", "Iúri", "Aléxis", "Kévim", "Mikael", "Aron",
43.    "Luís", "Israel", "Lucca",
44.    "Ulisses", "Alonso", "Thomas", "Célio", "Helder", "Dennis", "Mohammad",
45.    "Rúdi", "Rayan", "Damien",
46.    "Edmundo", "Mohamed", "Adrian", "Ary", "Noé", "Agostinho", "Nataniel", "
47.    Valentino", "Maurício",

```



27. "Fabrício", "Denilson", "Felipe", "Theo", "Éder", "Orlando", "Pietro", "Joabe", "Ravi", "Isaque",  
 28. "Rogério", "Américo", "Enrique", "Raphael", "Aaron", "Alexandro", "Juliano", "Breno", "Jónatas", "Jair",  
 29. "Ianis", "Isael", "Robert", "Salomão", "Ícaro", "Tobias", "Amadeu", "Júnior", "Kenzo", "Nathan", "Kauã",  
 30. "Ibrahim", "Alexander", "Omar", "Anselmo", "Fred", "Élson", "Zion", "Théo", "Jayden", "Eduard", "Válter",  
 31. "Henry", "Mickael", "Diniz", "Mateo", "Samir", "Lúcio", "Kennedy", "Tierri", "Messias", "Hélio", "Abdul",  
 32. "Humberto", "Antonio", "Celso", "Marlon", "Jerónimo", "Kauan", "Timóteo", "Haniel", "Iago", "Stefan",  
 33. "Eliel", "Armando", "Giovanni", "Dominic", "Derick", "Jayson", "Liedson", "Daniil", "Adonai", "Hernâni",  
 34. "Absalão", "Jessé", "Ethan", "Hazael", "Yu", "Damian", "Maksym", "Mamado", "Angelo", "Amir", "Giani",  
 35. "Yannick", "Evan", "Azael", "Zaqueu", "Kelton", "Maksim", "Milan", "Maxim", "Ruan", "Jaden", "Anderson",  
 36. "Yanni", "Matviy", "Chris", "Marcelino", "Apolo", "Silvestre", "Aires", "Brayan", "Jairo", "Ayaan",  
 37. "Timur", "Alessandro", "Alfredo", "Martín", "Keven", "Kiamy", "Lázaro", "Max", "Alan", "Bento", "Saúl",  
 38. "Isaias", "Lukas", "Kian", "Djeyson", "Darius", "Mohammed", "Matei", "Eliezer", "Aryan", "Pablo",  
 39. "Youssef", "Anthony", "Luiz", "Armaan", "Kleyton", "Evandro", "Angélico", "Artyom", "Josias", "Nilson",  
 40. "Marvin", "Máximo", "Marley", "Kévin", "Délcio", "Benício", "Yohan", "Deivid", "Uriel", "Raphaël",  
 41. "Vladimir", "Lopo", "Nicollas", "Oziel", "Steven", "Nikita", "Aleksandr", "Inácio", "Vladislav",  
 42. "Prince", "Abdullah", "George", "Dante", "Derek", "Axel", "Eurico", "Zayn", "Johnny", "Elijah",  
 43. "Baltasar", "Gastão", "Felix", "Nicholas", "Ioan", "Fabiano", "Rubim", "Delmar", "Michael", "Sancho",  
 44. "Arsénio", "Francesco", "Armindo", "Jack", "Valentin", "Rubi", "Christopher", "Eder", "Abdoulaye",  
 45. "Maximus", "Neymar", "Jan", "Richard", "Reinaldo", "Ion", "Delfim", "Nazar", "Jacinto", "Juan", "Ezio",  
 46. "Emídio", "Virgílio", "Kelson", "Davy", "Bogdan", "Sílvio", "Alexandr", "Diamantino", "Louis",  
 47. "Emmanuel", "Iulian", "Oscar", "Tito", "Nilton", "Nélson", "Gusttavo", "Estevão", "Aarnav", "Jeferson",  
 48. "Joseph", "Agrim", "Lamarana", "Helton", "Édi", "Wallace", "Danilson", "Keyson", "Belchior", "Dérick",  
 49. "Walter", "Albert", "Fernão", "Shayan", "Mario", "Kael", "Mustafa", "Georgi", "Anael", "Francis",  
 50. "Esteban", "Enoque", "Hermano", "Kiari", "Milton", "Ibrahima", "Calebe", "Simon", "Ezra", "Yannis",  
 51. "Mauri", "Emílio", "Tom", "Willian", "Hamilton", "Kayden", "Tyron", "Abrão", "Elielson", "Laércio",  
 52. "Hao", "Guransh", "Eli", "Atanas", "Eddy", "Ming", "Ivanilson", "Jefferson", "Fabian", "Marcel",  
 53. "Vasile", "Kimi", "Bartolomeu", "Jia", "Cédric", "Arnaldo", "Valter", "Eliseu", "Kyami", "Edir", "Jóni",  
 54. "Léon", "Mariano", "Kyle", "Cael", "Kirill", "Nikolai", "Henzo", "Rodney", "Michel", "Lev", "Lui", "Yi",  
 55. "Rodolfo", "Ben", "Nathaniel", "Camilo", "Giovany", "Kushal", "Samoa", "Alcino", "Jun", "Filip", "Túlio",  
 56. "Kaio", "Élton", "Elói", "Abrão", "Atílio", "Asael", "Nélio", "Arman", "Cristovão", "Jadir", "Viriato",

```

57.         "Oleksandr", "Eden", "Ionut", "Abílio", "Ahmad", "Ayrton", "Jimmy", "Gel
son", "Eusébio", "Éric",
58.         "Teófilo", "Ilídio", "Luka", "Antoine", "Harry", "Luke", "Finn", "Hilári
o", "Jordan", "Antônio",
59.         "Aayush", "Maximiano", "Brahim", "Zhen", "Andriy", "Gilberto", "Tyler",
"Avelino", "Kevyn", "Arian",
60.         "Lian", "Antero", "Nicolae", "Milton", "Pavel", "Jonata", "Adrián", "Rom
ão", "Janilson", "Dario",
61.         "Damião", "Stephen", "Raj", "Zé", "Alvin", "Mathias", "Osvaldo", "Gianlu
ca", "Edwin", "Jamie", "Russell",
62.         "Ravidson", "Destiny", "Érico", "Kyan", "Mizael", "Gilson", "Ari", "Dina
rte", "Patrício", "Giorgi",
63.         "Yaroslav", "Teotônio", "Adalberto", "Santhiago", "Amaro", "Laurindo", "
Zihao", "Nikolas", "Kylia",
64.         "Catalin", "Djayson", "Mouhamed", "Claúdio", "Bilal", "Kristian", "Oséia
s", "Aarav", "Jeremias", "Ishan",
65.         "Valdemar", "Imran", "Nilo", "Kenny", "Riyan", "Aman", "Januário", "Rian
", "Nicola", "Matthias"]
66.     r = random.randint(0, len(nomes_p) - 1)
67.     return str(nomes_p[r])
68.
69.
70. def gerarnomepF():
71.     nomes_f = ["Maria", "Matilde", "Leonor", "Carolina", "Beatriz", "Mariana", "Ana", "
Sofia", "Inês", "Margarida",
72.         "Francisca", "Lara", "Laura", "Alice", "Joana", "Clara", "Diana", "Madal
ena", "Mara", "Luana",
73.         "Benedita", "Constança", "Mafalda", "Bianca", "Íris", "Sara", "Camila",
"Gabriela", "Rita", "Eva",
74.         "Carlota", "Letícia", "Vitória", "Ema", "Luísa", "Catarina", "Ariana", "
Yasmin", "Yara", "Carminho",
75.         "Miriam", "Marta", "Alicia", "Rafaela", "Luena", "Victória", "Núria", "K
yara", "Helena", "Valentina",
76.         "Júlia", "Bárbara", "Isabel", "Nicole", "Daniela", "Lia", "Bruna", "Meli
ssa", "Noa", "Jéssica", "Teresa",
77.         "Iara", "Raquel", "Filipa", "Mia", "Érica", "Luna", "Isis", "Caetana", "
Alexandra", "Mélanie", "Pilar",
78.         "Juliana", "Kelly", "Isabela", "Adriana", "Aurora", "Débora", "Soraia",
"Eduarda", "Lorena", "Áurea",
79.         "Olivia", "Amélia", "Emília", "Naiara", "Frederica", "Irina", "Renata",
"Anita", "Larissa", "Petra",
80.         "Emma", "Sarah", "Vera", "Tatiana", "Julieta", "Nádia", "Mayara", "Salom
é", "Samira", "Rebeca",
81.         "Fabiana", "Patrícia", "Simone", "Paloma", "Aline", "Liliana", "Nair", "
Luz", "Luciana", "Andreia",
82.         "Isabella", "Dalila", "Chloe", "Ester", "Flor", "Safira", "Erika", "Elis
a", "Anna", "Rosa", "Mónica",
83.         "Cláudia", "Lúcia", "Abigail", "Ângela", "Rosarinho", "Sophia", "Carla",
"Alana", "Micaela", "Bia",
84.         "Fátima", "Maiara", "Emily", "Eliana", "Carmo", "Dânia", "Marisa", "Eric
a", "Cíntia", "Amália", "Raissa",
85.         "Viviane", "Serena", "Vanessa", "Clarisse", "Denise", "Carina", "Violeta
", "Eliane", "Rute", "Flávia",
86.         "Kiara", "Iris", "Liara", "Susana", "Cristina", "Cecília", "Giovana", "N
ayara", "Márcia", "Oriana",
87.         "Milena", "Victoria", "Lídia", "Dara", "Manuela", "Cristiana", "Estrela"
, "Elena", "Lea", "Liana",
88.         "Angélica", "Solange", "Telma", "Dayane", "Sónia", "Antónia", "Verónica"
, "Marina", "Aléxia", "Ariane",

```

89. "Isa", "India", "Paula", "Simara", "Liane", "Ellen", "Lívia", "Daria", "Neuza", "Lais", "Tânia", "Leila",  
90. "Sandra", "Priscila", "Martina", "Riana", "Melanie", "Tamara", "Samanta", "Valéria", "Jénifer", "Ariele",  
91. "Maísa", "Deise", "Maia", "Taís", "Carmen", "Briana", "Glória", "Amanda", "Ciara", "Doriana", "Esther",  
92. "Thaís", "Clarinha", "Jacinta", "Guiomar", "Esmeralda", "Brenda", "Aisha", "Anaísa", "Esperança",  
93. "Viviana", "Elsa", "Lua", "Natacha", "Olivia", "Oceana", "Luzia", "Ashley", "Tamara", "Concha", "Nina",  
94. "Noemi", "Cloe", "Maya", "Rafaella", "Rayanne", "Ísis", "Alicia", "Zoé", "Sabrina", "Estela", "Maira",  
95. "Lya", "Naísa", "Dora", "Cloé", "Charlotte", "Angelina", "Taísa", "Samar", "Penélope", "Lília", "Luisa",  
96. "Chloé", "Jennifer", "Suri", "Stella", "Maura", "Eunice", "Martinha", "Alysha", "Aura", "Maryam",  
97. "Karina", "Mayra", "Alma", "Josiane", "Eleonor", "Anaís", "Vânia", "Laís", "Graça", "Rayssa", "Mercedes",  
98. "Alissa", "Ária", "Lisandra", "Alexa", "Adelaide", "Miranda", "Naíma", "Yohanna", "Lígia", "Stephanie",  
99. "Adelina", "Iris", "Melina", "Lina", "Alexia", "Cátia", "Liz", "Jussara", "Mila", "Carol", "Dulce",  
100. "Fatumata", "Cândida", "Pérola", "Isaura", "Taíssa", "Jade", "Joyce", "Lavínia", "Suéli", "Irene",  
101. "Kayla", "Nídia", "Aurea", "Anabela", "Michele", "Emilly", "Miriam", "Hellen", "Tiffany", "Mercês",  
102. "Milana", "Nancy", "Naomi", "Rebecca", "Jacira", "Katie", "Nara", "Anastasia", "Leticia", "Leandra",  
103. "Giovanna", "Morgana", "Anamar", "Jessica", "Tiara", "Anaya", "Matilda", "Zoe", "Alessia", "Andrea",  
104. "Aminata", "Gabriella", "Flora", "Magda", "Ayla", "Adriela", "Rania", "Evelyn", "Luiza", "Elisabete",  
105. "Joelma", "Marlene", "Tiana", "Melany", "Keyla", "Fernanda", "Malika", "Dádiva", "Allana", "Nayra",  
106. "Natália", "Natalia", "Cinara", "Michelle", "Diva", "Hannah", "Marisol", "Lucília", "Stela", "Luara",  
107. "Janice", "Sol", "Kataleya", "Sheila", "Cataleya", "Cassandra", "Bella", "Florabela", "Celina", "Aida",  
108. "Milene", "Alina", "Melyssa", "Kaylane", "Branca", "Iriana", "Dafne", "Amora", "Liliane", "Janaína",  
109. "Suria", "Lana", "Ariel", "Sílvia", "Kimberly", "Siena", "Silvana", "Weza", "Evelina", "Haniela",  
110. "Veronica", "Guadalupe", "Rayane", "Alyssa", "Mariam", "Sophie", "Joice", "Camilla", "Alba", "Kieza",  
111. "Jasmina", "Veronika", "Khadija", "Hadassa", "Polina", "Yasmine", "Cármén", "Nadine", "Sienna", "Ariela",  
112. "Zara", "Elisabeth", "Seerat", "Maimuna", "Ânia", "Klara", "Ália", "Melinda", "Brianna", "Ava", "Nour",  
113. "Ândria", "Aicha", "Zita", "Tabita", "Nalini", "Rossana", "Isadora", "Sharon", "Laryssa", "Taynara",  
114. "Barbara", "Lucy", "Andreea", "Dina", "Maitê", "Lola", "Lena", "Teresinha", "Ellie", "Elvira", "Sahara",  
115. "Aylla", "Catalina", "Nicolly", "Marília", "Stephany", "Rúbia", "Noémi", "Filomena", "Eliza", "Agatha",  
116. "Denisa", "Daisi", "Fábia", "Olga", "Aaliyah", "Emilia", "Nayla", "Liany", "Sasha", "Nikol", "Muriel",  
117. "Pietra", "Latifa", "Aliça", "Lira", "Tician", "Leah", "Lucía", "Edna", "Stefany", "Nuna", "Daiana",  
118. "Quélia", "Leia", "Radija", "Emanuela", "Fatima", "Thayla", "Ivan", "Kira", "Anabel", "Amy", "Cássia",

```

119.         "Sarai", "Arina", "Geovana", "Pandora", "Francesca", "Letícia", "
    Alda", "Estefânia", "Amira", "Mirela",
120.         "Lunna", "Hoorain", "Dália", "Zaira", "Cora", "Kailany", "Zoey",
    "Suely", "Sehajpreet", "Ionara",
121.         "Jandira", "Indira", "Mathilde", "Hawa", "Linda", "Debora", "Alek
    sandra", "Dominique", "Anair",
122.         "Heloísa", "Gisela", "Jana", "Delia", "Rosário", "Mágui", "Stefan
    ia", "Inara", "Lidiana", "Cristal",
123.         "Cheila", "Jasmeen", "Letizia", "Laureana", "Anaëlle", "Vilma", "
    Manha", "Andra", "Amelia", "Ayumi",
124.         "Melody", "Yasmina", "Assunção", "Genésia", "Nayma", "Nadia", "Né
    lia", "Antonia", "Crystal", "Yana",
125.         "Kamila", "Thayra", "Élia", "Ayesha", "Lurdes", "Emanuelle", "Ary
    ana", "Anastácia", "Jael", "Aliana",
126.         "Zélia", "Eveline", "Otília", "Perla", "Chelsea", "Yi", "Kendra",
    "Hayla", "Viktoria", "Jil", "Fedra",
127.         "Ivy", "Samaritana", "Marwa", "Miguel", "Aliyha", "Josefa", "Pol
    iana", "Karen", "Laysla", "Ranya",
128.         "Lilian", "Josefina", "Maribel", "Piedade", "Amina", "Lyah", "Dje
    lissa", "Suzi", "Viktoria", "Jasmeet",
129.         "Adélia", "Isís", "Rahyssa", "Stéphanie", "Orquídea", "Shaira", "
    Adama", "Graciana", "Ludmila",
130.         "Yangchen", "Emilie", "Rose", "Céline", "Dayra", "Anaisa", "Alisa
    ", "Aya", "Íris", "Anastacia",
131.         "Lavinia", "Jacyara", "Alisha", "Hossana", "Ananda", "Soraya", "M
    alak", "Arianna", "Karolina", "Marie",
132.         "Florinda", "Samantha", "Karine", "Kaya", "Ayani", "Bela", "Ohana
    ", "Angela", "Jia", "Mikaela",
133.         "Isabelly", "Rahela", "Dilnura", "Elzira", "Mariama", "Elizabeth"
    , "Anais", "Oumou", "Louisa",
134.         "Estefany", "Layra", "Thaíssa", "Cidália", "Selma", "Olinda", "Ka
    tia", "Grace", "Mankirat", "Girisha",
135.         "Lueji", "Heloisa", "Zlata", "Roberta", "Louise", "Ticiane", "Pra
    pti", "Melânia", "Lyara", "Séfora",
136.         "Tamar", "Mellyssa", "Katerina", "Dominika", "Izabel", "Suzana",
    "Yaqi", "Elizabet", "Ágata",
137.         "Valentine", "Amélie", "Vivian", "Neusa", "Gaia", "Deva", "Raina"
    , "Dária", "Lícia", "Stacy", "Elisete",
138.         "Elisama", "Hareem", "Eshal", "Inaaya", "Andressa", "Kelly", "Hél
    ia", "Giulia", "Yiyi", "Quessia",
139.         "Lyana", "Yassna", "Iasmin", "Zilda", "Myriam", "Adriele", "Cleid
    e", "Sujana", "Andreina", "Gabrielly",
140.         "Mel", "Janira", "Geovanna", "Adele", "Celeste", "Nayr", "Aliya",
    "Raíssa", "Alycia", "Berenice",
141.         "Cadija", "Hafsa", "Virgínia", "Bebiana", "Anny", "Luenna", "Nels
    a", "Izabella", "Julia", "Neide",
142.         "Evelin", "Simona", "Jessie", "Surya", "Gina", "Ayra", "Rosalina"
    , "Jane", "Lisa", "Tainara", "Jasmine",
143.         "Miria", "Mellanie", "Ndeye"]
144.         r = random.randint(0, len(nomes_f) - 1)
145.         return str(nomes_f[r])
146.
147.
148.     def geraridade():
149.         return random.randint(1, 100)
150.
151.
152.     def gerarapelidos():
153.         apelidos = ["Abasto", "Abelho", "Abranches", "Abrantes", "Abreu", "Adão", "A
    darga", "Afonso", "Águeda", "Aguíar",

```

154. "Aires", "Albernaz", "Albuquerque", "Alcaide", "Alcântara", "Alc  
oforado", "Aldeia", "Aleixo", "Alencar",  
155. "Almada", "Almeida", "Alpuim", "Alvarenga", "Álvares", "Álvaro",  
"Alvelos", "Alves", "Alvim", "Amado",  
156. "Amaral", "Amarante", "Amaro", "Amoedo", "Amorim", "Andrade", "A  
nes", "Angelim", "Anhaia", "Anjos",  
157. "Anlicoara", "Antas", "Antunes", "Aquino", "Aragão", "Aranha", "  
Arantes", "Araújo", "Areosa", "Argolo",  
158. "Arouca", "Arruda", "Assis", "Assunção", "Ataíde", "Atilano", "A  
veiro", "Avelar", "Ávila", "Azambuja",  
159. "Azenha", "Azeredo", "Azevedo", "Bacelar", "Baía", "Baião", "Bai  
rros", "Baldaia", "Balsemão",  
160. "Bandeira", "Banha", "Baptista", "Barata", "Barateiro", "Barbalh  
o", "Barbosa", "Barcelos", "Barra",  
161. "Barreira", "Barreiros", "Barrico", "Barrela", "Barreno", "Barre  
to", "Barroca", "Barrocas",  
162. "Barroqueiro", "Barros", "Barroso", "Basílio", "Bastos", "Batata  
", "Beiriz", "Belchior", "Belchiorinho",  
163. "Belém", "Belmonte", "Belo", "Beltrão", "Bencatel", "Benevides",  
"Bensaúde", "Bentes", "Bettencourt",  
164. "Bento", "Berenguer", "Bernardes", "Bessa", "Bezerra", "Bezerril  
", "Bicalho", "Bicudo", "Bilhalva",  
165. "Bingre", "Bivar", "Boaventura", "Boeira", "Boga", "Bogado", "Bo  
nilha", "Bonito", "Borba", "Borges",  
166. "Borja", "Botelho", "Botica", "Boto", "Bouça", "Bouças", "Brandã  
o", "Braga", "Bragança", "Branco",  
167. "Brás", "Brásio", "Brasil", "Breia", "Brião", "Brites", "Brito",  
"Brochado", "Brum", "Bugalho",  
168. "Bulhões", "Bulhosa", "Cabeça", "Cabeça,de,Vaca", "Cabral", "Cab  
reira", "Cachão", "Cachoeira",  
169. "Caçoilo", "Cadaval", "Cadavez", "Caeira", "Caeiro", "Caetano",  
"Café", "Caiado", "Caires", "Calado",  
170. "Calçada", "Caldas", "Caldeira", "Calheiros", "Camacho", "Câmara  
", "Camargo", "Camarinho", "Cambezes",  
171. "Camelo", "Camilo", "Caminha", "Campelo", "Campos", "Canadas", "  
Canário", "Cancela", "Candal",  
172. "Candeias", "Canedo", "Caneira", "Canejo", "Canela", "Cangueiro"  
", "Canhão", "Caniça", "Cantanhede",  
173. "Canto", "Caparica", "Capistrano", "Capucho", "Cardim", "Cardoso  
", "Carlos", "Carmo", "Carmona",  
174. "Carneiro", "Coronel", "Carqueijeiro", "Carrasco", "Carrasqueira  
", "Carreira", "Carregueiro",  
175. "Carreiro", "Carrilho", "Carrromeu", "Cartaxo", "Carvalhais", "Ca  
rvalhal", "Carvalheira", "Carvalheiro",  
176. "Carvalho", "Carvalhosa", "Carvalhoso", "Casado", "Casalinho", "  
Cascais", "Casqueira", "Castanheira",  
177. "Castanho", "Castanheda", "Castelhano", "Castelo", "Castelo,Bran  
co", "Castilho", "Castilhos", "Castro",  
178. "Catela", "Cavadas", "Cavaco", "Cavalheiro", "Cedraz", "Cedro",  
"Cerqueira", "Cerveira", "César",  
179. "Cesário", "Chagas", "Chainho", "Chamusca", "Charneca", "Chaves"  
", "Chousa", "Cidreira", "Cipriano",  
180. "Cirne", "Cisneiros", "Clementino", "Cobra", "Coelho", "Coimbra"  
", "Colaço", "Colares", "Conceição",  
181. "Conde", "Condorcet", "Cordeiro", "Correia", "Corte-  
Real", "Cortês", "Cortesão", "Costa,da,Costa",  
182. "Cotrim", "Couceiro", "Coutinho", "Couto", "Covilhã", "Covinha",  
"Craveiro", "Cruz", "Cunha", "Curvelo",  
183. "Custódio", "Curado", "Damasceno", "Damásio", "Dâmaso", "Dantas"  
", "Delgado", "Dias", "Diegues", "Dinis",

184. "Dourado", "Doutel", "Doutis", "Domingos", "Domingues", "Dornele  
s", "Duarte", "Durão", "Durães",  
185. "Dutra", "Eanes", "Eiró", "Escobar", "Espargosa", "Esparteiro",  
"Espinosa", "Espírito", "Santo",  
186. "Esteves", "Estrada", "Estrela", "Faia", "Fagundes", "Falcão", "  
Faleiro", "Faria", "Farias", "Farinha",  
187. "Faro", "Fartaria", "Faustino", "Fazendeiro", "Feijó", "Feira",  
"Feitosa", "Félix", "Fernandes",  
188. "Ferraço", "Ferrão", "Ferraz", "Ferreira", "Ferro", "Festas", "F  
iães", "Fialho", "Fidalgo", "Figueira",  
189. "Figueiredo", "Figueiró", "Figueiroa", "Filgueiras", "Filipe", "  
Fitas", "Fiúza", "Flávio", "Flores",  
190. "Fogaça", "Fonseca", "Fontes", "Fontinha", "Fontoura", "Foquiço"  
"Fortes", "Fortunato", "Frade",  
191. "Fraga", "Fragoso", "Frajuca", "Franca", "França", "Franco", "Fr  
anca", "Franqueira", "Frazão", "Freire",  
192. "Freiria", "Freitas", "Freixo", "Frias", "Frois", "Frota", "Furq  
uim", "Furtado", "Gabeira", "Gadelha",  
193. "Gago", "Galante", "Galindo", "Galvão", "Gama", "Gameiro", "Garc  
ez", "Garcia", "Garrau", "Garrido",  
194. "Gaspar", "Gentil", "Gil", "Ginjeira", "Girão", "Godinho", "Godo  
i", "Gois", "Gomes", "Gomide",  
195. "Gonçalves", "Gorjão", "Goulart", "Gouveia", "Goulão", "Graça",  
"Grande", "Grangeia", "Granja",  
196. "Granjeiro", "Gravato", "Grilo", "Guedelha", "Guedes", "Guerra",  
"Guerreiro", "Guilheiro", "Guimarães",  
197. "Gusmão", "Guterres", "Henriques", "Hernandes", "Hilário", "Hipó  
lito", "Holanda", "Homem", "Horta",  
198. "Igrejas", "Ilha", "Imperial", "Inácio", "Inês", "Infante", "Jam  
andas", "Janes", "Jardim", "Jesus",  
199. "Jobim", "Jordão", "Jorge", "Júdice", "Junqueira", "Keil", "Lace  
rda", "Lage", "Lages", "Lago", "Lagoa",  
200. "Lagos", "Lalanda", "Lamego", "Lameira", "Lameirinhas", "Lameira  
s", "Lamenha", "Lampreia", "Lancastre",  
201. "Landim", "Lara", "Laranjeira", "Lários", "Laureano", "Leal", "L  
eão", "Leiria", "Leitão", "Leite",  
202. "Leme", "Lemes", "Lemos", "Lessa", "Letras", "Liberato", "Lima",  
"Linhares", "Lindim", "Lins", "Lira",  
203. "Lisboa", "Lobato", "Lobo", "Loio", "Lopes", "Lourenço", "Lourei  
ro", "Lousã", "Lousada", "Lousado",  
204. "Lucas", "Lucena", "Lustosa", "Luz", "Macedo", "Macena", "Machad  
o", "Macieira", "Maciel", "Madeira",  
205. "Madrugã", "Madureira", "Mafra", "Magalhães", "Maia", "Mainha",  
"Maior", "Malafaia", "Malheiro",  
206. "Malho", "Malta", "Mamouros", "Mangueira", "Mansilha", "Manso",  
"Mantas", "Maranhão", "Marçal",  
207. "Marcondes", "Marinho", "Marins", "Mariz", "Marmou", "Marques",  
"Marreiro", "Marroquim", "Martinho",  
208. "Martins", "Mascarenhas", "Mata", "Mateus", "Matias", "Matos", "  
Matosinhos", "Matoso", "Medeiros",  
209. "Medina", "Meira", "Meireles", "Melancia", "Melgaço", "Mena", "M  
elo", "Mendes", "Mendonça", "Meneses",  
210. "Mesquita", "Mexia", "Miguel", "Miguéis", "Mieiro", "Milhões", "  
Milheirão", "Milheirico", "Milheiro",  
211. "Minho", "Miranda", "Mirandela", "Modesto", "Moita", "Mondragão"  
"Monforte", "Monjardim", "Monsanto",  
212. "Mont Alverne", "Monte", "Montenegro", "Monteiro", "Morais", "Mo  
rão", "Moreira", "Moreno", "Morgado",  
213. "Mortágua", "Mota", "Moura", "Mourão", "Mourato", "Mourinho", "M  
outinho", "Muniz", "Murteira",

214. "Murtinho", "Muxagata", "Narvais", "Nascimento", "Natal", "Naves",  
 ", "Nazário", "Negrao", "Negreiros",  
 215. "Negromonte", "Neiva", "Neres", "Neto", "Neves", "Ninharelos",  
 "Nobre", "Nóbrega", "Nogueira", "Noite",  
 216. "Nolasco", "Noieto", "Norões", "Noronha", "Novais", "Nunes", "Ol  
 aio", "Oleiro", "Olivares", "Oliveira",  
 217. "Onofre", "Ornelas", "Orriça", "Osório", "Ourique", "Ouro", "Out  
 eiro", "Pacheco", "Padilha", "Pádua",  
 218. "Paião", "Pais", "Paiva", "Paixão", "Palha", "Palhares", "Palma"  
 ", "Palmeira", "Palos", "Parafita",  
 219. "Paranhos", "Pardo", "Paredes", "Parente", "Parracho", "Parreir"  
 ", "Passarinho", "Passos", "Pastana",  
 220. "Patrício", "Paula", "Paulos", "Paz", "Peçanha", "Pêcego", "Pede  
 rneiras", "Pedro", "Pedroso", "Pegado",  
 221. "Peixoto", "Penha", "Penteado", "Pequeno", "Peralta", "Perdigão"  
 ", "Pereira", "Pescada", "Peseiro",  
 222. "Pessoa", "Pestana", "Picanço", "Picão", "Pimenta", "Pimentel",  
 "Pinhal", "Pinheiro", "Pinho", "Pinto",  
 223. "Piteira", "Pires", "Poças", "Ponte", "Pontes", "Porciúncula", "  
 Portela", "Porto", "Portugal", "Póvoas",  
 224. "Prada", "Prado", "Prates", "Prestes", "Proença", "Protásio", "P  
 rudente", "Pureza", "Quadros",  
 225. "Quaresma", "Queiroga", "Queirós", "Quental", "Quesado", "Quina"  
 ", "Quinaz", "Quinta", "Quintal",  
 226. "Quintana", "Quintanilha", "Quintas", "Quintais", "Quintão", "Qu  
 inteiro", "Quintela", "Quinterno",  
 227. "Quinzeiro", "Quirino", "Rabelo", "Ramalho", "Raminhos", "Ramire  
 s", "Ramos", "Rangel", "Raposo",  
 228. "Rebello", "Rebimbas", "Rebocho", "Rebotim", "Rebouças", "Redondo"  
 ", "Regalado", "Rego", "Regodeiro",  
 229. "Regueira", "Rei", "Reino", "Reis", "Remígio", "Resende", "Ribas"  
 ", "Ribeiro", "Rico", "Rijo", "Rios",  
 230. "Robalinho", "Robalo", "Roçadas", "Rocha", "Rodovalho", "Rodrigu  
 es", "Rolim", "Roriz", "Rosa", "Rosado",  
 231. "Rosário", "Rosmaninho", "Rua", "Ruas", "Ruela", "Rufino", "Sá",  
 "Sabala", "Sabrosa", "Sacadura",  
 232. "Sacramento", "Salazar", "Saldanha", "Sales", "Salgado", "Salgue  
 iro", "Salvado", "Saloi", "Salomão",  
 233. "Saltão", "Sampaio", "Sanches", "Sandinha", "Santana", "Santarém"  
 ", "Santiago", "Santos", "Saraiva",  
 234. "Sardinha", "Sardo", "Sarmento", "Seabra", "Seixas", "Semedo", "  
 Serpa", "Serralheiro", "Serro",  
 235. "Sesimbra", "Setúbal", "Severiano", "Severo", "Silva", "Silveira"  
 ", "Silvestre", "Simas", "Simões",  
 236. "Simão", "Sintra", "Sítima", "Sequeira", "Soares", "Sobral", "So  
 breira", "Sodré", "Soeiro", "Sousa",  
 237. "Souto", "Souto, Maion", "Soveral", "Soverosa", "Tabanez", "Tabor  
 da", "Tabosa", "Talhão", "Tavares",  
 238. "Taveira", "Taveiros", "Távora", "Teixeira", "Tedim", "Teles", "  
 Telinhos", "Temes", "Teodoro", "Terra",  
 239. "Teves", "Tigre", "Tinoco", "Toledo", "Tomé", "Torquato", "Torra  
 do", "Torreiro", "Torres", "Toscano",  
 240. "Travassos", "Toste", "Trigueiro", "Trindade", "Tristão", "Tuna"  
 ", "Uchoa", "Ulhoa", "Úria", "Urias",  
 241. "Valadão", "Valadares", "Valadim", "Valcácer", "Valcanaia", "Val  
 e", "Valente", "Valentim", "Valério",  
 242. "Valgueiro", "Valido", "Valim", "Valverde", "Varanda", "Varão", "  
 Varejão", "Varela", "Vargas",  
 243. "Vasconcelos", "Vasques", "Vaz", "Veiga", "Velasco", "Velasques"  
 ", "Veleda", "Veloso", "Ventura",

244. "Vergueiro", "Veríssimo", "Viana", "Vidal", "Vides", "Vidigal",  
 "Viegas", "Vieira", "Vigário",  
 245. "Vila-  
 Chã", "Vilaça", "Vilalobos", "Vilanova", "Vilante", "Vilar", "Vilarica", "Vilarinho",  
 246. "Vilas,Boas", "Vilaverde", "Vilela", "Vilhena", "Vinhas", "Vital",  
 ", "Viveiros", "Xavier", "Ximenes",  
 247. "Xisto", "Zagalo", "Zambujal", "Zarco", "Abasto", "Abelho", "Abr  
 anches", "Abrantes", "Abreu", "Adão",  
 248. "Adarga", "Afonso", "Águeda", "Aguiar", "Aires", "Albernaz", "Al  
 buquerque", "Alcaide", "Alcântara",  
 249. "Alcoforado", "Aldeia", "Aleixo", "Alencar", "Almada", "Almeida"  
 , "Alpuim", "Alvarenga", "Álvares",  
 250. "Álvaro", "Alvelos", "Alves", "Alvim", "Amado", "Amaral", "Amara  
 nte", "Amaro", "Amoedo", "Amorim",  
 251. "Andrade", "Anes", "Angelim", "Anhaia", "Anjos", "Anlicoara", "A  
 ntas", "Antunes", "Aquino", "Aragão",  
 252. "Aranha", "Arantes", "Araújo", "Areosa", "Argolo", "Arouca", "Ar  
 ruda", "Assis", "Assunção", "Ataíde",  
 253. "Atilano", "Aveiro", "Avelar", "Ávila", "Azambuja", "Azenha", "A  
 zeredo", "Azevedo", "Bacelar", "Baía",  
 254. "Baião", "Bairros", "Baldaia", "Balsemão", "Bandeira", "Banha",  
 "Baptista", "Barata", "Barateiro",  
 255. "Barbalho", "Barbosa", "Barcelos", "Barra", "Barreira", "Barreir  
 os", "Barrico", "Barrela", "Barreno",  
 256. "Barreto", "Barroca", "Barrocas", "Barroqueiro", "Barros", "Barr  
 oso", "Basílio", "Bastos", "Batata",  
 257. "Beiriz", "Belchior", "Belchiorinho", "Belém", "Belmonte", "Belo  
 ", "Beltrão", "Bencatel", "Benevides",  
 258. "Bensaúde", "Bentes", "Bettencourt", "Bento", "Berenguer", "Bern  
 ardes", "Bessa", "Bezerra", "Bezerril",  
 259. "Bicalho", "Bicudo", "Bilhalva", "Bingre", "Bivar", "Boaventura"  
 , "Boeira", "Boga", "Bogado", "Bonilha",  
 260. "Bonito", "Borba", "Borges", "Borja", "Botelho", "Botica", "Boto  
 ", "Bouça", "Bouças", "Brandão",  
 261. "Braga", "Bragança", "Branco", "Brás", "Brásio", "Brasil", "Brei  
 a", "Brião", "Brites", "Brito",  
 262. "Brochado", "Brum", "Bugalho", "Bulhões", "Bulhosa", "Cabeça", "  
 Cabeça,de,Vaca", "Cabral", "Cabreira",  
 263. "Cachão", "Cachoeira", "Caçoilo", "Cadaval", "Cadavez", "Caeira"  
 , "Caeiro", "Caetano", "Café", "Caiado",  
 264. "Caires", "Calado", "Calçada", "Caldas", "Caldeira", "Calheiros"  
 , "Camacho", "Câmara", "Camargo",  
 265. "Camarinho", "Cambezes", "Camelo", "Camilo", "Caminha", "Campelo  
 ", "Campos", "Canadas", "Canário",  
 266. "Cancela", "Candal", "Candeias", "Canedo", "Caneira", "Canejo",  
 "Canela", "Cangueiro", "Canhão",  
 267. "Caníça", "Cantanhede", "Canto", "Caparica", "Capistrano", "Capu  
 cho", "Cardim", "Cardoso", "Carlos",  
 268. "Carmo", "Carmona", "Carneiro", "Coronel", "Carqueijeiro", "Carr  
 asco", "Carrasqueira", "Carreira",  
 269. "Carregueiro", "Carreiro", "Carrilho", "Carromeu", "Cartaxo", "C  
 arvalhais", "Carvalhal", "Carvalheira",  
 270. "Carvalheiro", "Carvalho", "Carvalhosa", "Carvalhoso", "Casado",  
 "Casalinho", "Cascais", "Casqueira",  
 271. "Castanheira", "Castanho", "Castanheda", "Castelhano", "Castelo"  
 , "Castelo,Branco", "Castilho",  
 272. "Castilhos", "Castro", "Catela", "Cavadas", "Cavaco", "Cavalheir  
 o", "Cedraz", "Cedro", "Cerqueira",  
 273. "Cerveira", "César", "Cesário", "Chagas", "Chainho", "Chamusca",  
 "Charneca", "Chaves", "Chousa",



274. "Cidreira", "Cipriano", "Cirne", "Cisneiros", "Clementino", "Cobra", "Coelho", "Coimbra", "Colaço",  
 275. "Colares", "Conceição", "Conde", "Condorcet", "Cordeiro", "Correia", "Corte-Real", "Cortês", "Cortesão",  
 276. "Costa, da Costa", "Cotrim", "Couceiro", "Coutinho", "Couto", "Covilhã", "Covinha", "Craveiro", "Cruz",  
 277. "Cunha", "Curvelo", "Custódio", "Curado", "Damasceno", "Damásio", "Dâmaso", "Dantas", "Delgado", "Dias",  
 278. "Diegues", "Dinis", "Dourado", "Doutel", "Doutis", "Domingos", "Domingues", "Dorneles", "Duarte",  
 279. "Durão", "Durães", "Dutra", "Eanes", "Eiró", "Escobar", "Espargosa", "Esparteiro", "Espinosa",  
 280. "Espírito Santo", "Esteves", "Estrada", "Estrela", "Faia", "Fagundes", "Falcão", "Faleiro", "Faria",  
 281. "Farias", "Farinha", "Faro", "Fartaria", "Faustino", "Fazendeiro", "Feijó", "Feira", "Feitosa", "Félix",  
 282. "Fernandes", "Ferraço", "Ferrão", "Ferraz", "Ferreira", "Ferro", "Festas", "Fiães", "Fialho", "Fidalgo",  
 283. "Figueira", "Figueiredo", "Figueiró", "Figueiroa", "Filgueiras", "Filipe", "Fitas", "Fiúza", "Flávio",  
 284. "Flores", "Fogaça", "Fonseca", "Fontes", "Fontinha", "Fontoura", "Foquiço", "Fortes", "Fortunato",  
 285. "Frade", "Fraga", "Fragoso", "Frajuca", "Franca", "França", "Francisco", "Franca", "Franqueira", "Frazão",  
 286. "Freire", "Freiria", "Freitas", "Freixo", "Frias", "Frois", "Frota", "Furquim", "Furtado", "Gabeira",  
 287. "Gadelha", "Gago", "Galante", "Galindo", "Galvão", "Gama", "Gameiro", "Garcez", "Garcia", "Garrau",  
 288. "Garrido", "Gaspar", "Gentil", "Gil", "Ginjeira", "Girão", "Godinho", "Godoi", "Góis", "Gomes",  
 289. "Gomide", "Gonçalves", "Gorjão", "Goulart", "Gouveia", "Goulão", "Graça", "Grande", "Grangeia",  
 290. "Granja", "Granjeiro", "Gravato", "Grilo", "Guedelha", "Guedes", "Guerra", "Guerreiro", "Guilheiro",  
 291. "Guimarães", "Gusmão", "Guterres", "Henriques", "Hernandes", "Hilário", "Hipólito", "Holanda", "Homem",  
 292. "Horta", "Igrejas", "Ilha", "Imperial", "Inácio", "Inês", "Infante", "Jamandas", "Janes", "Jardim",  
 293. "Jesus", "Jobim", "Jordão", "Jorge", "Júdice", "Junqueira", "Keil", "Lacerda", "Lage", "Lages", "Lago",  
 294. "Lagoa", "Lagos", "Lalanda", "Lamego", "Lameira", "Lameirinhas", "Lameiras", "Lamenha", "Lampreia",  
 295. "Lancastre", "Landim", "Lara", "Laranjeira", "Lários", "Laureano", "Leal", "Leão", "Leiria", "Leitão",  
 296. "Leite", "Leme", "Lemes", "Lemos", "Lessa", "Letras", "Liberato", "Lima", "Linhares", "Lindim", "Lins",  
 297. "Lira", "Lisboa", "Lobato", "Lobo", "Loio", "Lopes", "Lourenço", "Loureiro", "Lousã", "Lousada",  
 298. "Lousado", "Lucas", "Lucena", "Lustosa", "Luz", "Macedo", "Macena", "Machado", "Macieira", "Maciel",  
 299. "Madeira", "Madruga", "Madureira", "Mafra", "Magalhães", "Maia", "Mainha", "Maior", "Malafaia",  
 300. "Malheiro", "Malho", "Malta", "Mamouros", "Mangueira", "Mansilha", "Manso", "Mantas", "Maranhão",  
 301. "Marçal", "Marcondes", "Marinho", "Marins", "Mariz", "Marmou", "Marques", "Marreiro", "Marroquim",  
 302. "Martinho", "Martins", "Mascarenhas", "Mata", "Mateus", "Matias", "Matos", "Matosinhos", "Matoso",  
 303. "Medeiros", "Medina", "Meira", "Meireles", "Melancia", "Melgaço", "Mena", "Melo", "Mendes", "Mendonça",

304. "Meneses", "Mesquita", "Mexia", "Miguel", "Miguéis", "Mieiro", "Milhões", "Milheirão", "Milheiriço",  
305. "Milheiro", "Minho", "Miranda", "Mirandela", "Modesto", "Moita", "Mondragão", "Monforte", "Monjardim",  
306. "Monsanto", "Mont Alverne", "Monte", "Montenegro", "Monteiro", "Morais", "Morão", "Moreira", "Moreno",  
307. "Morgado", "Mortágua", "Mota", "Moura", "Mourão", "Mourato", "Mourinho", "Moutinho", "Muniz",  
308. "Murteira", "Murtinho", "Muxagata", "Narvais", "Nascimento", "Natal", "Naves", "Nazário", "Negrão",  
309. "Negreiros", "Negromonte", "Neiva", "Neres", "Neto", "Neves", "Ninharelhos", "Nobre", "Nóbrega",  
310. "Nogueira", "Noite", "Nolasco", "Noieto", "Norões", "Noronha", "Novais", "Nunes", "Olaio", "Oleiro",  
311. "Olivares", "Oliveira", "Onofre", "Ornelas", "Orriça", "Osório", "Ourique", "Ouro", "Outeiro",  
312. "Pacheco", "Padilha", "Pádua", "Paião", "Pais", "Paiva", "Paixão", "Palha", "Palhares", "Palma",  
313. "Palmeira", "Palos", "Parafita", "Paranhos", "Pardo", "Paredes", "Parente", "Parracho", "Parreir",  
314. "Passarinho", "Passos", "Pastana", "Patrício", "Paula", "Paulos", "Paz", "Peçanha", "Pêcego",  
315. "Pederneiras", "Pedro", "Pedroso", "Pegado", "Peixoto", "Penha", "Penteado", "Pequeno", "Peralta",  
316. "Perdigão", "Pereira", "Pescada", "Peseiro", "Pessoa", "Pestana", "Picanço", "Picão", "Pimenta",  
317. "Pimentel", "Pinhal", "Pinheiro", "Pinho", "Pinto", "Piteira", "Pires", "Poças", "Ponte", "Pontes",  
318. "Porciúncula", "Portela", "Porto", "Portugal", "Póvoas", "Prada", "Prado", "Prates", "Prestes",  
319. "Proença", "Protásio", "Prudente", "Pureza", "Quadros", "Quaresma", "Queiroga", "Queirós", "Quental",  
320. "Quesado", "Quina", "Quinaz", "Quinta", "Quintal", "Quintana", "Quintanilha", "Quintas", "Quintais",  
321. "Quintão", "Quinteiro", "Quintela", "Quinterno", "Quinzeiro", "Quirino", "Rabelo", "Ramalho",  
322. "Raminhos", "Ramires", "Ramos", "Rangel", "Raposo", "Rebelo", "Rebimbas", "Rebocho", "Rebotim",  
323. "Rebouças", "Redondo", "Regalado", "Rego", "Regodeiro", "Regueira", "Rei", "Reino", "Reis", "Remígio",  
324. "Resende", "Ribas", "Ribeiro", "Rico", "Rijo", "Rios", "Robalinho", "Robalo", "Roçadas", "Rocha",  
325. "Rodovalho", "Rodrigues", "Rolim", "Roriz", "Rosa", "Rosado", "Rosário", "Rosmaninho", "Rua", "Ruas",  
326. "Ruela", "Rufino", "Sá", "Sabala", "Sabrosa", "Sacadura", "Sacramento", "Salazar", "Saldanha", "Sales",  
327. "Salgado", "Salgueiro", "Salvado", "Saloio", "Salomão", "Saltão", "Sampaio", "Sanches", "Sandinha",  
328. "Santana", "Santarém", "Santiago", "Santos", "Saraiva", "Sardinha", "Sardo", "Sarmiento", "Seabra",  
329. "Seixas", "Semedo", "Serpa", "Serralheiro", "Serro", "Sesimbra", "Setúbal", "Severiano", "Severo",  
330. "Silva", "Silveira", "Silvestre", "Simas", "Simões", "Simão", "Sinha", "Sítima", "Sequeira", "Soares",  
331. "Sobral", "Sobreira", "Sodré", "Soeiro", "Sousa", "Souto", "Souto, Maior", "Soveral", "Soverosa",  
332. "Tabanez", "Taborda", "Tabosa", "Talhão", "Tavares", "Taveira", "Taveiros", "Távora", "Teixeira",  
333. "Tedim", "Teles", "Telinhos", "Temes", "Teodoro", "Terra", "Teves", "Tigre", "Tinoco", "Toledo", "Tomé",

```

334.         "Torquato", "Torrado", "Torreiro", "Torres", "Toscano", "Travass
os", "Toste", "Trigueiro", "Trindade",
335.         "Tristão", "Tuna", "Uchoa", "Ulhoa", "Úria", "Urias", "Valadão",
"Valadares", "Valadim", "Valcácer",
336.         "Valcanaia", "Vale", "Valente", "Valentim", "Valério", "Valgueir
o", "Valido", "Valim", "Valverde",
337.         "Varanda", "Varão", "Varejão", "Varela", "Vargas", "Vasconcelos"
, "Vasques", "Vaz", "Veiga", "Velasco",
338.         "Velasques", "Veleda", "Veloso", "Ventura", "Vergueiro", "Veriss
imo", "Viana", "Vidal", "Vides",
339.         "Vidigal", "Viegas", "Vieira", "Vigário", "Vila-
Chã", "Vilaça", "Vilalobos", "Vilanova", "Vilante",
340.         "Vilar", "Vilarica", "Vilarinho", "Vilas,Boas", "Vilaverde", "Vi
lela", "Vilhena", "Vinhas", "Vital",
341.         "Viveiros", "Xavier", "Ximenes", "Xisto", "Zagalo", "Zambujal",
"Zarco"]
342.     r = random.randint(0, len(apelidos) - 1)
343.     return str(apelidos[r])
344.
345.
346.     def geraradress():
347.         lista = ['Rua de', 'Avenida de', 'Praca', 'Travessa']
348.         r = random.randint(0, len(lista) - 1)
349.         return "{} {} n{}".format(lista[r], gerarapelidos(), random.randint(1, 50))
350.
351.
352.     if __name__ == '__main__':
353.         i = 2000000000
354.
355.         with open('pessoa.sql', 'w') as f:
356.             for h in range(1667):
357.                 f.write(
358.                     "INSERT INTO pessoa VALUES ({},'{} {}', '{}', {}, '{}', {});\n".
format(i, gerarnomepM(), gerarapelidos(),
359.                     "Masculino", geraridade(),
360.                     geraradress(),random.randint(1,308)))
361.                 i= i +1
362.                 f.write(
363.                     "INSERT INTO pessoa VALUES ({},'{} {} {}', '{}', {}, '{}',{});\n
".format(i, gerarnomepM(), gerarapelidos(),gerarapelidos(),
364.                     "Masculino", geraridade(),
365.                     geraradress(),random.randint(1,308)))
366.                 i= i +1
367.
368.                 f.write(
369.                     "INSERT INTO pessoa VALUES ({},'{} {} {} {}', '{}', {}, '{}', {}
);\n".format(i, gerarnomepM(), gerarnomepM(), gerarapelidos(),
370.                     gerarapelidos(),
371.                     "Masculino", geraridade(),
372.                     geraradress(),random.randint(1,308)))
373.                 i= i +1
374.                 f.write(

```

```

375.             "INSERT INTO pessoa VALUES ({},'{} {}', '{}', {}, '{}', {});\n".
format(i, gerarnomepF(), gerarapelidos(),
376.     "Femenino", geraridade(),
377.     geraraddress(),random.randint(1,308)))
378.         i= i +1
379.         f.write(
380.             "INSERT INTO pessoa VALUES ({},'{} {} {} {}', '{}', {}, '{}', {});\n
".format(i, gerarnomepF(), gerarapelidos(),gerarapelidos(),
381.     "Femenino", geraridade(),
382.     geraraddress(),random.randint(1,308)))
383.         i= i +1
384.         f.write(
385.             "INSERT INTO pessoa VALUES ({},'{} {} {} {} {}', '{}', {}, '{}', {}
);\n".format(i, gerarnomepF(), gerarnomepF(), gerarapelidos(),
386.     gerarapelidos(),
387.     "Femenino", geraridade(),geraraddress(),random.randint(1,308)))
388.         i = i +1
389.         f.close()
390.

```

```

1. # cria um sql que contem todas as regiões e os seus hab
2. regioao = ['Região Norte' , 'Região de Lisboa' , 'Região Centro' , 'Região do Alentejo' , '
Região do Algarve'
3.     , 'Região Autónoma da Madeira' , 'Região Autónoma dos Açores']
4.
5. populacao = ['3818722' , '2808414' , '2348453' , '776339' , '395208' , '245012' , '244006']
6.
7. if __name__ == '__main__':
8.
9.     k = 1
10.     with open('regiao.sql', 'w') as f:
11.         for i in range(len(regiao) - 1):
12.             f.write("INSERT INTO REGIAO VALUES ({}, '{}', {});\n".format(k, regioao[i],
populacao[i]))
13.             k = k + 1
14.     f.close()

```

```

1. # cria um sql que contem todas os sintomas do covid 19
2. sintomas = ['Febre' , 'Tosse seca' , 'Cansaço' , 'Dor de cabeça' , 'Perda de gosto e olfato'
, 'Garganta inflamada' ,
3.     'Nariz entupido' , 'Tensao e dores musculares' , 'Diarreia' , 'Dificuldades Re
spiratorias' , 'Pressao ou dor no peito']
4.
5. if __name__ == '__main__':
6.     with open('sintomas.sql', 'w') as f:
7.         k = 1
8.         for i in range(len(sintomas)-1):
9.             f.write('INSERT INTO sintoma VALUES ({}, "{}")\n'.format(k,sintomas[i]))

```

```
10.         k += 1
11.     f.close()
```

## 12.3 ANEXO 3 – EXCEL

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1					60%	30%	40%	20%	30%							
2	1	febre		1	7				14		*random com os vários tipo de motivos					
3	2	tosse		2	12						*varia em cada alteração na planilha					
4	3	cansaço		3		14	5		3							
5	4	perda olfato		4	5			4	2							
6	5	perda paladar		5	10	13										
7	6	Tensão e dores musculares		6	5	2										
8	7	dores garganta		7	13			3								
9	8	diarreia		8	10		7	13								
10	9	conjuntivite		9			4									
11	10	dor de cabeça		10												
12	11	irritação pela		11	3		9									
13	12	difficuldade respiratória ou falta de ar		12	3											
14	13	pressao ou dor no peito;		13				9								
15	13	pressao ou dor no peito;		14	7											
16	14	sem sintomas		15			10									
17				16	2	4			14							
18				17				3								
19				18	4	9	9		9							
20				19												
21				20												
22				21	7				4							
23				22												
24				23	8		6		1							
25				24	3	8		8								
26				25	12		13		7							
27				26			13									
28				27			14	14								
29				28	9											
30				29												
31				30	8											
32				31	7		9		14							

[illegible]

E1220		INSERT INTO tiposPositivo VALUES												
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id_tipo_positi	nome_tipo_positi	id_resulta											
1220	1219	RECUPERADO	5		INSERT INTO tiposPositivo (1219,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1219,'RECUPERADO');	5;							
1221	1220	RECUPERADO	12		INSERT INTO tiposPositivo (1220,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1220,'RECUPERADO');	12;							
1222	1221	RECUPERADO	13		INSERT INTO tiposPositivo (1221,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1221,'RECUPERADO');	13;							
1223	1222	RECUPERADO	14		INSERT INTO tiposPositivo (1222,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1222,'RECUPERADO');	14;							
1225	1224	RECUPERADO	22		INSERT INTO tiposPositivo (1224,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1224,'RECUPERADO');	22;							
1226	1225	RECUPERADO	29		INSERT INTO tiposPositivo (1225,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1225,'RECUPERADO');	29;							
1227	1226	RECUPERADO	32		INSERT INTO tiposPositivo (1226,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1226,'RECUPERADO');	32;							
1229	1228	RECUPERADO	36		INSERT INTO tiposPositivo (1228,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1228,'RECUPERADO');	36;							
1230	1229	RECUPERADO	40		INSERT INTO tiposPositivo (1229,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1229,'RECUPERADO');	40;							
1236	1235	RECUPERADO	83		INSERT INTO tiposPositivo (1235,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1235,'RECUPERADO');	83;							
1237	1236	RECUPERADO	89		INSERT INTO tiposPositivo (1236,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1236,'RECUPERADO');	89;							
1244	1383	RECUPERADO	152		INSERT INTO tiposPositivo (1383,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1383,'RECUPERADO');	152;							
1246	1245	RECUPERADO	159		INSERT INTO tiposPositivo (1245,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1245,'RECUPERADO');	159;							
1248	1247	RECUPERADO	163		INSERT INTO tiposPositivo (1247,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1247,'RECUPERADO');	163;							
1250	1249	RECUPERADO	171		INSERT INTO tiposPositivo (1249,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1249,'RECUPERADO');	171;							
1251	1250	RECUPERADO	179		INSERT INTO tiposPositivo (1250,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1250,'RECUPERADO');	179;							
1253	1252	RECUPERADO	202		INSERT INTO tiposPositivo (1252,'RECUPERADO')	INSERT INTO tiposPositivo VALUES (1252,'RECUPERADO');	202;							

## 12.4 ANEXO 4 – STORED PROCEDURE

--Mostra o valor de incidência do concelho selecionado

```
Create PROCEDURE [dbo].[Valordeincidencia]
@concelho VARCHAR (30)
AS
SELECT CONVERT (DECIMAL (5, 0), SUM(totals.conf) * 1.0 /
SUM([Utente].[concelho].quantHabitantes) * 100000) AS Valor_de_incidencia
FROM [Utente].[concelho], (SELECT count(*) AS conf
FROM Resultado.tiposPositivo, Resultado.resultado,
Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
WHERE Resultado.resultado.idResultado =
Resultado.tiposPositivo.idResultado
AND Resultado.resultado.idResultado =
Notificacao.pessoaNotificacaoResultado.idResultado
AND Notificacao.pessoaNotificacaoResultado.idPessoa
= Utente.pessoa.idPessoa
AND Utente.pessoa.idConcelho =
Utente.concelho.idConcelho
AND Utente.concelho.nomeConcelho = @concelho
AND resultado.Resultado.nomeResultado = 'POSITIVO')
COLLATE Latin1_General_CI_AI
AS totals
WHERE [Utente].[concelho].nomeConcelho = @concelho COLLATE Latin1_General_CI_AI
GROUP BY totals.conf;
```

--Devolve o numero de recuperados por idade e género

```
Create PROCEDURE [dbo].[UtenteRecuperadosGeneroIdade]
@genero VARCHAR (10), @idade1 INT, @idade2 INT
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Utente.pessoa.idade > @idade1
            AND Utente.pessoa.idade < @idade2
            AND Utente.pessoa.genero = @genero
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO' COLLATE
Latin1_General_CI_AI;
END
```

--devolve os positivos por idade e género

```
Create PROCEDURE [dbo].[UtentePositivoGeneroIdade]
@genero VARCHAR (10), @idade1 INT, @idade2 INT
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Utente.pessoa.idade > @idade1
            AND Utente.pessoa.idade < @idade2
            AND Utente.pessoa.genero = @genero COLLATE Latin1_General_CI_AI
            AND Resultado.nomeResultado = 'POSITIVO';
END
```

--mostra o resultado da pessoa pelo o nif

```
Create PROCEDURE [dbo].[UtentePessoaNifCases]
@Utentepessoanif INT
AS
BEGIN
    SELECT Utente.pessoa.nif,
           Utente.pessoa.nomePessoa,
           Resultado.nomeResultado
    FROM    Utente.pessoa, Notificacao.pessoaNotificacaoResultado, Resultado.resultado
    WHERE   Utente.pessoa.idPessoa = Notificacao.pessoaNotificacaoResultado.idPessoa
```

```

        AND Notificacao.pessoaNotificacaoResultado.idResultado =
Resultado.resultado.idResultado
        AND Utente.pessoa.nif = @Utentepessoanif;
END

-- devolve os positivos por género e concelho

Create PROCEDURE [dbo].[UtentePessoaGeneroCases]
@Utentepessoagenero VARCHAR (10)
AS
BEGIN
    SELECT count(*) AS Total
    FROM Resultado.tiposPositivo, Resultado.resultado,
Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
        AND Resultado.resultado.idResultado =
Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND Resultado.nomeResultado = 'Positivo'
        AND Utente.pessoa.genero = @Utentepessoagenero;
END

--permite inserir um campo encripted

CREATE PROCEDURE [dbo].[UtentePessoaEncripted]
@idPessoa SMALLINT, @nif INT, @nomePessoa VARCHAR (100), @genero VARCHAR (10), @idade
INT, @morada VARCHAR (100), @idConcelho SMALLINT
AS
BEGIN
    INSERT INTO [Utente].[pessoa] ([idPessoa], [nif], [nomePessoa], [genero], [idade],
[morada], [idConcelho])
    VALUES (@idPessoa, @nif, @nomePessoa, @genero, @idade,
@morada, @idConcelho);
END

-- devolve os obitos por género e idade

CREATE PROCEDURE [dbo].[UtenteObitosGeneroIdade]
@genero VARCHAR (10), @idade1 INT, @idade2 INT
AS
BEGIN
    SELECT count(*) AS Total
    FROM Resultado.tiposPositivo, Resultado.resultado,
Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
        AND Resultado.resultado.idResultado =
Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND Utente.pessoa.idade > @idade1
        AND Utente.pessoa.idade < @idade2
        AND Utente.pessoa.genero = @genero
        AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO';
END

```



--mostra os óbitos por idade

```
CREATE PROCEDURE [dbo].[UtenteIdadeObitos]
@idade1 INT, @idade2 INT
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO'
            AND Utente.pessoa.idade > @idade1
            AND Utente.pessoa.idade < @idade2;
END
```

--devolve o total de recuperados por concelho

```
CREATE PROCEDURE [dbo].[UtenteConselhorRecuperados]
@conselho VARCHAR (50)
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO' COLLATE
    Latin1_General_CI_AI
            AND Utente.pessoa.genero = @conselho;
END
```

-- devolve o total de recuperados por concelho e pela data

```
CREATE PROCEDURE [dbo].[UtenteConselhorNewRecuperados]
@conselho VARCHAR (50), @dataResultado DATE
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Utente.concelho, Resultado.tiposPositivo
    WHERE   Resultado.tiposPositivo.idResultado = Resultado.resultado.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
```

```

        AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO'
        AND Utente.concelho.nomeConcelho = @conselho COLLATE Latin1_General_CI_AI
        AND dataResultado LIKE @dataResultado;
END

--devolve os óbitos por data e concelho

ALTER PROCEDURE [dbo].[UtenteConselhorNewÓbitos]
@conselho VARCHAR (50), @dataResultado DATE
AS
BEGIN
    SELECT count(*) AS Total
    FROM Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Utente.concelho, Resultado.tiposPositivo
    WHERE Resultado.tiposPositivo.idResultado = Resultado.resultado.idResultado
        AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO'
        AND Utente.concelho.nomeConcelho = @conselho COLLATE Latin1_General_CI_AI
        AND dataResultado LIKE @dataResultado;
END

-- devolve os óbitos por conselho

CREATE PROCEDURE [dbo].[UtenteConselhoObitos]
@conselho VARCHAR (50)
AS
BEGIN
    SELECT count(*) AS Total
    FROM Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
        AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO'
        AND Utente.pessoa.genero = @conselho;
END

--Mostra o total de ativos por concelho

CREATE PROCEDURE [dbo].[UtenteconselhoCaseAtivos]
@UtenteconcelhonomeConcelho VARCHAR (50)
AS
BEGIN
    SELECT concelho.nomeConcelho AS localidade,
        count(*) AS Ativos
    FROM Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
        AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND nomeTiposPositivo = 'ATIVO'

```

```

        AND Utente.concelho.nomeConcelho LIKE @UtenteconcelhonomeConcelho COLLATE
Latin1_General_CI_AI
    GROUP BY nomeConcelho;
END

```

--devolve os positivos por idade

```

CREATE PROCEDURE [dbo].[UtendeidadeCases]
@idade1 INT, @idade2 INT
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Resultado.nomeResultado = 'Positivo'
            AND Utente.pessoa.idade > @idade1
            AND Utente.pessoa.idade < @idade2;
END

```

--Devolve o Recuperados

```

CREATE PROCEDURE [dbo].[TotalRecuperados]
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO' COLLATE
Latin1_General_CI_AI;
END

```

--Devolve os positivos por regiões

```

CREATE PROCEDURE [dbo].[TotalPositivoRegiao]
AS
BEGIN
    SELECT    count(*) AS Total,
            nomeRegiao
    FROM      Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Utente.concelho, Utente.regiao
    WHERE     Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
            AND Utente.concelho.idRegiao = Utente.regiao.idRegiao
            AND Resultado.nomeResultado = 'POSITIVO'

```

```

        GROUP BY nomeRegiao;
END

--Devolve o total de ÓBITOS

CREATE PROCEDURE [dbo].[TotalObitos]
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.tiposPositivo, Resultado.resultado,
    Notificacao.pessoaNotificacaoResultado, Utente.pessoa
    WHERE   Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO';
END

--Devolve recuperados por data

CREATE PROCEDURE [dbo].[TotalNewRecuperados]
@dataResultado DATE
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Resultado.tiposPositivo
    WHERE   Resultado.tiposPositivo.idResultado = Resultado.resultado.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO'
            AND dataResultado LIKE @dataResultado;
END

--devolve os óbitos por data

CREATE PROCEDURE [dbo].[TotalNewÓbitos]
@dataResultado DATE
AS
BEGIN
    SELECT count(*) AS Total
    FROM    Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Resultado.tiposPositivo
    WHERE   Resultado.tiposPositivo.idResultado = Resultado.resultado.idResultado
            AND Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
            AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
            AND Resultado.tiposPositivo.nomeTiposPositivo = 'ÓBITO'
            AND dataResultado LIKE @dataResultado;
END

--devolve o total de positivos por data

CREATE PROCEDURE [dbo].[TotalNewCases]
@dataResultado DATE

```

```

AS
BEGIN
    SELECT count(*) AS Total
    FROM [Resultado].[resultado]
    WHERE Resultado.nomeResultado = 'POSITIVO'
        AND dataResultado LIKE @dataResultado;
END

--Devolve o total de ativos

ALTER PROCEDURE [dbo].[TotalCasosAtivos]
AS
SELECT t.Ativos - count(f.idPessoa)
FROM (SELECT Utente.pessoa.idPessoa
      FROM Utente.pessoa, Notificacao.pessoaNotificacaoResultado,
Resultado.resultado, Resultado.tiposPositivo
      WHERE Utente.pessoa.idPessoa =
Notificacao.pessoaNotificacaoResultado.idPessoa
          AND Notificacao.pessoaNotificacaoResultado.idResultado =
Resultado.resultado.idResultado
          AND Resultado.resultado.idResultado =
Resultado.tiposPositivo.idResultado
          AND Resultado.resultado.nomeResultado = 'POSITIVO'
      GROUP BY Utente.pessoa.idPessoa
      HAVING Count(Utente.pessoa.idPessoa) > 1) AS f, (SELECT count(*) AS Ativos
FROM
Resultado.tiposPositivo
WHERE
Resultado.tiposPositivo.nomeTiposPositivo = 'ATIVO') AS t
GROUP BY t.Ativos;

--devolve o o estado do concelho referente a incidencia

CREATE PROCEDURE [dbo].[RiscoporConselho]
@concelho VARCHAR (30)
AS
DECLARE @tconf AS INT;
DECLARE @tesse AS INT;
SELECT @tesse = count(*)
FROM Resultado.tiposPositivo, Resultado.resultado,
Notificacao.pessoaNotificacaoResultado, Utente.pessoa, Utente.concelho
WHERE Resultado.resultado.idResultado = Resultado.tiposPositivo.idResultado
    AND Resultado.resultado.idResultado =
Notificacao.pessoaNotificacaoResultado.idResultado
    AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
    AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
    AND Utente.concelho.nomeConcelho = @concelho COLLATE Latin1_General_CI_AI
    AND resultado.Resultado.nomeResultado = 'POSITIVO';
SELECT @tconf = CONVERT (DECIMAL (5, 0), SUM(@tesse) * 1.0 /
SUM([Utente].[concelho].quantHabitantes) * 100000)
FROM [Utente].[concelho]
WHERE [Utente].[concelho].nomeConcelho = @concelho COLLATE Latin1_General_CI_AI;
IF @tconf < 240
    SELECT 'Moderado' AS Resultado;
ELSE
    IF @tconf < 480

```

```

        AND @tconf > 240
        SELECT 'Risco Elevado' AS Resultado;
ELSE
    IF @tconf > 480
        AND @tconf < 960
        SELECT 'Risco Muito Elevado' AS Resultado;
ELSE
    IF @tconf > 960
        SELECT 'Extremamente Elevado' AS Resultado;

--Devolve os positivos que o motivo foi contacto

CREATE PROCEDURE [dbo].[ResultadoTotalPositivoContacto]
AS
BEGIN
    SELECT count(*) AS Total
    FROM [Notificacao].[pessoaNotificacaoResultado], [Notificacao].[notificacao],
Resultado.resultado
    WHERE [Notificacao].[pessoaNotificacaoResultado].idNotificacao =
[Notificacao].[notificacao].idNotificacao
        AND [Notificacao].[pessoaNotificacaoResultado].idNotificacao =
Resultado.resultado.idResultado
        AND resultado.nomeResultado = 'POSITIVO'
        AND Notificacao.notificacao.idMotivo = 3;
END

--devolve os positivos

CREATE PROCEDURE [dbo].[ResultadoresultadoTotalCasosConfirmados]
AS
BEGIN
    SELECT count(*) AS Total
    FROM [Resultado].[resultado]
    WHERE Resultado.nomeResultado = 'POSITIVO';
END

--devolve os recuperados por concelho e por data

CREATE PROCEDURE [dbo].[ResultadoNewRecuperadosConselho]
@conselho VARCHAR (50), @dataResultado DATE
AS
BEGIN
    SELECT count(*) AS Total
    FROM Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
Utente.concelho, Resultado.tiposPositivo
    WHERE Resultado.tiposPositivo.idResultado = Resultado.resultado.idResultado
        AND Resultado.resultado.idResultado =
Notificacao.pessoaNotificacaoResultado.idResultado
        AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
        AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
        AND Resultado.tiposPositivo.nomeTiposPositivo = 'RECUPERADO'
        AND Utente.concelho.nomeConcelho = @conselho COLLATE Latin1_General_CI_AI

```

```

        AND dataResultado LIKE @dataResultado;
END

--devolve os positivos por concelho e data

CREATE PROCEDURE [dbo].[ResultadoNewCasesConselho]
@conselho VARCHAR (50), @dataResultado DATE
AS
BEGIN
    SELECT *
    FROM Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Utente.concelho
    WHERE Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
    AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
    AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
    AND Resultado.resultado.nomeResultado = 'POSITIVO'
    AND Utente.concelho.nomeConcelho = @conselho COLLATE Latin1_General_CI_AI
    AND dataResultado LIKE @dataResultado;
END

--devolve os positivos por concelho

CREATE PROCEDURE [dbo].[ResultadoConselhoCasosConfirmados]
@conselho VARCHAR (50)
AS
BEGIN
    SELECT count(*) AS Ativos
    FROM Resultado.resultado, Notificacao.pessoaNotificacaoResultado, Utente.pessoa,
    Utente.concelho
    WHERE Resultado.resultado.idResultado =
    Notificacao.pessoaNotificacaoResultado.idResultado
    AND Notificacao.pessoaNotificacaoResultado.idPessoa = Utente.pessoa.idPessoa
    AND Utente.pessoa.idConcelho = Utente.concelho.idConcelho
    AND Resultado.resultado.nomeResultado = 'POSITIVO'
    AND Utente.concelho.nomeConcelho LIKE @conselho COLLATE Latin1_General_CI_AI
    GROUP BY nomeConcelho;
END

--quando a população chegar a 1%,20%,50%,75% de infetados envia um mail e printa na consola

ALTER PROCEDURE [dbo].[porcpoo]
AS
BEGIN
    DECLARE @thabit AS INT;
    DECLARE @tconf AS INT;
    SELECT @tconf = COUNT(*)
    FROM [ProjetoBasedeDados2].[Resultado].[resultado]
    WHERE nomeResultado = 'POSITIVO';
    SELECT @thabit = sum(Utente.regiao.quantHabitantes)
    FROM Utente.regiao;
END
IF (@thabit / 100 = @tconf)
BEGIN
    PRINT 'Chegou a 1% da população infetada';

```

```

EXECUTE msdb.dbo.sp_send_dbmail @profile_name = 'EmailTP2', @recipients =
'Ivanvieig14@hotmail.com', @body = 'AVISO

!! A BASE DADOS DETECTOU QUE ULTRAPASSOU 1% DA POPULAÇÃO !!', @subject = 'Automated
Success Message';
END
ELSE
IF (@thabit * 20 / 100 = @tconf)
BEGIN
PRINT 'Chegou a 20% da população infetada';
EXECUTE msdb.dbo.sp_send_dbmail @profile_name = 'EmailTP2', @recipients =
'Ivanvieig14@hotmail.com', @body = 'AVISO

!! A BASE DADOS DETECTOU QUE ULTRAPASSOU 20% DA POPULAÇÃO !!', @subject = 'Automated
Success Message';
END
ELSE
IF (@thabit * 50 / 100 = @tconf)
BEGIN
PRINT 'Chegou a 50% da população infetada';
EXECUTE msdb.dbo.sp_send_dbmail @profile_name = 'EmailTP2', @recipients =
'Ivanvieig14@hotmail.com', @body = 'AVISO

!! A BASE DADOS DETECTOU QUE ULTRAPASSOU 50% DA POPULAÇÃO !!', @subject = 'Automated
Success Message';
END
ELSE
IF (@thabit * 75 / 100 = @tconf)
BEGIN
PRINT 'Chegou a 75% da população infetada';
EXECUTE msdb.dbo.sp_send_dbmail @profile_name = 'EmailTP2',
@recipients = 'Ivanvieig14@hotmail.com', @body = 'AVISO

!! A BASE DADOS DETECTOU QUE ULTRAPASSOU 75% DA POPULAÇÃO !!', @subject = 'Automated
Success Message';
END

-- faz o índice rt
CREATE PROCEDURE [dbo].[IndiceRT]
AS
BEGIN
SELECT sum(med.t) * 1.0 / COUNT(med.t)
FROM (SELECT count(idContacto) AS t
FROM Notificacao.contacto
WHERE Notificacao.contacto.idContacto <> 10002
GROUP BY contacto.idContacto) AS med;
END

```



## 12.5 ANEXO 5 – SEGURANÇA

```
1.  --CRIAÇÃO LOGIN E USUÁRIO
2.
3.  CREATE LOGIN UserCrud WITH PASSWORD = '123';
4.
5.  CREATE LOGIN UserRead WITH PASSWORD = '123';
6.
7.  CREATE LOGIN UserAdm WITH PASSWORD = '123';
8.
9.  CREATE LOGIN UserJrDb WITH PASSWORD = '123';
10.
11. USE ProjetoBasedeDados2;
12. GO
13.
14. CREATE USER UserJrDb FOR LOGIN [UserJrDb];
15.
16. USE ProjetoBasedeDados2;
17. GO
18.
19. CREATE USER UserCrud FOR LOGIN [UserCrud];
20.
21. USE ProjetoBasedeDados2;
22. GO
23.
24. CREATE USER UserRead FOR LOGIN [UserRead];
25.
26. USE ProjetoBasedeDados2;
27. GO
28.
29. CREATE USER UserAdm FOR LOGIN [UserAdm];
30.
31. USE ProjetoBasedeDados2;
32. GO
33.
34.
35. --PERMISSOES SELECT PARA USERCRUD E P USEREAD
36.
37. GRANT SELECT ON [Notificacao].[contacto] TO [UserCrud]
38. GRANT SELECT ON [Notificacao].[contacto] TO [UserRead]
39. GRANT SELECT ON [Notificacao].[motivo] TO [UserCrud]
40. GRANT SELECT ON [Notificacao].[motivo] TO [UserRead]
41. GRANT SELECT ON [Notificacao].[notificacao] TO [UserCrud]
42. GRANT SELECT ON [Notificacao].[notificacao] TO [UserRead]
43. GRANT SELECT ON [Notificacao].[notificacaoSintomas] TO [UserCrud]
44. GRANT SELECT ON [Notificacao].[notificacaoSintomas] TO [UserRead]
45. GRANT SELECT ON [Notificacao].[pessoaNotificacaoResultado] TO [UserCrud]
46. GRANT SELECT ON [Notificacao].[pessoaNotificacaoResultado] TO [UserRead]
47. GRANT SELECT ON [Notificacao].[sintomas] TO [UserCrud]
48. GRANT SELECT ON [Notificacao].[sintomas] TO [UserRead]
49. GRANT SELECT ON [Resultado].[resultado] TO [UserCrud]
50. GRANT SELECT ON [Resultado].[resultado] TO [UserRead]
51. GRANT SELECT ON [Resultado].[tiposPositivo] TO [UserCrud]
52. GRANT SELECT ON [Resultado].[tiposPositivo] TO [UserRead]
53. GRANT SELECT ON [Utente].[concelho] TO [UserCrud]
54. GRANT SELECT ON [Utente].[concelho] TO [UserRead]
55. GRANT SELECT ON [Utente].[pessoa] TO [UserCrud]
56. GRANT SELECT ON [Utente].[pessoa] TO [UserRead]
57. GRANT SELECT ON [Utente].[regiao] TO [UserCrud]
```

```

58. GRANT SELECT ON [Utente].[regiao] TO [UserRead]
59.
60. --PERMISSOES CRUD PARA USERCRUD(SELECT JÁ FOI DADO ANTERIORMENTE)
61.
62. GRANT INSERT ON [Notificacao].[contacto] TO [UserCrud]
63. GRANT ALTER ON [Notificacao].[contacto] TO [UserCrud]
64. GRANT DELETE ON [Notificacao].[contacto] TO [UserCrud]
65. GRANT INSERT ON [Notificacao].[motivo] TO [UserCrud]
66. GRANT ALTER ON [Notificacao].[motivo] TO [UserCrud]
67. GRANT DELETE ON [Notificacao].[motivo] TO [UserCrud]
68. GRANT INSERT ON [Notificacao].[notificacao] TO [UserCrud]
69. GRANT ALTER ON [Notificacao].[notificacao] TO [UserCrud]
70. GRANT DELETE ON [Notificacao].[notificacao] TO [UserCrud]
71. GRANT INSERT ON [Notificacao].[notificacaoSintomas] TO [UserCrud]
72. GRANT ALTER ON [Notificacao].[notificacaoSintomas] TO [UserCrud]
73. GRANT DELETE ON [Notificacao].[notificacaoSintomas] TO [UserCrud]
74. GRANT INSERT ON [Notificacao].[pessoaNotificacaoResultado] TO [UserCrud]
75. GRANT ALTER ON [Notificacao].[pessoaNotificacaoResultado] TO [UserCrud]
76. GRANT DELETE ON [Notificacao].[pessoaNotificacaoResultado] TO [UserCrud]
77. GRANT INSERT ON [Notificacao].[sintomas] TO [UserCrud]
78. GRANT ALTER ON [Notificacao].[sintomas] TO [UserCrud]
79. GRANT DELETE ON [Notificacao].[sintomas] TO [UserCrud]
80. GRANT INSERT ON [Resultado].[resultado] TO [UserCrud]
81. GRANT ALTER ON [Resultado].[resultado] TO [UserCrud]
82. GRANT DELETE ON [Resultado].[resultado] TO [UserCrud]
83. GRANT INSERT ON [Resultado].[tiposPositivo] TO [UserCrud]
84. GRANT ALTER ON [Resultado].[tiposPositivo] TO [UserCrud]
85. GRANT DELETE ON [Resultado].[tiposPositivo] TO [UserCrud]
86. GRANT INSERT ON [Utente].[concelho] TO [UserCrud]
87. GRANT ALTER ON [Utente].[concelho] TO [UserCrud]
88. GRANT DELETE ON [Utente].[concelho] TO [UserCrud]
89. GRANT INSERT ON [Utente].[pessoa] TO [UserCrud]
90. GRANT ALTER ON [Utente].[pessoa] TO [UserCrud]
91. GRANT DELETE ON [Utente].[pessoa] TO [UserCrud]
92. GRANT INSERT ON [Utente].[regiao] TO [UserCrud]
93. GRANT ALTER ON [Utente].[regiao] TO [UserCrud]
94. GRANT DELETE ON [Utente].[regiao] TO [UserCrud]
95. --permissaocrud ver stored procedure
96.
97. USE ProjetoBasedeDados2;
98. GRANT EXECUTE ON OBJECT::[dbo].[UtentepessoaEncripted]
99.     TO [UserCrud]
100.     GO
101.     USE ProjetoBasedeDados2;
102.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoConselhoCasosConfirmados]
103.         TO [UserCrud]
104.         GO
105.     USE ProjetoBasedeDados2;
106.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoNewCasesConselho]
107.         TO [UserCrud]
108.         GO
109.     USE ProjetoBasedeDados2;
110.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoNewRecuperadosConselho]
111.         TO [UserCrud]
112.         GO
113.     USE ProjetoBasedeDados2;
114.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoresultadoTotalCasosConfirmados]
115.         TO [UserCrud]
116.         USE ProjetoBasedeDados2;
117.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoTotalPositivoContacto]

```

```

118.         TO [UserCrud]
119.     USE ProjetoBasedeDados2;
120.     GRANT EXECUTE ON OBJECT::[dbo].[RiscoporConselho]
121.         TO [UserCrud]
122. GO
123.
124.     USE ProjetoBasedeDados2;
125.     GRANT EXECUTE ON OBJECT::[dbo].[TotalCasosAtivos]
126.         TO [UserCrud]
127.
128.     USE ProjetoBasedeDados2;
129.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewCases]
130.         TO [UserCrud]
131.
132.     USE ProjetoBasedeDados2;
133.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewÓbitos]
134.         TO [UserCrud]
135.
136.     USE ProjetoBasedeDados2;
137.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewRecuperados]
138.         TO [UserCrud]
139.
140.     USE ProjetoBasedeDados2;
141.     GRANT EXECUTE ON OBJECT::[dbo].[TotalObitos]
142.         TO [UserCrud]
143.
144.     USE ProjetoBasedeDados2;
145.     GRANT EXECUTE ON OBJECT::[dbo].[TotalPositivoRegiao]
146.         TO [UserCrud]
147.
148.     USE ProjetoBasedeDados2;
149.     GRANT EXECUTE ON OBJECT::[dbo].[TotalRecuperados]
150.         TO [UserCrud]
151.
152.     USE ProjetoBasedeDados2;
153.     GRANT EXECUTE ON OBJECT::[dbo].[UtendeidadeCases]
154.         TO [UserCrud]
155.     USE ProjetoBasedeDados2;
156.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteconselhoCaseAtivos]
157.         TO [UserCrud]
158.     USE ProjetoBasedeDados2;
159.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhoObitos]
160.         TO [UserCrud]
161.     USE ProjetoBasedeDados2;
162.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorNewÓbitos]
163.         TO [UserCrud]
164.
165.     USE ProjetoBasedeDados2;
166.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorNewRecuperados]
167.         TO [UserCrud]
168.     USE ProjetoBasedeDados2;
169.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorRecuperados]
170.         TO [UserCrud]
171.     USE ProjetoBasedeDados2;
172.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteIdadeObitos]
173.         TO [UserCrud]
174.
175.     USE ProjetoBasedeDados2;
176.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteObitosGenero]
177.         TO [UserCrud]

```

```

178.
179.     USE ProjetoBasedeDados2;
180.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteObitosGeneroIdade]
181.         TO [UserCrud]
182.
183.     USE ProjetoBasedeDados2;
184.     GRANT EXECUTE ON OBJECT::[dbo].[UtentePessoaNifCases]
185.         TO [UserCrud]
186.
187.     USE ProjetoBasedeDados2;
188.     GRANT EXECUTE ON OBJECT::[dbo].[UtentePositivoGeneroIdade]
189.         TO [UserCrud]
190.
191.     --permissao para o usuario jrdb ver as definições
192.     GRANT VIEW DEFINITION TO [UserJrDb]
193.
194.     --permissao para o usuario read excut stored procedure
195.     USE ProjetoBasedeDados2;
196.     GRANT EXECUTE ON OBJECT::[dbo].[IndiceRT]
197.         TO [UserRead]
198.     GO
199.     USE ProjetoBasedeDados2;
200.     GRANT EXECUTE ON OBJECT::[dbo].[porcpoo]
201.         TO [UserRead]
202.     GO
203.
204.     USE ProjetoBasedeDados2;
205.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoConselhoCasosConfirmados]
206.         TO [UserRead]
207.     GO
208.     USE ProjetoBasedeDados2;
209.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoNewCasesConselho]
210.         TO [UserRead]
211.     GO
212.
213.     USE ProjetoBasedeDados2;
214.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoNewRecuperadosConselho]
215.         TO [UserRead]
216.     GO
217.
218.     USE ProjetoBasedeDados2;
219.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoreResultadoTotalCasosConfirmados]
220.         TO [UserRead]
221.
222.     USE ProjetoBasedeDados2;
223.     GRANT EXECUTE ON OBJECT::[dbo].[ResultadoTotalPositivoContacto]
224.         TO [UserRead]
225.
226.     USE ProjetoBasedeDados2;
227.     GRANT EXECUTE ON OBJECT::[dbo].[RiscoporConselho]
228.         TO [UserRead]
229.     GO
230.
231.     USE ProjetoBasedeDados2;
232.     GRANT EXECUTE ON OBJECT::[dbo].[TotalCasosAtivos]
233.         TO [UserRead]
234.
235.     USE ProjetoBasedeDados2;
236.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewCases]
237.         TO [UserRead]

```

```

238.
239.     USE ProjetoBasedeDados2;
240.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewÓbitos]
241.     TO [UserRead]
242.
243.     USE ProjetoBasedeDados2;
244.     GRANT EXECUTE ON OBJECT::[dbo].[TotalNewRecuperados]
245.     TO [UserRead]
246.
247.     USE ProjetoBasedeDados2;
248.     GRANT EXECUTE ON OBJECT::[dbo].[TotalObitos]
249.     TO [UserRead]
250.
251.     USE ProjetoBasedeDados2;
252.     GRANT EXECUTE ON OBJECT::[dbo].[TotalPositivoRegiao]
253.     TO [UserRead]
254.
255.     USE ProjetoBasedeDados2;
256.     GRANT EXECUTE ON OBJECT::[dbo].[TotalRecuperados]
257.     TO [UserRead]
258.
259.     USE ProjetoBasedeDados2;
260.     GRANT EXECUTE ON OBJECT::[dbo].[UtendeidadeCases]
261.     TO [UserRead]
262.
263.     USE ProjetoBasedeDados2;
264.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteconselhoCaseAtivos]
265.     TO [UserRead]
266.
267.     USE ProjetoBasedeDados2;
268.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhoObitos]
269.     TO [UserRead]
270.
271.     USE ProjetoBasedeDados2;
272.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorNewÓbitos]
273.     TO [UserRead]
274.
275.     USE ProjetoBasedeDados2;
276.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorNewRecuperados]
277.     TO [UserRead]
278.
279.     USE ProjetoBasedeDados2;
280.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteConselhorRecuperados]
281.     TO [UserRead]
282.
283.     USE ProjetoBasedeDados2;
284.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteIdadeObitos]
285.     TO [UserRead]
286.
287.     USE ProjetoBasedeDados2;
288.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteObitosGenero]
289.     TO [UserRead]
290.
291.     USE ProjetoBasedeDados2;
292.     GRANT EXECUTE ON OBJECT::[dbo].[UtenteObitosGeneroIdade]
293.     TO [UserRead]
294.
295.     USE ProjetoBasedeDados2;
296.     GRANT EXECUTE ON OBJECT::[dbo].[UtentePessoaNifCases]
297.     TO [UserRead]

```

```

298.
299.     USE ProjetoBasedeDados2;
300.     GRANT EXECUTE ON OBJECT::[dbo].[UtentePositivoGeneroIdade]
301.     TO [UserRead]
302.
303.     USE ProjetoBasedeDados2;
304.     GRANT EXECUTE ON OBJECT::[dbo].[Valordeincidencia]
305.     TO [UserRead]

```

```

1.  --
    ROLES EM NIVEL DE BD, PARA UM USUÁRIO Júnior de base de dados SOMENTE VISUALIZAR AS TAB
    ELAS (SELECT)
2.
3.
4.  USE [ProjetoBasedeDados2]
5.  GO
6.  CREATE ROLE [TableReader] AUTHORIZATION [UserJrDb]
7.  GO
8.  USE [ProjetoBasedeDados2]
9.  GO
10. ALTER ROLE [TableReader] ADD MEMBER [UserJrDb]
11. GO
12. use [ProjetoBasedeDados2]
13. GO
14. GRANT SELECT ON [Utente].[regiao] TO [TableReader]
15. GO
16. use [ProjetoBasedeDados2]
17. GO
18. GRANT SELECT ON [Resultado].[tiposPositivo] TO [TableReader]
19. GO
20. use [ProjetoBasedeDados2]
21. GO
22. GRANT SELECT ON [Notificacao].[motivo] TO [TableReader]
23. GO
24. use [ProjetoBasedeDados2]
25. GO
26. GRANT SELECT ON [Notificacao].[notificacao] TO [TableReader]
27. GO
28. use [ProjetoBasedeDados2]
29. GO
30. GRANT SELECT ON [Notificacao].[notificacaoSintomas] TO [TableReader]
31. GO
32. use [ProjetoBasedeDados2]
33. GO
34. GRANT SELECT ON [Notificacao].[contacto] TO [TableReader]
35. GO
36. use [ProjetoBasedeDados2]
37. GO
38. GRANT SELECT ON [Utente].[concelho] TO [TableReader]
39. GO
40. use [ProjetoBasedeDados2]
41. GO
42. GRANT SELECT ON [Notificacao].[pessoaNotificacaoResultado] TO [TableReader]
43. GO
44. use [ProjetoBasedeDados2]
45. GO
46. GRANT SELECT ON [Resultado].[resultado] TO [TableReader]

```

```

47. GO
48. use [ProjetoBasedeDados2]
49. GO
50. GRANT SELECT ON [Notificacao].[sintomas] TO [TableReader]
51. GO
52. use [ProjetoBasedeDados2]
53. GO
54. GRANT SELECT ON [Utente].[pessoa] TO [TableReader]
55. GO
56. --ROLES EM NIVEL DE SERVIDOR, PARA UM USUÁRIO Júnior de base de dados
57.
58. USE [master]
59.
60. GO
61.
62. CREATE SERVER ROLE [JrDBA] AUTHORIZATION [sa]
63.
64. GO
65.
66. ALTER SERVER ROLE [JrDBA] ADD MEMBER [UserJrDb]
67.
68. GO
69.
70. use [master]
71.
72. GO
73.
74. GRANT CREATE AVAILABILITY GROUP TO [JrDBA]
75.
76. GO
77.
78. use [master]
79.
80. GO
81.
82. GRANT VIEW SERVER STATE TO [JrDBA]
83.
84. GO
85.
86. use [master]
87.
88. GO
89.
90. GRANT VIEW ANY DATABASE TO [JrDBA]
91.
92. GO
93.
94. use [master]
95.
96. GO
97.
98. GRANT VIEW ANY DEFINITION TO [JrDBA]
99.
100.      GO
101.
102.      use [master]
103.
104.      GO
105.
106.      GRANT CONTROL SERVER TO [JrDBA]

```

```

107.
108.      GO

1.  --criação de schemas
2.  USE [ProjetoBasedeDados2]
3.  GO
4.
5.  /***** Object:  Schema [Utente]      Script Date: 27/01/2021 17:06:41 *****/
6.  CREATE SCHEMA [Utente]
7.  GO
8.
9.  USE [ProjetoBasedeDados2]
10. GO
11.
12. /***** Object:  Schema [Resultado]    Script Date: 27/01/2021 17:06:35 *****/
13. CREATE SCHEMA [Resultado]
14. GO
15.
16. USE [ProjetoBasedeDados2]
17. GO
18.
19. /***** Object:  Schema [Notificacao]  Script Date: 27/01/2021 17:06:29 *****/
20. CREATE SCHEMA [Notificacao]

```

## 12.6 ANEXO 6— TRIGGERS

```

1.  --Verifica se passou 1% da populacao
2.  CREATE TRIGGER [Resultado].[checkpoo]
3.      ON [Resultado].[resultado]
4.      AFTER INSERT, UPDATE
5.      AS BEGIN
6.          SET NOCOUNT ON;
7.          EXECUTE [porcpoo] ;
8.      END

```

--Cria um campo em tiposdepositivo que fica como ativo se o resultado é positivo

```

ALTER TRIGGER [Resultado].[resultado_tipospositivos]
ON [Resultado].[resultado]
AFTER INSERT, UPDATE
AS BEGIN
    SET NOCOUNT ON;
    DECLARE @valor AS INT;
    DECLARE @valor1 AS INT;
    DECLARE @name AS VARCHAR (12);
    SET @valor = 0;
    SET @valor1 = 0;
    SELECT @valor = idresultado
    FROM inserted;
    SELECT @name = nomeResultado

```



```

FROM inserted;
SELECT @valor1 = count(*) + 1
FROM [ProjetoBasedeDados2].[Resultado].[tiposPositivo];
IF (@name = 'POSITIVO')
    INSERT INTO [Resultado].[tiposPositivo]
    VALUES (@valor1, 'ATIVO', @valor, GETDATE());
END

--Cria um campo em resultados que fica como aguardado
CREATE TRIGGER [Notificacao].[createresul]
ON [Notificacao].[notificacao]
AFTER INSERT
AS BEGIN
    SET NOCOUNT ON;
    DECLARE @valor AS INT;
    SET @valor = 0;
    SELECT @valor = idNotificacao
    FROM inserted;
    INSERT INTO [Resultado].[resultado]
    VALUES (@valor, 'Aguardando', GETDATE());
END

```

## 12.7 ANEXO 7 – BACKUPS

```

1. BACKUP DATABASE [ProjetoBasedeDados2] TO DISK = N'E:\Program Files\Microsoft SQL Serve
r\MSSQL15.MSSQLSERVER\MSSQL\Backup\TPBD2.bak' WITH NOFORMAT, NOINIT, NAME = N'ProjetoB
asedeDados2-Completo Banco de Dados Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
2. GO

```

```

1. BACKUP DATABASE [ProjetoBasedeDados2] TO DISK = N'E:\Program Files\Microsoft SQL Serve
r\MSSQL15.MSSQLSERVER\MSSQL\Backup\TPBD2.bak' WITH DIFFERENTIAL, NOFORMAT, NOINIT, N
AME = N'ProjetoBasedeDados2-Completo Banco de Dados Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
2. GO

```

```

1. BACKUP LOG [ProjetoBasedeDados2] TO DISK = N'E:\Program Files\Microsoft SQL Server\MSS
QL15.MSSQLSERVER\MSSQL\Backup\TPBD2.bak' WITH NOFORMAT, NOINIT, NAME = N'ProjetoBasede
Dados2-Completo Banco de Dados Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
2. GO
3. USE [master]
4. BACKUP LOG [ProjetoBasedeDados2] TO DISK = N'E:\Program Files\Microsoft SQL Server\MSS
QL15.MSSQLSERVER\MSSQL\Backup\ProjetoBasedeDados2_LogBackup_2021-01-27_12-52-
37.bak' WITH NOFORMAT, NOINIT, NAME = N'ProjetoBasedeDados2_LogBackup_2021-01-27_12-
52-37', NOSKIP, NOREWIND, NOUNLOAD, NORECOVERY, STATS = 5
5. RESTORE DATABASE [ProjetoBasedeDados2] FROM DISK = N'A:\es.bak' WITH FILE = 1, NOUNL
OAD, STATS = 5
6.
7. GO

```

## 12.8 ANEXO 8 – FILEGROUPS

```
1. USE [ProjetoBasedeDados2]
2. GO
3. ALTER DATABASE [ProjetoBasedeDados2] MODIFY FILE (NAME=N'ProjetoBasedeDados2', NEWNAME=
  N'BDTP')
4. GO
5. USE [master]
6. GO
7. ALTER DATABASE [ProjetoBasedeDados2] ADD FILEGROUP [FG_data]
8. GO
9. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'data1', FILENAME = N'E:\Progra
  m Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\data1.ndf' , SIZE = 10240KB
  , MAXSIZE = 262144000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_data]
10. GO
11. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'data2', FILENAME = N'E:\Progra
  m Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\data2.ndf' , SIZE = 10240KB
  , MAXSIZE = 262144000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_data]
12. GO
13. ALTER DATABASE [ProjetoBasedeDados2] ADD FILEGROUP [FG_data1]
14. GO
15. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'data3', FILENAME = N'E:\Progra
  m Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\data3.ndf' , SIZE = 10240KB
  , MAXSIZE = 262144000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_data1]
16. GO
17. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'data4', FILENAME = N'E:\Progra
  m Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\data4.ndf' , SIZE = 10240KB
  , MAXSIZE = 262144000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_data1]
18. GO
19. ALTER DATABASE [ProjetoBasedeDados2] ADD FILEGROUP [FG_Indices]
20. GO
21. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'indice1', FILENAME = N'E:\Prog
  ram Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\indice1.ndf' , SIZE = 102
  40KB , MAXSIZE = 5120000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_Indices]
22. GO
23. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'indice2', FILENAME = N'E:\Prog
  ram Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\indice2.ndf' , SIZE = 102
  40KB , MAXSIZE = 5120000KB , FILEGROWTH = 10%) TO FILEGROUP [FG_Indices]
24. GO
25. ALTER DATABASE [ProjetoBasedeDados2] ADD FILEGROUP [FG_localidade]
26. GO
27. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'localidade', FILENAME = N'E:\P
  rogram Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\localidade.ndf' , SIZE
  = 3072KB , MAXSIZE = 51200KB , FILEGROWTH = 10%) TO FILEGROUP [FG_localidade]
28. GO
29. ALTER DATABASE [ProjetoBasedeDados2] ADD FILE ( NAME = N'BDTP1', FILENAME = N'E:\Progra
  m Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\BDTP.ndf' , SIZE = 8192KB ,
  FILEGROWTH = 65536KB ) TO FILEGROUP [PRIMARY]
30. GO
```

```

1. --NON CLUSTERED
2. CREATE NONCLUSTERED INDEX Notificacao_idmotivo_index ON notificacao ( idMotivo ASC )
3. WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = ON, ONLINE = OFF) ON [FG_indices]
4. GO
5.
6. CREATE NONCLUSTERED INDEX resultado_index ON resultado ( dataResultado ASC )
7. WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = ON, ONLINE = OFF) ON [FG_indices]
8. GO

1. --CLUSTERED
2. ALTER TABLE regiao DROP CONSTRAINT PK_regiao
3. GO
4. ALTER TABLE regiao ADD CONSTRAINT [PK_regiao] PRIMARY KEY CLUSTERED
5. ( [idRegiao] ASC)
6. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
  EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
7. ON [FG_localidade]
8. GO
9.
10. ALTER TABLE concelho DROP CONSTRAINT PK_concelho
11. GO
12. ALTER TABLE concelho ADD CONSTRAINT PK_concelho PRIMARY KEY CLUSTERED
13. ( [idConcelho] ASC)
14. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
  EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
15. ON [FG_localidade]
16. GO
17.
18. ALTER TABLE pessoa DROP CONSTRAINT [PK_pessoa]
19. GO
20. ALTER TABLE pessoa ADD CONSTRAINT [PK_pessoa] PRIMARY KEY CLUSTERED
21. ( [idPessoa] ASC)
22. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
  EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
23. ON FG_data
24. GO
25.
26.
27. ALTER TABLE contacto DROP CONSTRAINT [PK_contacto]
28. GO
29. ALTER TABLE contacto ADD CONSTRAINT [PK_contacto] PRIMARY KEY CLUSTERED
30. ( [idPessoa],[idContacto] ASC)
31. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
  EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
32. ON FG_data
33. GO
34.
35.
36. ALTER TABLE [dbo].[tiposPositivo] DROP CONSTRAINT [PK_tiposPositivo]
37. GO
38. ALTER TABLE [dbo].[tiposPositivo] ADD CONSTRAINT [PK_tiposPositivo] PRIMARY KEY CLUSTER
  ED
39. ( [idTiposPositivo] ASC)
40. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
  EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
41. ON FG_data1
42. GO

```

```

43.
44. ALTER TABLE [dbo].[pessoaNotificacaoResultado] DROP CONSTRAINT [PK_pessoaNotificacaoRes
    ultado]
45. GO
46. ALTER TABLE [dbo].[pessoaNotificacaoResultado] ADD CONSTRAINT [PK_pessoaNotificacaoResu
    ltado] PRIMARY KEY CLUSTERED
47. ( [idPessoa],[idNotificacao],[idResultado] ASC)
48. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
    EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
49. ON FG_data1
50. GO
51.
52. ALTER TABLE [dbo].[sintomas] DROP CONSTRAINT [PK_sintomas]
53. GO
54. ALTER TABLE [dbo].[sintomas] ADD CONSTRAINT [PK_sintomas] PRIMARY KEY CLUSTERED
55. ( [idSintomas] ASC)
56. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
    EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
57. ON FG_data
58. GO
59.
60. ALTER TABLE [dbo].[motivo] DROP CONSTRAINT [PK_motivo]
61. GO
62. ALTER TABLE [dbo].[motivo] ADD CONSTRAINT [PK_motivo] PRIMARY KEY CLUSTERED
63. ( [idMotivo] ASC)
64. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
    EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
65. ON FG_data
66. GO
67.
68. ALTER TABLE [dbo].[motivo] DROP CONSTRAINT [PK_motivo]
69. GO
70. ALTER TABLE [dbo].[motivo] ADD CONSTRAINT [PK_motivo] PRIMARY KEY CLUSTERED
71. ( [idMotivo] ASC)
72. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
    EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
73. ON FG_data
74. GO
75.
76. ALTER TABLE [dbo].[resultado] DROP CONSTRAINT [PK_resultado]
77. GO
78. ALTER TABLE [dbo].[resultado] ADD CONSTRAINT [PK_resultado] PRIMARY KEY CLUSTERED
79. ( [idResultado] ASC)
80. WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_K
    EY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
81. ON FG_data1
82. GO

```

```

1. -- Criar FK
2. ALTER TABLE [Utente].[concelho] WITH CHECK ADD CONSTRAINT [FK_idRegiao]
3. FOREIGN KEY ([idRegiao]) REFERENCES [Utente].[regiao]);
4. GO
5. ALTER TABLE [Utente].[concelho] CHECK CONSTRAINT [FK_idRegiao]
6.
7.
8. ALTER TABLE [Notificacao].[notificacao] WITH CHECK ADD CONSTRAINT [FK_idMotivo]
9. FOREIGN KEY ([idMotivo]) REFERENCES [Notificacao].[motivo]([idMotivo]);

```

```

10. GO
11. ALTER TABLE [Notificacao].[notificacao] CHECK CONSTRAINT [FK_idMotivo]
12.
13.
14. ALTER TABLE [Notificacao].[notificacaoSintomas] WITH CHECK ADD CONSTRAINT [FK_idNotifi
    cacao]
15. FOREIGN KEY ([idNotificacao]) REFERENCES [Notificacao].[notificacao]([idNotificacao]);
16. GO
17. ALTER TABLE [Notificacao].[notificacaoSintomas] CHECK CONSTRAINT [FK_idNotificacao]
18.
19.
20.
21. ALTER TABLE [Notificacao].[notificacaoSintomas] WITH CHECK ADD CONSTRAINT [FK_idSintom
    as]
22. FOREIGN KEY ([idSintomas]) REFERENCES [Notificacao].[sintomas]([idSintomas]);
23. GO
24. ALTER TABLE [Notificacao].[notificacaoSintomas] CHECK CONSTRAINT [FK_idSintomas]
25.
26.
27. ALTER TABLE [Utente].[pessoa] WITH CHECK ADD CONSTRAINT [FK_idConcelho]
28. FOREIGN KEY ([idConcelho]) REFERENCES [Utente].[concelho]([idConcelho]);
29. GO
30. ALTER TABLE [Utente].[pessoa] CHECK CONSTRAINT [FK_idConcelho]
31.
32.
33. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] WITH CHECK ADD CONSTRAINT [FK_i
    dPessoa]
34. FOREIGN KEY ([idPessoa]) REFERENCES [Utente].[pessoa]([idPessoa]);
35. GO
36. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] CHECK CONSTRAINT [FK_idPessoa]
37.
38.
39. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] WITH CHECK ADD CONSTRAINT [FK_n
    otificacao]
40. FOREIGN KEY ([idResultado]) REFERENCES [Resultado].[resultado]([idResultado]);
41. GO
42. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] CHECK CONSTRAINT [FK_notificaca
    o]
43.
44.
45. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] WITH CHECK ADD CONSTRAINT [FK_i
    dnotificacao1]
46. FOREIGN KEY ([idNotificacao]) REFERENCES [Notificacao].[notificacao]([idNotificacao]);
47. GO
48. ALTER TABLE [Notificacao].[pessoaNotificacaoResultado] CHECK CONSTRAINT [FK_idnotifica
    cao1]
49.
50.
51. ALTER TABLE [Resultado].[tiposPositivo] WITH CHECK ADD CONSTRAINT [FK_idresultado]
52. FOREIGN KEY ([idResultado]) REFERENCES [Resultado].[resultado]([idResultado]);
53. GO
54. ALTER TABLE [Resultado].[tiposPositivo] CHECK CONSTRAINT [FK_idresultado]
55.
56.
57. ALTER TABLE [Notificacao].[contacto] WITH CHECK ADD CONSTRAINT [FK_contacto] FOREIGN
    KEY([idPessoa])
58. REFERENCES [Utente].[pessoa] ([idPessoa])

```

```

59. GO
60. ALTER TABLE [Notificacao].[contacto] CHECK CONSTRAINT [FK_contacto]
61. GO

---Read only
USE [ProjetoBasedeDados2]
GO
declare @readonly bit
SELECT @readonly=convert(bit, (status & 0x08)) FROM sysfilegroups WHERE
groupname=N'FG_localidade'
if(@readonly=0)
    ALTER DATABASE [ProjetoBasedeDados2] MODIFY FILEGROUP [FG_localidade] READONLY
GO

USE [master]
GO
declare @readonly bit
SELECT @readonly=convert(bit, (status & 0x08)) FROM sysfilegroups WHERE
groupname=N'FG_localidade'
if(@readonly=0)
    ALTER DATABASE [ProjetoBasedeDados2] MODIFY FILEGROUP [FG_localidade] READONLY
GO

1. SELECT object_name(i.[object_id]) as Name_of_Object,
2. i.name as Index_Name,
3. i.type_desc as Index_Type,
4. f.name as Name_of_Filegroup,
5. a.type as Object_Type,
6. f.type,
7. f.type_desc
8. FROM sys.filegroups as f
9. INNER JOIN sys.indexes as i
10. ON f.data_space_id = i.data_space_id
11. INNER JOIN sys.all_objects as a
12. ON i.object_id = a.object_id
13. WHERE a.type = 'U'

1. --Finding User Created Tables
2. SELECT o.[name], o.[type], i.[name], i.[index_id], f.[name] FROM sys.indexes i
3. INNER JOIN sys.filegroups f
4. ON i.data_space_id = f.data_space_id
5. INNER JOIN sys.all_objects o
6. ON i.[object_id] = o.[object_id] WHERE i.data_space_id = f.data_space_id
7. AND o.type = 'U' -- User Created Tables
8. GO

```

## 12.9 ANEXO 9 – REPORT BUILDER



# SARS-CoV-2 / COVID-19

## Relatório de situação Geral

Data de Publicação: 26/01/2021 21:33:02 (dados até 2020-08-09 )

Número de Casos Confirmados:	284
Total de casos Ativos:	130
Total de casos Recuperados:	113
Total de óbitos:	41

Novos casos Ativos	Novos casos Recuperados	Novos Óbitos
1	0	0

Índice RT = 1,606903485254

Positivos por Contágio = 65

Gráfico de Casos Confirmado por Região

Total

