UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Anonimização Automática de Dados Estruturados

Francisco Eduardo do Couto Soares Ramos

Mestrado em Engenharia Informática

Trabalho de Projeto orientado por:
Doutor João Ricardo Martins Ferreira da Silva

2022

Agradecimentos

Gostava de expressar a minha profunda gratidão e apreciação ao meu orientador Dr. João Ricardo Silva pelos seus conselhos, ideias, compreensão e acompanhamento durante todo o tempo, sem o qual não seria possível realizar este projeto. Também não conseguiria chegar ao fim, ultrapassando vários obstáculos encontrados durante a tese, sem a ajuda dos orientadores da Trust Systems: Ana Guimarães e João Bernardo que me providenciaram todo o apoio necessário, juntamente com o seu conhecimento e a sua experiência. A todos os colegas da empresa que me foram ajudando e acompanhando o meu muito obrigado.

Tive o prazer de começar e acabar o meu percurso académico com esta tese na Faculdade de Ciências, onde fui sempre bem recebido pela faculdade e pude ganhar toda a preparação e conhecimento, dado pelos vários professores, para chegar ao fim desta jornada. Estou agradecido a todos os colegas com os quais percorri este caminho, ajudando-nos mutuamente em vários projetos e aprendendo uns com os outros trabalhando em equipa.

Por fim, à minha família, todos os meus amigos e, especialmente, à minha namorada que me encorajaram, motivaram e acreditaram em mim todo este tempo, o meu muito obrigado.

Resumo

Neste projeto foi estudada e desenvolvida uma solução para a anonimização automática de dados estruturados, que é um requisito legal para a partilha de dados com informação sensível sobre indivíduos. A solução é capaz de detetar um conjunto abrangente de campos de uma base de dados que contém dados que podem ser associados a algum indivíduo e anonimizá-los.

A falta de ferramentas de anonimização que possam tornar todo este processo mais rápido é ainda um problema para muitas entidades que o fazem manualmente, sendo este um processo que poderá ser bastante moroso ou até inviável. Para tentar resolver este problema, foi desenvolvida uma framework, que é descrita ao longo desta tese, baseando-se numa base de dados existente com dados fictícios e avaliada numa nova base de dados não conhecida. Esta solução sugere que campos da base de dados devem ser anonimizados, tendo o utilizador que validar a sugestão e escolher o processo de anonimização a aplicar, sendo que diferentes métodos têm diferente complexidade e impacto na utilidade dos dados anonimizados.

Em síntese, esta solução irá permitir uma maior privacidade dos dados pessoais de indivíduos e um processo mais rápido e automático para a tarefa de anonimização de dados estruturados.

Palavras-chave: Anonimização, Deteção de Dados Pessoais, Pseudonimização, Dados Estruturados, Privacidade

Abstract

In this project a solution was studied and developed for the automatic anonymisation of structured data, which is a legal requirement for sharing data with sensitive information about individuals. The solution is able to detect a comprehensive set of fields from a database which contains data that can be associated to individuals and anonymise them.

The lack of anonymisation tools that can make this whole process faster is still a problem for many entities that do it manually, being a process that can be quite time consuming or even unfeasible. To try solve this problem, a framework was developed, which is described throughout this thesis, based on an existing database with dummy data and evaluated on a new unknown database. This solution suggests which fields in the database should be anonymised. The user have to validate the suggestion and choose the anonymisation process to apply, with different methods having different complexity and impact on the usefulness of the anonymised data.

In summary, this solution will allow greater privacy of individual's personal data and a faster and more automatic process for the task of anonymising structured data.

Keywords: Anonymization, Personal Data Detection, Pseudonymization, Structured Data, Privacy

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Este capítulo providencia uma introdução de alto-nível ao trabalho realizado, explicando o seu contexto e motivação, sumarizando as contribuições principais, dando o enquadramento institucional e apresentando a estrutura do relatório.

1.1 Contexto e motivação

O direito à privacidade encontra-se consagrado na Declaração Universal dos Direitos Humanos [19, Art. 12], e diferentes países impõem variadas restrições ao uso e disseminação de dados pessoais sem o consentimento da pessoa a que tais dados se referem. Por exemplo, a *General Data Protection Regulation (GDPR)* [10] é uma regulamentação que impõe restrições a organizações em relação à recolha e partilha de dados relativos às pessoas da União Europeia. A privacidade dos dados deve por isso ser uma prioridade para qualquer instituição que opere no contexto europeu.

A vasta quantidade de dados que instituições e empresas recolhem e armazenam são um recurso extremamente valioso para análise e investigação [8], e estes dados podem também ser usados para melhorar os serviços providenciados e melhor alinhá-los com os desejos dos utilizadores. Porém, isto tem de ser conseguido sem violar os requisitos de privacidade e expor a identidade de indivíduos [6].

Para possibilitar o uso e estudo de dados sem expor identidades, estes têm primeiro de ser anonimizados através de um processo que remova a possibilidade de identificar os indivíduos referenciados. Fazê-lo manualmente é extremamente moroso, e até inviável para as grandes quantidades de dados produzidos hoje em dia, o que motivou o desenvolvimento de técnicas para anonimização automática de dados [13].

Estas técnicas, que serão descritas em maior pormenor no Capítulo 2, têm de conseguir anonimizar os dados sem porém os inutilizar para análise, o que poderia acontecer com um processo de anonimização trivial que se limitasse a pura e simplesmente eliminar toda e qualquer informação sensível presente nos dados.

1.2 Contribuições

Neste projeto foi desenvolvida uma *framework* para a anonimização de bases de dados. Esta *framework* automatiza o processo de deteção dos campos a serem anonimizados, reduzindo assim o esforço da parte do utilizador e o tempo necessário para o processo. Tendo identificado os campos a anonimizar, a *framework* permite a aplicação de diferentes processos de anonimização (ver Secção 2.1.3) dependendo do tipo de campo, assegurando sempre a consistência dos dados anonimizados. O resultado final da aplicação da *framework* é uma cópia anonimizada da base de dados que pode ser usada sem expor a identidade de indivíduos.

1.3 Enquadramento institucional

Este projeto foi realizado no âmbito do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa, sendo o projeto final do Mestrado. É uma proposta dada por uma empresa e aceite pela faculdade formando uma parceria para um projeto final de curso.

O projeto foi realizado na empresa Trust Systems [18], uma empresa tecnológica que se insere na área da Segurança da Informação e que se especializa em fornecer soluções em ambientes empresariais e governamentais com o objetivo de garantir a segurança dos ativos de informação.

No desenvolvimento do projeto, como caso de uso, foi usada uma base de dados de um projeto da empresa designado *TeamingUp*, tendo esta base de dados sido preenchida com dados fictícios. A intenção, porém, é ir além deste caso de uso e ter uma *framework* genérica que possa no futuro ser aplicada a outras bases de dados.

1.4 Estrutura do relatório

O resto deste relatório segue a seguinte estrutura: O Capítulo 2 introduz conceitos base sobre anonimização e cobre o trabalho relacionado, incluindo algumas das ferramentas disponíveis. O Capítulo 3 mostra quais os objetivos do projeto, o planeamento e os desvios ocorridos. O Capítulo 4 descreve todo o trabalho que foi feito, mostra o funcionamento da framework e apresenta uma avaliação da solução criada. Finalmente, o Capítulo 5 conclui o relatório, apresenta um comentário crítico e aponta algumas linhas de trabalho futuro.

Capítulo 2

Anonimização

Este capítulo descreve em que consiste a tarefa de anonimização, introduzindo conceitos importantes como os diferentes tipos de identificadores pessoais, a utilidade, a consistência, a k-anonimidade, e os vários processos de anonimização que podem ser aplicados. É também feito um levantamento do trabalho relacionado, cobrindo em especial as diferentes ferramentas de anonimização disponíveis.

2.1 Anonimização

A anonimização é um processo através do qual dados são alterados com o objetivo de os tornar não pessoais, isto é, de forma a que não possam ser associados a alguém [5]. A informação a anonimizar divide-se em três categorias: identificadores diretos, quase-identificadores e atributos sensíveis [13].

Identificadores diretos são únicos para cada indivíduo, e incluem atributos como o nome próprio, o número de cartão de cidadão ou número de telemóvel, que permitem identificar o indivíduo sem precisar de informação adicional. Assim sendo, identificadores nesta categoria devem sempre ser anonimizados.

Quase-identificadores são dados, como o código postal, sexo, nacionalidade, cidade de residência ou profissão que, isoladamente, não permitem identificar um indivíduo mas que, combinados entre si, reduzem o leque de indivíduos possíveis e podem até permitir a identificação. Por exemplo, como Golle [11] sublinha, especificar o sexo, código postal e data de nascimento basta para permitir identificar unicamente cerca de 63% da população dos Estados Unidos da América.

Atributos sensíveis são dados, como informações de saúde ou de religião, que são protegidos sob certas molduras legais, não podendo ser recolhidos nem usados sem o consentimento do sujeito.

É de notar que existem ainda descrições indiretas, como "o vigésimo Presidente da República Portuguesa", que permitem identificar um indivíduo sem recorrer a identificadores como os descritos acima. Detetar e anonimizar este tipo de descrições está fora do âmbito deste projeto.

2.1.1 Estruturação dos dados

Os dados sobre os quais os processos de anonimização são aplicados podem ser caracterizados como sendo dados **estruturados** ou dados **não estruturados**.

Por dados estruturados entendem-se quaisquer dados sobre os quais já está imposto algum tipo de organização, como acontece com dados tabulados ou com bases de dados relacionais. A anonimização destes dados tende a ser um desafio mais simples pois a estruturação já existente permite mais facilmente localizar os atributos a anonimizar e determinar qual o seu tipo.

Por outro lado, os dados não estruturados não têm qualquer organização que lhes esteja pré-imposta. São aquilo a que poderíamos chamar "texto livre" e frequentemente incluem descrições indiretas. A anonimização automática deste tipo de dados apresenta-se como sendo um desafio mais difícil para a comunidade científica e é ainda alvo de investigação ativa (ver [13] para um levantamento recente da literatura da área).

Este projeto foca-se apenas na anonimização de dados estruturados, nomeadamente bases de dados relacionais, ficheiros com dados tabulados e ficheiros JSON, sendo que no resto deste relatório usaremos "base de dados" como termo genérico para referir este tipo de dados.

2.1.2 Utilidade

Na anonimização existe um *trade-off* a ter em conta entre maximizar a **utilidade** dos dados anonimizados e maximizar a privacidade dos mesmos [6]. Remover a informação acerca de indivíduos leva, necessariamente, à perda de alguma da utilidade dos dados pois diminui o ganho de informação que se pode retirar de uma análise dos dados [13]. Como um exemplo extremo, todos os campos de uma base de dados podiam ser apagados, o que certamente garantiria anonimidade mas com o custo da perda total de utilidade.

É importante preservar tanto quanto possível esta utilidade para que os utilizadores que vão trabalhar com os dados anonimizados possam analisá-los, usá-los em testes, retirar conclusões adequadas e não ficar com dados pobres que não servirão os seus propósitos. Assim sendo, anonimizar toda a informação, como no exemplo extremo dado acima, nem sempre é o correto nem o que se pretende, e o nível de anonimização ideal depende do propósito a que os dados estão destinados.

Por exemplo, dados sobre a saúde de pessoas são tipicamente para anonimizar, no entanto se o objetivo da análise for recolher estatísticas sobre a incidência de uma dada

doença na população, estes atributos sensíveis não devem ser completamente eliminados dos dados.

2.1.3 Processos de anonimização

Tendo identificado os dados a anonimizar, é necessário considerar a forma como estes serão anonimizados. Seguindo a terminologia usada por Lison et alia [13], os dados podem ser anonimizados por supressão, pseudonimização ou generalização.

Suprimir consiste em eliminar por completo os dados a anonimizar [17]. Nalgumas variantes deste processo, e como forma de preservar um mínimo de utilidade dos dados anonimizados, podem usar-se marcadores de tipo semântico para substituir o dado suprimido, por exemplo substituindo nomes próprios por "NOME" e números de contribuinte por "NIF".

Pseudonimizar consiste em substituir dados a anonimizar por pseudónimos [20]. Isto consiste tipicamente em substituir um nome próprio por outro, por exemplo "André" por "Hugo", mas o conceito pode ser alargado a, por exemplo, substituir um número por outro, mantendo quaisquer restrições de formato que se apliquem (e.g. o número de dígitos ou a paridade), trocar um endereço de email por um endereço gerado automaticamente mas corretamente formado, etc. Se desejarmos que o processo de pseudonimização seja aplicado de forma consistente (ver Secção 2.1.4), o mapeamento entre os valores originais e os valores anonimizados correspondentes deve ser guardado à parte e em segurança [13] ou deve ser feito usando algum tipo de função de *hashing* que, por ser irreversível, preserva a anonimidade.

Generalizar consiste em tornar um atributo mais geral e, portanto, menos informativo e aplicável a um conjunto mais largo de indivíduos. Isto pode ser feito através da substituição de um valor por outro mais geral ou mascarando parte do valor do atributo [17]. Por exemplo, um código postal como "4331-576" pode ser generalizado substituindo-o pelo nome do concelho correspondente ou mascarando os últimos dígitos (e.g. "4331-xxx"), um valor de idade em anos pode ser trocado por um intervalo de idades (e.g. "35-45 anos"), e o nome de uma doença substituído por um termo mais geral (e.g. substituir "linfoma" por "tumor").

Cada um destes processos de anonimização tem uma complexidade de implementação diferente, assim como diferente impacto na utilidade dos dados anonimizados produzidos. A supressão dos dados é claramente o processo mais simples, mas é também aquele que mais reduz a utilidade dos dados resultantes, sendo tipicamente usado apenas para aqueles atributos que não são de todo relevantes para a análise que se pretende fazer aos

dados, evitando assim ter de aplicar processos alternativos mais complexos. Para os restantes atributos, o processo escolhido é tipicamente alguma forma de pseudonimização ou generalização.

2.1.4 Consistência

Para possibilitar uma análise correta dos dados anonimizados, é importante assegurar a **consistência** do processo de anonimização.

Um processo de anonimização é consistente quando a relação entre as várias entradas das tabelas se mantém dos dados originais para os anonimizados. Anonimizar o número de cartão de crédito "1234" para "9814" sempre que aparecer este cartão numa base de dados é um exemplo onde se mantém a consistência.

Manter a consistência entre as entradas depois de anonimizadas e entre diferentes execuções do processo de anonimização maximiza a utilidade da informação, podendo no entanto aumentar o risco de ataque, devido aos ficheiros que são necessários guardar para correlacionar a informação.

2.1.5 k-anonimidade

A k**-anonimidade** [17] é uma propriedade de um conjunto de dados que dá uma medida do nível de ambiguidade neles presente, sendo que neste contexto a presença de ambiguidade assegura um certo grau de anonimidade.

Mais concretamente, um conjunto de dados tem a propriedade de k-anonimidade se qualquer combinação de quase-identificadores que queiramos especificar ocorrer pelo menos k vezes nos dados. Ou seja, qualquer combinação de quase-identificadores, por mais extensa e detalhada que seja, será sempre compatível com pelo menos k entradas da base de dados, não podendo portanto ser ligada a um único indivíduo.

A Figura 2.1 (adaptada de [7]) mostra um exemplo de 2-anonimidade. Os dados originais (tabela à esquerda) foram anonimizados através de um processo de generalização de forma a que qualquer combinação de valores de quase-identificadores QI_n anonimizados (tabela à direita) resulte sempre em pelo menos 2 entradas, impedindo assim a atribuição de um atributo sensível S_n a um indivíduo específico.

2.2 Trabalho relacionado

Muita da literatura científica sobre anonimização foca-se em dados não estruturados, que são um desafio em termos de processamento da linguagem mas que saem fora do âmbito deste trabalho. No que toca à anonimização de dados estruturados, o maior foco da literatura encontra-se em técnicas que asseguram certas propriedades como a k-anonimidade.

ID	QI_1 Age	QI_2 Zip	S_1 Disease
1	5	15	Flu
2	15	25	Fever
3	28	28	Diarrhea
4	25	15	Fever
5	22	28	Flu
6	32	35	Fever
7	38	32	Flu
8	35	25	Diarrhea

ID	QI_1 Age	QI_2 Zip	S_1 Disease
1	0-20	10-30	Flu
2	0-20	10-30	Fever
3	20-30	10-30	Diarrhea
4	20-30	10-30	Fever
5	20-30	10-30	Flu
6	30-40	20-40	Fever
7	30-40	20-40	Flu
8	30-40	20-40	Diarrhea

Figura 2.1: Exemplo de dados com 2-anonimidade

Saramati propõe[16] uma formalização de conceitos base como quase-identificadores, k-anonimidade, generalização e supressão, assim como um algoritmo que encontra a generalização ótima, ou seja o mínimo de generalização necessária para atingir k-anonimidade, mas que não é viável aplicar em tempo real. Já Bayardo e Agrawal[4] apresentam um algoritmo alternativo que acha a solução ótima e que é praticável, mesmo num conjunto de dados não trivial.

Uma nova *framework* foi feita para a geração de dados sintéticos (fictícios) de registos de saúde electrónicos que se aproxima a um conjunto de dados real [21]. O modelo criado é chamado de *ADS-GAN (anonymization through data synthesis using generative adversarial networks)* e é proposto que possa ser usado para a partilha de dados deste tipo de forma legal e segura, proporcionando um avanço mais rápido da inteligência artificial na medicina.

Mondrian [12] é um algoritmo eficiente para atingir a k-anonimidade. Recentemente, foi apresentada uma nova solução baseada neste algoritmo que não necessita de conhecer todo o conjunto de dados e é implementada através de programação paralela [8]. Foi concebida com o propósito de anonimizar grandes conjuntos de dados proporcionando, através dos resultados, escalabilidade, quando comparada ao algoritmo *Mondrian*, e sem perder utilidade dos dados.

2.2.1 Ferramentas de anonimização

Existem já várias ferramentas de anonimização disponíveis. Para limitar o número de ferramentas a estudar, uma série de requisitos e características desejáveis foram impostos logo à partida.

- Apenas serão consideradas ferramentas *open-source*, permitindo assim que sejam feitos ajustes ao código, se tal se revelar necessário. No entanto, é importante evitar ferramentas que, mesmo sendo de código aberto, sejam antiquadas, pois tal poderia

requerer muito trabalho de correção de *bugs* e vulnerabilidades antes de as poder usar para o projeto.

- Serão excluídas ferramentas que apenas produzam gráficos e estatísticas que sumarizam os dados (e.g. Privacy Integrated Queries (PINQ) [14] e Private data Sharing Interface (PSI) [9]), mesmo que de forma anonimizada, pois o objetivo do projeto é criar uma solução que produza uma cópia anonimizada da base de dados.

- Serão também excluídas ferramentas que sejam baseadas apenas em *web services* providenciados por terceiros pois tal iria requerer que os dados a anonimizar tivessem que, a dado ponto, ser tratados por entidades externas à empresa Trust Systems.

- Finalmente, será desejável que a ferramenta tenha suporte para Windows e que seja baseada em Java, dado que estes são o sistema operativo e a linguagem de programação mais usados na empresa.

Tendo em conta estes requisitos, foram consideradas três ferramentas de anonimização para estudo, sendo que todas são implementadas em Java: Amnesia [2], ARX Data Anonymization Tool [15] e Anonimatron [3].

Amnesia: Suporta Windows e Linux. Permite gerar, carregar, editar e guardar hierarquias que são usadas para generalizar quase-identificadores. Suporta também supressão e pseudonimização (sendo os campos simplesmente substituídos por caracteres aleatórios ou por símbolos, mascarando-os). Funciona com o modelo de privacidade k-anonimidade, sendo o objetivo da ferramenta deixar os dados k-anonimizados, para um certo k especificado pelo utilizador. Não tem suporte para correr diretamente sobre bases de dados relacionais, apenas permitindo o carregamentos de ficheiros com os dados.

ARX: Suporta Windows, Linux e MacOS. Suporta vários métodos de transformação de dados: generalização através de hierarquias, supressão, escolher uma amostra aleatória do conjunto de dados, etc. Disponibiliza vários modelos de privacidade, sendo um deles k-anonimidade. É compatível com bases de dados SQL e ficheiros. Das três ferramentas consideradas é a única que tem métodos para analisar o risco de reidentificação e para analisar a utilidade dos dados anonimizados.

Anonimatron: Suporta Windows, Linux e MacOS. Permite apenas a pseudonimização de nomes, emails e números. Não tem modelos de privacidade como as outras ferramentas consideradas, mas é capaz de correr sobre dados em bases de dados ou em ficheiros, e é a única das três ferramentas que mantém a consistência na anonimização entre diferentes execuções.

	Amnesia	ARX	Anonimatron
linguagem	Java	Java	Java
PostgreSQL	incompatível	compatível	compatível
consistência	não	não	sim
pseudonimização	parcial	não	sim
facilidade de extensão	$\star$	$\star\star$	$\star\star\star$

Tabela 2.1: Sumário das características das ferramentas

Um sumário das características mais relevantes destas ferramentas pode ser visto na Tabela 2.1. É de notar que nenhuma destas ferramentas é capaz de automaticamente detetar quais os campos a anonimizar numa base de dados. Isto é, todas esperam que os campos a anonimizar sejam especificados pelo utilizador.

Capítulo 3

Metodologia e planeamento

Este capítulo apresenta a metodologia seguida assim como o planeamento inicialmente proposto e os desvios ocorridos.

3.1 Metodologia

Como referido na Secção 2.2.1, nenhuma das ferramentas de anonimização consideradas é capaz de detetar quais os campos da base de dados que devem ser anonimizados. Assim sendo, a solução de anonimização desenvolvida neste trabalho tem de começar por um passo de deteção automática dos campos da base de dados que possam potencialmente conter informação sensível.

A deteção automática dos campos a anonimizar será feita recorrendo a padrões e a heurísticas. Para atributos com um formato bem definido, como o endereço de email, o número de telefone ou o número de cartão de crédito, serão usados padrões especificados por expressões regulares. Para atributos que não possam ser definidos através de um padrão, o processo de deteção irá recorrer a heurísticas baseadas em "dicionários" (i.e. listas de palavras). Por exemplo, para detetar se uma coluna da base de dados contém nomes próprios, as entradas dessa coluna serão comparadas com um dicionário de nomes próprios, sendo que se várias das entradas forem encontradas no dicionário, é provável que a coluna contenha nomes próprios.

O resultado deste primeiro passo de deteção automática dos campos a anonimizar será um conjunto de campos da base de dados que, plausivelmente, correspondem a informação que pode ter de ser anonimizada. Este conjunto de campos será apresentado ao utilizador da ferramenta como uma sugestão dos campos a anonimizar, tendo o utilizador que a validar com base no uso pretendido para os dados anonimizados. É também neste ponto que o utilizador tem de escolher que processo de anonimização aplicar a cada campo, de entre aqueles descritos na Secção 2.1.3.

Tendo um conjunto validado de campos a anonimizar, o processo de anonimização propriamente dito pode ser delegado a uma das ferramentas de anonimização estudadas

(ver Secção 2.2.1), sendo que pode vir a ser necessário estender esta ferramenta de forma a que suporte novos tipos de dados.

Para a avaliação do projeto irá ser usada uma versão da base de dados para correr a ferramenta final e analisarmos a sua eficácia, tanto a detetar os campos a anonimizar, como a tornar os dados anónimos.

3.1.1 Anonimização estática vs. dinâmica

Pretende-se que a ferramenta possibilite dois modos de funcionamento. No modo de **anonimização estática** o processo de anonimização resulta na criação de uma cópia anonimizada da base de dados, sendo esse o resultado que é partilhado. No modo de **anonimização dinâmica** o processo de anonimização é aplicado *on-the-fly* ao resultado de *queries* realizadas sobre a base de dados, nunca havendo a criação de uma cópia anonimizada da base de dados.

Cada modo de funcionamento tem propósitos diferentes e a escolha do modo a utilizar será feita pelo utilizador e dependerá do caso de uso. Embora os diferentes modos de funcionamento não alterem os métodos de anonimização a aplicar, será preciso ter em consideração questões de eficiência. Por exemplo, certos processos que asseguram um dado nível de k-anonimidade podem ser demasiado pesados para correr de forma dinâmica em bases de dados muito grandes [8].

3.1.2 A base de dados TeamingUp

O TeamingUp é um projeto da empresa Trust Systems que foi usado como caso de estudo para construir e testar a *framework* de anonimização. No âmbito do projeto TeamingUp, a Trust Systems desenvolveu uma aplicação para a gestão de colaboradores de empresas e contratos dos mesmos, tendo por isso de recorrer a uma base de dados.

Esta base de dados, a que chamaremos base de dados TeamingUp, é composta por várias tabelas, nomeadamente uma tabela de colaboradores, uma de emails dos colaboradores, uma de empresas, uma de gestores e uma de contratos. Um diagrama da estrutura da base de dados pode ser visto na Figura 3.1.

Naturalmente, estas tabelas contêm muita informação pessoal e sensível acerca dos trabalhadores de uma empresa, como nomes, NIF, NIB, morada, nível de educação, remuneração, emails, números de telemóvel, passaporte, nacionalidade, estado civil, data de aniversário, dados de IRS, etc., fazendo desta base de dados um bom caso de uso para a ferramenta de anonimização desenvolvida.

Para efeitos do projeto atual, esta base de dados foi populada com dados fictícios.

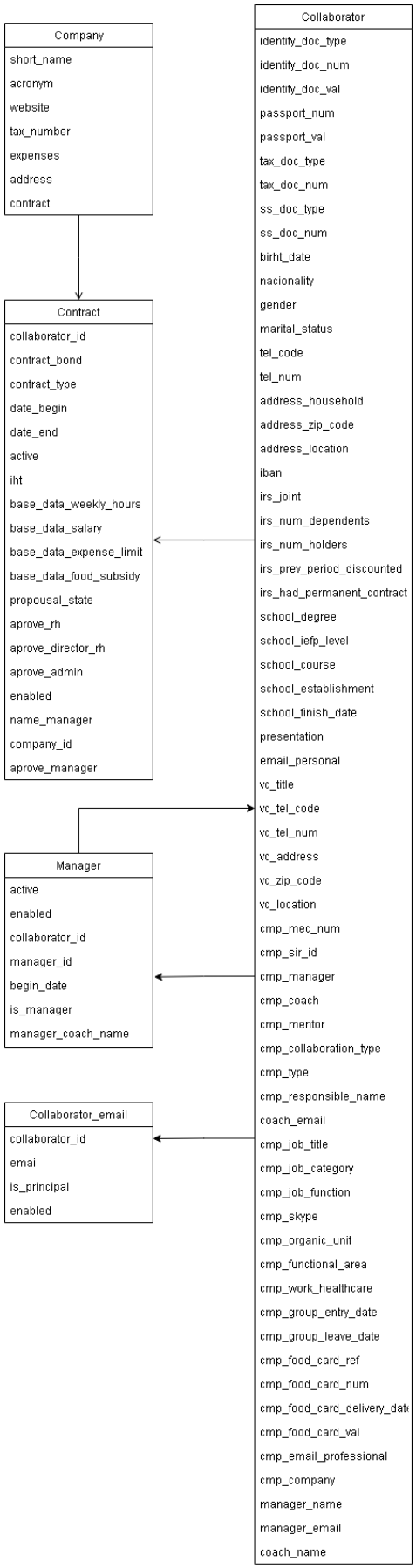

Figura 3.1: Modelo da base de dados TeamingUp

3.2 Planeamento

O planeamento inicial do projeto encontra-se representado na Figura 3.2.

No início do projeto, há uma fase de familiarização com a proposta com a duração aproximada de uma semana e meia. Esta fase serve para conhecer melhor a proposta do projeto, juntamente com os orientadores da empresa e da faculdade, assim como o ambiente e arquitetura de desenvolvimento de projetos da empresa.

A isto segue-se a pesquisa do trabalho relacionado, quer através da leitura de artigos sobre o tópico quer procurando possíveis soluções e ferramentas já existentes; e a experimentação com estas soluções e ferramentas encontradas, com vista a perceber o que conseguem e fazer e quais as suas eventuais limitações.

Terminada a pesquisa e experimentação, passa-se à escrita do relatório preliminar do projeto, onde se resume o trabalho realizado até ao momento, sendo esta fase encerrada com uma apresentação do relatório até à segunda semana de Janeiro.

Apresentado o relatório preliminar, passa-se então para a fase mais longa, com a duração aproximada de 3 meses, onde é realizada a implementação propriamente dita da solução escolhida. Perto do fim da fase de implementação há um maior foco em testes à ferramenta.

Finda a implementação e os testes, é feita uma avaliação final da solução sobre uma base de dados nova, que não foi vista durante a implementação. É importante avaliar o desempenho da ferramenta sobre uma base de dados diferente daquela usada durante a implementação, e à qual a ferramenta e o seu programador não foram expostos antes, pois isto dá-nos uma melhor ideia de como a ferramenta se portará quando em uso.

Finalmente, no período de meados de Maio até ao final de Junho, o foco estará na escrita deste relatório final.

3.2.1 Desvios ocorridos

Comparando o trabalho realizado com o planeamento apresentado na secção anterior, a entrega e apresentação do relatório preliminar estenderam-se um pouco no tempo, tendo este sido apresentado no final de Janeiro. Esta alteração atrasou um pouco o projeto, fazendo com que a fase de implementação começasse um pouco mais tarde.

A implementação prolongou-se até meados de Maio, devido a vários obstáculos encontrados e alterações que tiverem de ser feitas. Isto levou a uma maior sobreposição das tarefas na fase final do projeto. Além de realizar testes durante a fase implementação, como inicialmente planeado, a avaliação sobre a base de dados nova (isto é, não conhecida previamente) foi realizada durante a escrita do relatório final, que se iniciou em Junho.

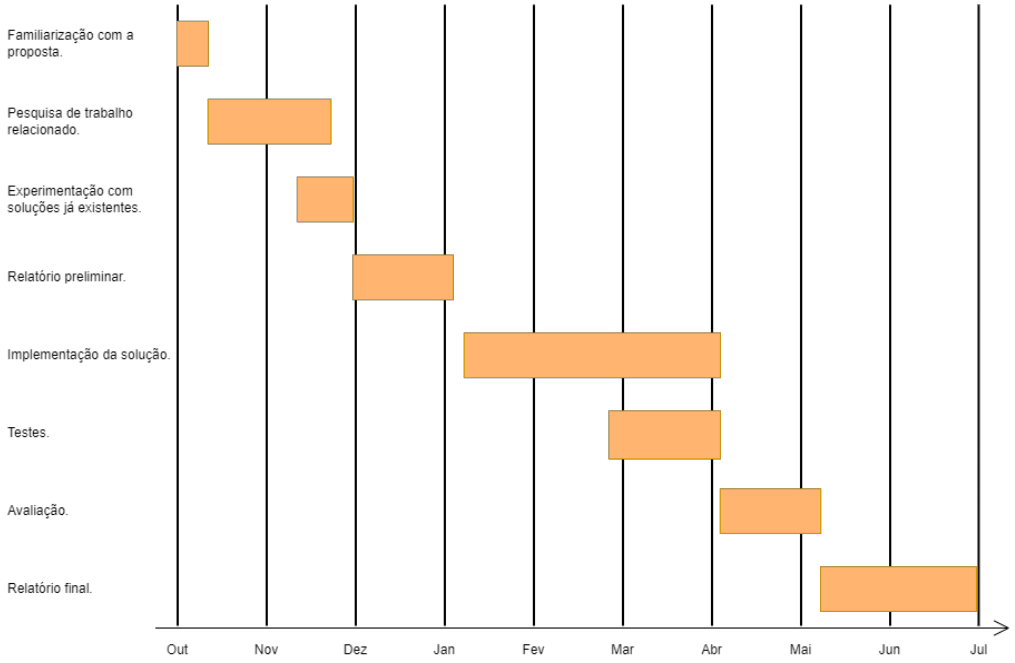

Figura 3.2: Diagrama do planeamento

Descarte da anonimização dinâmica

Como referido na Secção 3.1.1, estava planeado permitir que a ferramenta conseguisse anonimizar quer de forma estática (criando uma cópia anonimizada da base de dados) quer de forma dinâmica (*on-the-fly*, sem criar uma cópia).

Porém, durante o decorrer do projeto, a intenção da empresa no que toca ao modo de anonimização dinâmica foi sendo clarificada. Nomeadamente, o que se pretende com a anonimização dinâmica é que quando um utilizador faz um pedido à base de dados os dados retornados sejam anonimizados dinamicamente *dependendo do papel que está atribuído a esse utilizador num determinado projeto*. Por exemplo, um utilizador normal do sistema vê os dados anonimizados, enquanto um utilizador que é administrador tem acesso aos dados reais.

Ora, como cada projeto ou *back-end* tem o seu próprio sistema de papéis de utilizadores e direitos, é difícil criar uma solução geral para este problema. Uma possibilidade seria adicionar ao *back-end* uma camada de anonimização de dados entre a "camada de negócio" e a camada de acesso aos dados, mas esta implementação variaria de projeto para projeto e representaria um custo adicional de tempo e esforço que iria além do escopo inicialmente previsto para este projeto.

Assim sendo, foi decidido descartar o modo de anonimização dinâmica, sendo que a ferramenta funcionará apenas no modo de anonimização estática.

Capítulo 4

Trabalho realizado

Este capítulo apresenta em detalhe o trabalho que foi feito. Começa pela análise e desenho da solução, passa à descrição da implementação, e termina com a avaliação dos resultados obtidos.

4.1 Análise

A parte inicial do trabalho realizado esteve focada na leitura de vários artigos, tanto mais antigos como mais recentes, o que permitiu entender vários conceitos sobre o tema da anonimização e conhecer os métodos aplicados. Este trabalho inicial envolveu também um estudo e análise exploratória das ferramentas de anonimização disponíveis.

Depois de realizar algumas experiências com as ferramentas indicadas na Secção 2.2.1, o Anonimatron revelou-se como sendo a escolha adequada para os propósitos deste projeto. As razões para a escolha do Anonimatron tocam em vários aspetos desta ferramenta, listados de seguida:

- O Anonimatron pode ser diretamente aplicado a uma base de dados (PostgreSQL, no caso do projeto atual), o mesmo não acontecendo com a Amnesia, que apenas suporta ficheiros com o conjunto de dados. É conveniente poder executar a anonimização conectando-se diretamente à base de dados para que não seja necessário gastar recursos e tempo a replicar o conteúdo da base de dados para um ficheiro que será dado à ferramenta.

- O Anonimatron é a única ferramenta, de entre as ferramentas consideradas, capaz de assegurar a consistência na anonimização entre diferentes execuções do processo de anonimização e na mesma execução do processo.

- O Anonimatron tem um algoritmo de pseudonimização melhor do que o da ferramenta Amnesia (que se limita a substituir os atributos a anonimizar por caracteres aleatórios).

- Finalmente, o Anonimatron é também a ferramenta que, na análise exploratória preliminar, me pareceu ser mais facilmente extensível em comparação com as outras, visto que permite criar módulos para anonimizar novos tipos de dados através da simples definição de uma classe Java, tendo como requisito implementar apenas uns métodos de uma interface (ver Secção 4.3.4).

Como referido na Secção 2.2.1, nenhuma destas ferramentas é capaz de detetar quais os campos a anonimizar. Todas esperam que os campos a anonimizar já estejam definidos.

4.2 Desenho

Para proceder à deteção dos campos suscetíveis à anonimização, começa-se por recolher da base de dados a informação sobre que tabelas existem e quais as colunas que as constituem.

Percorrem-se todas as colunas aplicando algoritmos com dicionários e expressões regulares para as entradas de cada coluna, com o objetivo de encontrar correspondências com um determinado tipo de dado. Por fim, exportam-se esses metadados para serem utilizados na ferramenta de anonimização. Antes de começar a anonimização, cria-se uma cópia da base de dados e essa cópia é que será anonimizada.

A ferramenta Anonimatron começa por receber a informação necessária para correr e anonimiza as colunas pretendidas. Esta ferramenta foi também estendida para incluir novos algoritmos de anonimização (ver Secção 4.3.4).

A Figura 4.1 apresenta um esquema de alto nível de como a *framework* está construída. Nesta figura as caixas são os processos de deteção e anonimização e estes são independentes um do outro. Os cilindros representam as bases de dados e as setas pretas, entre estes e os processos, significam que são requisitados dados pelos processos várias vezes. A seta branca entre os processos simboliza o envio dos metadados de um processo para o outro e a seta branca entre as bases de dados simboliza o envio dos dados ou o processo de cópia.

4.3 Implementação

Neste secção, irá ser descrito com maior detalhe toda a implementação e o trabalho referido na Secção 4.2.

A implementação foi feita alternando o foco entre a deteção e a anonimização, isto é, implementava a deteção de alguns tipos de dados, por exemplo Nomes, Emails e Moradas, e depois implementava a anonimização dos mesmos na ferramenta Anonimatron.

Antes de se implementar cada método para detetar e anonimizar os vários tipos de dados foi feita uma análise e estudo sobre as características de cada um deles para perceber como se podia detetá-los e anonimizá-los de forma a que os dados pareçam realistas.

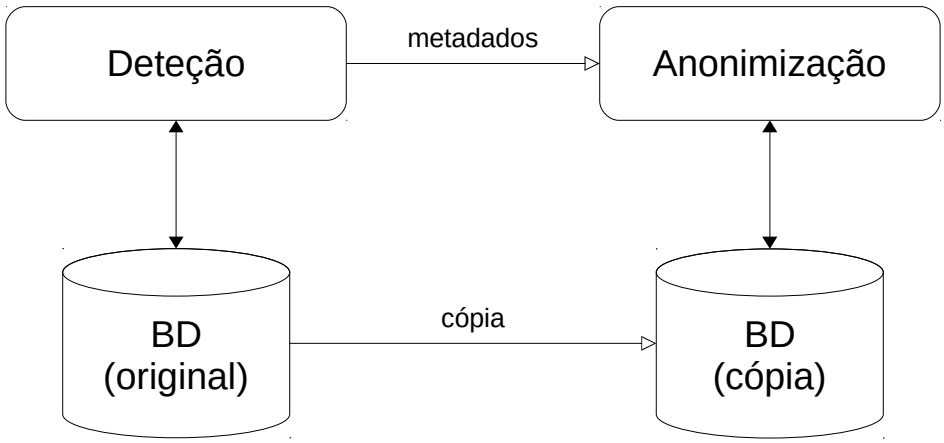

Figura 4.1: Diagrama com o funcionamento da *framework*

4.3.1 Ferramentas e *frameworks* de suporte

Para este projeto, fiz uso das seguintes ferramentas e *frameworks* de suporte:

- **Apache Maven** (https://maven.apache.org/), mais conhecido apenas por **Maven**, é uma ferramenta para ajudar a construir e gerir projetos de *software*. Embora suporte várias linguagens de programação, é maioritariamente usado em projetos Java. Esta ferramenta tem como objetivo facilitar o *build* e compilação do projeto para o desenvolvedor, lidando com dependências e automatizando várias tarefas. A Maven é usado no projeto Java da deteção e da anonimização.

- **Docker** (https://www.docker.com/) é um *software* que permite construir, testar, implementar e partilhar aplicações. O *software* fica empacotado em *docker images* que podem ter bibliotecas, código, ferramentas, entre outros. Um *container* é uma instância de uma *docker image* sendo que uma imagem descreve a aplicação e como ela irá correr. As bases de dados deste projeto estão contidas dentro de *containers*.

- **Dbeaver** é uma ferramenta *open-source* para trabalhar com bases de dados, com o propósito de visualizar e administrar os dados, suportando várias bases de dados populares. Foi usado durante todo o desenvolvimento do projeto para criar e alterar os dados fictícios para testes e visualizar o resultados da anonimização.

4.3.2 Copiar a base de dados

A base de dados do projeto TeamingUp é uma base de dados PostgreSQL que se encontra num *docker container*. Para criar uma cópia desta base de dados num outro *container* é necessário ter o PostgreSQL e o Docker instalados na máquina.

É criado um ficheiro *docker-compose*, que é usado para correr o novo *docker container* que é uma instância de uma *docker image* que irá conter a base de dados copiada. Este

ficheiro usa uma *docker image* com versão 10.7 de PostgreSQL retirada do Docker Hub[1] e é definido nele qual a palavra-passe para aceder à nova da base de dados.

Este processo começa colocando o *container* em cima com uma base de dados padrão. De seguida são executados dois comandos que vêm na instalação de PostgreSQL: *pg_dump* e *psql*. O primeiro extrai a base de dados para um ficheiro com uma sequência de comandos SQL que replicam a base de dados, e o segundo usa esse ficheiro para criar a nova base de dados, cópia da original.

4.3.3 Detetar os campos a anonimizar

Antes de tudo, é feita uma ligação à base de dados através das credenciais de acesso fornecidas num ficheiro *.properties* para se obter informação acerca de que tabelas existem, as respetivas colunas e o tipo de dados que cada uma guarda. Essa informação é guardada num mapa chave-valor, sendo as chaves os nomes das tabelas e os valores uma lista de Coluna, tendo cada objeto Coluna guardado o nome, o tipo de dados e uma *flag* que mudará o seu valor quando for validada por um algoritmo para anonimização.

Por uma questão de eficiência, todos os algoritmos de deteção verificam apenas um subconjunto das entradas de uma coluna. Mais concretamente, apenas as 100 primeiras entradas *com valores válidos* de uma dada coluna são verificadas. Por exemplo, se nas 100 primeiras entradas existir uma entrada com valor "null" e outra com uma string vazia, então essas duas não contarão para o limite de 100 entradas válidas, e irão então ser lidas mais duas entradas válidas antes de parar o algoritmo.

Depois de um algoritmo ter terminado de percorrer a coluna, é necessário decidir se a coluna deve ser assinalada como contendo dados sensíveis. Após alguma experimentação ao longo do desenvolvimento, decidiu-se a condição para assinalar se a coluna contém dados sensíveis ou não. A decisão é feita com base no número de entradas reconhecidas, segundo esta condição:

- Se, das 100 primeiras entradas *com valores válidos* (i.e. sem serem "null" ou strings vazias), pelo menos 70% forem confirmadas pelo algoritmo como sendo um dado sensível, então a coluna é assinalada para anonimização.

A informação de que uma coluna foi detetada para o processo de anonimização é guardada, ativando a *flag* do respetivo objeto Coluna, dispensando que os restantes algoritmos considerem essa coluna.

No final do processo de deteção é construído um ficheiro XML com as tabelas e respetivas coluna detetadas para anonimização. Este XML terá também uma *tag* em cada elemento *column* com o nome da coluna e o tipo de anonimização, do Anonimatron, sugerido para usar.

[1] O Docker Hub (https://hub.docker.com/) é um repositório de *container images* já preparadas a usar.

Os únicos tipos de dados que não seguem este processo são as datas e os *timestamps*. Para estes, não é necessário percorrer quaisquer entradas pois a própria base de dados já especifica o tipo da coluna (tipo "date" ou tipo "timestamp").

Métodos baseados em expressões regulares

Foram construído métodos para detetar os seguintes tipos de dados baseados em expressões regulares: números de telemóveis portugueses, emails, números de cartão de cidadão (CC), números de identificação fiscal (NIF), números de identificação de segurança social (NISS), códigos postais portugueses, e IBANs dos países pertencentes à *Single Euro Payments Area* (SEPA).

Para detetar números de telemóvel portugueses, emails, números de cartão de cidadão, códigos postais e IBANs foi usada uma expressão regular para cada um deles, verificando se uma entrada corresponde exatamente à expressão regular, isto é, uma entrada só passa na regra se não tiver mais nada no campo para além do que se encontra na regra. Por exemplo, uma entrada "1801-239" será detetada como um código postal, no entanto se a entrada for "Rua Amarela, 1801-239 Lisboa" já não será detetado pela regra, pois não irá procurar pela informação de código postal no meio de frases. As expressões regulares também são insensíveis a maiúsculas e minúsculas, para cobrir casos onde possam estar dados guardados de forma diferente.

Números de telemóvel. Os números de telemóvel portugueses são detetados pela seguinte expressão regular.

```
^(?:(?:\+|00)351)?9[1236]\d{7}$
```

Isto é, um número é reconhecido como sendo um número de telemóvel português se o primeiro dígito é um 9 seguido de 3, 6, 2 ou 1, sendo que os últimos sete dígitos podem ter valores de 0 a 9. O número pode, opcionalmente, ser prefixado pelo indicativo do país nas suas várias formas possíveis ("+351", "00351" ou "351").

Emails. Os endereços de email são detetados pela seguinte expressão regular.

```
^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$
```

Números de cartão de cidadão. Os números de cartão de cidadão são detetados pela seguinte expressão regular.

```
^([0-9]{7,8}|[0-9]{7,8}[0-9][A-Z0-9]{2}[0-9])$
```

Este padrão encontra correspondência com uma sequência com 7 (números mais antigos) ou 8 dígitos, sendo assim o número validado como um cartão de cidadão. Caso a premissa anterior não se verifique, é verificado se é uma sequência com um tamanho de 11 (números mais antigos) ou 12 caracteres com o seguinte formato: começa com 7 ou 8 dígitos seguido de 1 dígito (dígito de controlo), 2 caracteres que podem ser letras ou números (versão do CC) e por fim 1 dígito (dígito de controlo).

Números de identificação fiscal. Os números de identificação fiscal são detetados pela seguinte expressão regular.

```
^\d{9}$
```

Este padrão é bastante simples, limitando-se a procurar correspondência com qualquer número que tenha 9 dígitos.

Números de identificação de segurança social. Os números de identificação fiscal são detetados pela seguinte expressão regular.

```
^(\d{9}|\d{11})$
```

Procura-se a correspondência com um número de 9 dígitos (números mais antigos) ou 11 dígitos (números mais recentes).

É de notar que, quer este padrão quer o anterior, fazem correspondência com números com 9 dígitos. Mais abaixo voltaremos ao assunto de tipos de dados que podem ser reconhecidos por mais de um padrão, e qual a solução adotada.

Código postal. Os códigos postais são detetados pela seguinte expressão regular.

```
^[1-9]\d{3}-\d{3}$
```

Números IBAN. Os números internacionais de conta bancária são detetados por uma expressão regular que restringe o formato com base no código do país. A expressão regular é uma disjunção das seguintes sub-expressões regulares sendo que, dependendo do código do país (as duas primeiras letras), o resto do IBAN tem diferentes formatos.

```
^(?:((?:IT|SM)\d{2}[A-Z]{1}\d{22})|
(NL\d{2}[A-Z]{4}\d{10})|
(LV\d{2}[A-Z]{4}\d{13})|
((?:BG|GB|IE)\d{2}[A-Z]{4}\d{14})|
(GI\d{2}[A-Z]{4}\d{15})|
(RO\d{2}[A-Z]{4}\d{16})|
```

```
(MT\d{2}[A-Z]{4}\d{23})|
(NO\d{13})|
((?:DK|FI)\d{16})|
((?:SI)\d{17})|
((?:AT|EE|LU|LT)\d{18})|
((?:HR|LI|CH)\d{19})|
((?:DE|VA)\d{20})|
((?:AD|CZ|ES|MD|SK|SE)\d{22})|
(PT\d{23})|
((?:IS)\d{24})|
((?:BE)\d{14})|
((?:FR|MC|GR)\d{25})|
((?:PL|HU|CY)\d{26}))$
```

Em todos estes algoritmos, antes de ser usada a expressão regular, são removido os espaços que possam existir numa entrada entre palavras ou caracteres.

Métodos baseados em heurísticas e listas

A deteção com base em heurísticas e listas permite detetar **moradas** e **nomes** portugueses.

São usadas listas de palavras para detetar as moradas e nomes, procurando por palavras destas listas nas entradas, através de uma expressão regular com a palavra que se está à procura. Estas palavras, ao contrário dos métodos explicados na secção anterior, são detetadas quando se encontram no meio de frases. Além disso, as palavras são reconhecidas mesmo que tenham letras maiúsculas ou minúsculas (i.e. a busca é *case insensitive*).

Por exemplo, tanto a entrada "Avenida Pinhal do Lago, nº32", como a entrada "praça" serão reconhecidas como uma morada pelo algoritmo, porque quer a palavra "Avenida" quer a palavra "Praça" se encontram na lista de palavras usada para detetar moradas.

A lista de nomes usada na deteção contém 2344 nomes próprios e apelidos portugueses e a lista usada para as moradas contém 26 palavras de ocorrência comum em moradas. A lista de nomes próprios e de apelidos foi compilada por mim, com base numa pesquisa dos nomes mais comuns portugueses, tendo sido posteriormente estendida com uma lista de nomes fornecida pela empresa. A lista de palavras para as moradas foi construída com base em buscas por várias moradas ao longo do país, em território continental e ilhas.

Dependência da ordem

A ordem pela qual os algoritmos são corridos sobre uma dada coluna é importante, pois pode mudar o tipo que é atribuído à coluna.

Por exemplo, o algoritmo da deteção de nomes pode reconhecer uma coluna com moradas como sendo uma coluna de nomes, porque muitas moradas contêm nomes de

pessoas. Isto pode também acontecer para os algoritmos que detetam NIFs e NISSs, pois ambos procuram por números com 9 dígitos.

Por este motivo o algoritmo de deteção de moradas é executado primeiro que o de nomes e que o de códigos postais e o algoritmo de NIFs primeiro que o de NISSs, relembrando que não se volta a correr mais nenhum algoritmo numa coluna que já foi detetada por algum algoritmo.

4.3.4 Estender o Anonimatron

O código da ferramenta Anonimatron foi retirado por inteiro do GitHub.[2] O objetivo era criar novos tipos de anonimização, que não existiam na ferramenta, para diferentes tipos de dados. Como é preferência da empresa a pseudonimização dos dados, optou-se por usar este tipo de anonimização em todos as novas extensões da ferramenta.

Para criar um novo *anonymizer* (o termo usado no Anonimatron para uma classe que implementa um tipo de anonimização) é necessário criar uma nova classe Java que implemente a interface Anonymizer e registar este novo anonymizer na classe AnonymizerService para que se torne visível para a ferramenta e seja possível usá-lo.

A interface Anonymizer obriga a que sejam implementados os seguintes dois métodos:

getType() que devolve uma String que é o nome do anonymizer. Este nome é usado no XML de configuração que o Anonimatron recebe (ver Secção 4.3.5) para indicar que anonymizer aplicar a cada coluna da base de dados.

anonymize(Object from, int size, boolean shortlived) que recebe como parâmetros a entrada da base de dados a ser anonimizada (from), o tamanho máximo que a nova entrada gerada pode ter (size) e um flag que indica que não queremos manter a consistência desta entrada na próxima anonimização (shortlived), não sendo então guardada no ficheiro de sinónimos.

Este método deve devolver um objeto Synonym, com os seguintes atributos: o valor retornado pelo **getType()**, o objecto from, um objeto com a nova entrada gerada (podendo ser String, Integer, Date, entre outros), e a flag shortlived. Este Synonym irá ser usado posteriormente pelo Anonimatron para modificar a entrada da base de dados e guardar as informações necessárias para manter a consistência entre diferentes execuções de anonimizações.

Anonymizers criados

Os novo anonymizers criados no âmbito deste trabalho foram:

[2] O GitHub (https://github.com/) é uma plataforma online de desenvolvimento de *software* baseada no sistema de controlo de versão Git.

MobilePhoneNumberPTAnonymizer para pseudonimizar números de telemóveis portugueses. Gera uma String que começa com "91", "92", "93" ou "96", seguida de sete dígitos aleatórios.

FiscalDocumentNumPTAnonymizer para pseudonimizar NIFs. Gera uma String com nove dígitos, sendo que o primeiro dígito é igual ao primeiro dígito do número original. Os sete dígitos seguintes são escolhidos aleatoriamente e o último dígito, o dígito de controlo, é calculado usando uma função para que o NIF gerado seja válido. O cálculo do dígito de controlo usa os oito dígitos já adquiridos e pode ser visto no Algoritmo 1 apresentado de seguida.

Algorithm 1 Cálculo do dígito de controlo do NIF

Require: $value =$ número de oito dígitos
  for $i = 0, 1, \ldots, 7$ **do**
    $sum \leftarrow sum + (value[i] \times (8 + 1 - i)$
  end for
  $mod \leftarrow sum \bmod 11$
  if mod igual a 0 ou 1 **then**
    return 0
  else
    return $11 - mod$
  end if

IdentityDocumentNumPTAnonymizer para a pseudonimização de números de cartão de cidadão. Gera uma String com doze caracteres começando com oito dígitos aleatórios e depois um dígito de controlo usando o mesmo cálculo já usado para o dígito de controlo do NIF (ver Algoritmo 1). Os dois caracteres seguintes são duas letras maiúsculas do alfabeto português, escolhidas aleatoriamente.[3] Esta escolha de usar apenas as letras torna a pseudonimização mais realista. O último carácter é um dígito de controlo do número todo e é calculado através da função descrita no Algoritmo 2 que usa um mapa para converter as letras maiúsculas para valores numéricos, tendo "A" o valor 10 por ordem crescente até ao "Z" com valor 35, enquanto que dígitos são mapeados para os seus respetivos valores numéricos.

SocialDocumentNumPTAnonymizer para pseudonimizar números de segurança social. Gera um número aleatório com onze dígitos.

FirstNamePTAnonymizer para pseudonimizar um nome português. Recorre a uma lista com 2248 nomes próprios, escolhendo um deles aleatoriamente. Esta classe estende

[3] Este código de dois carateres representa a emissão do cartão de cidadão. Na primeira emissão, este código é "ZZ", na segunda é "ZY", na terceira é "ZX", etc., seguindo a mesma lógica decrescente para cada vez que se renova o documento [1].

Algorithm 2 Cálculo do dígito de controlo do CC

Require: $value =$ uma palavra com nove dígitos seguidos de duas letras maiúsculas
  for $i = 10, 9, \ldots, 0$ **do**
    $d \leftarrow$ mapear($value[i]$)
    if $i \bmod 2$ igual a 0 **then**
      $d \leftarrow d \times 2$
      if $d > 9$ **then**
        $d \leftarrow d - 9$
      end if
    end if$sum \leftarrow sum + d$
  end for
  return $(10 - (sum \bmod 10)) \bmod 10$

uma classe abstrata, também criada no âmbito deste projeto, designada *AbstractNameRandom*.

A classe ***AbstractNameRandom*** foi implementada para que possa receber ficheiros com um nome em cada linha e ser estendida por diferentes *anonymizers* com diferentes algoritmos que usam listas.

LastNamePTAnonymizer para pseudonimizar um apelido português. Funciona de forma similar à classe anterior, recorrendo a uma lista de 96 apelidos e escolhendo um deles aleatoriamente. Esta classe estende a classe abstrata *AbstractNameRandom*.

FullNamePTAnonymizer para pseudonimizar um nome completo português. Este *anonymizer* recorre às duas classes anteriores, FirstNamePTAnonymizer e LastNamePTAnonymizer, para gerar dois nomes próprios e um apelido.

AddressPTAnonymizer para pseudonimizar moradas portuguesas. Estende a classe *AbstractNameRandom* para usar uma lista de palavras para construir uma morada falsa. Irá gerar moradas como "Rua Cláudio Tavares nº15", usando, também, as listas de nomes próprios e apelidos e uma lista de 22 palavra que costumam ser as primeiras palavras de uma morada (e.g. "avenida", "rua", etc).

Começa por adicionar uma palavra da lista de moradas, por exemplo "Alameda". Depois, de forma aleatória, decide se a morada terá um ou dois nomes próprios, que são obtidos de forma aleatória da lista de nomes, como por exemplo "Filipe Tomás". Depois dos nomes próprios é adicionado um apelido, novamente escolhido aleatoriamente da lista de apelidos, por exemplo "Araújo". Finalmente é acrescentado "nºx" com $1 \leq x \leq 30$, escolhido aleatoriamente, chegando à morada final "Alameda Filipe Tomás Araújo nº7", por exemplo.

EmailAddressPTAnonymizer para pseudonimizar emails portugueses. Este anonymizer gera nomes de conta de email de acordo com uma de três formas *template*, esco-

lhida aleatoriamente: i) nome próprio seguido de apelido; ii) nome próprio seguido de uma letra aleatória, um ponto final e o apelido; ou iii) dois nomes próprios, um ponto final e um apelido. Por exemplo, "franciscoramos", "franciscot.ramos" ou "franciscoeduardo.ramos". Finalmente, ao nome de conta gerado é adicionado um de quatro domínios possíveis, também escolhido aleatoriamente: "@gmail.com", "@hotmail.com", "@outlook.com" ou "@yahoo.com".

ZipCodePTAnonymizer para pseudonimizar códigos postais portugueses. Gera uma String começando com um número aleatório de 1000 a 9999 seguido de um hífen e de mais três dígitos aleatórios, por exemplo "9732-017".

IbanPTAnonymizer para pseudonimizar IBANs portugueses. O IBAN português tem vinte e cinco caracteres, por exemplo "PT5000596169595976238678030". Os primeiros quatro caracteres "PT50" são sempre os mesmos para IBANs portugueses e o número do IBAN sem estes quatro caracteres é chamado de Número de Identificação Bancária (NIB). Os quatro dígitos seguintes, "0059" neste exemplo, são o código do banco. No processo de anonimização está guardado num array com vinte e quatro códigos de bancos, sendo escolhido um aleatoriamente para gerar um IBAN. Os quinze dígitos seguintes, "616959762386780" neste exemplo, são gerados aleatoriamente. Os últimos dois dígitos, "30" neste exemplo, são dígitos de controlo, sendo gerados através de um cálculo (ver Algoritmo 3) usando o número já construído exceto os primeiros quatro caracteres ("PT50"), isto é, o NIB exceto os dois últimos dígitos que irão ser calculados.

Algorithm 3 Cálculo dos dois dígitos de controlo do IBAN

Require: $value$ = NIB exceto os dois últimos dígitos, total de dezanove dígitos
  for $i = 0, 1, \ldots, 18$ **do**
    $d \leftarrow value[i]$
    $d \leftarrow d \times 10 \bmod 97$
  end for
  $d \leftarrow 98 - (d \times 10 \bmod 97)$
  if $d > 9$ **then**
    return $toString(d)$
  else
    return "0" + $toString(d)$
  end if

4.3.5 Correr a Anonimização

No final da deteção é então gerado um ficheiro XML que é necessário o Anonimatron receber para o processo de anonimização. Um excerto deste ficheiro pode ser visto na Figura 4.2.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<configuration jdbcurl="jdbc:postgresql://localhost:13000/postgres" password="secret_password" userid="postgres">
    <table name="collaborator">
        <column name="identity_doc_num" type="ID_CC_DOC_NUMBER_PT"/>
        <column name="identity_doc_val" type="DATE"/>
        <column name="passport_val" type="DATE"/>
        <column name="tax_doc_num" type="FISCAL_DOC_NUMBER_PT"/>
        <column name="ss_doc_num" type="SOCIAL_DOC_NUMBER_PT"/>
        <column name="birth_date" type="DATE"/>
        <column name="tel_num" type="MOBILE_PHONE_NUMBER_PT"/>
        <column name="address_household" type="ADDRESS_PT"/>
        <column name="address_zip_code" type="ZIP_CODE_PT"/>
        <column name="iban" type="IBAN_PT"/>
        <column name="school_finish_date" type="DATE"/>
        <column name="email_personal" type="EMAIL_ADDRESS_PT"/>
        <column name="vc_tel_num" type="MOBILE_PHONE_NUMBER_PT"/>
        <column name="vc_zip_code" type="ZIP_CODE_PT"/>
        <column name="cmp_responsible_name" type="FULL_NAME_PT"/>
        <column name="coach_email" type="EMAIL_ADDRESS_PT"/>
        <column name="cmp_work_healthcare" type="DATE"/>
        <column name="cmp_group_entry_date" type="DATE"/>
        <column name="cmp_group_leave_date" type="DATE"/>
        <column name="cmp_food_card_num" type="FISCAL_DOC_NUMBER_PT"/>
        <column name="cmp_food_card_delivery_date" type="DATE"/>
        <column name="cmp_food_card_val" type="DATE"/>
        <column name="manager_name" type="FULL_NAME_PT"/>
        <column name="manager_email" type="EMAIL_ADDRESS_PT"/>
        <column name="coach_name" type="FULL_NAME_PT"/>
    </table>
    <table name="collaborator_email">
        <column name="email" type="EMAIL_ADDRESS_PT"/>
    </table>
```

Figura 4.2: Excerto do ficheiro gerado pela deteção.

A informação começa com um elemento *configuration* que tem os dados de conexão da base de dados que irá ser anonimizada, que é a cópia da base de dados original. Seguem-se elementos *table*, um por cada tabela da base de dados, com os respetivos sub-elementos *column* que indicam o nome das colunas detetadas e os nomes dos *anonimyzers* que o Anonimatron irá aplicar a essas colunas. É neste ponto, alterando o nome do *anonymizer* indicado pelo atributo *type*, que o utilizador pode escolher que processo de anonimização é aplicado à coluna.

Para correr a anonimização, basta executar um *script* que irá fazer a cópia da base de dados e depois irá executar o Anonimatron para anonimizar a cópia da base de dados, usando os algoritmos indicados no ficheiro XML para cada coluna.

Um excerto da base de dados, antes e depois do processo de anonimização ser executado, pode ser visto na Figura 4.3. Neste exemplo, é de realçar como o processo de pseudonimização produz valores com um aspeto realista e formato válido, e como as garantias de consistência dadas pelo Anonimatron fazem com que valores iguais no excerto original (“António Costa”) sejam pseudonimizados para valores iguais (“Erique Eponina Abreu”).

identity_doc_num	address_household	iban	manager_name
12345678	estrada amália rodrigues 25, 2ºB	PT50123412341234123405	António Costa
17022686 7 ZX9	Grande Praça do Duarte, nº 25, 1ºDt	PT50001058066465114519769	António Costa
1234567	7th Street nº3 A	pt50000773710344732262491	Marta Alves
16406873 2 ZZ5	Rua Joaquim, nº 12, 3ºC	PT50000761681959105719989	Helton Saidi
34125097ZX6	Rua das Flores, nº1	BG18RZBB91550123456789	Beatriz Couto Rego

(a) Excerto original

identity_doc_num	address_household	iban	manager_name
524959862KE4	Campo Simoneta Acácio Assunção nº 3	PT50001017686726842048185	Erique Eponina Abreu
139155392KL7	Rampa Carela Fred Cruz nº 1	PT50003862650842739454006	Erique Eponina Abreu
581119991KZ6	Praça Josiana Moreira nº 23	PT50003680437080618721047	Loela Ingeburga Pereira
221895523ID6	Vereda Sátira Gaspar nº 4	PT50005912958274585260410	Cárin Dulcídia Antunes
478602561YY8	Jardim Celina Oriana Simões nº 15	PT50003439665590909418069	Carlota Nicanor Pinho

(b) Excerto anonimizado

Figura 4.3: Excerto da base de dados antes e depois da anonimização

4.3.6 Estrutura do projeto

Inicialmente, pretendia-se que um único projeto Java usando Maven tratasse da deteção e da anonimização, usando o Anonimatron como uma biblioteca Maven. No entanto, foram surgindo dificuldades e necessidades que só podiam ser resolvidas alterando o próprio código Java da ferramenta Anonimatron, pois este não se encontra desenhado da forma mais conveniente para ser estendido como biblioteca. Decidimos, então, alterar a forma como estávamos a construir a *framework*, passando de um único projeto Maven para dois projectos Maven, um com a deteção e outro com a anonimização.

Ficámos assim com a seguinte estrutura geral:

- Uma pasta com o código da deteção e o respetivo JAR.
- Uma pasta com o código do Anonimatron estendido e o respetivo JAR.
- Uma pasta com os recursos (ficheiros) necessários, como dicionários, ficheiros XML e um ficheiro *properties* a ser preenchido pelo utilizador com as informações para a ligação à base de dados que será anonimizada.
- Uma pasta com os *scripts* necessários para correr a *framework*.

Esta alteração da estrutura facilitou, também, a integração de novos algoritmos de anonimização e simplificou bastante a compilação e execução dos mesmos no Anonimatron. Antes da alteração era necessário colocar a classe java, com o novo algoritmo de anonimização, na pasta principal da ferramenta, juntamente com o respetivo ficheiro de compilação *.class* e também era necessário acrescentar uma linha a indicar esse ficheiro no XML que o Anonimatron recebe.

```
db.url=jdbc:postgresql://localhost:14020/postgres
db.user=bob
db.password=bobpassword
```

Figura 4.4: Ficheiro *.properties* com os dados de conexão à base de dados.

4.4 Execução da framework

Nesta secção irá ser demonstrado, passo a passo, o funcionamento por inteiro da *framework*, corrida com a base de dados fictícia do TeamingUp.

Antes de correr a *framework*, é necessário especificar como aceder à base de dados (url, user e password). Esta informação é colocada num ficheiro *.properties* designado *config.properties* na pasta *resources*. Na Figura 4.4 podemos ver um exemplo do ficheiro *config.properties*.

Começa-se por correr o processo de deteção executando o *script runDetection.bat*, que se encontra na pasta *scripts*, que irá criar o ficheiro *DetectionToUse.xml* com os metadados sobre os campos detetados. Na Figura 4.5 podemos observar algum do output do processo de deteção onde se indica a tabela e depois a coluna que irá ser verificada, assim como os métodos de deteção aplicados às entradas dessa coluna.

O processo de deteção acaba dando a informação de que foi criado um ficheiro XML, como aquele exemplificado na Figura 4.2, com os metadados sobre as colunas identificadas como contendo dados pessoais. Neste ponto, o utilizador pode editar este ficheiro para ajustar os *anonymizers* que ião ser aplicados a cada coluna, apagar as linhas referentes a colunas que não se quer anonimizar, ou acrescentar colunas que não tenham sido detetadas.

Por fim, executa-se o *script runCopyAnonimatron.bat* que irá fazer uma cópia da base de dados e correr a ferramenta Anonimatron sobre esta cópia. O processo da cópia começa sempre por tentar apagar o *container* que irá criar para o caso de este já existir, isto é, de já ter sido corrida a *framework* anteriormente. Na Figura 4.6 vemos que depois de inicializar o *container* com a base de dados vazia, que irá executar um comando para copiar e que pede as palavras passe para se autenticar em ambas as bases de dados.

A Figura 4.7 mostra o fim do processo da cópia. As primeiras linhas da consola informam da alteração de tabelas, seguidas de informação dada pelo Anonimatron de que terminou o seu trabalho de anonimização.

O ficheiro README que acompanha a *framework*, e que descreve este processo de forma mais minuciosa, pode ser visto no Anexo B.

```
181654 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT ZipCodePT ---
181654 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT IBAN ---
181654 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT Date ---
181654 [main] INFO org.example.detection.business.DetectionHandler  - TABLE: addrfeat
181654 [main] INFO org.example.detection.business.DetectionHandler  - COLUMN: the_geom
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT EmailPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT AddressPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT NamePT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT MobilePhoneNumberPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT FiscalDocumentNumPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT SocialDocumentNumPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT IdentityDocumentNumPT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT ZipCodePT ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT IBAN ---
181656 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT Date ---
181657 [main] INFO org.example.detection.business.DetectionHandler  - TABLE: addrfeat
181657 [main] INFO org.example.detection.business.DetectionHandler  - COLUMN: gid
181659 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT EmailPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT AddressPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT NamePT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT MobilePhoneNumberPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT FiscalDocumentNumPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT SocialDocumentNumPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT IdentityDocumentNumPT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT ZipCodePT ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT IBAN ---
181660 [main] INFO org.example.detection.business.DetectionHandler  - --- DETECT Date ---
181670 [main] INFO org.example.detection.business.DetectionHandler  - Done creating XML File
181681 [main] INFO org.example.detection.business.DetectionHandler  - Done creating XML File
```

Figura 4.5: Fim do processo de deteção.

```
C:\devenv\teseproject\scripts\postgres-compose>call down.bat

C:\devenv\teseproject\scripts\postgres-compose>rem docker-compose config

C:\devenv\teseproject\scripts\postgres-compose>docker-compose down
[+] Running 2/2
 - Container postgres-compose_anonymization_db_1  Removed                                                    1.4s
 - Network postgres-compose_default               Removed                                                    0.7s

C:\devenv\teseproject\scripts\postgres-compose>call delete-volume.bat

C:\devenv\teseproject\scripts\postgres-compose>docker volume rm postgres-compose_db
postgres-compose_db

C:\devenv\teseproject\scripts\postgres-compose>call start.bat

C:\devenv\teseproject\scripts\postgres-compose>rem docker-compose config

C:\devenv\teseproject\scripts\postgres-compose>docker-compose up -d
[+] Running 3/3
 - Network postgres-compose_default               Created                                                    0.9s
 - Volume "postgres-compose_db"                   Created                                                    0.0s
 - Container postgres-compose-anonymization_db-1  Started                                                    2.4s

C:\devenv\teseproject\scripts\postgres-compose>ping 127.0.0.1 -n 11  1>nul

C:\devenv\teseproject\scripts\postgres-compose>pg_dump -h localhost -p 14021 -U postgres -W   | psql -h localhost -p 130
00 -U postgres
Password: Password for user postgres:
```

Figura 4.6: Início do processo da cópia.

```
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE

C:\devenv\teseproject\scripts\postgres-compose>cd ..

C:\devenv\teseproject\scripts>call runAnonimatronJar.bat

C:\devenv\teseproject\scripts>cd ../anonimatron-1.15

C:\devenv\teseproject\anonimatron-1.15>call java8
Java 8 activated.
Reading Synonyms from C:\devenv\teseproject\anonimatron-1.15\..\resources\synonyms.xml ...[done].

Anonymization process started

Jdbc url      : jdbc:postgresql://localhost:13000/postgres
Database user : postgres
To do         : 7 tables.

Anonymizing table 'company', total progress  [100%, ETA 12:10:05 PM] PM]
Anonymization process completed.

Writing Synonyms to C:\devenv\teseproject\anonimatron-1.15\..\resources\synonyms.xml ...[done].
```

Figura 4.7: Fim do processo de anonimização.

4.5 Avaliação

Para a avaliação, é importante correr a ferramenta sobre um conjunto de dados novo, diferente daquele usado durante o desenvolvimento, pois isto dá-nos uma melhor ideia do desempenho "real" da ferramenta, que virá a ser aplicada sobre bases de dados diferentes daquela que guiou a implementação.

Assim, foi fornecida pela empresa uma base de dados semelhantes à base de dados do TeamingUp. Usando esta nova base de dados, a deteção encontrou as seguintes colunas das várias tabelas:

Colunas detetadas

- Na tabela *creationarea*:

  reponsable detetada como uma coluna de nomes;

  email detetada como uma coluna de emails.

- Na tabela *users*:

  UserName detetada como uma coluna de nomes.

- Na tabela *employeesaccesses*:

parameter detetada como uma coluna de emails.

- Na tabela *dependents*:

  Name detetada como uma coluna de nomes.

  Birthday detetada como uma coluna de datas.

- Na tabela *employees*:

  ShortName detetada como uma coluna de nomes.

  EntryDate detetada como uma coluna de datas.

  FIN detetada como uma coluna de NIFs.

  emailContact detetada como uma coluna de emails.

  TelephoneNumber detetada como uma coluna de números de telemóveis.

  ExitDate detetada como uma coluna de datas.

  GroupEntry detetada como uma coluna de datas.

- Na tabela *managers*:

  managerName detetada como uma coluna de nomes.

  managerEmail detetada como uma coluna de emails.

  managerLogin detetada como uma coluna de nomes.

- Na tabela *icaretemp*:

  Name detetada como uma coluna de nomes.

  Email detetada como uma coluna de emails.

  CCNumber detetada como uma coluna de CCs.

  ExpirationDate detetada como uma coluna de datas.

  SSN detetada como uma coluna de números de NISSs.

  Birthday detetada como uma coluna de datas.

  BankId detetada como uma coluna de IBANs.

  AcademicDegreeDate detetada como uma coluna de datas.

  FoodCardRef detetada como uma coluna de NIFs.

Destas colunas detetadas existem algumas que foram detetadas e não o deviam ter sido e outras que foram detetadas mas com o tipo errado:

- A coluna *FoodCardRef* foi detetada como sendo uma coluna de NIFs. Existe apenas uma entrada válida e essa entrada foi detetada como um NIF.
- As colunas *responsable*, *UserName* e *managerLogin* foram detetadas como contendo nomes próprios embora contenham usernames como "luis.p.amorim" e "joana.vilaverde".

No primeiro caso, o problema está em parte no facto de a base de dados de teste ter poucas entradas, o que faz com que o critério que o algoritmo verifica ao decidir se deve reconhecer a coluna seja aceite com base em poucos falsos positivos. No entanto, são casos indicativo de que é sempre possível que valores numéricos possam, por mero acaso, ter um formato que coincide com os tipos reconhecidos pela ferramenta.

No segundo caso, os usernames são detetados por conterem nomes próprios como parte da sequência.[4] Esta deteção errada acaba por ser "útil", porque estes dados podem ajudar a identificar um indivíduo ou vir a ser usados em acessos a algum sistema ou plataforma.

Colunas que escaparam à deteção

Neste processo de avaliação, não foram detetadas as seguintes colunas, apesar de conterem dados potencialmente sensíveis sobre indivíduos:

- Na tabela *academiclevels*:

  levelName dados sobre o nível de escolaridade de um indivíduo.

- Na tabela *civilstate*:

  civilName dados sobre o estado civil de um indivíduo.

  civilstateIRS dados sobre o estado civil de um indivíduo.

- Na tabela *genders*:

  genderName dados sobre o sexo de um indivíduo.

- Na tabela *icaretemp*:

  Address morada de um indivíduo.

  ZipCode código postal de um indivíduo.

  Passport número de passaporte de um indivíduo.

  Birthplace localidade onde nasceu o indivíduo.

[4] Como mencionado na Secção 4.3.3, os métodos baseados em listas—ao invés dos métodos baseados em expressões regulares—aceitam correspondência com *substrings* e são *case-insensitive*.

Nacionality nacionalidade de um indivíduo.

Os dados sobre a escolaridade, estado civil, sexo, número de passaporte, localidades e nacionalidades são dados que não têm a sua deteção implementada, por isso era de esperar que não fossem detetados.

A coluna *Address* não foi detetada como uma coluna de moradas porque as entradas da base de dados não são moradas portuguesas. Na verdade, são palavras com caracteres aleatórios, como "Mm. ukw Bwaftri Tº1 Rghba 6 P/H Wmu".

A coluna *ZipCode* não foi detetada como uma coluna de códigos postais, porque, como já foi explicado na Secção 4.3.3, a deteção de códigos postais apenas deteta um código postal numérico em isolamento e não o como parte de uma sequência maior. Um exemplo de uma entrada desta coluna é "2665-305 Milharado", que, por ter a palavra "Milharado", já não é detetado como um código postal.

Sumário

A base de dados usada para a avaliação tem vinte e cinco tabelas e são percorridas cento e cinquenta e sete colunas com o total de 1112 entradas no processo de deteção.

Resumindo, vinte e cinco colunas foram detetadas como colunas que contém dados pessoais. Destas vinte e cinco, quatro foram detetadas erradamente, como explicado acima. Nove colunas com dados sensíveis não foram assinaladas para o processo de anonimização, pelas razões já mencionadas.

Se excluirmos as colunas que nunca seriam detetadas, por ainda não ter sido implementado a deteção para esse tipo de dados, e a coluna *Address* por não ter moradas em português (também não foi implementado), então apenas uma coluna não foi detetada, sendo esta a coluna *ZipCode*.

Detetar colunas em excesso não é tão problemático quanto falhar na deteção de colunas, dado que as colunas detetadas passam sempre por um processo de validação.

Das quatro colunas que foram detetadas incorretamente, se pensarmos que três delas são importantes de serem alteradas pelo processo de anonimização, então só uma foi realmente detetada desnecessariamente.

O tempo que demorou a correr a deteção dos campos numa média de cinco execuções é 7,270 segundos, numa máquina com um processador com a velocidade de 2,42 GHz.

Capítulo 5

Conclusão

Este capítulo encerra o relatório com um sumário em comentário crítico ao trabalho realizado, e apresentado várias linhas de possível trabalho futuro.

5.1 Sumário

O direito à privacidade encontra-se consagrado na Declaração Universal dos Direitos do Humanos e, na Europa em particular, a *General Data Protection Regulation (GDPR)*, que regulamenta a recolha e partilha de dados relativos a pessoas da União Europeia, impõe várias restrições quanto ao que pode ser feito com dados pessoais.

Para que os dados pessoais em bases de dados sejam protegidos, não exponham a identidade de nenhum indivíduo e possam ser usados para estudos e melhoramento de serviços mesmo sem o consentimento da pessoa, necessitam, em primeiro lugar, de ser anonimizados para remover a possibilidade de identificação de indivíduos. Este processo pode ser inviável para as enormes quantidades de dados adquiridas nos dias de hoje, se for feito manualmente, pois, para além do tempo gasto em anonimizar os dados, existe ainda um esforço considerável em reconhecer os dados que se devem anonimizar. Estas considerações motivam o desenvolvimento de técnicas automáticas de anonimização.

Depois de uma análise das soluções de anonimização disponíveis, foi escolhida a ferramenta Anonimatron, que é capaz de anonimizar automaticamente e de forma consistente os dados de uma base de dados. Esta ferramenta, porém, requer que os campos a anonimizar já se encontrem identificados. Assim, neste projeto foi desenvolvida uma solução para a deteção automática de campos a anonimizar. Foi também necessário estender o Anonimatron para anonimizar novos tipos dados.

Depois de descrito o funcionamento e a implementação da solução automática que deteta e anonimiza dados pessoais e de ter sido testada, mostrou-se ser possível encontrar uma solução para o problema e aferir que se conseguiu alcançar a maioria dos objetivos e demonstrar um bom funcionamento desta solução, apesar de o seu desempenho não ser perfeito.

5.2 Comentário crítico

Existem, naturalmente, limitações e pontos que podem ser melhorados no funcionamento da *framework*.

- Existem ainda tipos de dados que não são detetados. Os tipos de dados a reconhecer pela *framework* foram escolhidos tendo em conta aqueles que achámos serem mais importantes no caso de uso explorado (a base de dados TeamingUp), assim como tentando englobar dados com diferentes características (padrões regulares e nomes).
- Os processos de anonimização por supressão e generalização não foram considerados aquando da criação de novos *anonymizers*. Foi decidido ter o foco apenas na pseudonimização, dado que este processo preserva uma maior utilidade dos dados.
- Existem limitações no que toca ao suporte de diferentes sistemas de bases de dados, sendo que a *framework* atual apenas aceita PostgreSQL. Este trabalho, no entanto, conseguiu demonstrar a viabilidade da abordagem de anonimização usada pela *framework*, e a extensão a outros sistemas de bases de dados não acrescentaria nada a esse aspeto concreto, ficando então para trabalho futuro.

Posto isto, e apesar de não terem sido realizados testes com medidas de risco, a avaliação que foi feita mostra que a solução implementada consegue detetar e anonimizar campos da base de dados, contribuindo assim para a preservação da privacidade.

5.3 Trabalho futuro

Este projeto estabeleceu as bases da framework de anonimização e demostrou a sua viabilidade. No entanto, há ainda muito por explorar e estender, assim como possibilidades de melhorias ao que foi feito. Nesta secção irei referir alguns dos caminhos de trabalho futuro mais relevantes.

Dependências e suporte de software

Para facilitar o desenvolvimento e uso da framework, é útil reduzir a dependência da máquina concreta onde a framework está a ser executada. Por exemplo, será conveniente contornar a necessidade de instalar localmente o PostgreSQL.

Para conseguir isto, é recomendado, no futuro, levantar um docker container para "simular" a máquina onde se corre a ferramenta e que comunica com os outros dois containers que contém as bases de dados.

De momento, a framework suporta apenas o sistema de base dados relacional PostgreSQL. É importante que outros sistemas como MySQL, MariaDB ou Oracle, possam ter suporte para a realização da cópia de um container para outro.

Novos tipos de dados

Estender os tipos de dados reconhecidos é muito importante e algo que, provavelmente, estará sempre a ser construído ao longo do tempo à medida que a framework for aplicada a novas bases de dados.

Além dos métodos de deteção já usados, uma ideia a explorar é estender as heurísticas de forma a que levem em consideração o nome da coluna sob análise. Embora não possamos controlar os nomes de coluna que o criador da base de dados escolheu, é plausível assumir que sejam nomes indicativos do conteúdo da coluna, e possam ser usados como componente da heurística.

Como um exemplo concreto encontrado neste projeto, foi feita uma pesquisa sobre o formato de número de passaportes portugueses para os tentar detetar. No entanto, não se conseguiu encontrar nenhuma definição do formato dos números de passaporte, impedindo assim de criar alguma expressão regular para os detetar. Caso não exista um formato fixo para números de passaporte portugueses, seria de tentar usar o nome da coluna para a reconhecer como uma coluna com números de passaporte.

Interface com o utilizador

Uma melhoria que beneficiaria os utilizadores desta solução seria uma interface gráfica para o utilizador, tornando tudo mais amigável, rápido e intuitivo.

Performance

Em termos de performance, pode-se tentar tornar mais veloz o processo de deteção através de paralelização correndo várias tarefas ao mesmo tempo dividindo-se por colunas ou tabelas, por exemplo, e usando novas métricas para limitar a procura na base de dados.

Existe outra possibilidade que pode ser feita pelos utilizadores ou, no futuro, implementada no processo de deteção, que consiste em guardar o ficheiro XML ou a informação deste ficheiro que a ferramenta cria com os campos detetados. Isto pode ser sempre feito pelos utilizadores, para que em futuros processos de anonimização sobre a mesma base de dados o resultado do processo de deteção possa ser reutilizado, tendo em atenção alterações que possam acontecer na base de dados ao longo do tempo ou implementar uma nova funcionalidade que poderá guardar esta informação da deteção na base de dados para ser usada novamente sem necessidade de correr o processo de deteção novamente.

Anonimização dinâmica

Como referido na Secção 3.2.1, a anonimização dinâmica foi descartada devido ao esforço extra que seria necessário para a implementar, dentro do âmbito. Ainda assim, alguma análise exploratória feita durante este projeto apontou para duas vias promissoras para implementar esta funcionalidade: Uma consiste em criar uma camada num projeto entre

a camada de dados e a "camada de negócio" e a outra consiste em recorrer à linguagem SQL para que a anonimização seja feita no momento em que o pedido é feito à base de dados dependendo de certos papeis dos utilizadores.

A adição desta funcionalidade necessita de uma maior investigação e análise para escolher mais conscientemente o melhor caminho a seguir.

Apêndice A

Listas

A.1 Termos de moradas

- Rua
- Praça
- Praca
- Avenida
- Travessa
- Beco
- Calçada
- Calcada
- Cais
- Largo
- Campo
- Estrada
- Caminho
- Canada
- Rampa
- Bairro
- Vereda
- Escadinhas
- Jardim
- Viela
- Alameda
- Quinta
- Praceta
- Urbanização
- Urbanizaçao
- Urbanizacao

A.2 Nomes próprios

- Aarão
- Abdénago
- Abdul
- Abel
- Abelâmio
- Abelardo
- Abigail
- Abílio
- Abna
- Abraão
- Abraim
- Abrão
- Absalão
- Acácio
- Ácil
- Acilino
- Acílio
- Açucena
- Acúrsio
- Ada
- Adail
- Adalberto
- Adalgisa
- Adália
- Adalsindo
- Adalsino
- Adamantino
- Adamastor
- Adão
- Adelaide
- Adélia
- Adélio
- Adelindo
- Adelina
- Adelino
- Adelmo
- Ademar
- Adeodato
- Adério
- Adérito
- Adiel
- Ádila
- Adília
- Adílio
- Adner
- Adolfo
- Adonai
- Adonias
- Adonilo
- Adónis
- Adorino
- Adosinda
- Adriana
- Adriano
- Adriel
- Adrien
- Adrualdo
- Adruzilo
- Afonsino
- Afonsina
- Afonso
- Afra
- Afrânio
- Afre
- Africana
- Africano
- Ágata
- Agenor
- Agna
- Agnelo
- Agnes
- Agostinho
- Águeda
- Aida
- Aidé
- Aires
- Airiza
- Airton
- Aitor
- Aixa
- Aladino
- Alaíde
- Alamiro
- Alan
- Alana
- Alano
- Alão
- Alba
- Albano
- Alberico
- Alberta
- Albertina
- Alberto
- Alcibíades
- Alcides
- Alcina
- Alcindo
- Alcino
- Alcíone
- Aldaír
- Aldara
- Aldemar
- Aldenir
- Aldenora
- Alder
- Aldo
- Aldo
- Aldónio
- Aldora
- Alegria
- Aleixa
- Aleta
- Aleu
- Alex
- Alexa
- Alexandra
- Alexandre
- Alexandrina
- Alexandrino
- Alexandro
- Aléxia
- Alexina
- Aléxio
- Aléxis
- Alfeu
- Alfreda
- Ália
- Aliana
- Aliça
- Alice
- Alícia
- Alida
- Alina
- Aline
- Alípio
- Alírio
- Alisande
- Álison
- Alita
- Alítio
- Alito
- Alivar
- Alix
- Alma
- Almara
- Almesinda
- Almira
- Almiro
- Aloís
- Aloísio

- Alpoim
- Altina
- Altino
- Alva
- Alvarim
- Alvarina
- Alvarino
- Alvário
- Alvino
- Alzira
- Amadeu
- Amadis
- Amado
- Amador
- Amália
- Amanda
- Amandina
- Amara
- Amarildo
- Amarílio
- Amarílis
- Amaro
- Amauri
- Amável
- Amélia
- Amelina
- América
- Américo
- Aminadabe
- Amor
- Amora
- Amorim
- Amorina
- Amorzinda
- Amós
- Ana
- Anabel
- Anabela
- Anael
- Anaíce
- Anaíde
- Anaim
- Anair
- Anaís
- Anaisa
- Anaísa
- Analdina
- Anália
- Analice
- Analide
- Analisa
- Anamar
- Anania
- Ananias
- Anás
- Anatilde
- André
- Andrea
- Andreia
- Andreias
- Andreína
- Andrelina
- Andreo
- Andrés
- Andresa
- Ândria
- Aneide
- Anésia
- Anfílito
- Anfíloco
- Angel
- Ângela
- Angélica
- Angélico
- Angelina
- Angelita
- Ângelo
- Ânia
- Aniana
- Anícia
- Anielo
- Aníria
- Anísia
- Anísio
- Anita
- Anolido
- Anquita
- Anselmo
- Anteia
- Antelmo
- Antera
- Antero
- Antonela
- Antonelo
- Antónia
- Antonieta
- Antonina
- António
- Anunciação
- Anunciada
- Anuque
- Anusca
- Aparecida
- Aparício
- Ápio
- Apolinário
- Apolo
- Aprígio
- Aquil
- Aquila
- Áquila
- Aquiles
- Aquilino
- Aquira
- Arabela
- Araci
- Aradna
- Aramis
- Arão
- Arcádio
- Arcanjo
- Arcelino
- Arcélio
- Arcílio
- Ardingue
- Argemiro
- Argentina
- Argentino
- Ari
- Ária
- Ariadna
- Ariadne
- Ariana
- Ariane
- Ariel
- Ariele
- Arinda
- Arine
- Ariosto
- Arisberto
- Aristides
- Aristóteles
- Arlanda
- Arlete
- Armandina
- Armandino
- Armando
- Armelim
- Arménia
- Arménio
- Armindo
- Aron
- Arquimedes
- Arquimínio
- Arquimino
- Arsénio
- Artemisa
- Artemísia
- Artur
- Aruna
- Ary
- Ascenso
- Asélio
- Áser
- Ásia
- Assis
- Assunção
- Assunta
- Astrid
- Astride
- Ataíde
- Atanásio
- Atão
- Atenais
- Átila
- Átina
- Aubri
- Audete
- Aura
- Áurea
- Aurélia
- Aureliana
- Áureo
- Aurete
- Auriana
- Ausenda
- Ausendo
- Austrelino
- Auta
- Auxília
- Ava
- Avelino
- Aventino
- Axel
- Azélio
- Aziz
- Azuil
- Baldemar
- Baldomero
- Banduíno
- Baltasar
- Baqui
- Barac
- Barão
- Bárbara
- Bárbora
- Barcino
- Bartolina
- Bartolomeu
- Basília
- Basílio
- Basilissa
- Bastião
- Batista
- Beanina
- Beatriz
- Bebiana
- Bebiano
- Bela
- Belchior
- Belém
- Belina
- Belinda
- Belisa
- Bendavida
- Benedita
- Benedito
- Benevenuto
- Benícia
- Benicio
- Benigna
- Benilde
- Benita
- Benjamim
- Benjamina
- Bento
- Benvinda
- Berardo
- Berengária
- Berilo
- Bernadete
- Bernardete
- Bernardim
- Bernardina
- Bernardino
- Bernardo
- Bérnia
- Bertila
- Bertilde
- Bertina
- Bertino
- Berto
- Bertolino
- Betânia
- Bétia
- Betina
- Betino
- Beto
- Betsabé
- Bia
- Biana
- Bianca
- Bianor
- Bibiana
- Bibili
- Bijal
- Bina
- Bitia
- Blandina
- Blásia
- Boanerges
- Boavida
- Bóris
- Branca
- Brandão
- Brás
- Brásia
- Bráulio
- Brázia
- Brena
- Brenda
- Breno
- Brian
- Briana
- Brícia
- Brígida
- Brígido
- Brigite
- Briolanjo
- Briosa
- Brites
- Brizida
- Bruce
- Bruna
- Bruno
- Brunilde
- Bryan
- Cássia
- Cael
- Caetana
- Caetano
- Caia
- Caíco
- Caio
- Caleb
- Calila
- Calisto
- Camélia
- Camila
- Candice
- Cândido
- Cânia
- Canto
- Capitolina
- Carela
- Cáren
- Cárin
- Carina
- Carisa
- Carísia
- Carissa
- Cárita
- Carla
- Carlinda

- Carlo
- Carlos
- Carlota
- Carmélia
- Carmelina
- Carmelinda
- Carmelita
- Cármen
- Carmério
- Carmezinda
- Carmim
- Carmina
- Carminda
- Carminho
- Carmo
- Carmorinda
- Carol
- Carole
- Carolina
- Carsta
- Cassandra
- Cássia
- Cassiano
- Cassilda
- Cássio
- Casta
- Castelina
- Castelino
- Castor
- Castorina
- Catalina
- Catarina
- Catarino
- Caterina
- Cátia
- Catila
- Catilina
- Cecília
- Cedrico
- Célia
- Celina
- Celinia
- Celino
- Célio
- Celísio
- Celsa
- Célsio
- Celso
- Celto
- Ceres
- Cesaltina
- Cesária
- Cesarina
- Cesário
- Césaro
- Chantal
- Charbel
- Cheila
- Chema
- Chloe
- Cibele
- Cícero
- Cid
- Cidália
- Cidalina
- Cidálio
- Cidalisa
- Cildo
- Cília
- Cílio
- Cinara
- Cínara
- Cinderela
- Cinira
- Cíntia
- Cipora
- Circe
- Círia
- Cirila
- Cirilo
- Ciro
- Cita
- Cizina
- Clara
- Clarina
- Clarinda
- Clarindo
- Clarinha
- Clarisse
- Claudemira
- Claudemiro
- Cláudia
- Claudiana
- Claudiano
- Cláudio
- Cleia
- Cleide
- Clélia
- Clélio
- Clemência
- Cleodice
- Cleonice
- Cleópatra
- Clésia
- Clésio
- Clícia
- Clício
- Clídio
- Clife
- Climénia
- Clívia
- Cloe
- Cloé
- Clorinda
- Clorindo
- Clóvis
- Colete
- Conceição
- Concha
- Consolação
- Constança
- Constância
- Constâncio
- Consulino
- Cora
- Corália
- Corálio
- Cordélia
- Corina
- Corino
- Córita
- Córito
- Corsino
- Cosete
- Cosme
- Cremilda
- Cremilde
- Crestila
- Crisália
- Crisálida
- Crisanta
- Crisante
- Crispim
- Cristela
- Cristele
- Cristene
- Cristiana
- Cristiano
- Cristofe
- Cristóforo
- Cristolinda
- Cristóvão
- Cursino
- Dácia
- Dácio
- Dafne
- Dagmar
- Dagoberto
- Daina
- Daisi
- Dália
- Daliana
- Dalida
- Dalila
- Dalinda
- Dalva
- Dámaris
- Damas
- Damião
- Damien
- Dana
- Dânia
- Daniana
- Dariana
- Daniel
- Daniela
- Danila
- Danilo
- Dante
- Dara
- Darcília
- Dárcio
- Dario
- Dário
- Darlene
- Darnela
- Darque
- Davi
- David
- Davide
- Davina
- Davínia
- Débora
- Décia
- Décimo
- Deise
- Deivid
- Dejalme
- Dejanira
- Délcio
- Dele
- Delfim
- Delfino
- Délia
- Deliana
- Délio
- Delisa
- Delmano
- Delmar
- Delmina
- Delmina
- Delminda
- Delmira
- Delmiro
- Demelza
- Deméter
- Demétria
- Demétrio
- Dener
- Denil
- Denis
- Denisa
- Denise
- Deodata
- Deodete
- Deolindo
- Deonilde
- Deotila
- Deótila
- Dércio
- Derocila
- Deusdedito
- Dhruva
- Dialina
- Diamantina
- Diamantino
- Diana
- Didaco
- Dídia
- Didiana
- Diego
- Dieter
- Digna
- Digno
- Dilan
- Dilermando
- Diliana
- Dilsa
- Dimas
- Dina
- Diná
- Dinamene
- Dinarda
- Dinarta
- Dinarte
- Dineia
- Dinis
- Dino
- Dinora
- Dioclécia
- Diocleciana
- Diocleciano
- Dioclécio
- Diogo
- Diomar
- Dione
- Dionilde
- Dionísia
- Dionísio
- Dioniso
- Dionisodoro
- Dirce
- Dircea
- Dircila
- Dírio
- Dirque
- Disa
- Ditza
- Diva
- Divo
- Diza
- Djalma
- Djalme
- Djalmo
- Djamila
- Dólique
- Dolores
- Domingas
- Domingos
- Domínico
- Domitila
- Domitília
- Domitilo
- Donaldo
- Donatila
- Donato
- Donzélia
- Donzília
- Donzílio
- Dora
- Dorabela
- Doralice
- Doriana
- Dóriclo
- Dorina
- Dorinda
- Dorindo
- Dorine
- Dorino
- Dóris
- Dorisa
- Dositeu
- Drusila
- Druso
- Duarte
- Duartina
- Duílio
- Dulce
- Dulcelina
- Dulcídia
- Dulcina
- Dulcineia
- Dulcínio
- Dúlia

- Dúnia
- Dúnio
- Durbalino
- Durval
- Durvalina
- Durvalino
- Eárine
- Eberardo
- Eda
- Eder
- Éder
- Edéria
- Edgar
- Édi
- Edina
- Edine
- Édipo
- Edir
- Edite
- Edith
- Edma
- Edmero
- Edmur
- Edna
- Edo
- Eduarda
- Eduardo
- Eduartino
- Eduina
- Eduíno
- Edvino
- Egídio
- Egil
- Eglantina
- Eládio
- Elana
- Elca
- Elda
- Eleazar
- Electra
- Eleia
- Eleine
- Elena
- Eleonor
- Eleonora
- Eleutério
- Elgar
- Eli
- Élia
- Eliab
- Eliana
- Eliane
- Eliano
- Elias
- Elícia
- Eliete
- Eliezer
- Élin
- Elina
- Eline
- Élio
- Elioenai
- Elisa
- Elisabeta
- Elisabete
- Elisabeth
- Elisama
- Eliseba
- Elisete
- Eliseu
- Elísia
- Elisiário
- Elma
- Elmano
- Elmar
- Elmer
- Elmira
- Eloá
- Elodia
- Elódia
- Elói
- Eloisa
- Elpídio
- Elsa
- Elsinda
- Élsio
- Élson
- Élton
- Eluína
- Elva
- Elvina
- Elvino
- Elza
- Elzeário
- Elzo
- Ema
- Emanuel
- Emanuela
- Emaús
- Emídia
- Emídio
- Emília
- Emiliana
- Emo
- Encarnação
- Eneias
- Enes
- Engelécia
- Engrácio
- Énia
- Enide
- Enilda
- Énio
- Enoque
- Enrique
- Enzo
- Éola
- Eponina
- Ercília
- Ercílio
- Eric
- Erica
- Érica
- Erico
- Érico
- Erik
- Erika
- Erique
- Éris
- Erméria
- Ermitério
- Ernâni
- Esaú
- Esmeralda
- Esmeraldo
- Esméria
- Especiosa
- Esperança
- Estanislau
- Estéfana
- Estefânia
- Estéfano
- Estela
- Estélio
- Ester
- Estêvão
- Estrela
- Etel
- Étel
- Etelca
- Etéria
- Eudora
- Eufémia
- Eularina
- Eulógio
- Eunice
- Eurica
- Eurico
- Eurídice
- Eustácio
- Eutália
- Eva
- Evaldo
- Evandra
- Evandro
- Evangelino
- Evangelista
- Evelácio
- Evelásio
- Evelina
- Eveline
- Evélio
- Evêncio
- Everaldo
- Everardo
- Évila
- Expedito
- Ezequiel
- Ezequiela
- Fábia
- Fabiana
- Fabiano
- Fabião
- Fábio
- Fabíola
- Fabrícia
- Fabrício
- Falco
- Fani
- Fânia
- Fantina
- Fara
- Farida
- Fátima
- Faustino
- Fausto
- Feba
- Febe
- Fédora
- Fedra
- Felícia
- Feliciana
- Felicidade
- Felicíssimo
- Felisbela
- Felisbina
- Felismina
- Félix
- Feliz
- Ferdinando
- Fernandina
- Fernandino
- Fernando
- Fernão
- Ferrer
- Fiama
- Fidélia
- Fidélio
- Filémon
- Filena
- Filino
- Filinto
- Filipa
- Filipe
- Filipo
- Filomena
- Filomeno
- Filoteu
- Fiona
- Firmino
- Firmo
- Flamínia
- Flávia
- Flávio
- Flor
- Flora
- Florbela
- Florença
- Florencia
- Florentino
- Flória
- Floriana
- Floripes
- Florisa
- Florisbela
- Florival
- Fradique
- Francília
- Francina
- Francisca
- Francisco
- Franclim
- Franco
- Franklim
- Franklin
- Franklino
- Fred
- Frede
- Frederica
- Frederico
- Fredo
- Fúlvio
- Gabi
- Gabínia
- Gabínio
- Gabino
- Gabriel
- Gabriela
- Gaela
- Gaele
- Gaia
- Gáil
- Gala
- Galiana
- Galiano
- Galileu
- Gamaliel
- Gaori
- Gaorii
- Garcia
- Gardela
- Garibaldo
- Gaspar
- Gastão
- Gávio
- Gedeão
- Geisa
- Genciana
- Genésia
- Genésio
- Gentil
- Georgeta
- Georgete
- Geórgia
- Georgina
- Georgino
- Geralda
- Geraldina
- Geraldino
- Geraldo
- Gerberta
- Gerberto
- Gerda
- Germana
- Germano
- Gersão
- Gerson
- Gerta
- Gertrudes
- Gervásia
- Gervásio
- Giana
- Giani
- Giulia
- Gil
- Gilberta
- Gilda
- Gildásio
- Gildo
- Gileade
- Gilma
- Gilmeno
- Gina
- Ginestal

- Gino
- Gioconda
- Giovana
- Giovani
- Giraldina
- Girel
- Gisela
- Giselda
- Gisete
- Gislena
- Gislene
- Gláucia
- Glenda
- Glicínia
- Gloriosa
- Goma
- Gomes
- Gonçala
- Gonçalo
- Gonzaga
- Goreti
- Graça
- Grácia
- Graciana
- Graciano
- Graciela
- Graciete
- Graciliana
- Graciliano
- Gracinda
- Grácio
- Graciosa
- Gravelina
- Gregória
- Gregório
- Greta
- Grimanesa
- Guadalupe
- Gualdim
- Gualter
- Gueir
- Guendolina
- Gui
- Guida
- Guido
- Guilherme
- Guimar
- Guislena
- Guislene
- Gumersinda
- Gumersindo
- Gumesindo
- Gusmão
- Gustavo
- Guterre
- Habacuc
- Habacuque
- Hadassa
- Haidé
- Hália
- Hamilton
- Haraldo
- Harolda
- Haroldo
- Hazael
- Hebe
- Héber
- Heda
- Hédila
- Hedviges
- Heitor
- Hélada
- Hélade
- Heládia
- Heládio
- Helda
- Heldemaro
- Hélder
- Heldo
- Helena
- Helénico
- Heleno
- Helga
- Heli
- Hélia
- Heliana
- Helier
- Hélio
- Heliodora
- Heliodoro
- Hélmut
- Heloísa
- Helvécia
- Helvécio
- Hélvia
- Hélvio
- Hemexi
- Hemetéria
- Hemetério
- Hemitéria
- Hemitério
- Henoch
- Henrique
- Henriqueta
- Heralda
- Heraldo
- Herberta
- Herberto
- Herculana
- Herculano
- Herédio
- Herénia
- Herénio
- Heriberta
- Heriberto
- Herlander
- Hérman
- Hermana
- Hermânia
- Hermano
- Hermenegilda
- Hermenegildo
- Hermenerica
- Hermenerico
- Hermes
- Hermínia
- Hermínio
- Hermitério
- Hernâni
- Hersília
- Hersílio
- Hervê
- Higina
- Higino
- Hilária
- Hilário
- Hildeberta
- Hildeberto
- Hildebrando
- Hildegarda
- Hildegardo
- Hilma
- Hipólita
- Hipólito
- Hirondino
- Hólger
- Homero
- Honorata
- Honorato
- Honorina
- Honorino
- Horácia
- Horácio
- Hortense
- Hortênsia
- Hortênsio
- Hugo
- Huguete
- Hulda
- Iag
- Iago
- Ian
- Iana
- Ianis
- Iara
- Iasmin
- Iasmina
- Ibérico
- Iberina
- Ícaro
- Ida
- iddy
- Idália
- Idalina
- Idálio
- Idário
- Idavide
- Idélia
- Idélso
- Idília
- Idrisse
- Igelcemina
- Ignez
- Igor
- Ilca
- Ilda
- Ildo
- Ilídia
- Ilídio
- Ilsa
- Ilse
- Ilundi
- Ima
- Indalécio
- Indaleta
- India
- Indira
- Indro
- Inês
- Infante
- Inga
- Ingeburga
- Ingo
- Ingrid
- Ingride
- Ingue
- Inocência
- Inocêncio
- Inoi
- Io
- Iolanda
- Ionara
- Ione
- Ioque
- Iracema
- Iráis
- Ireneia
- Iria
- Iriana
- Irina
- Irineu
- Íris
- Irisalva
- Irma
- Irmino
- Isa
- Isaac
- Isabel
- Isabela
- Isabelina
- Isac
- Isadora
- Isael
- Isaí
- Isalda
- Isália
- Isalina
- Isandro
- Isaque
- Isaura
- Isaurinda
- Isauro
- Isidoro
- Isidro
- Isilda
- Isildo
- Isis
- Ismael
- Ismália
- Isolda
- Isolete
- Isolina
- Isolino
- Israel
- Italo
- Iúri
- Iva
- Ivan
- Ivana
- Ivânia
- Ivanoel
- Ivanoela
- Iven
- Ivete
- Ivo
- Ivone
- Izalino
- Jabes
- Jabim
- Jacira
- Jacó
- Jacob
- Jacobina
- Jácome
- Jacqueline
- Jader
- Jadir
- Jael
- Jaime
- Jair
- Jairo
- Jalmira
- James
- Jamila
- Jamília
- Jamim
- Janai
- Janaína
- Janardo
- Jandira
- Janete
- Jani
- Jânia
- Janice
- Janina
- Janine
- Janique
- Jansénio
- Januário
- Jaque
- Jaquelina
- Jaqueline
- Jaques
- Jarbas
- Jardel
- Jasão
- Jasmim
- Jasmina
- Jeanete
- Jéni
- Jénifer
- Jerónimo
- Jerusa
- Jessé
- Jéssica
- Jesualdo
- Jesus
- Jetro
- Jezabel
- Jil
- Jitendra
- Jó
- Joab
- Joabe

- Joana
- Joaninha
- João
- Joaquim
- Joás
- Job
- Jocelina
- Jocelino
- Jociano
- Joel
- Joela
- Joele
- Joelma
- Jofre
- Joice
- Jonas
- Jonatã
- Jónatas
- Jóni
- Joraci
- Jordana
- Jordano
- Jordão
- Jorge
- Jorgina
- Jório
- Jorja
- Josabete
- Josafat
- Josana
- Joscelina
- Joscelino
- José
- Josefa
- Joséfa
- Josefina
- Josefo
- Joselene
- Josélia
- Joselina
- Joselindo
- Joselino
- Josete
- Josiana
- Josiane
- Josias
- Josina
- Josselina
- Josselino
- Josuana
- Josué
- Jovelina
- Jovelino
- Jovito
- Judá
- Judas
- Juliana
- Juliano
- Julião
- Julinda
- Júlio
- Julita
- Juna
- Júnia
- Júnio
- Juno
- Juraci
- Jussara
- Juvenal
- Juventino
- Karen
- Karina
- Katarina
- Katia
- Katie
- Kelly
- Kevin
- Kyara
- Laércio
- Laertes
- Laila
- Laira
- Lais
- Lana
- Lara
- Larissa
- Laura
- Laureana
- Laureano
- Laurénio
- Laurentino
- Lauriano
- Laurina
- Laurinda
- Laurine
- Lauro
- Lavínia
- Lázaro
- Lea
- Leão
- Leal
- Leandra
- Leandro
- Leanor
- Léccio
- Lécio
- Leena
- Leila
- Lélia
- Lemuel
- Lénia
- Lénio
- Lenira
- Leo
- Leoberto
- Leocádia
- Leolina
- Leoménia
- Leonardina
- Leonardo
- Leôncio
- Leone
- Leonel
- Leonete
- Leónia
- Leonício
- Leonida
- Leonídia
- Leonídio
- Leonila
- Leonilda
- Leonilde
- Leonília
- Leonisa
- Leonor
- Leonora
- Leontina
- Leta
- Letícia
- Letízia
- Levi
- Levina
- Lhuzie
- Lia
- Liana
- Liane
- Lianor
- Liara
- Liberal
- Liberalina
- Liberdade
- Libéria
- Libertária
- Libertário
- Liberto
- Líbia
- Lici
- Lícia
- Lícidas
- Licínia
- Liciniano
- Licínio
- Lício
- Lídia
- Lidiana
- Lídio
- Lidório
- Liduína
- Liete
- Lígia
- Lígio
- Lilá
- Lila
- Lília
- Lilian
- Liliana
- Liliane
- Liliano
- Liliete
- Lilite
- Lina
- Linda
- Lindorfo
- Lindoro
- Lineia
- Linete
- Lineu
- Lino
- Línton
- Lira
- Lis
- Lisa
- Lisana
- Lisandra
- Lisandro
- Lisdália
- Liseta
- Lisete
- Lisuarte
- Lito
- Lívia
- Liz
- Lizélia
- Lízi
- Lízie
- Loela
- Loide
- Lólia
- Lopo
- Loredana
- Lorena
- Lorenzo
- Loreta
- Lorina
- Lorine
- Lorival
- Lourença
- Lourenço
- Lourival
- Lua
- Luamar
- Luana
- Lubélia
- Luca
- Lucas
- Lucélia
- Lucelinda
- Lucena
- Lucete
- Lúcia
- Lucialina
- Luciana
- Lucileine
- Lucília
- Lucilina
- Lucílio
- Lucina
- Lucínio
- Lúcio
- Lucíola
- Ludgero
- Ludmila
- Ludovico
- Ludovino
- Luela
- Luena
- Luís
- Luísa
- Luisete
- Luizete
- Lumena
- Luna
- Lurdes
- Lurdite
- Lusa
- Lussinga
- Lutero
- Lutgarda
- Luz
- Luzia
- Luzinira
- Luzio
- Madalena
- Mafalda
- Magali
- Magda
- Mamede
- Manel
- Manuel
- Manuela
- Mara
- Márcia
- Marcilene
- Márcio
- Marco
- Marcos
- Marcela
- Marcelo
- Marcolina
- Margarida
- Maria
- Mariana
- Mariano
- Marilda
- Marília
- Marina
- Mário
- Marisa
- Marlene
- Marli
- Marta
- Martim
- Martinho
- Mateus
- Matias
- Matilde
- Maurício
- Maura
- Mauro
- Máxima
- Máximo
- Maximiliano
- Maximino
- Mécia
- Melânia
- Melinda
- Melissa
- Melquisedeque
- Mem
- Mercedes
- Merrelho
- Miguel
- Miguelina
- Milena
- Mileide
- Milu
- Micael
- Micaela
- Michele
- Minervina
- Miriam
- Moacir
- Moisés
- Mónica

- Morgana
- Murilo
- Miru
- Nádia
- Nadine
- Nair
- Napoleão
- Natacha
- Natália
- Natalina
- Natércia
- Natividade
- Nazaré
- Nelson
- Nestor
- Neusa
- Neuza
- Nélia
- Nicanor
- Nicolas
- Nicolau
- Nídia
- Nilza
- Nivaldo
- Noa
- Noah
- Noé
- Noel
- Noémia
- Norberto
- Normando
- Nuno
- Octávio
- Octávia
- Odete
- Odilia
- Ofélia
- Olavo
- Olívia
- Olívio
- Oliveira
- Olga
- Omar
- Ondina
- Ordonho
- Orestes
- Oriana
- Otília
- Óscar
- Orlando
- Osóri
- Osvaldo
- Ovídio
- Paloma
- Palmira
- Palmiro
- Pandora
- Parcidio
- Párias
- Pascoal
- Poliana
- Patrícia
- Patrício
- Paulina
- Paulino
- Paula
- Paulo
- Paulino
- Pedro
- Petra
- Penélope
- Pépio
- Piedade
- Plácido
- Plínio
- Políbio
- Polibe
- Porfírio
- Prião
- Priscila
- Quaiela
- Quar
- Quéli
- Quélia
- Querubim
- Quezia
- Quévin
- Quiliano
- Quim
- Quintino
- Quirilo
- Quirina
- Quirino
- Quírio
- Quitéria
- Quitério
- Rafael
- Rafaelo
- Rafaela
- Ramão
- Ramiro
- Raimundo
- Raquel
- Raul
- Rebeca
- Regina
- Reginaldo
- Reinaldo
- Reinamor
- Remo
- Renan
- Renata
- Renato
- Ricardina
- Ricardo
- Rita
- Roberta
- Roberto
- Rodolfo
- Rodrigo
- Rogério
- Romão
- Romano
- Rómulo
- Ronaldo
- Roque
- Roquita
- Rosa
- Rosália
- Rosalina
- Rosalinda
- Rosana
- Rossana
- Rosário
- Rosaura
- Roseli
- Rúben
- Rubim
- Rudi
- Rufus
- Rui
- Russel
- Rute
- Ruca
- Sabina
- Sabino
- Sabrina
- Sacramento
- Sadi
- Sadraque
- Sadrudine
- Safia
- Safira
- Salazar
- Salemo
- Sales
- Salete
- Sáli
- Salima
- Salma
- Salomão
- Salomé
- Salomite
- Salúquia
- Salustiano
- Salustiniano
- Salvação
- Salvador
- Salvador
- Salviano
- Salvina
- Samanta
- Samara
- Samaritana
- Samaritano
- Samir
- Samira
- Samuel
- Sancha
- Sancho
- Sância
- Sancler
- Sandra
- Sandrina
- Sandrino
- Sandro
- Sansão
- Santana
- Santelmo
- Santiago
- Santos
- Sara
- Sarah
- Sarai
- Sarina
- Sário
- Sásquia
- Sássia
- Sátia
- Sátira
- Sátiro
- Saúl
- Saula
- Saulina
- Saulo
- Sauro
- Sávio
- Sebastiana
- Sebastião
- Secundino
- Séfora
- Segismundo
- Selena
- Selene
- Selénia
- Selesa
- Selésia
- Selésio
- Seleso
- Selma
- Selmo
- Semíramis
- Sena
- Sénia
- Sénio
- Seomara
- Serafim
- Serafina
- Serena
- Serenela
- Sérgio
- Sesinando
- Sesira
- Severiano
- Severino
- Sextina
- Sheila
- Sibila
- Siddártha
- Sidnei
- Sidónio
- Sidraque
- Siena
- Sifredo
- Silas
- Silvana
- Silvandira
- Silvano
- Silvério
- Silvestre
- Sílvia
- Sílvia
- Silviana
- Silviano
- Sílvio
- Simão
- Simara
- Simaura
- Simauro
- Simeão
- Simone
- Simoneta
- Simplício
- Sindulfo
- Sinésio
- Sira
- Síria
- Sirla
- Sisenando
- Sisínio
- Sisnando
- Sívio
- Socorro
- Sócrates
- Soeiro
- Sofia
- Sol
- Solana
- Solange
- Solano
- Soledade
- Solene
- Solôngia
- Sónia
- Soraia
- Sotero
- Stela
- Stelina
- Suati
- Suéli
- Sulamita
- Sunamita
- Suraje
- Surendralal
- Suri
- Suria
- Susana
- Susana
- Susana
- Susano
- Suse
- Susete
- Susi
- Tauane
- Tabita
- Taciana
- Taciano
- Tácio
- Tadeu
- Taís
- Taísa
- Taíssa
- Talia
- Tálio
- Talita
- Tamar
- Tamara
- Tamár
- Tâmiris
- Tanagra
- Tânia
- Tarcísio
- Tarina
- Tarsício

- Tásia
- Tasso
- Tatiana
- Tatiano
- Tejala
- Teliano
- Telma
- Telmo
- Telo
- Teodemiro
- Teodomiro
- Teodora
- Teodoro
- Teófilo
- Tércio
- Teresa
- Teresca
- Teresina
- Teresinha
- Terezinha
- Teseu
- Tiago
- Tiana
- Tiara
- Tibério
- Tibúrcio
- Ticiana
- Ticiano
- Tierri
- Timóteo
- Tirsa
- Tirso
- Tirza
- Tita
- Titânia
- Tito
- Tobias
- Toledo
- Tomás
- Tomé
- Toni
- Torcato
- Torpécia
- Torquato
- Traciana
- Trajano
- Trasila
- Trindade
- Tristão
- Tude
- Túlio
- Túlipa
- Turgo
- Takeo
- Valdemar
- Valentim
- Valentina
- Valéria
- Valério
- Valmor
- Valter
- Vanda
- Vanessa
- Vânia
- Vasco
- Vera
- Veríssimo
- Verónica
- Vérter
- Vestina
- Vianei
- Vicência
- Vicenta
- Vicente
- Victor
- Victória
- Vida
- Vidal
- Vidálio
- Vidaúl
- Vila
- Vilator
- Vili
- Vilma
- Vílmar
- Vílson
- Vinícia
- Vinício
- Violante
- Violeta
- Violinda
- Virgílio
- Virgínio
- Virgulino
- Viriato
- Vital
- Vitália
- Vitaliano
- Vitálio
- Vitiza
- Vito
- Vítor
- Vitória
- Vitório
- Vivalda
- Vivaldo
- Vivelinda
- Viveque
- Viviana
- Viviane
- Vivilde
- Vivina
- Vladimiro
- Xavier
- Zacarias
- Zahra
- Zaido
- Zaida
- Zaira
- Zaíro
- Zamy
- Zaqueu
- Zara
- Zará
- Zarco
- Zardilaque
- Zarina
- Zé
- Zeferina
- Zélia
- Zelinda
- Zélio
- Zena
- Zenaida
- Zenaide
- Zénia
- Zerá
- Zila
- Zilda
- Zília
- Zilma
- Zita
- Ziza
- Zoa
- Zobaida
- Zoé
- Zola
- Zora
- Zoraida
- Zubaida
- Zubeida
- Zulaia
- Zuleica
- Zulmira

A.3 Apelidos

- Silva
- Santos
- Ferreira
- Pereira
- Oliveira
- Costa
- Rodrigues
- Martins
- Jesus
- Sousa
- Fernandes
- Gonçalves
- Gomes
- Lopes
- Marques
- Alves
- Almeida
- Ribeiro
- Pinto
- Carvalho
- Teixeira
- Moreira
- Correia
- Mendes
- Nunes
- Soares
- Vieira
- Monteiro
- Cardoso
- Rocha
- Neves
- Coelho
- Cruz
- Cunha
- Pires
- Ramos
- Reis
- Simões
- Antunes
- Matos
- Fonseca
- Machado
- Araújo
- Barbosa
- Tavares
- Lourenço
- Castro
- Figueiredo
- Azevedo
- Freitas
- Magalhães
- Henriques
- Lima
- Guerreiro
- Batista
- Pinheiro
- Faria
- Miranda
- Barros
- Morais
- Nogueira
- Esteves
- Anjos
- Baptista
- Campos
- Mota
- Andrade
- Brito
- Sá
- Nascimento
- Leite
- Abreu
- Borges
- Melo
- Vaz
- Pinho
- Vicente
- Gaspar
- Assunção
- Maia
- Moura
- Valente
- Domingues
- Garcia
- Carneiro
- Loureiro
- Neto
- Amaral
- Branco
- Leal
- Pacheco
- Macedo
- Paiva
- Matias
- Amorim
- Torres

Apêndice B

README

DB Anonymization

This is an automatic framework that detects sensitive information in a database and creates an anonymized copy of it.

What You Need

Java

You need to have two versions of Java installed on your machine:

- jdk-17 or latest
  You must put the jdk-17 folder path in your computer's environment variables:

  > In the System variables, find or create the variable named JAVA_HOME and put, in its value, the path (eg: C:\Program Files\Java\jdk-17).
  > Then add the following line to the Path variable:
  > %JAVA_HOME%\bin

- jdk1.8
  There is a folder called **scriptsJava** in the scripts folder.
  Inside this folder there is a file named **java8.bat**. You should right click and choose the edit option. The second line should be something like this:

```
>set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_321
```

Check that this path is the correct path for your installation of jdk1.8 version.
If you have a different java 1.8 version than the **jdk1.8.0_321**, such as **jdk1.8.0_331**, replace the folder name in this line:

```
>set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_331
```

You must put the **scriptsJava** folder path in your computer's environment variables:
In the System variables, in the Path variable put the path *C:\(...)\scripts\scriptsJava* .

Postgresql

You need PostgreSQL installed on your machine and to put in the Path environment variable the path to the bin folder of your PostgreSQL installation.

Docker

You need Docker installed.
After the installation, in the hidden icon that appears in the taskbar, check that when you click with the right mouse button you see the option "Switch to Windows containers...". If "Linux" appears instead of "Windows", left click on it so that it changes.

How to Run

- First, you **need to have a .properties file** in resources folder named *config.properties* (resources/config.properties). This file must have the necessary data for the connection to your DB.

  > The file must have the following format:
  > db.url=jdbc:postgresql://localhost:14020/postgres
  > db.user=bob
  > db.password=bobpassword

- Second, **for detection** run the file scripts/runDetection.bat.
  This will create a *DetectionToUse.xml* file in
  the resources folder with metadata about the database fields that were
  detected

- Third, check the *resources/DetectionToUse.xml* file and delete the lines with the columns **you don't want to anonymize**.
  If an XML element referring to **a table doesn't have any column elements**, then this table element must be deleted!

  > You can also change the type in each column to use another type of anonymization.
  > A list of existing anonymization types can be found below in the section Anonymization Types.

Example of a *DetectionToUse.xml* file with two element tables and how to use:

```
1 <table name="company">
2   <column name="tax_number" type="FISCAL_DOC_NUMBER_PT"/>
3 </table>
4 <table name="collaborator">
5   <column name="identity_doc_num" type="ID_CC_DOC_NUMBER_PT"/>
6   <column name="identity_doc_val" type="DATE"/>
7   <column name="passport_val" type="DATE"/>
8   <column name="tax_doc_num" type="FISCAL_DOC_NUMBER_PT"/>
9 </table>
```

- Now, if we don't want to anonymise the *tax_doc_num* column in the *collaborator* table, then we can delete the line corresponding to the column (line 8):

```
1 <table name="company">
2   <column name="tax_number" type="FISCAL_DOC_NUMBER_PT"/>
3 </table>
4 <table name="collaborator">
5   <column name="identity_doc_num" type="ID_CC_DOC_NUMBER_PT"/>
6   <column name="identity_doc_val" type="DATE"/>
7   <column name="passport_val" type="DATE"/>
8 </table>
```

- If you don't want to anonymise the *tax_number* column, we will delete that line. As it was the only column within the XML table element suggested for anonymisation, then we need to delete the *company* table element.

```
1 <table name="collaborator">
2   <column name="identity_doc_num" type="ID_CC_DOC_NUMBER_PT"/>
3   <column name="identity_doc_val" type="DATE"/>
4   <column name="passport_val" type="DATE"/>
5 </table>
```

- Let's now change the anonymization type of the *identity_doc_num* column from ID_CC_DOC_NUMBER_PT to RANDOMCHARACTERS.

```
1 <table name="collaborator">
2   <column name="identity_doc_num" type="RANDOMCHARACTERS"/>
3   <column name="identity_doc_val" type="DATE"/>
4   <column name="passport_val" type="DATE"/>
5 </table>
```

- Finally, to **for anonymization** run the file scripts/runCopyAnonimatron.bat.
  You will need to enter your database password, which can be found in the properties file, and press Enter. There will be no output and you should immediately enter the password of the duplicate database, which is *my-secret-pw*.
  This will create an anonymised copy of the database in a docker container.

> jdbcurl="jdbc:postgresql://localhost:13000/postgres" password="secret_password" userid="postgres"

Anonymization Types

Name	Type	Input
MobilePhoneNumberPTAnonymizer	MOBILE_PHONE_NUMBER_PT	Any string
FiscalDocumentNumPTAnonymizer	FISCAL_DOC_NUMBER_PT	Any string
IdentityDocumentNumPTAnonymizer	ID_CC_DOC_NUMBER_PT	Any string
SocialDocumentNumPTAnonymizer	SOCIAL_DOC_NUMBER_PT	Any string
FirstNamePTAnonymizer	FIRST_NAME_PT	Any string
LastNamePTAnonymizer	LAST_NAME_PT	Any string
FullNamePTAnonymizer	FULL_NAME_PT	Any string
AddressPTAnonymizer	ADDRESS_PT	Any string
EmailAddressPTAnonymizer	EMAIL_ADDRESS_PT	Any string
ZipCodePTAnonymizer	ZIP_CODE_PT	Any string
IbanPTAnonymizer	IBAN_PT	Any string
CharacterStringAnonymizer	RANDOMCHARACTERS	Any string
CharacterStringPrefetchAnonymizer	PREFETCHCHARACTERS	Any string
CountryCodeAnonymizer	COUNTRY_CODE	Any string
DateAnonymizer	DATE	Valid date
DigitStringAnonymizer	RANDOMDIGITS	Any string
DutchBankAccountAnononymizer	DUTCHBANKACCOUNT	Any string
DutchBSNAnononymizer	BURGERSERVICENUMMER	Number or string
DutchZipCodeAnonymizer	DUTCH_ZIP_CODE	Any string
ElvenNameGenerator	ELVEN_NAME	Any string
EmailAddressAnonymizer	EMAIL_ADDRESS	Any string
IbanAnonymizer	IBAN	Any string
RomanNameGenerator	ROMAN_NAME	Any string
StringAnonymizer	STRING	Any string
UkPostCodeAnonymizer	UK_POST_CODE	Any string
UUIDAnonymizer	UUID	Any string

Anonymization

The open source tool Anonimatron is used for anonymization, which has been extended to support new algorithms.

Bibliografia

[1] Validação de número de documento do cartão de cidadão, 2009. https://www.autenticacao.gov.pt/documents/20126/115760/Validacao+de+Numero+de+Documento+do+Cartao+de+Cidadao.pdf.

[2] Amnesia. Amnesia anonymization tool, 2021. https://amnesia.openaire.eu/.

[3] Anonimatron. Anonimatron, 2021. https://realrolfje.github.io/anonimatron/.

[4] R.J. Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *21st International Conference on Data Engineering (ICDE'05)*, pages 217–228, 2005.

[5] Raphaël Chevrier, Vasiliki Foufi, Christophe Gaudet-Blavignac, Arnaud Robert, Christian Lovis, et al. Use and understanding of anonymization and de-identification in the biomedical literature: scoping review. *Journal of medical Internet research*, 21(5):e13484, 2019.

[6] Graham Cormode and Divesh Srivastava. Anonymized data: Generation, models, usage. SIGMOD '09, page 1015–1018, New York, NY, USA, 2009. Association for Computing Machinery.

[7] d0nut. Attacks on applications of k-anonymity — for the rest of us, 2019. https://d0nut.medium.com/attacks-on-applications-of-k-anonymity-for-the-rest-of-us-426d3b751

[8] Sabrina De Capitani di Vimercati, Dario Facchinetti, Sara Foresti, Gianluca Oldani, Stefano Paraboschi, Matthew Rossi, and Pierangela Samarati. Scalable distributed data anonymization. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 401–403. IEEE, 2021.

[9] Marco Gaboardi, James Honaker, Gary King, Kobbi Nissim, Jonathan Ullman, Salil Vadhan, and Jack Murtagh. Psi: a private data sharing interface. In *Theory and Practice of Differential Privacy*, New York, NY, 2016 2016.

[10] GDPR. Regulation (EU) 2016/679 of the European Parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016.

[11] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, pages 77–80, 2006.

[12] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 25–25, 2006.

[13] Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4188–4203, Online, August 2021. Association for Computational Linguistics.

[14] Frank D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, page 19–30, New York, NY, USA, 2009. Association for Computing Machinery.

[15] Fabian Prasser, Johanna Eicher, Helmut Spengler, Raffael Bild, and Klaus A. Kuhn. Flexible data anonymization using arx—current status and challenges ahead. *Software: Practice and Experience*, 50(7):1277–1304, 2020.

[16] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[17] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[18] Trust Systems. Trust systems company web site, 2022. `https://www.trustsystems.pt/`.

[19] United Nations. *Universal Declaration of Human Rights*. dec 1948.

[20] Elena Volodina, Yousuf Ali Mohammed, Sandra Derbring, Arild Matsson, and Beáta Megyesi. Towards privacy by design in learner corpora research: A case of on-the-fly pseudonymization of Swedish learner essays. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 357–369. ACL, 2020.

[21] Jinsung Yoon, Lydia N. Drumright, and Mihaela van der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE Journal of Biomedical and Health Informatics*, 24(8):2378–2388, 2020.