



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO

Jairo Nascimento de Sousa Filho

**Geração de dados sintéticos utilizando a
aplicação Blocks: simulando dados discrepantes
e faltantes.**

Belém

2019

Jairo Nascimento de Sousa Filho

**Geração de dados sintéticos utilizando a aplicação
Blocks: simulando dados discrepantes e faltantes.**

Monografia apresentada na Faculdade de
Computação do Instituto de Ciências Exa-
tas e Naturais como requisito parcial para
obtenção do grau de Bacharel.

Universidade Federal do Pará

Orientador: Prof. Dr. Carlos Gustavo Resque dos Santos

Belém

2019

Solicite sua ficha catalográfica em: <<http://bcficat.ufpa.br/>>

Jairo Nascimento de Sousa Filho

**Geração de dados sintéticos utilizando a aplicação
Blocks: simulando dados discrepantes e faltantes.**

Monografia apresentada na Faculdade de
Computação do Instituto de Ciências Exa-
tas e Naturais como requisito parcial para
obtenção do grau de Bacharel.

Conceito: _____

Belém, 1 de janeiro de 2019.

BANCA EXAMINADORA

Prof. Dr. Carlos Gustavo Resque dos Santos - Orientador
UFPA

Nome Convidado 1
SIGLA INSTITUIÇÃO

Nome Convidado 2
SIGLA INSTITUIÇÃO

Escreva sua dedicatória aqui.

Agradecimentos

Escrever Agradecimentos aqui.

*“Escreva sua epígrafe aqui”
(Fulano de Tal, 19XX)*

Resumo

Segundo a (ABNT, 2003), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Keywords: latex. abntex. text editoration.

Lista de ilustrações

Figura 1.	Exemplo de escala de discrepância (Adaptado de Aggarwal)	33
Figura 2.	Exemplo de ruído e de anomalia.	34
Figura 3.	Exemplo das classificações de ruído (adaptado de Rathi)	34
Figura 4.	Visão Geral de geração de dados.	35
Figura 5.	Visão Geral de geração de dados do Sketchpad.	36
Figura 6.	Exemplo de árvore de decisão para jogar tennis criado a partir de regras encontradas em um conjunto de dados.	37
Figura 7.	Exemplo da interface do usuário para configuração do gerador de dados.	37
Figura 8.	Fluxo de passos para geração dos dados sintéticos.	38
Figura 9.	Comparação da média dos dados reais e sintéticos na simulação paramétrica e não paramétrica.	39
Figura 10.	Fluxo de passos para geração dos dados sintéticos.	40
Figura 11.	Fluxo de passos para geração dos dados sintéticos.	40
Figura 12.	Usando o DTM Data Generator. Fonte: DTM Data Generator	42
Figura 13.	Usando o Redgate SQL Data Generator. Fonte: Red Gate SQL Data Generator.	43
Figura 14.	Usando o Microsoft Visual Studio. Fonte: anranik.	44
Figura 15.	Usando o dbForge Test Data Generator. Fonte: anranik.	45
Figura 16.	Usando o Mockaroo. Fonte: anranik.	46
Figura 17.	Diagrama de Caso de uso do Blocks Data Generator. Fonte: o autor.	47
Figura 18.	Fluxograma de utilização do Blocks Data Generator. Fonte: Yvan Brito, 2019.	50
Figura 19.	Ilustrando a leitura dos marcos dos quartis. O tamanho do espaço entre os quartis ou entre 0 ou o 100 é o valor da probabilidade de um número ser desse espaço. Fonte: O Autor.	55
Figura 20.	Conhecendo os elementos da tela principal do Blocks Data Generator, na sua versão para Windows. Fonte: O Autor.	57
Figura 21.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	58
Figura 22.	Conhecendo os elementos da <i>context menu</i> na aba do modelo. Fonte: O Autor.	59
Figura 23.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	60
Figura 24.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	60

Figura 25. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor. 61

Lista de quadros

Lista de tabelas

Tabela 1.	Exemplo de dados ausentes MCAR	32
Tabela 2.	Exemplo de dados ausentes MAR	32
Tabela 3.	Propriedade dos geradores do Blocks Data Generator	51
Tabela 4.	Resumo dos geradores do Blocks Data Generator	62

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
abnTeX	ABsurdas Normas para TeX

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

Sumário

1	INTRODUÇÃO	27
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	Dados Sintéticos	29
2.2	Proveção dos dados	29
2.2.1	Arquivo	29
2.2.2	Web Service	30
2.3	Dados Ausentes	31
2.3.1	MCAR	31
2.3.2	MAR	31
2.3.3	MNAR	32
2.4	Dados Discrepantes	32
2.4.1	Dados Ruidosos e Anômalos	33
3	TRABALHOS RELACIONADOS	35
3.1	Trabalhos acadêmicos relacionados	35
3.2	Aplicações relacionadas	41
4	ARQUITETURA DO PROJETO	47
4.1	Casos de uso do sistema	47
5	PROTÓTIPO	49
5.1	Tipos de Geradores de Dados	50
5.1.1	Sequencial	52
5.1.2	Aleatório	52
5.1.3	Funcional	52
5.1.4	Acessórios	53
5.1.5	Geométrico	53
5.1.6	Baseado em dados reais	54
5.2	Modos de Geração de Dados	54
5.2.1	<i>Streaming Data</i>	54
5.2.2	Web Service	55
5.3	Modos para Visualização de Dados	56
5.3.1	Preview	56
5.3.2	Módulo de Visualização Externo e Integralizado	56
5.4	Estrutura da Interface do Blocks	56

5.4.1	Mensagens para o usuário	59
5.4.2	Atalhos do Teclado	59
6	RESULTADOS	63
7	CONCLUSÃO	65
	REFERÊNCIAS	67

1 Introdução

Este documento e seu código-fonte são exemplos de referência de uso da classe `abntex2` e do pacote `abntex2cite`. O documento exemplifica a elaboração de trabalho acadêmico (tese, dissertação e outros do gênero) produzido conforme a ABNT NBR 14724:2011 *Informação e documentação - Trabalhos acadêmicos - Apresentação*.

A expressão “Modelo Canônico” é utilizada para indicar que `abnTEX2` não é modelo específico de nenhuma universidade ou instituição, mas que implementa tão somente os requisitos das normas da ABNT. Uma lista completa das normas observadas pelo `abnTEX2` é apresentada em (ARAÚJO, 2015a).

Sinta-se convidado a participar do projeto `abnTEX2`! Acesse o site do projeto em [<http://www.abntex.net.br/>](http://www.abntex.net.br/). Também fique livre para conhecer, estudar, alterar e redistribuir o trabalho do `abnTEX2`, desde que os arquivos modificados tenham seus nomes alterados e que os créditos sejam dados aos autores originais, nos termos da “The L^AT_EX Project Public License”¹.

Encorajamos que sejam realizadas customizações específicas deste exemplo para universidades e outras instituições — como capas, folha de aprovação, etc. Porém, recomendamos que ao invés de se alterar diretamente os arquivos do `abnTEX2`, distribua-se arquivos com as respectivas customizações. Isso permite que futuras versões do `abnTEX2` não se tornem automaticamente incompatíveis com as customizações promovidas. Consulte (ARAÚJO, 2015b) para mais informações.

Este documento deve ser utilizado como complemento dos manuais do `abnTEX2` (ARAÚJO, 2015a; ARAÚJO, 2015c; ARAÚJO, 2015d) e da classe `memoir` (WILSON; MADSEN, 2010).

Esperamos, sinceramente, que o `abnTEX2` aprimore a qualidade do trabalho que você produzirá, de modo que o principal esforço seja concentrado no principal: na contribuição científica.

Equipe `abnTEX2`

Lauro César Araújo

¹ [<http://www.latex-project.org/lppl.txt>](http://www.latex-project.org/lppl.txt)

2 Fundamentação Teórica

Neste capítulo é abordado em mais detalhes sobre a literatura dos dados sintéticos, discrepâtes, faltantes, bem como de arquivos, e serviços como *Web Service* e base de dados.

2.1 Dados Sintéticos

Dados sintéticos foi definido como "qualquer dado produzido o qual possa ser aplicado a uma dada situação que não foi obtido por mensuração direta.". (EDUCATION, 2016) Em seu trabalho, Rubin (RUBIN, 1993) a introduziu um conjunto de dados completamente sintético. Em suma, seu objetivo era tornar anônimo os domicílios que participaram do censo daquela época. A questão da confidencialidade sempre foi uma característica necessária para dados divulgados, principalmente para dados sensíveis. Os dados sintéticos possuem a possibilidade de serem alterados mantendo a mesma ideia, logo, representa os dados reais originais. Essa característica que ajudou na popularização dos dados sintéticos.

A necessidade de dados sintéticos podem ser de várias formas, desde a escassez de dados reais ou indisponibilidade; para teste de dados não usuais; para evitar lidar com questões de privacidade dos dados; teste de aplicação sem precisar modificar dados da aplicação de produção; criar teste de estresse da aplicação com *Big Data* antes de criar versão para produção; bem como não precisar adicionar os dados de teste manualmente. (KUMAR, 2019)

A aplicabilidade dos dados sintéticos é ilimitada e é bastante explorada por setores cujos dados são sensíveis como a financeiro (LOPEZ-ROJAS; AXELSSON, 2012) e de saúde. (BERGEAT et al., 2014) Também são muito bem aplicáveis para exaustivos testes de segurança, os quais são necessários vários casos de teste pesquisador/analista de teste tem controle suficiente das características (fórmulas matemáticas ou regras de geração) e pode usar em um sistema de detecção de fraudes, por exemplo. (BARSE; KVARNSTROM; JONSSON, 2003)

2.2 Provedor dos dados

2.2.1 Arquivo

Gerar os dados não é o suficiente, para isso, é necessário oferecer uma forma pronta de uso para o usuário. Para isso, pode ser utilizado os arquivos. Segundo Tanenbaum

(TANENBAUM; FILHO, 1995) arquivos são unidades lógicas de informação criadas por processos e gerenciados por sistemas operacionais. Também é um mecanismo de abstração ao usuário para leitura e escrita em disco. Para que isso funcione, são adotados algumas convenções.

A primeira são os sistemas de arquivos. Basicamente, um sistema operacional adota um sistema de arquivos para personalizar a questão da leitura e escrita. (TANENBAUM; FILHO, 1995) Também, um arquivo possui uma extensão (nome.extensão) cuja esta dá mais informações a respeito do conteúdo do arquivo.

A exemplo de extensão de arquivo há o JSON (BRAY, 2017) (CROCKFORD, 2003) (Javascript Object Notation, ou em português Notação de Objecto Javascript) lançado em 2002, é uma formatação leve para troca de dados. O uso é facilitado tanto para seres humano quanto para máquina. O JSON é um formato de texto que é independente de linguagem, mas foi baseado no objeto provido do Javascript (ECMA-262, 1999).

Quanto aos tipos de dados suportados, o JSON (BRAY, 2017) é uma sequência de tokens. Os tipos de tokens aceitos é do tipo *object*, *array*, *string*, *number* e nomes literais como *false*, *true* e *null*.

Outra extensão de arquivo é o CSV (SHAFRANOVICH, 2005) (comma-separated values, ou em português Valores Separados por Vírgula) o qual é um arquivo do tipo de texto MIME (Internet Media) (FREED J. KLENSIN, 1996) que utiliza a encodificação de caracteres US-ASCII (HAUSENBLAS E. WILDE, 2014). Ao longo dos anos, seu uso foi consolidado para exportar dados entre vários softwares de tabelas (Microsoft suíte para Apple Suíte, por exemplo). A padronização do CSV de morou a ocorrer e por isso, vários outros estilos surgiram, a exemplo, o uso do CSV com ponto-e-vírgula (;). Outros estilos foram criados a ponto de ser chamado de arquivo DSV (RAYMOND, 2003). Por conseguinte, outro estilo que teve notoriedade na troca de dados entre bancos de dados ou tabelas de dados foi o TSV (KORPELA, 2000). A ideia é similar ao CSV, porém é utilizado uma tabulação em vez de vírgula.

2.2.2 Web Service

Um *Web Service* (GROUP, 2004) é definido como um software criado para suportar interoperabilidade entre máquinas através da rede computadores. Também possui uma interface descrita em um formato processável por máquinas (WSDL) e um protocolo para comunicação (SOAP). (GROUP, 2004) Essa era a arquitetura utilizada em 2004. Atualmente é predominante o uso de REST que em vez de exportar serviços como o SOAP, exporta os dados em si e não necessita do WSDL. (STACKIFY, 2017)

2.3 Dados Ausentes

O termo dados ausentes ou dados faltantes significa que está faltando dados suficientes para se formar uma informação e, por conseguinte, compreender o fenômeno de interesse ao observar o conjunto de dados. (MCKNIGHT, 2007) Esses dados podem ser perdidos ou não coletados em todas as etapas de geração de dados como um participante desistindo ou não respondendo parte da pesquisa, o pesquisador esquecendo ou perdendo seu dispositivo de anotação, má operação ao salvar em dispositivos eletrônicos etc. (MCKNIGHT, 2007)

O grande impacto dos dados ausentes está nos resultados da pesquisa, isto é, se esta se tornará tendenciosa, inconclusiva ou inconsistente. (MCKNIGHT, 2007) Um exemplo seria uma pesquisa de salários de executivos, os quais são coletados o sexo, a idade, o cargo e o salário. E por quaisquer motivos, os executivos do sexo masculino de idade acima dos 40 anos que tinham altos cargos e salário abaixo da média resolvessem não responder qual o seu salário. Uma avaliação sem perceber e tratar esse fenômeno pode inferenciar que os homens mais velhos de altos cargos ganham na média ou acima da média, o que se caracterizaria uma pesquisa incondizente com a verdade.

Para compreender e lidar melhor com os dados ausentes foram definidos os mecanismos de dados ausentes. Esses mecanismos são conceituados como a probabilidade de uma resposta ser observada ou estar faltando (HANDBOOK..., 2014) Existem 3 mecanismos conhecidos como faltando de forma completamente aleatória - *Missing completely at random (MCAR)*; faltando de forma aleatória - *Missing at random (MAR)*; faltando de forma não aleatória - *Not missing at random (NMAR)*. (HANDBOOK..., 2014)

2.3.1 MCAR

Um dado faltante é classificado como MCAR quando a probabilidade da resposta está faltando não é relacionada com outros valores do conjunto de dados nem com os dados que deveriam ser coletados. (HANDBOOK..., 2014) Vale ressaltar que é muito difícil relacionar este mecanismos nos conjuntos de dados reais. (HANDBOOK..., 2014) (LITTLE et al., 2016) Como visto na tabela 1 os dados ausentes MCAR não apresentam correlação com outras propriedades para justificar o dado faltante. Portanto, não há como prever qual o valor do dado faltante.

2.3.2 MAR

Quanto ao MAR, este é definido como a probabilidade da resposta está faltando depende dos dados obtidos, mas não está relacionado com dados não coletados. Este é o mecanismo menos arriscado de se assumir, pois permite a predição de resultados. (HANDBOOK..., 2014) (LITTLE et al., 2016) Na tabela 2 é possível visualizar um exemplo de dados ausentes do mecanismo MAR. Neste caso, assume-se que há correlação

Tabela 1. Exemplo de dados ausentes MCAR

ID	Estação do ano	Fruta	Receita
1	Verão	Laranja	Alta
2	Inverno	Laranja	Baixa
3	Verão	Morango	Baixa
4	Inverno	Morango	Baixa
5	Outono		Baixa

entre os valores da tabela e por isso, por predição, assume-se que o valor faltante seja "Alta".

Tabela 2. Exemplo de dados ausentes MAR

ID	Estação do ano	Fruta	Receita
1	Verão	Laranja	Alta
2	Primavera	Laranja	Alta
3	Verão	Limão	Alta
4	Inverno	Limão	Baixa
5	Verão	Laranja	

2.3.3 MNAR

Quanto ao MNAR, este é definido como a probabilidade da resposta está relacionada com os dados não coletados. (HANDBOOK..., 2014) (LITTLE et al., 2016) Isto é, por algum motivo que não está no conjunto de dados, há dados ausentes. Este mecanismo permite a geração de hipóteses para justificar a ausencia desses dados. Ainda na tabela 2 visualiza-se um exemplo de dados ausentes do mecanismo MNAR. O fato da receita de laranja não ter sido divulgada neste registro pode indicar que o produtor não queira preocupar os possíveis investidores (ou partes interessadas no agronegócio) devido uma possível baixa nos rendimentos.

2.4 Dados Discrepantes

Dados discrepantes ou *outliers* são dados que são significativamente diferentes dos outros dados do conjunto de dados. (AGGARWAL, 2012) Também conhecidos como anomalias, dados desviantes, ou discordantes na literatura, esses dados podem ser gerados, em geral, quando o sistema se comporta de forma não usual. (AGGARWAL, 2012) Por isso, a presença e a frequência de dados discrepantes também são informações relevantes para com o conjunto de dados. Exemplos desta relevância são para sistemas de detecção

de invasão, fraudes de cartão de crédito, diagnósticos médicos, estudos geológicos etc. (AGGARWAL, 2012)

Para identificar os dados discrepantes é um pouco mais subjetivo, isto é, mais dependente de critérios feitos por quem está avaliando, assim como de qual aplicação está sendo extraído o conjunto de dados. (AGGARWAL, 2012) Contudo, existe um espectro de dados normais para discrepantes como pode ser visto na figura 1. Nesta figura, justamente o limiar entre os normais para os ruídos e anormalias não são precisamente definidos, mas algoritmos de detecção de discrepância podem dar pontuação de discrepância para cada dado e utilizar este nível. (AGGARWAL, 2012)

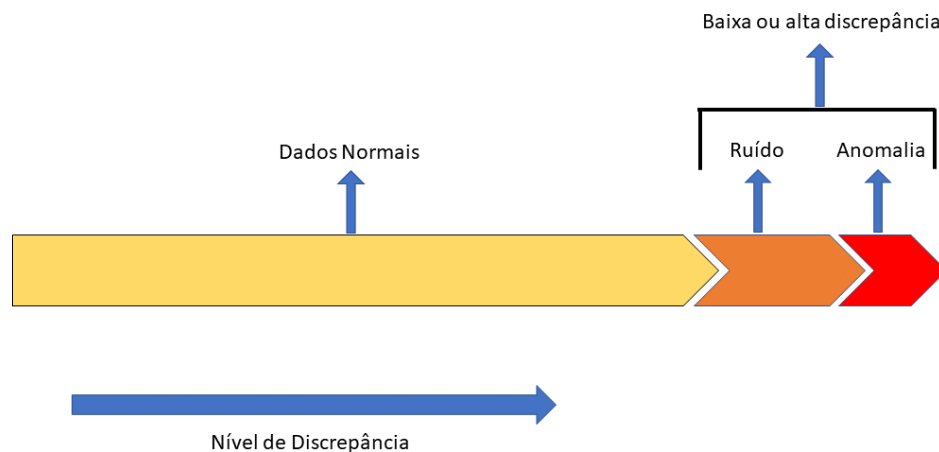


Figura 1. Exemplo de escala de discrepância (Adaptado de Aggarwal)

E se os dados discrepantes não forem tratados, eles podem gerar problemas como redução da precisão do modelo de dados, aumentar a complexidade do modelo, dificultar e legibilidade dos dados também. (AGGARWAL, 2012) (RATHI, 2019) E para tratá-los, as formas convencionais são remoção de instâncias, filtro de dimensões, combinar essas formas convencionais com algoritmos de validação - como o k-fold - ou detecção de anomalias - como baseado em clusterização, SMV ou densidade.

2.4.1 Dados Ruidosos e Anômalos

Dados ruidosos são dados indesejáveis, dimensões ou instâncias que não estão relacionadas com o fenômeno estudado. (RATHI, 2019) Em geral, dados ruidosos fazem com que algoritmos de aprendizado de máquina encontrem padrões incoerentes. (RATHI, 2019) Dados ruidosos e dados anômalos diferenciam-se, basicamente, na sua facilidade de

percepção em uma visualização e no seu grau de impacto ao inferir sobre os dados. Na figura 2, o item A é um exemplo de dado ruidoso, pois desvia-se levemente do padrão dos dados - uma reta. Quanto ao Item B este descaracteriza significativamente o padrão dos dados - este é um exemplo de dado anômalo.

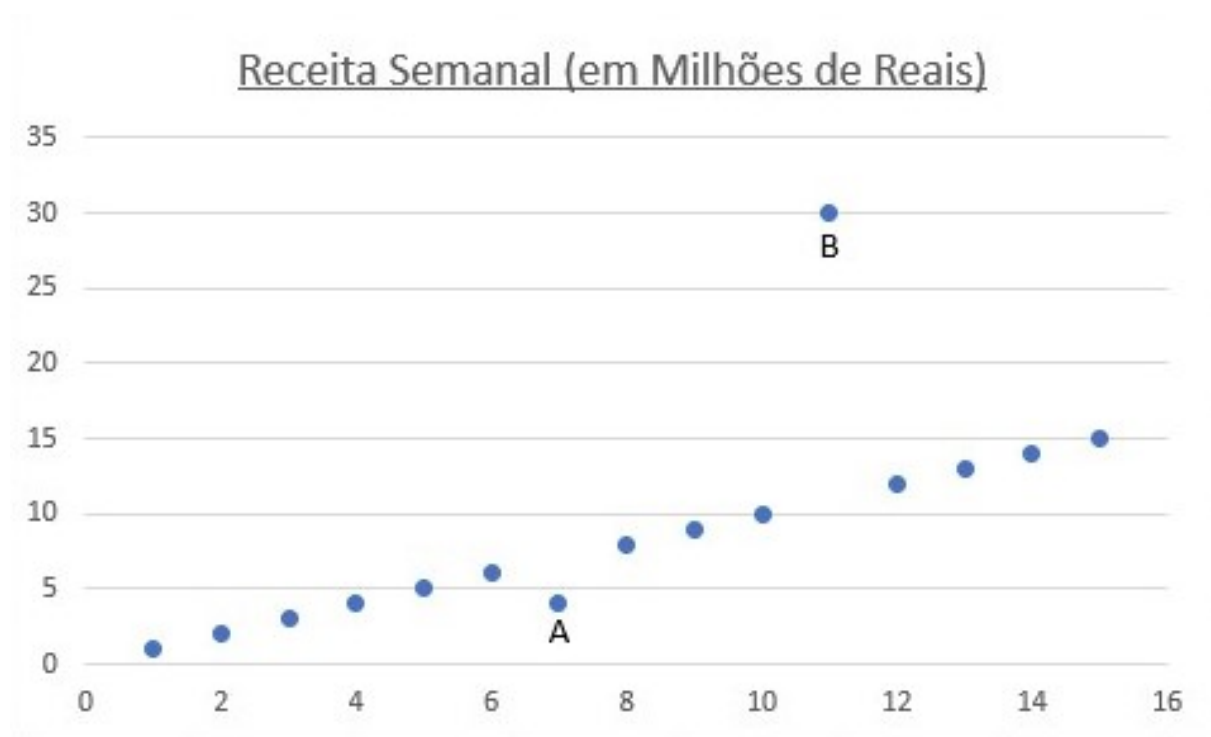


Figura 2. Exemplo de ruído e de anomalia.

Em dados tabulares, é possível classificar esses dados de 3 formas segundo (RATHI, 2019) (ver figura 3), sendo estas abnormalidades, dimensões irrelevantes ou instâncias ruidas. As abnormalidades são irregularidades tanto nas dimensões dos dados ou no evento estudado. As características irrelevantes são aquelas que não ajudam a explicar o fenômeno. E as instâncias ruidas são aquelas que desviam a forma dos outros dados. (RATHI, 2019)

Índice	Dimensão1	Dimensão2	Dimensão3	Dimensão4	Dimensão..	DimensãoN-1	DimensãoN	Evento			
Registro1											
Registro2											
Registro3											
Registro4											
Registro5											
Registro6											
Registro..											
RegistroN-1											
RegistroN											

Figura 3. Exemplo das classificações de ruído (adaptado de Rathi)

3 Trabalhos Relacionados

Nesse capítulo serão apresentados trabalhos acadêmicos e aplicações as quais possuem correlação com meu trabalho. Para isso, foram feitas análises sobre os trabalhos para identificar semelhanças e diferenças pontuais. Essa análise foi feita com o fim de comparar os trabalhos e identificar contribuições, suprir necessidades e captar trabalhos futuros.

3.1 Trabalhos acadêmicos relacionados

Albuquerque et al. (ALBUQUERQUE; LOWE; MAGNOR, 2011) descreveu um *framework* capaz de gerar dados sintéticos multidimensionais. O sistema (ver figura 4) recebe um *input* que representa algumas propriedades do conjunto de dados como número de dimensões, uma distribuição de dados padrão, tipo de dado de cada dimensão entre outros. A partir disso, é criada uma função densidade de probabilidade, com o fim de gerar um conjunto de dados padrão. Essas funções podem ser ajustadas e modeladas através de objetos. Também, essas funções podem ser de 1, 2 ou 3 dimensões. Adicionalmente, pode-se haver ruídos, para simular as irregularidades encontradas em conjunto de dados reais.

O framework apresentado também possui uma interface gráfica para auxiliar o usuário a configurar o conjunto de dados, bem como gerá-lo. Contudo, não foi encontrado uma interface para pré-visualização dos futuros dados gerados. Quanto aos tipos de dados, estes são restritos aos numéricos, quer sejam inteiros ou de ponto flutuante.

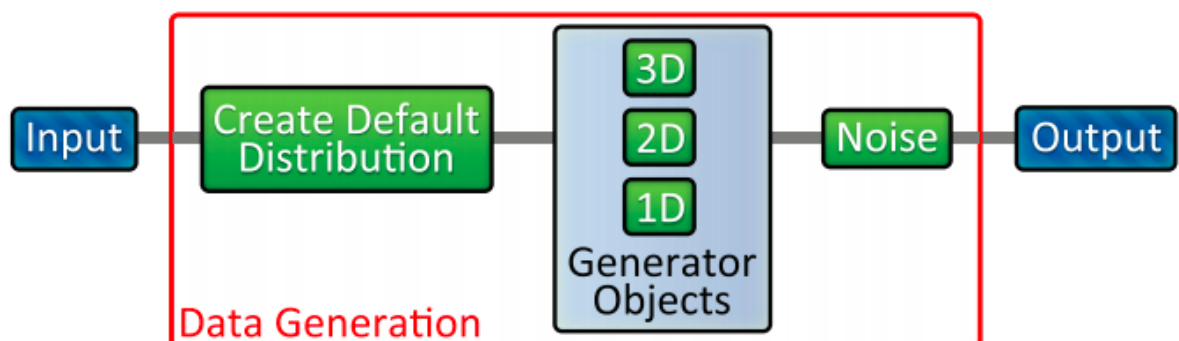


Figura 4. Visão Geral de geração de dados.

Wang et al. (WANG; RUCHIKACHORN; MUELLER, 2013) apresentou uma aplicação cujo principal diferencial é a capacidade de modelar, através de desenho, o comportamento das dimensões do conjunto de dados sintéticos. A priori, o usuário pode

iniciar o processo de geração através do zero, de um conjunto de dados já existente, ou um conjunto de dados aleatório. A partir disso, o usuário visualiza os dados no gráfico - que pode ser as coordenadas paralelas ou o *scatterplot* - e pode modificá-lo através de cliques e arrastos. Por conseguinte, os dados podem ser gerados e isto também serve como retroalimentação do sistema. Na figura 5 é possível visualizar a visão geral do funcionamento do SketchPad.

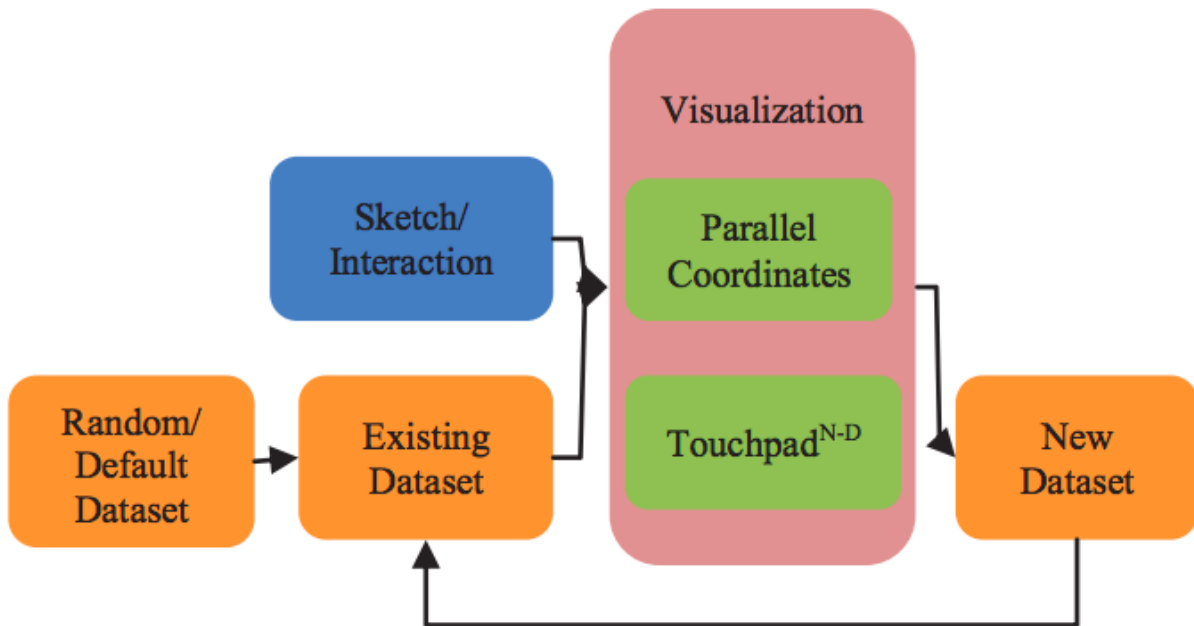


Figura 5. Visão Geral de geração de dados do Sketchpad.

Liu (LIU et al., 2016) criou um gerador de dados sintéticos a partir de avaliação de regras de aprendizagem. O sistema funciona criando regras de aprendizagem - usando algoritmos de árvore de decisão como o ID3 - baseado em dados de entrada contruindo correlações entre os dados. Na figura 6 é possível visualizar uma árvore de decisão. Durante a leitura do conjunto de dados de entrada é feita a árvore de decisão e, concomitantemente, são geradas as regras de aprendizagem. Essas regras são utilizadas para gerar amostras de dados sintéticos.

Garcia e Millán (GARCIA; MILLAN, 2011) criaram um sistema par gerar dados sintéticos pensado para desenvolvedores que buscam testar de forma eficiente e exaustiva a sua aplicação. Esses dados pode ser configurados (ver figura 7) de acordo com as preferências do usuário. As dimensões de dados seguem alguns padrões como a partir de fontes externas (Arquivos, Bibliotecas, Base de dados) Sequencial, Constante, Funcional, Intervalo ou Lista de valores.

Kofinas et al. (KOFINAS; SPYROPOULOU; LASPIDOU, 2018) criou uma metodologia para gerar dados sintéticos para simular consumo de água. A metodologia é avaliada através de algoritmos de validação - como a visualização dos resultados e fórmulas.

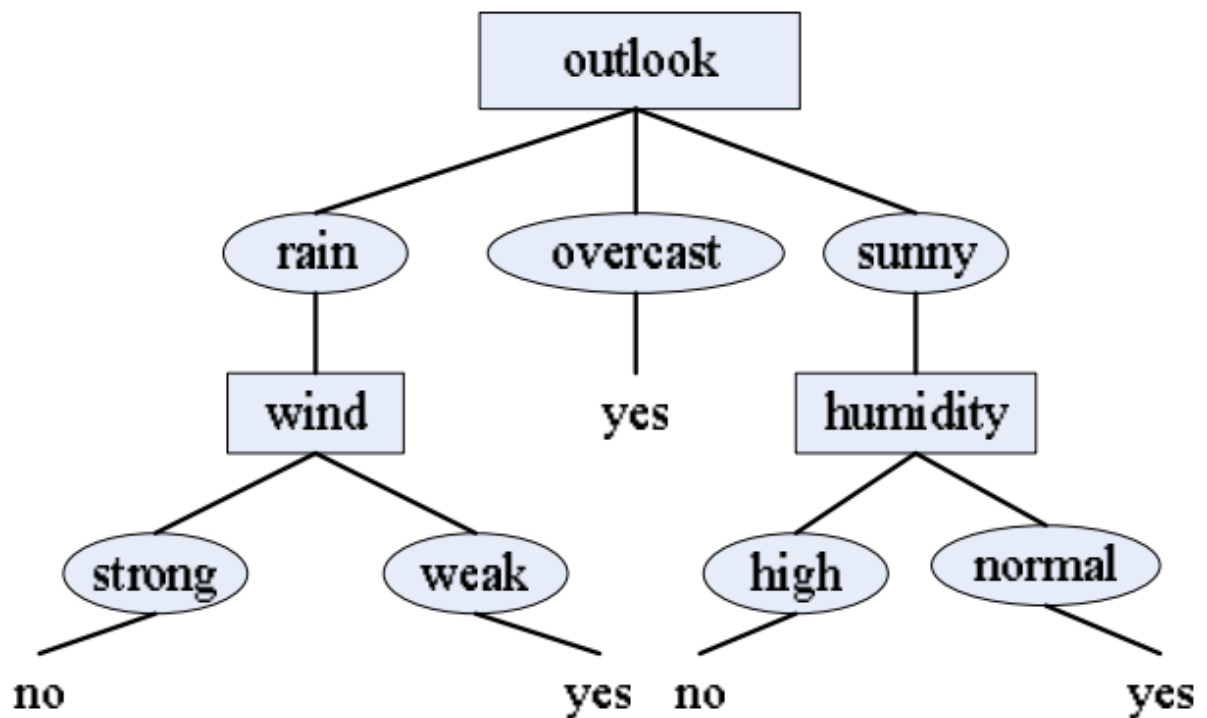


Figura 6. Exemplo de árvore de decisão para jogar tennis criado a partir de regras encontradas em um conjunto de dados.

Detalle

Información del campo

Nombre de la tabla	afiliado
Nombre del campo	apellido
Tipo de dato	varchar(200)
Permite valores nulos	No
Valor por defecto	no definido

Establecer restricciones

Porcentaje valores nulos:

Fuente de datos:

Configuración de fuente de datos

Conexión archivo:

Tipo de acceso:

Establecer

Estructura de la BDO

- afiliado
 - id
 - nombre
 - apellido
 - sexo
 - fecha_nacimiento
 - tipo_afiliacion
 - ingreso_mensual
 - cita_medica
 - medico
 - vacunacion

Figura 7. Exemplo da interface do usuário para configuração do gerador de dados.

Como pode ser visto na figura 8, a geração dos dados é feita a partir de 2 fases. A fase 1 serve, basicamente, para investigar a distribuição dos dados. Esta fase, primeiramente transforma dados números em séries temporais de 30 segundos. Em alguns casos, não há

registro, para isso, é criada uma tabela de incidentes e posteriormente uma probabilidade de existência de registro para que seja encontrada as classes usadas para construção do histograma de Pearson (DEAN; ILLOWSKY, 2009), por fim, são comparadas funções de distribuição com a atual com o fim de encontrar a que mais se aproxima.

Para a fase 2 cuida da geração de dados sintéticos propriamente. Basicamente, o sistema utiliza a distribuição criada na fase um para gerar os dados para 24h, respeitando as características diferenciadas para dias de semana e finais de semana.

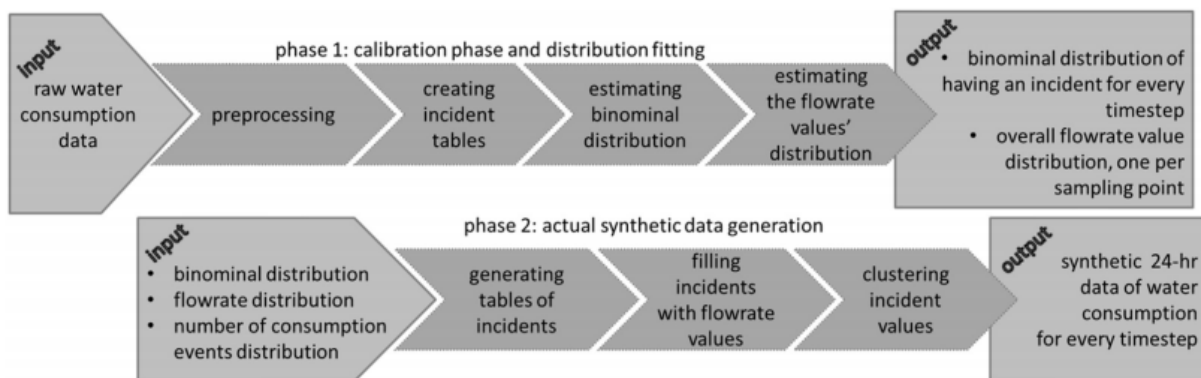


Figura 8. Fluxo de passos para geração dos dados sintéticos.

No trabalho feito por Sakshaug and Raghunathan (SAKSHAUG; RAGHUNATHAN, 2014) foi aplicado um procedimento de simulação não paramétrica para geração de dados sintéticos de variáveis contínuas com o foco em pequenas áreas geográficas. Segundo a avaliação do autor os dados sintéticos tiveram validade moderadamente alta em seus testes, mas ressalta a limitação do método não paramétrico. Em geral, os dados sintéticos se mostram promissores para geração de dados sintéticos para pequenas áreas geográficas, mas faltam testes mais aprofundados como dados de pesquisa em larga escala para substituir os dados reais por dados sintéticos em centros de dados de pesquisa. Na figura 9 é possível observar a comparação dos resultados da média da simulação paramétrica e da não paramétrica para cada atributo. Na simulação não paramétrica as médias dos dados sintéticos e reais ficam bem próximas, com exceção da idade (*age*), apresentando um bom resultado para a troca de dados reais para dados sintéticos.

Similarmente ao Blocks Data Generator, o projeto Threat Streaming Generator (TSG) (WHITING; HAACK; VARLEY, 2008) visa criar um gerador de dados sintéticos realistas com foco em dados para testes. É mostrado o fluxograma dos processos do TSG na figura 10. Primeiramente são definidos qual o tipo de conjunto de dados vai ser gerado. Em seguida são dadas 3 possibilidades ao usuário de inserir o ambiente e a ameaça: manualmente, através da ferramenta TSG e outras fontes. Por fim, esses dados são analisados por especialistas os quais são responsáveis pela qualidade do conjunto de dados gerado.

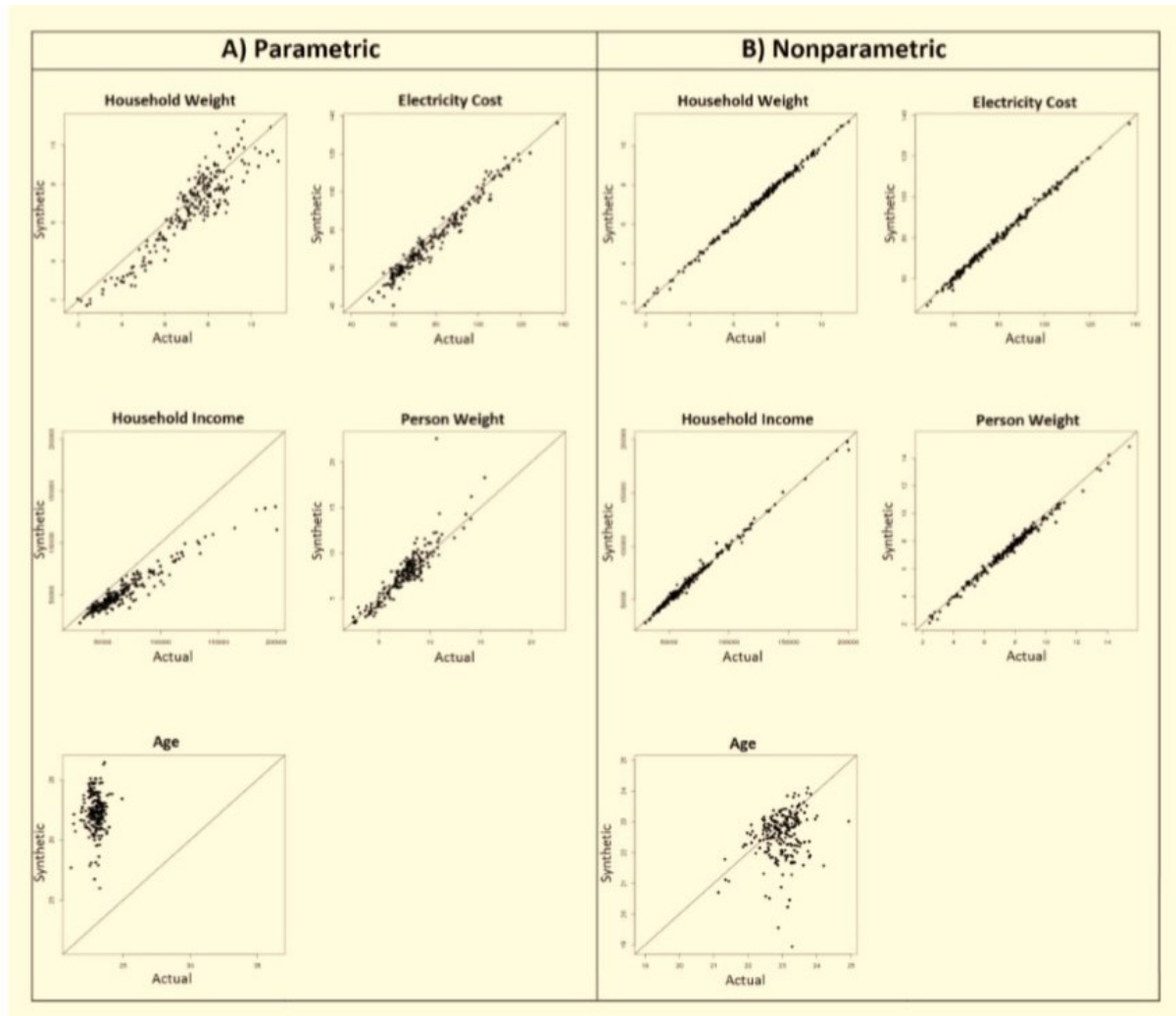


Figura 9. Comparação da média dos dados reais e sintéticos na simulação paramétrica e não paramétrica.

Com o foco em CARS (Context-Aware Recommender Systems - Sistemas de Recomendação sensíveis ao contexto), o DataGenCARS (RODRÍGUEZ-HERNÁNDEZ et al., 2017) é uma ferramenta para gerar dados sintéticos de forma flexível e prática. Permitindo que o usuário possa inserir os tipos perfis do usuário, tipos de contexto e itens, misturar dados sintéticos e dados reais com o fim de aumentar o realismo dos dados gerados. Na imagem 11 mostra-se, de forma geral, o funcionamento do DataGenCARS. De início a ferramenta mostra que é possível, opcionalmente, expandir outros conjuntos de dados bem como analisá-los estatisticamente. De qualquer modo, deve ser definido os esquemas de contexto usuários, itens e configuração da geração para que se tenha o conjunto de dados.

Pensando em oferecer confidencialidade dos dados governamentais, Larsen and Hockett (LARSEN; HUCKETT, 2012) desenvolveram um gerador de dados sintéticos que alia regressão de quantis com imputação *hot deck* e troca de classificação. A predição de regressão de quantis é feita para proteger dados sensíveis a partir de dados não sensíveis,

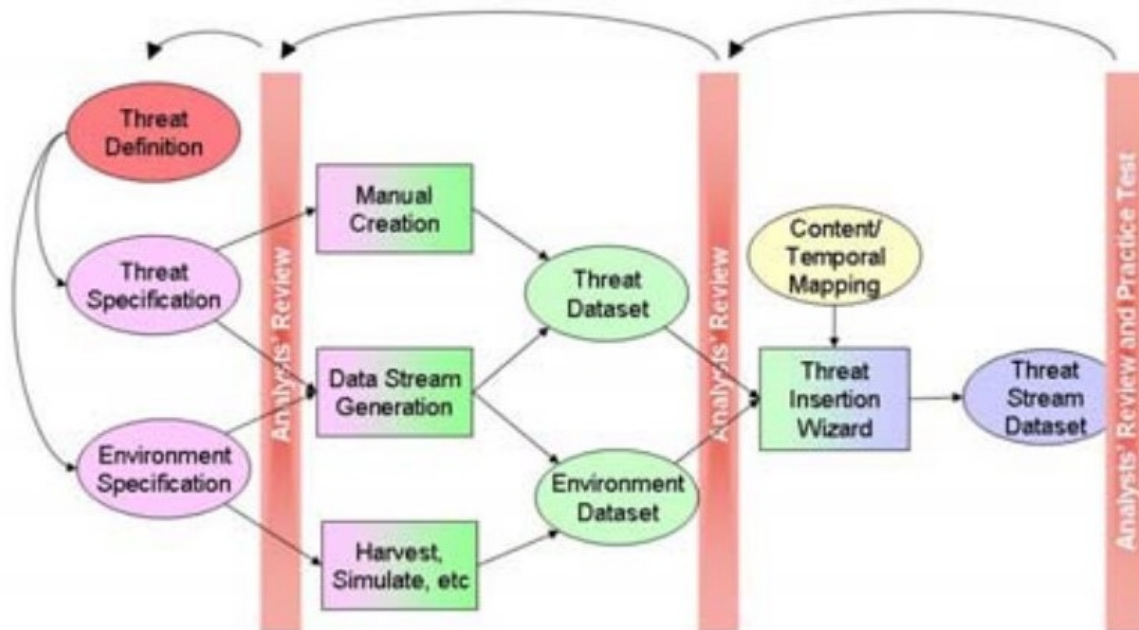


Figura 10. Fluxo de passos para geração dos dados sintéticos.

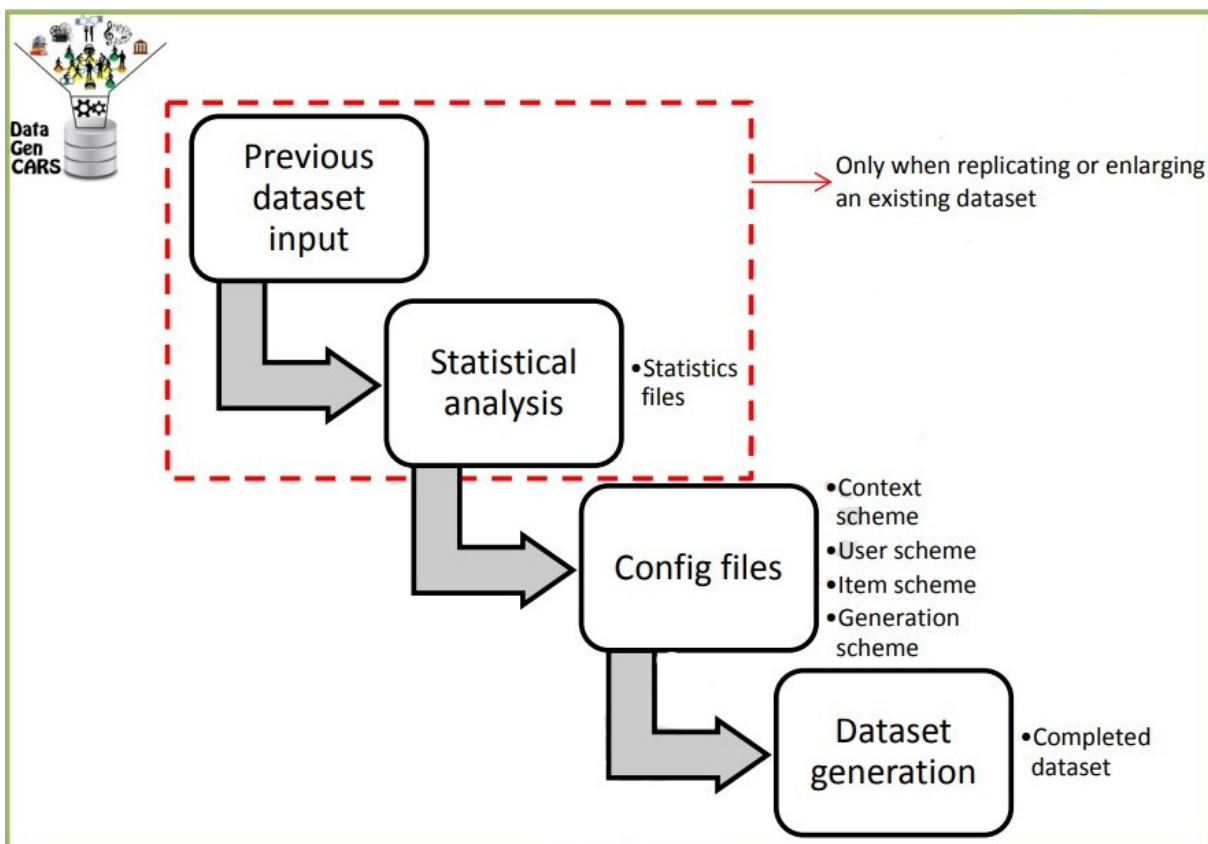


Figura 11. Fluxo de passos para geração dos dados sintéticos.

isto é, o conjunto de dados finais são compostos por dimensões reais - os não sensíveis - e sintéticos - para dimensões sensíveis. Em alguns casos os dados sintéticos são similares

aos reais e para isso foi utilizado a imputação *hot deck* junto com a troca de classificação, para garantir a aleatoriedade e confidencialidade dos dados.

3.2 Aplicações relacionadas

DTM Data Generator (SOFT., 2019) é uma plataforma de geração de dados sintéticos que existe de 1998. Esta possui suporte para geração de dados em arquivos, em banco de dados, também para *Big Data*. Possui suporte multiplataforma, através do modo *multiplatform runtime*, contudo é limitado quando comparado à versão Windows, o qual suporta a versão para servidor também. É válido destacar que é um software essencialmente pago, isto é, existem versões gratuitas - demonstrações, para ser mais exato - mas limitadas. Além disso, há categorias de versões pagas, que vão desde limitações de geração (Standart - Professional) à vantagens mais técnicas (Professional - Enterprise).

O DTM Data Generator possui uma vasta coleção de funcionalidades, as quais liberadas de acordo com as versões pagas. Adotando a versão mais cara, a lista de *features* é composta por geração de dados em JSON, XML, CSV ou geração por separador customizado. Também permite gerar dados por arquivo DSN (Database Source Name), gerar dados por linha de comando, e gerar um arquivo SQL para não seja necessário conexão com banco de dados.

É possível gerar cerca de 9.2 sextilhões de registros por *rule*, modos de atualizar dados existentes (adicionar, substituir e *Data Scrambling*), e suporte para bibliotecas de dados realistas. A plataforma disponibiliza entrada de dados através de SQL, XML, JSON, pela WEB através de HTTP ou FTP, XLSM, arquivos de texto e scripts em Python. Também é possível visualizar e testar os dados gerados, bem como gerá-los nos principais arquivos de texto (TSV, CSV, "DSV", JSON, XML) e banco de dados. (MS SQL Server, Oracle, DB2, MySQL, PostgreSQL, Informix, Sybase, SQLite e Firebird)

Há uma suíte de produtos relacionados fornecidos pela DTM soft. Além do gerador de dados, há o gerador de dados XML para teste de aplicação (DTM Test XML Generator); um gerador de planilhas Excel (DTM Data Generator for Excel) testador exaustivo - teste de estresse - de banco de dados (DTM DB Stress); Bem como editor, visualizador (DTM Data Editor), comparador e sincronizador de banco de dados (DTM Data Comparer) entre outros.

O SQL Data Generator (LTD, 2019) é um software que compõe uma suíte de ferramentas (chamada de SQL Toolbelt) da Red Gate. O software é exclusivo para o ecossistema Windows, com suporte do Windows 7 ao 10, à versão para servidores do Windows, ao SQL Server (2008 ao 2017), .NET e Oracle. Este produto é distribuído através de licenças pagas e vitalícias, com atualizações gratuitas e, no mínimo 1 ano de suporte gratuito. Vale ressaltar que é possível testar o produto por 14 dias gratuitamente.

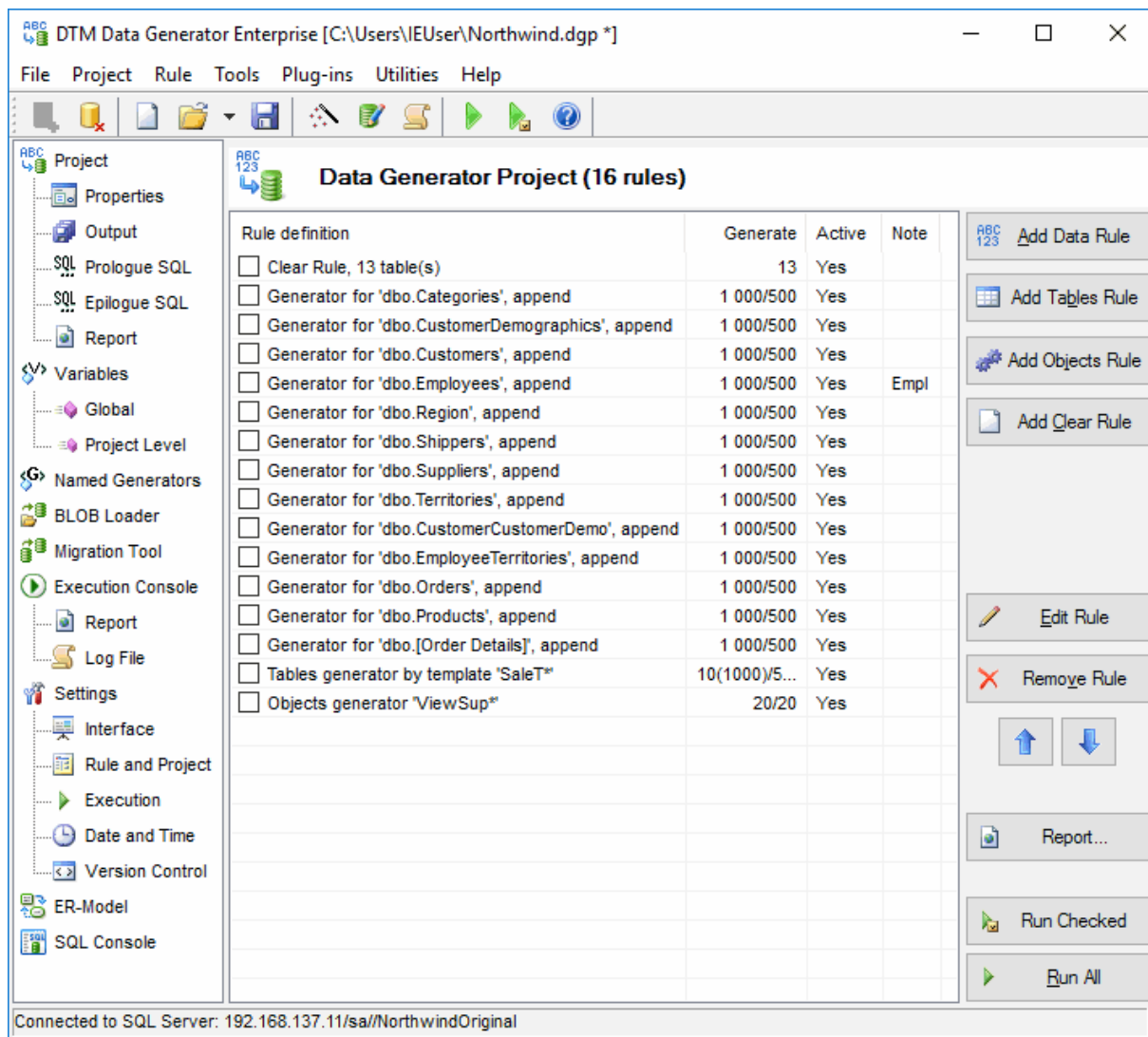


Figura 12. Usando o DTM Data Generator. Fonte: DTM Data Generator

O SQL Toolbelt tem funcionalidades bem delimitadas e a função do Data Generator é popular um banco de dados. A população acontece ao escolher, primeiramente, uma tabela do banco. A partir disso, escolhe-se um gerador para cada coluna da tabela. Um gerador tem classificação fortemente baseada na realidade, isto é, possui geradores como palavras relacionadas à compras, pagamentos, pessoas (primeiro e último nome), dado geográficos e afins. Contudo, também disponibiliza a geração a partir de expressões regulares *Regex generator* e scripts de python. Por se tratar de banco de dados, também há checagem e tratamento de *constraints*, *Foreign keys* e *Dependencies*. O SQL Data Generator também permite lidar com arquivos XML, quer seja para geração de valores XML, como utilizar como dados de entrada, além de mesclá-los com o *Regex generator*.

Quanto ao SQL Toolbelt oferecido pela Red Gate, ele conta com 2 modalidades, o completo com 14 programas e o *essentials* com 10. Entre os mais relevantes, pode-se citar o *SQL Data Compare*, *SQL Data Generator*, *SQL Test*, *SQL Backup Pro* e *SQL Scripts*

Manager.

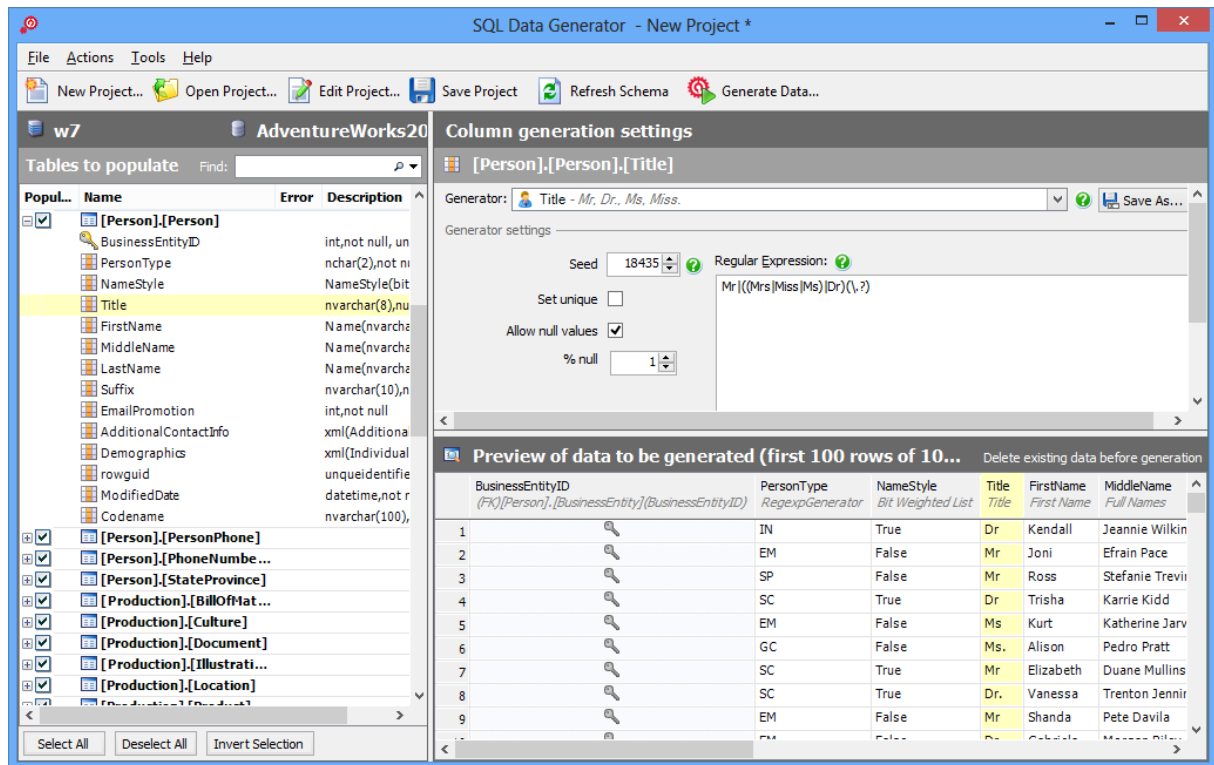


Figura 13. Usando o Redgate SQL Data Generator. Fonte: Red Gate SQL Data Generator.

Microsoft Visual Studio (MICROSOFT, 2019) é um pacote de programas da Microsoft para desenvolvimento de *software*. Este é composto por 4 versões (*Express*, *Professional*, *Premium*, *Ultimate*), e a opção de gerar dados para teste está disponível a partir da versão *Premium*. O foco é permitir que verifique o comportamento do banco de dados, sem relacioná-los com os dados da aplicação em produção.

Para gerar os dados de teste, deve-se utilizar os geradores de dados (*Data Generators*), que são correlacionados às tabelas do banco de dados. Os geradores podem ser dos mais primitivos (Binários, Inteiros, Data, *Float*), como de Imagem, Dinheiro, Expressão Regular, Categórico entre outros. Também é disponibilizado um Plano de Geração de Dados (*Data Generation Plan*), feito em XML, que contém informações do banco de dados, o tipo de dados de cada gerador e a quantidade de dados para ser gerado. Este plano serve basicamente para reutilização da lógica de teste.

Test Data Generator (DEVART, 2018) é uma ferramenta GUI (*Graphical User Interface*) pela dbForge para gerar dados de teste para banco de dados SQL desde 1997. O software possui mais de 200 geradores predefinidos e configuráveis os quais permitem a geração de dados mais inteligentes, isto é, mais próximos da realidade, como nomes, localização, dados de saúde e afins. Quanto à compatibilidade, este é exclusivo do ecossistema Windows, com suporte à versão 7 ao 10, do Windows Server 2008 ao 2019 e

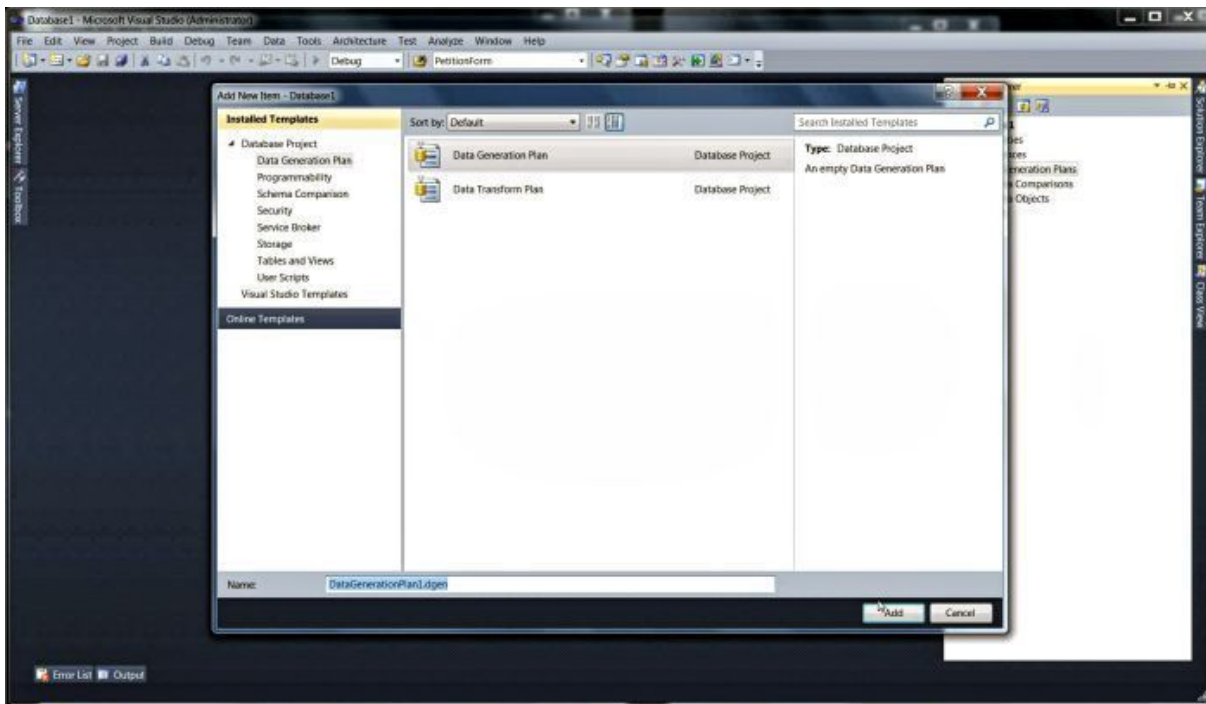


Figura 14. Usando o Microsoft Visual Studio. Fonte: anranik.

ao SQL Server Azure, 2008 ao 2017. Além da GUI, também há o suporte para geração de dados a partir da linha de comando. O produto é distribuído sob licenças pagas e vitalícias, porém, com suporte ao cliente com tempo limitado e com 30 dias gratuitos para avaliação.

Para usar o dbForge Test Data Generator, é preciso fazer uma conexão com banco de dados. A partir disso, utiliza-se os *Data Generators* para determinar o comportamento dos dados para determinada coluna da tabela selecionada no Banco de dados. Os Geradores de dados podem ser do tipo *emphBasics* e *emphAdvanced*. Do primeiro tipo, são formas mais próximas dos dados primitivos, como datas, texto *lorem ipsum*, JSON, *ReGex*. Já o avançado conta com número de cartão de crédito, aniversário, número de conta bancária internacional, IPv4, *hash* de senhas. A geração de dados resume-se à população de banco de dados, não há uma forma de exportar os dados em arquivos como CSV e JSON.

Há um suíte exclusivo para SQL Server, contudo também para Oracle, MySQL, PostgreSQL entre outros. Neste suíte, há várias ferramentas que auxiliam na manutenção, mas não, necessariamente, a geração de dados, a exemplode um *previewer*. Destes, pode-se citar um comparador de dados, criador de *querys*, um monitor - para supervisão do banco de dados - e afins.

Mockaroo (MOCKAROO, 2019b) é um *web site* e *framework* para desenvolver dados de teste. Há um total de 143 geradores, sendo a maioria considerados geradores realistas. Por ser um site, é possível acessá-lo por qualquer sistema operacional, dependendo apenas de conexão com a internet. O produto possui versões gratuita e pagas. - *Free*, *Silver*, *Gold*, *Enterprise* as quais variam no *host*, o qual pode ser do Mockaroo ou privado,

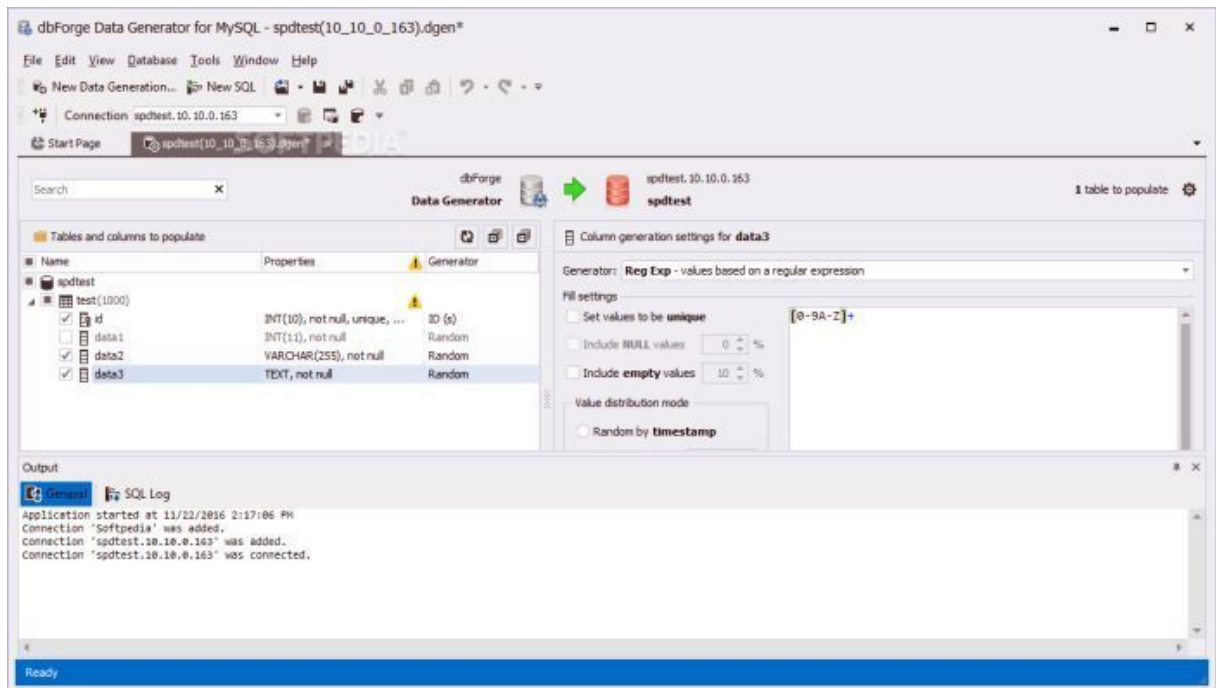


Figura 15. Usando o dbForge Test Data Generator. Fonte: anranik.

máximo de registros por download, velocidade de download e preço.

Na tela inicial, é possível escolher o nome da coluna, o tipo de gerador, algumas opções - valores em branco e funções a partir dos dados. Ainda nesta tela, encontra-se o botão para *download* dos dados, pré-visualização dos mesmo (mas sem gráficos), algumas configurações como quantidade de linhas, formato dos dados para *download*, botão para clone ou deleção de banco de dados, e importação de dados csv/Excel ou SQL.

Outro serviço interessante do Mockaroo é Mockaroo APIs (MOCKAROO, 2019a). Este consiste em baixar dados programaticamente através de requisições REST (*Representational State Transfer*). As requisições podem ser feitas de 2 formas, a *Generate API* - gera os dados através de um banco de dados salvo e os envia pelo corpo de uma requisição - e *Mock APIs* que basicamente, simula um *back-end* como tratamento de parâmetros e simulação de erros. É pensado para desenvolvimento ágil de aplicações *front-end*, isto é, sem perder muito tempo com o *back-end* a priori.

Field Name	Type	Options
<input type="text" value="id"/>	<input type="text" value="Row Number"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="text" value="first_name"/>	<input type="text" value="First Name"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="text" value="last_name"/>	<input type="text" value="Last Name"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="text" value="email"/>	<input type="text" value="Email Address"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="text" value="gender"/>	<input type="text" value="Gender"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×
<input type="text" value="ip_address"/>	<input type="text" value="IP Address v4"/>	blank: <input type="text" value="0"/> % <input type="text" value="fx"/> ×

Rows: Format: Line Ending: Include: ☒ header ☐ BOM

Want to save this for later? [Sign up for free.](#)

Figura 16. Usando o Mockaroo. Fonte: anranik.

4 Arquitetura do projeto

O software chamado de Blocks Data Generator é *Open Source* e está hospedado no GitHub em <http://github.com/gustavoresque/DataGenerator>. Em termos de organização do projeto, foi adotado o padrão de arquitetura de *software* MVC (*Model, View, Controller*), um modelo incremental de desenvolvimento, com reuniões diárias para discussão de problemas e melhorias, e utilização da ferramenta Trello (<https://trello.com>) para organização e persistência das informações.

Quanto ao desenvolvimento, foi utilizada a linguagem Javascript com foco para *Desktop*, através do *Framework* Electron (<https://electronjs.org/>). Também foi adicionado o *jQuery* para agilizar a codificação do projeto e o Node.js 10 (<https://nodejs.org/en/>) para acessar recursos do sistema operacional, para o desenvolvimento do *Web Service* e também para dar suporte ao Electron. Do Javascript foi utilizado o EcmaScript 6 (2015) e seus novos recursos como o desenvolvimento assíncrono com as *Promises* e *arrow functions*.

4.1 Casos de uso do sistema

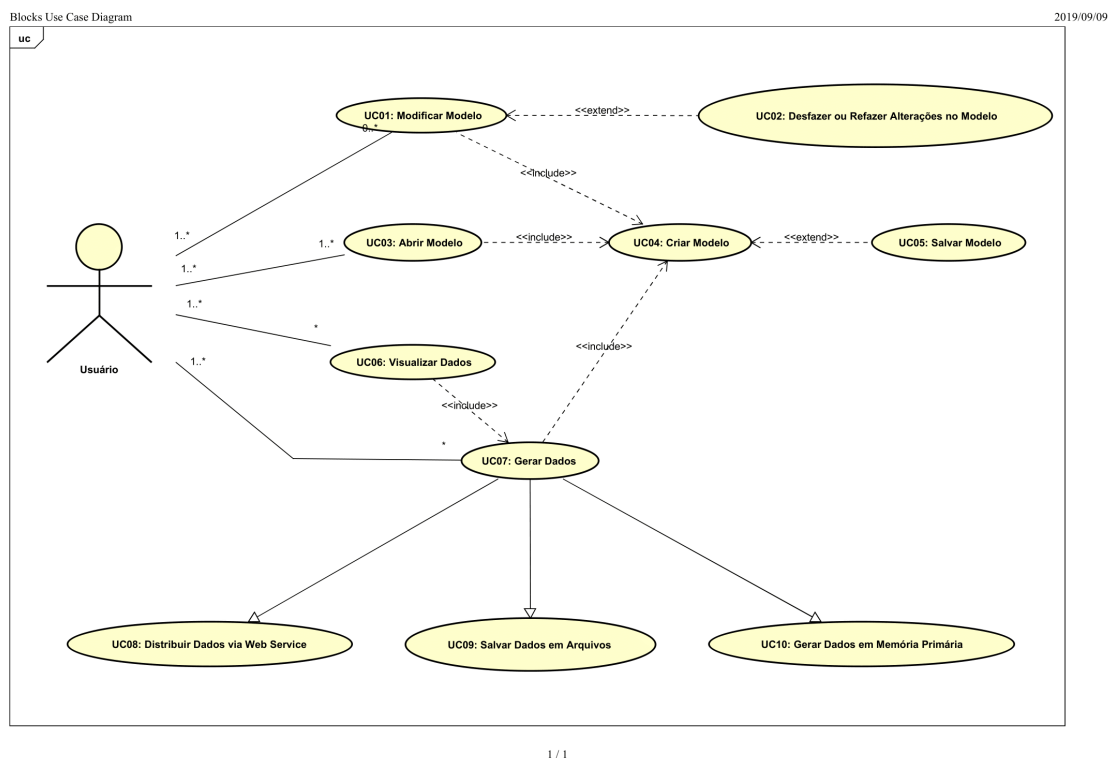


Figura 17. Diagrama de Caso de uso do Blocks Data Generator. Fonte: o autor.

"O objetivo de um diagrama de casos de uso na UML é demonstrar as diferentes maneiras pelas quais um usuário pode interagir com um sistema." (INC., 2019) Por conta disso, é utilizado o Diagrama de Casos de Uso para demonstrar as principais funcionalidades do protótipo Blocks Data Generator. Na figura 17 são encontrados um total de 10 casos de uso.

No UC01 mostra que é possível modificar um modelo, isto é, adicionar ou remover dimensões, alterar geradores e afins; também há inclusão do UC04 e a possibilidade de desfazer ou refazer as alterações (UC02). O UC04 representa a criação do modelo. Esta é considerada a funcionalidade original, pois é incluída por vários casos de uso e não inclui nenhuma funcionalidade. Outros casos de uso que possuem relação com UC04 é o UC03 e UC07 como inclusão e UC05 como extensão.

UC03 é uma funcionalidade para reaproveitamento do modelo, também está atrelado à validação de teste por outros pesquisadores. UC07 é o carro chefe do Blocks Data Generator, o qual refere-se à geração de dados, progredindo para o *Big Data* ou não. Esta geração pode ser feita, especificamente, de 3 formas: gerando dados em Memória, para ser utilizado dentro do sistema (UC10); salvar em arquivos - TSV, CSV e JSON - (UC09); e distribuir os dados através de requisições HTTP (UC08).

Incluindo a UC07, mais especificamente a UC10, a UC06 permite que o usuário visualize os dados. Assim, a visualização - a qual pode ser rápida (através do *preview*) ou detalhada (com o VisTechLib) - tendo um panorama do comportamento dos dados. Com isso, pode-se validar os dados, ter insights para aprimorar o modelo e afins.

5 Protótipo

Este capítulo é dedicado em explicar mais sobre o protótipo, seu fluxo de funcionamento, funcionalidades, mais detalhes sobre a interface do usuário entre outros. De modo geral, o protótipo é chamado de Blocks Data Generator e visa ser um gerador de dados sintéticos baseado em modelos de dados. Assim, o usuário pode manipular um ou mais modelos e cada modelo pode conter N dimensões, que por sua vez podem conter M geradores de dados encadeados.

Os geradores de dados podem gerar dados numéricos, categóricos, temporais etc (haverá uma seção específica para geradores) e o resultado de um gerador pode servir de entrada para outro gerador através de operadores. Os operadores podem ou não aplicar uma operação matemática (soma, subtração, divisão, multiplicação) ao resultado do gerador anterior - a leitura de anterior e posterior é da esquerda para a direita, respectivamente. Junto com os operadores, também há outras propriedades que variam de acordo com o gerador.

Ainda na modelagem das dimensões, é possível modificar seu nome, verificar o tipo do dado gerado pelo gerador, o ID e se está disponível para geração e visualização. Essa disponibilidade (chamado de *display*) foi feita para o caso de haver um modelo em que nem todas as dimensões sejam necessárias em determinado momento, mas também não queira perdê-las. Adicionalmente, é possível copiar e colar dimensões através de atalhos no teclado, bem como adicionar ou excluir dimensões, esta por só meio de um botão.

Também é disponibilizado um pré-visualizador de dados com apenas um gráfico - o coordenadas paralelas -, o qual é colorido e interativo e seu volume de dados é independente do volume de dados para ser gerado. Além do *preview*, há uma integralização com um visualizador de dados mais elaborado e com mais opções de visualização, o *VisTechLib*.

Outrossim, há um botão específico para gerar os dados em arquivos JSON, CSV, TSV ou por de requisições HTTP do tipo GET (*Web Service*). Vale ressaltar que é possível configurar, em ambiente dedicado, a quantidade de dados gerados, pré-visualizados, formato dos dados gerados, se contém a legenda dos dados no arquivo final e as propriedades do *Web Service*.

Na figura 18 é possível visualizar como foi pensada a utilização da aplicação. Na primeira raia, encontra-se como o usuário pode configurar o modelo. Basicamente, o usuário define as dimensões e os seus geradores. E o comportamento dos dados pode ser validado pelo *preview*. Caso seja necessário, o modelo pode ser atualizado a qualquer momento.

A segunda raia demonstra os caminhos para geração de dados. É possível a geração

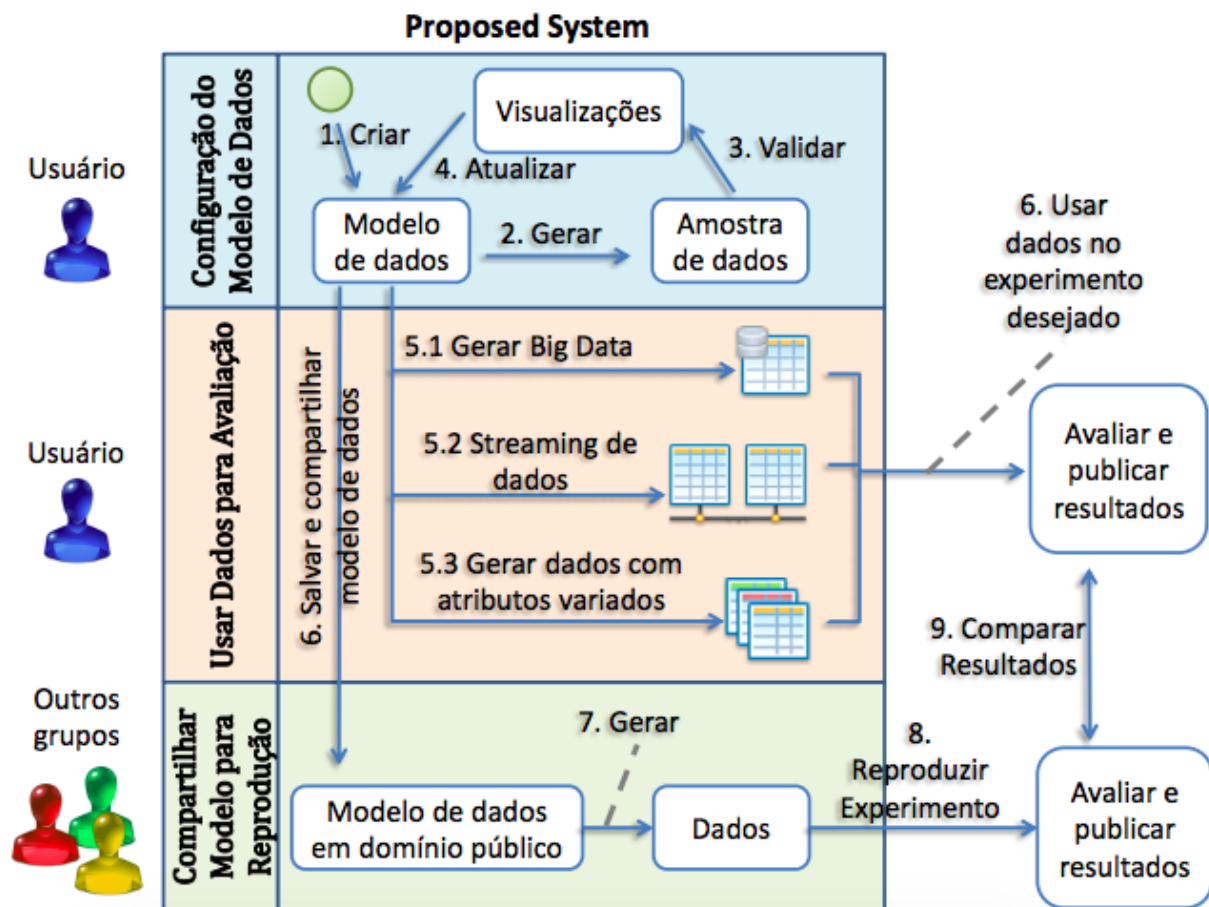


Figura 18. Fluxograma de utilização do Blocks Data Generator. Fonte: Yvan Brito, 2019.

de dados por *Big Data*, isto é, um grande conjunto de dados multidimensionais, apenas por meio de arquivos. Também a geração de dados por *Streaming*, o qual dar mais controle ao usuário sobre o processo de geração, e também é disponibilizado o *Web Service*. E o caminho 5.3 demonstra uma forma iterativa de geração de arquivos de dados, com mudanças programadas de atributos. Esses dados podem ser usados em experimentos, testes e afins, cujos resultados podem ser publicados.

A terceira raia funciona como um agrupamento das duas primeiras, mas externa. Isso para demonstrar que os processos anteriores podem ser replicados a partir do mesmo modelo de dados e o resultado pode ser comparado, facilitando a produção científica de pesquisadores.

5.1 Tipos de Geradores de Dados

Na tabela 3 são apresentados, de forma geral, todos os geradores do Blocks Data Generator. São apresentados seus nomes, suas categorias, o tipo de valores retornados, os tipos de parâmetros, se possui correlação - o que na prática é se um dos parâmetros é uma outra dimensão - , se são independentes - ou seja, se precisa encadear geradores para

Tabela 3. Propriedade dos geradores do Blocks Data Generator

Nome	Categoria	T. dos valores	Correlação	Independente
Constant	Sequencial	N ¹	Não	Sim
Counter	Sequencial	N ¹	Não	Sim
Fixed Time	Sequencial	T ³	Não	Sim
Sinusoidal Sequence	Sequencial	N ¹	Não	Sim
Custom Sequence	Sequencial	N ¹	Não	Sim
Poisson Time	Aleatório	T ³	Não	Sim
Uniform	Aleatório	N ¹	Não	Sim
Gaussian	Aleatório	N ¹	Não	Sim
Poisson	Aleatório	N ¹	Não	Sim
Bernoulli	Aleatório	N ¹	Não	Sim
Cauchy	Aleatório	N ¹	Não	Sim
Weighted Categorical	Aleatório	C ²	Não	Sim
Categorical	Aleatório	C ²	Não	Sim
Categorical Quantity	Aleatório	C ²	Não	Sim
Linear	Função	N ¹	Sim	Sim
Quadratic	Função	N ¹	Sim	Sim
Polynomial	Função	N ¹	Sim	Sim
Exponential	Função	N ¹	Sim	Sim
Logarithm	Função	N ¹	Sim	Sim
Sinusoidal	Função	N ¹	Sim	Sim
Categorical	Função	C ²	Sim	Sim
Piecewise	Função	N ¹	Sim	Sim
TimeLaps	Função	T ³	Sim	Sim
MCAR	Acessório	N ¹ , C ² ou T ³	Não	Não
MAR	Acessório	N ¹ , C ² ou T ³	Sim	Sim
MNAR	Acessório	N ¹ , C ² ou T ³	Não	Não
Noise	Acessório	N ¹ , C ² ou T ³	Não	Não
Constant Noise	Acessório	N ¹ , C ² ou T ³	Não	Não
Range Filter	Acessório	N ¹ , C ² ou T ³	Não	Não
Linear Scale	Acessório	N ¹ , C ² ou T ³	Não	Não
No Repeat	Acessório	N ¹ , C ² ou T ³	Não	Não
MinMax	Acessório	N ¹ , C ² ou T ³	Não	Não
Low-Pass Filter	Acessório	N ¹ , C ² ou T ³	Não	Não
Get Extra Value	Acessório	N ¹ , C ² ou T ³	Não	Não
CubicBezier	Geométrico	N ¹	Não	Sim
Path2D Stroke	Geométrico	N ¹	Não	Sim
Path2D Fill	Geométrico	N ¹	Não	Sim

funcionar corretamente. Na tabela 4 é mostrado um breve resumo para um entendimento abrangente do comportamento de cada gerador.

¹ Numérico

² Categórico

³ Temporal

5.1.1 Sequencial

Os geradores da categoria Sequencial geram valores encadeados dado um padrão. É possível gerar o próprio padrão a partir do gerador *Custom Sequence*, o qual você determina um valor Inicial (*Begin*), o valor Intervalar (*Step*), isto é, o qual vai ser incrementado ou decrementado dado uma Sentença (*sentence*) customizada.

Contudo, já são predefinidos alguns geradores como o *Constant*, o qual define um valor único de geração; o *Counter*, funciona como um contador, onde-se é definido o valor Inicial e o Intervalar; o *Fixed Time Generator* gera um intervalo de tempo, onde-se define o valor inicial (*init*), Intervalar e a máscara (*mask*), isto é, como o tempo deve ser formatado; o *Sinusoidal Sequence* gera de acordo com a função senoidal, que, além do valor Inicial e Intervalar, há o 'a' de Amplitude, 'b' de frequência angular e 'c' para representar a fase da onda.

5.1.2 Aleatório

A categoria aleatória de dados contém um significativo número de geradores, pois são mais fáceis de se dissociar da realidade, pelo caráter aleatório, mas também de reaproximar, pelo caráter probabilístico.

Esta categoria conta com geradores uniformes, isto é, a distribuição dos dados é equalizada; Também há um gerador de dados de tempo, parecido com o *Fixed Time Generator* com a diferença que o comportamento é definido pela fórmula de Poisson e que há mais duas configurações: unidade de tempo - a qual pode ser desde milissegundos a anos - e o lambda, advindo da fórmula. Há uma distribuição de poisson também, apenas com o lambda; É disponibilizado geradores de fórmulas clássicas com a normal (Gaussiana), Bernoulli, e Cauchy com seus devidos parâmetros.

Além de números, também é possível gerar dados categóricos (*Categorical*), dadas as palavras - também chamado de categorias - inicialmente. Similarmente há o *Weighted Categorical* que possui valores de probabilidade para cada palavra, e já o *Categorical Quantity*, em vez de probabilidade, define quantas vezes cada palavra deve aparecer.

5.1.3 Funcional

A categoria funcional (*Function*) serve para gerar dados de acordo com outra dimensão chamado de *input*, isto é, facilita a correlação entre dimensões. Para dados numéricos, disponibiliza-se as função de primeiro grau (*Linear Function*) e segundo grau (*Quadratic Function*), exponencial (*Exponential Function*), logarítmica (*Logarithm Function*) e a *Piecewise Function*, cuja função é definida por subfunções, e no caso, é possível definir o gerador desejado até um determinado valor chamado de *Intervals* e depois pode-se escolher outro gerador.

Para dados categóricos, há a função categórica (*Categorical Function*) e a *TimeLaps Function* a qual funciona de forma semelhante ao gerador *Piecewise Function*, só que utiliza uma quantidade de tempo como limiar - imagine uma corrida de fórmula 1 e cada vez que os carros passam pela linha de chegada eles completam uma volta. Esta volta é o limiar também chamado de *Laps* - e para o *input*, somente geradores de tempo.

5.1.4 Acessórios

Os geradores da categoria Acessórios (*Acessory*) foram pensados especialmente para serem concatenados com outros geradores, com o fim auxiliá-los. Entre os geradores acessórios, pode-se citar o *Missing Value*, o qual foi subdividido em 3 geradores: o *MCAR*, *MAR* e *MNAR*. O *MCAR* usa a probabilidade para definir qual dado será faltante; O *MAR* e o *MNAR* são similares. Eles trabalham de forma diferente para cada tipo de dado. Os tipos de dado são os Numéricos, os Categóricos e os Temporais. Para os Numéricos e os Temporais, é definido um intervalo no qual os dados vão ser faltantes. Para os dados Categóricos, é definida uma lista de categorias na qual todas essas categorias faltarão na geração de dados. Quanto a diferença entre *MAR* e *MNAR* está em que o *MAR* possui correlação com outra dimensão e é independente de outros geradores, enquanto o *MNAR* não possui correlação com outras dimensões, mas é dependente de outros geradores.

O *Noise Generator* que adiciona dados fora do padrão, conhecido como ruído, com uma determinada probabilidade, intensidade - o que ajuda a criar o nível de discrepância - e a partir de 3 distribuições: uniforme, gaussiana, e de Poisson. O *Constant Noise Generator* também adiciona ruídos, mas só que é um valor específico com determinada probabilidade de ser adicionado; o *Ranger Filter* permite retirar do conjunto de dados os valores que estão entre os valores de início (*Begin*) e fim (*End*); o *Linear Scale* permite que os determinados (selecionados através do *MinIn* e *MaxIn*) dados sejam escalados através do *minOut* e *MaxOut*.

O *No Repeat* retira dados repetidos do conjunto; o *MinMax* define quais valores serão os maiores e menores de acordo com os parâmetros dados; o *Low-Pass Filter* faz jus ao nome e filtra pela "Amplitude" do dado. Na prática, o valor sucessor é uma media ponderada (valor recebido pelo parâmetro *Smooth*) do valor anterior com o valor gerado; o *Get Extra Value* pega os retornos extras dos geradores que retornam mais que um valor.

5.1.5 Geométrico

Os geradores Geométricos (*Geometric*) permitem que seja gerado dados para desenho de polígonos etc. Para isso são disponibilizados três geradores. O primeiro deles (*Path2D Stroke*) gera dados bidimensionais a partir do polígono inserido pelo usuário. O segundo (*Path2D Fill*) gera os dados a partir do preenchimento do polígono inserido (ver

algoritmo *Floodfill*) pelo usuário. Quanto ao terceiro (*CubicBezier*), este gera dados para desenho de uma curva bezier cúbica a partir de seus pontos.

5.1.6 Baseado em dados reais

Para esta categoria, existe apenas um gerador, chamado como *Real Data Wrapper*. Basicamente, ele é criado automaticamente quando o usuário importa um conjunto de dados reais através de um CSV, por exemplo. Este gerador recebe tantos valores categóricos como numéricos e essa informação pode ser decidida automaticamente pelo gerador ou ser forçada pelo usuário. É possível gerar uma quantidade superior de dados em relação ao conjunto de dados real, para isso é feito um tratamento de imputação de dados. Esse tratamento é feito através de funções de geração, chamadas de *GenType*.

Essas funções pode ser do tipo *Standart*, que é pegar os dados do início ao fim de forma cíclica até chegar ao número desejado de registros. Também pode ser do tipo *Reverse* que ao invés do *Standart*, pega os dados do final ao início. É disponibilizado o modo aleatório (*Random*), e algumas variações.

A primeira variação é o *QuartileRandom*, que divide o conjunto de dados em 3 marcos e a probabilidade de se pegar um dado daquele quartil é proporcional ao tamanho do marco. A leitura dos Marcos pode ser visualizada na figura 19. Então, se o valor do espaço entre 0 e quartil 1 for 100, todos os dados gerados serão do primeiro 1/4 do conjunto de dados. A segunda variação é o *AverageRandom* que utiliza o valor da média e da variancia - [Média - Variancia, Média + Variancia] de um conjunto de dados numérico ou utiliza os N valores categóricos mais frequentes com distribuição uniforme.

5.2 Modos de Geração de Dados

Cada gerador possui um comportamento e este foi demonstrado na subseção anterior. Nessa seção é demonstrado como é feita a concretização desse comportamento, isto é, os dados em si. Gerar os dados é tão importante quanto gerar o modelo, visto que os dados gerados podem ser utilizados para testes de aplicações, por exemplo.

Os dados são gerados por *Web Service* e 2 em tipos de memória: a primária e a secundária. A memória primária serve de base para escrever na memória secundária e também alimenta as visualizações de dados. A memória secundária e o Web Service são descritos nas subseções a seguir.

5.2.1 Streaming Data

O protótipo unificou o processo de geração de dados em arquivos - antes era um modo padrão, bloqueante; e outro não bloqueante - para que otimize a geração de grande

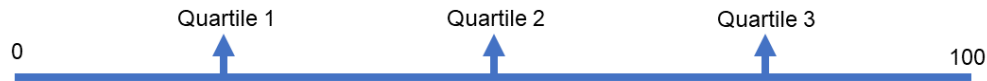


Figura 19. Ilustrando a leitura dos marcos dos quartis. O tamanho do espaço entre os quartis ou entre 0 ou o 100 é o valor da probabilidade de um número ser desse espaço. Fonte: O Autor.

volumes de dados (*Big Data*). Para isso, foi utilizado o conceito de *Streaming Data*, isto é, gerar o volume de dados aos poucos - em blocos - para que não haja estouro de memória primária. Também foi utilizado esse processo assincronamente, para que a interface de usuário não seja bloqueada durante o processo e permitir que o progresso da geração seja acompanhado.

O usuário escolhe a quantidade de dados a ser gerada e a aplicação define automaticamente a quantidade de blocos. Cada bloco é fixado em até 10.000 instâncias. O bloco é processado e armazenado temporariamente na memória primária até que ele esteja completo. Então, o bloco é escrito na memória secundária.

5.2.2 Web Service

Quanto ao *Web Service*, este foi pensado para facilitar o teste de aplicação. Cada modelo é independente, isto é, podem ser habilitados somente os modelos desejados para distribuição. E além da configuração por dentro do *software*, também é possível criar configurações temporárias para cada requisição, sem alterar as configurações do modelo.

Os parâmetros disponíveis para configuração temporária pela URI são: o nome do modelo, o formato dos dados e a quantidade de registro. É disponibilizado um ícone de aviso ao usuário quando um modelo está distribuindo dados via *Web Service* na aba do modelo. Um exemplo de URI para fazer requisição HTTP do tipo GET é: (<http:

`//localhost:8000/?modelid=MODEL_r6w2ffk3.mva&nsample=100&format=csv>`), nos quais "modelid" é o *ID* do modelo, "nsample" é a quantidade de instâncias desejada e "format" é o formato do arquivo desejado.

5.3 Modos para Visualização de Dados

O protótipo permite modelar os dados, gerá-los e também visualizá-los. Nessa seção são mostradas as duas formas de visualizar dados no Blocks. A primeira é mais simples, acessível e fica na tela inicial do gerador chamado de *Preview*. A outra é um programa a parte que é integrado ao Blocks, isto é, há o compartilhamento do modelo de dados.

5.3.1 Preview

O pré-visualizador de dados foi criado pensando em oferecer uma visualização rápida e abrangente do modelo de dados. Para isso, foi escolhido o gráfico Coordenadas Paralelas, por conta de sua característica de visualização prática de dados multidimensionais. Também foram adicionadas algumas características extras como diferenciação por cores (mapa de calor para dados numéricos e cor única para dados categóricos); filtro de dimensão, para seja visualizado apenas o que for necessário; escolha de dimensão como referencial, isto é, a partir da dimensão escolhida, verificar como os dados se comportam nas outras dimensões. Isso pode ser ativado tanto clicando sobre o nome da dimensão, quanto através do *ComboBox* acima do *preview*; também é possível recarregá-lo e desativá-lo, para travamentos quando for trabalhar com *Big Data*, por exemplo.

5.3.2 Módulo de Visualização Externo e Integralizado

O módulo chamado VisTechLib é um conjunto de técnicas de visualização reutilizáveis. Ela pode chamada por dentro do Blocks e já pode consumir os dados do modelo atual. Dentre as visualizações disponíveis pode-se citar as Coordenadas Paralelas, *Scatter Plot*, *Treemap*, *Sunburst*, *Bar Chart* entre outros. Como diferencial, algumas funcionalidades são adicionadas como detalhe sob demanda, *zoom*, marcação de dados (*Highlight*), múltiplas visualizações simultâneas, entre outras.

5.4 Estrutura da Interface do Blocks

O protótipo possui uma interface gráfica para *Desktop* e segue um modelo conhecido como SPA (*Single Page Application*). Isso significa que há uma tela principal (ver figura 20), e outras informações mais raras de serem consumidas aparecem através de *tabs*, *modals*, *alerts* e correlacionados.



Figura 20. Conhecendo os elementos da tela principal do Blocks Data Generator, na sua versão para Windows. Fonte: O Autor.

Na figura 20, pode-se encontrar os principais elementos para utilizar o Blocks Data Generator. Na marcação 1 (M1), são definidas as propriedades da dimensão, é possível visualizar informações e alterar o título. A M2 mostra o nome atual do modelo, ao seu lado há o símbolo de que este modelo está servindo dados (*Web Service*) (M3); M4 mostra que há alterações não salvas no modelo, a M5 é um botão para fechar o modelo e a M6 é para criar um novo modelo.

A marcação 7 agrupa as dimensões do modelo. A M8 é um exemplo de gerador, no formato de chip; a M9 é o botão responsável por criar e aumentar o encadeamento de geradores; e a M10 excluir um gerador da cadeia - é válido ressaltar que são se pode ter menos que um gerador. A M11 cuida do filtro de dimensões - quando este ícone estiver com uma reta na diagonal sobre o filtro, significa que a dimensão não será incluída na geração e nem visualização dos dados. A M12 é um dos botões responsáveis pela exclusão da dimensão do modelo, o outro botão é encontrado na M14, junto com os botões para reorganização de dimensões - significa poder mover uma dimensão para cima ou para baixo.

A M13 agrupa as configurações atuais do gerador. Tanto as configurações da M13 quanto da M1 seguem um padrão chamado *Two-way data binding*, o que significa que não há a necessidade de um botão de salvar os dados, toda alteração é salva automaticamente, prevalecendo a consistência em todo o sistema. Na M13 encontra-se o tipo de gerador, que apresenta uma lista de categorias que, por sua vez, cada uma apresenta uma lista de geradores (ver seção 4.1) e também as propriedades do gerador selecionado. A M15

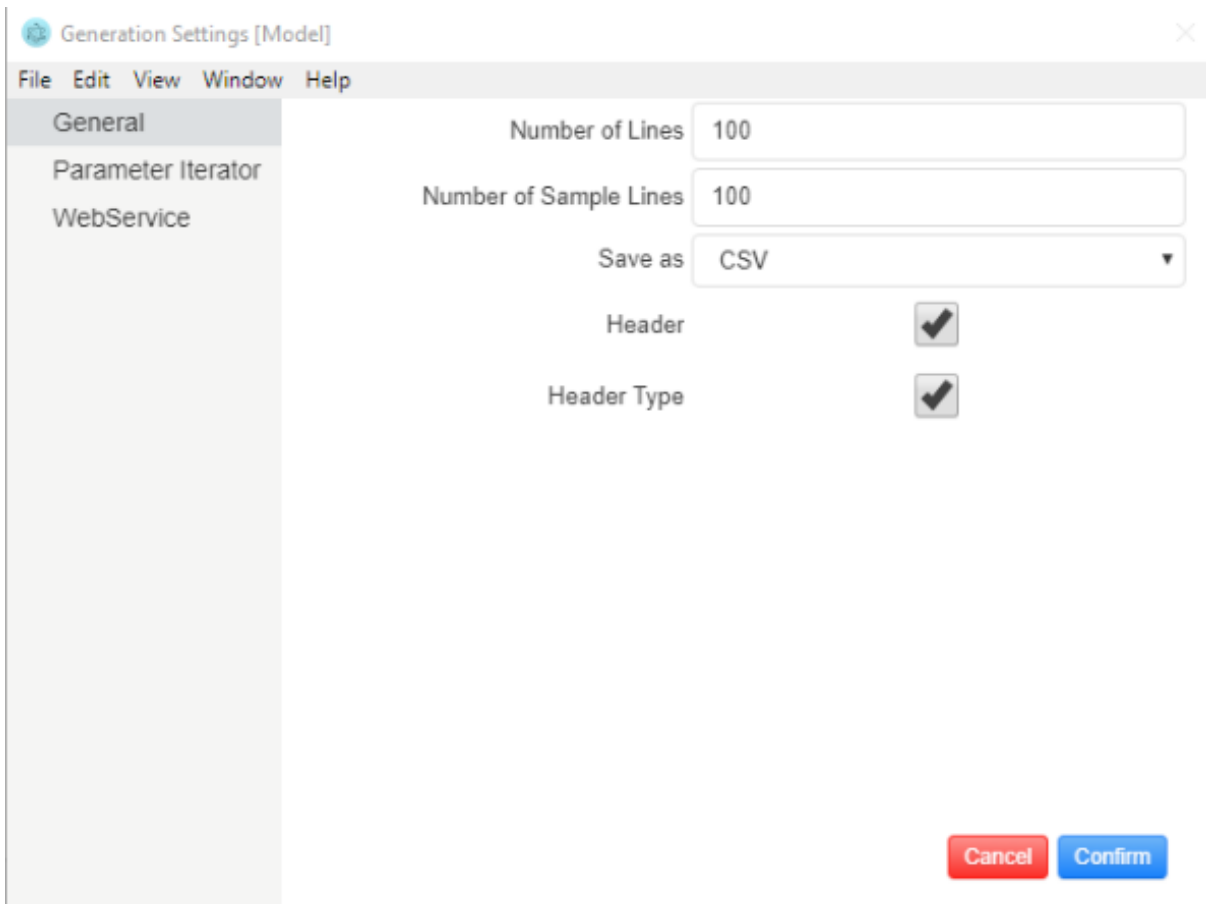


Figura 21. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

possui alguns botões que tornam a modelagem mais prática como um copiador de gerador, e (saber o que é magic painter!).

Na parte inferior da tela, é encontrado o pré-visualizador de dados. É utilizado o Coordenadas Paralelas (M20) (ver seção 4.3.1) como principal e único gráfico. A parte interativa deste gráfico se dá pelo *ComboBox* (M19) ou pelo clique no título (ver seção 4.3.1). Ainda sobre o *preview*, é possível escondê-lo (M16) e recarregá-lo (M17). Na marcação 21 encontra-se o botão para gerar os dados a partir do modelo atual e para manipular as algumas configurações o modelo atual.

Ao clicar com o botão direito do *mouse* no título do modelo (M2) aparece um menu, como visto na figura fig:contextMenu. Esse *context menu* permite renomear ou deletar o modelo, exportá-lo como arquivo .DOT e também manipular algumas informações para o Web Service Como copiar para a área de transferência o ID do modelo, uma URI padrão (*localhost*); ativar ou desativar o modelo pra *Web Service*, bem como abrir a URI em um software padrão do usuário.

Ao lado do botão (*Generate*) para iniciar a geração é encontrado outro botão com o ícone de engrenagem (M21). Na imagem 21, no lado esquerdo, encontra-se 3 seções. A

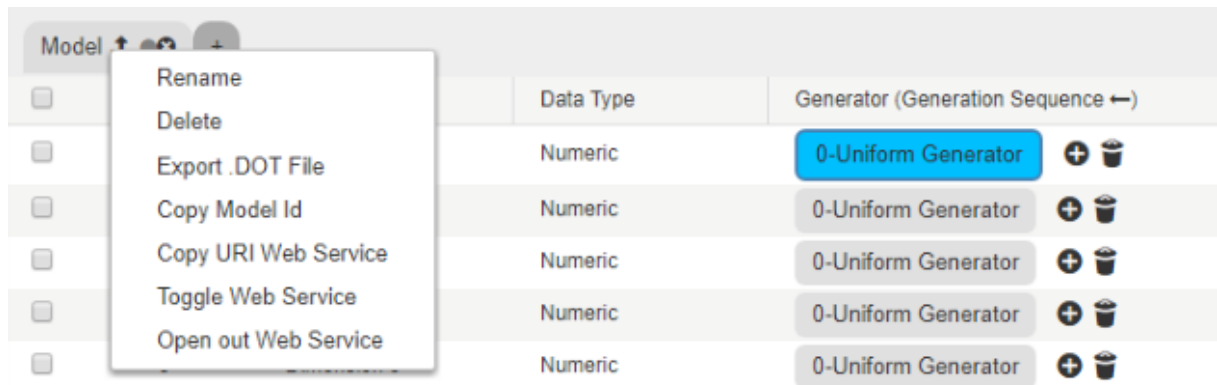


Figura 22. Conhecendo os elementos da *context menu* na aba do modelo. Fonte: O Autor.

primeira é dedicada para configurações gerais do modelo, como quantidade de dados para ser gerados, mostrado no *preview* ou formato dos dados. A segunda seção (*Parameter Iterator*) foi pensada para quem precisar criar mais de um arquivo variando apenas alguns parâmetros, de forma iterativa. A terceira seção é especial para o *Web Service* no qual, liga ou desliga o servidor ou troca a porta padrão para servir os dados.

5.4.1 Mensagens para o usuário

O sistema precisa avisar o usuário de falhas, perguntar sobre preferências e afins. Para isso, o Blocks utiliza-se de *Dialogs* (ver figura 23) para receber um caminho para salvar ou abrir um arquivo. Há um espaço dedicado na tela principal para mensagens advindas de um processo de geração de dados. Mensagem para preparação, progresso, finalização ou falha na geração de dados pode ser acompanhado pelo *Footer Display*. (ver figura 24). Para outros avisos mais genéricos como erros ou tarefas bem sucedidas, bem como avisos mais detalhados há o *Modal* (ver figura 25).

5.4.2 Atalhos do Teclado

Modelar um conjunto de dados pode ser exaustivo, portanto, alguns atalhos podem facilitar na prevenção e correção de erros, eficiência e economia de esforço e afins. Visando dar praticidade ao usuário, o Block Data Generator possui atalhos para criar novo modelo (Ctrl/Cmd + M); deletar modelo atual (Ctrl/Cmd + W); criar nova dimensão (Ctrl/Cmd + D); salvar modelo (Ctrl/Cmd + S).

E para termos de segurança no uso, há o desfazer/refazer com Ctrl/Cmd + Z e Ctrl/Cmd + Shift + Z respectivamente. Além de acesso pelo teclado, os atalhos podem ser acessados pela barra superior da tela inicial (M22). É válido ressaltar, ainda na questão de segurança de uso, que o Blocks Data Generator salva as mudanças no automaticamente em arquivo separado, o qual pode ser recuperado se não forem salvas/discardadas

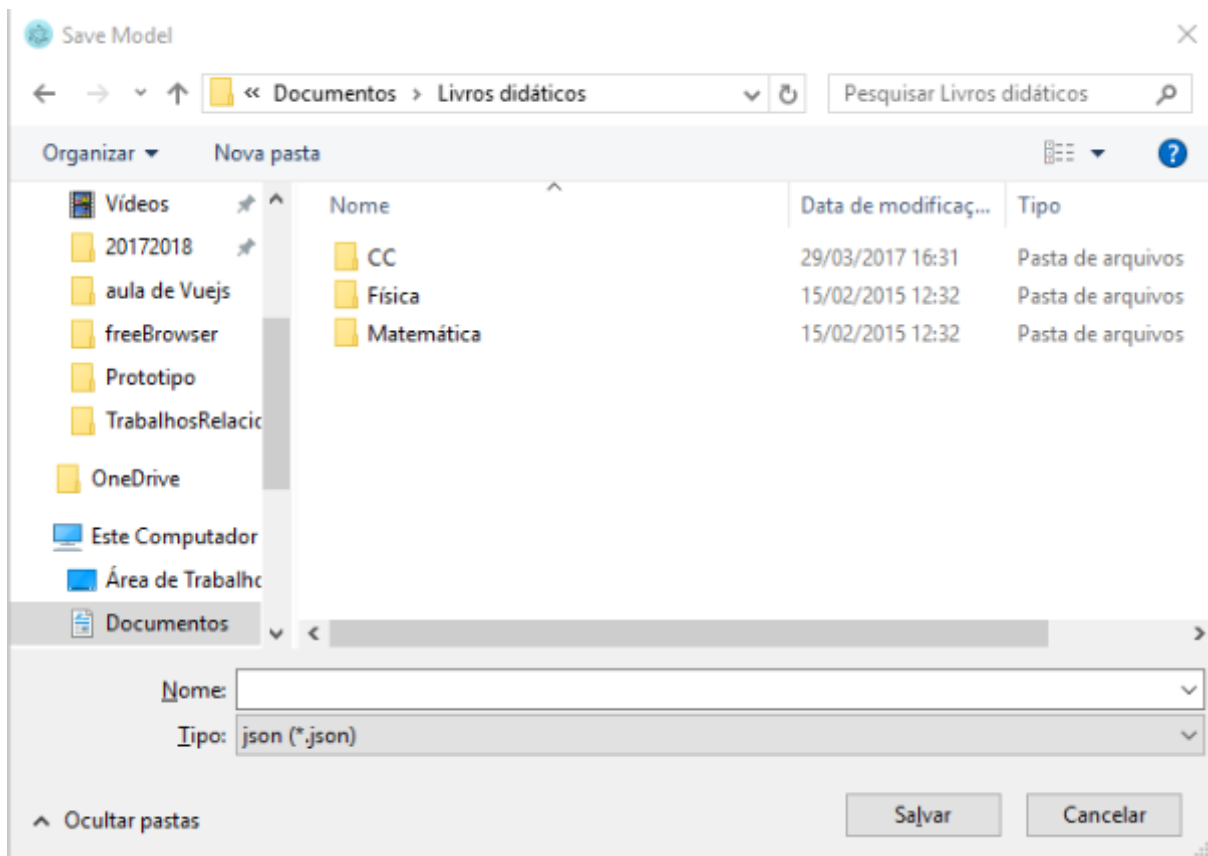


Figura 23. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

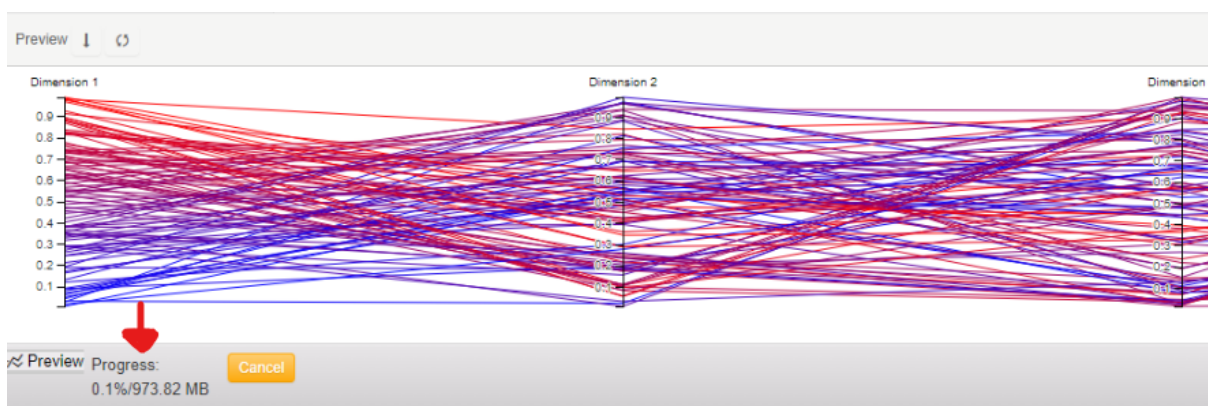


Figura 24. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

adequadamente.

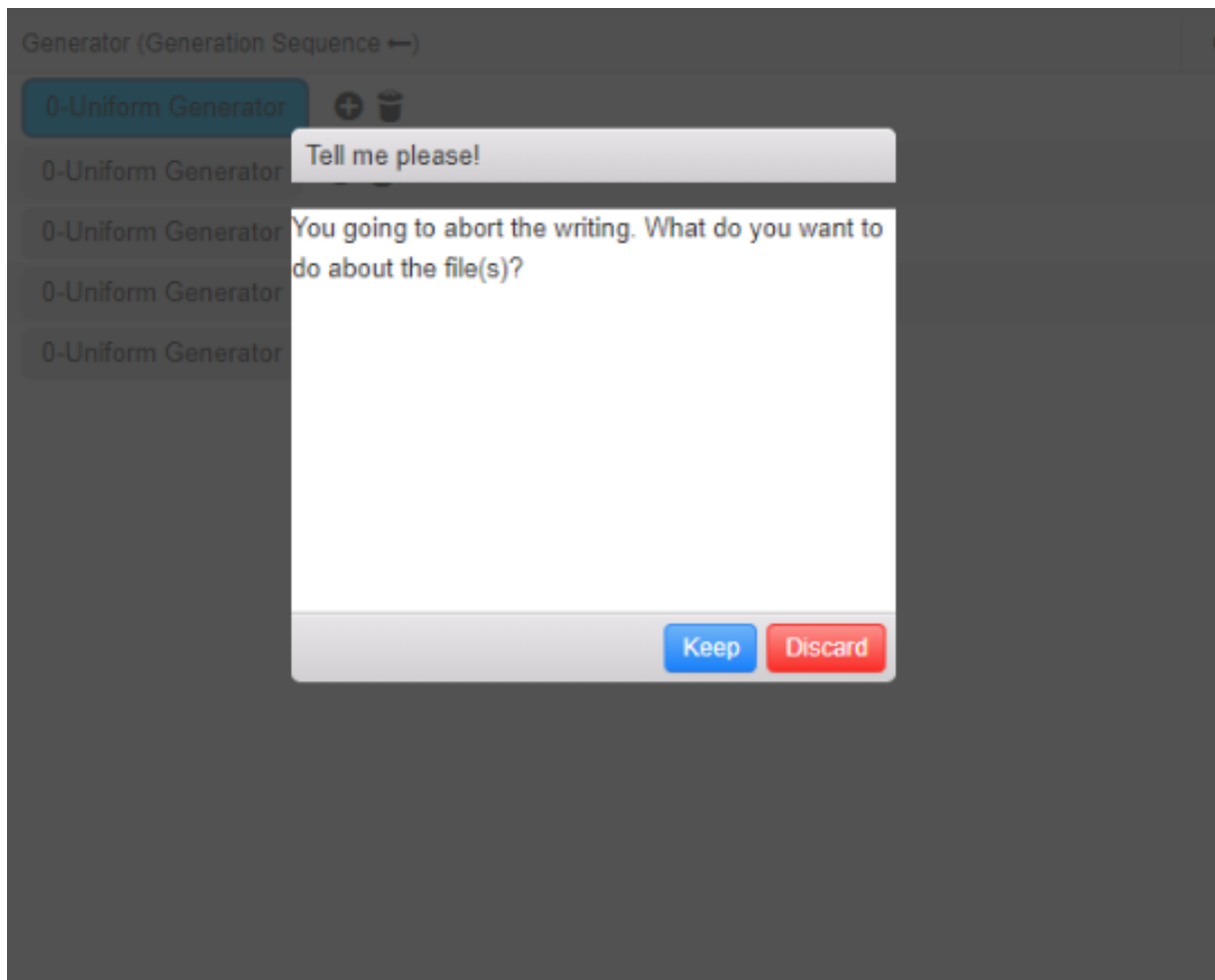


Figura 25. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

Tabela 4. Resumo dos geradores do Blocks Data Generator

Nome	Resumo
Constant	Sequência de números com sempre o mesmo número.
Counter	Sequência de números que é incrementada ou decrementada.
Fixed Time	Sequência de tempo que é incrementada ou decrementada.
Sinusoidal Sequence	Sequência de números de acordo com uma função senoidal.
Custom Sequence	Sequência de números cujo comportamento é dado pelo usuário.
Poisson Time	Gera valores aleatórios temporais em uma distribuição de de Poisson.
Uniform	Gera valores aleatórios númericos em uma distribuição uniforme.
Gaussian	Gera valores aleatórios númericos em uma distribuição de Gaussiana.
Poisson	Gera valores aleatórios númericos em uma distribuição de de Poisson.
Bernoulli	Gera valores aleatórios númericos em uma distribuição de de Bernoulli.
Cauchy	Gera valores aleatórios númericos em uma distribuição de de Cauchy.
Weighted Categorical	Gera valores aleatórios categóricos dadas as categorias com probabilidade ponderada.
Categorical	Gera valores aleatórios categóricos dadas as categorias.
Categorical Quantity	Gera valores aleatórios categóricos dadas as categorias definindo a quantidade de aparição de categorias.
Linear	Gera valores a partir de uma dimensão númerica em uma função linear.
Quadratic	Gera valores a partir de uma dimensão númerica em uma função quadrática.
Polynomial	Gera valores a partir de uma dimensão númerica em uma função polinomial.
Exponential	Gera valores a partir de uma dimensão númerica em uma função exponencial.
Logarithm	Gera valores a partir de uma dimensão númerica em uma função logarítmica.
Sinusoidal	Gera valores a partir de uma dimensão númerica em uma função senoidal.
Categorical	Gera valores a partir de uma dimensão categórica .
Piecewise	Gera valores a partir de uma dimensão númerica com limiar recebe dados de 2 outros geradores.
TimeLaps	Gera valores a partir de uma dimensão temporal que ao atingir o tempo definido ele dados de um novo gerador.
MCAR	Gera valores faltantes aleatoriamente com auxílio de um ou mais geradores dada uma probabilidade.
MAR	Gera valores faltantes com auxílio a partir de uma dimensão e com parâmetros definidos pelo usuário faltantes.
MNAR	Gera valores faltantes com auxílio de um ou mais geradores com parâmetros definidos pelo usuário para ter dados faltantes.
Noise	Gera valores ruidosos com auxílio de um mais geradores dada uma probabilidade e intensidade.
Constant Noise	Gera dados ruidosos com valor específico constante com auxílio de um mais geradores dada uma probabilidade.
Range Filter	Dado um ou mais geradores, os dados que estão no intervalo não são gerados.
Linear Scale	Dado um ou mais geradores, os dados são escalados de acordo com os parâmetros definidos.
No Repeat	Dado um ou mais geradores, são retirados os dados repetidos.
MinMax	Permite escolher qual será o valor máximo e o mínimo gerados.
Low-Pass Filter	Dado um ou mais geradores, este simula o resultado de um filtro passa-baixa.
Get Extra Value	Recebe o valor de dados multidimensionais.
CubicBezier	Gera dados para desenhar uma curva Bezier cúbica.
Path2D Stroke	Gera dados para desenhar a borda de um polígono.
Path2D Fill	Gera dados para preencher um polígono.

6 Resultados

Este capítulo é dedicado a apresentar como podem ser modelados dados discrepantes e faltantes no Blocks Data Generator. Também, serão apresentados os dados gerados e também visualização desses dados gerados na diversidade de métodos permitidos pelo protótipo.

Foram criados 3 bases de dados de contextos genéricos, simulando abstratamente contextos reais. Para criação dessas bases foram utilizados os geradores *Noise Generator* e o *Constant Noise Generator* para gerar dados discrepantes. E para criação de dados faltantes foram utilizados os geradores *MCAR*, *MAR* e *MNAR*.

7 Conclusão

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

Referências

AGGARWAL, C. C. An introduction to outlier analysis. In: *Outlier Analysis*. Springer New York, 2012. p. 1–40. Disponível em: <https://doi.org/10.1007/978-1-4614-6396-2_1>. Citado 2 vezes nas páginas 32 e 33.

ALBUQUERQUE, G.; LOWE, T.; MAGNOR, M. *Synthetic Generation of High-Dimensional Datasets*. Institute of Electrical and Electronics Engineers (IEEE), 2011. 2317–2324 p. Disponível em: <<https://doi.org/10.1109/tvcg.2011.237>>. Citado na página 35.

ARAUJO, L. C. *A classe abntex2: Modelo canônico de trabalhos acadêmicos brasileiros compatível com as normas ABNT NBR 14724:2011, ABNT NBR 6024:2012 e outras*. [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Citado na página 27.

ARAUJO, L. C. *Como customizar o abnTeX2*. 2015. Wiki do abnTeX2. Disponível em: <<https://github.com/abntex/abntex2/wiki/ComoCustomizar>>. Acesso em: 27 abr 2015. Citado na página 27.

ARAUJO, L. C. *O pacote abntex2cite: Estilos bibliográficos compatíveis com a ABNT NBR 6023*. [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Citado na página 27.

ARAUJO, L. C. *O pacote abntex2cite: tópicos específicos da ABNT NBR 10520:2002 e o estilo bibliográfico alfabético (sistema autor-data)*. [S.l.], 2015. Disponível em: <<http://www.abntex.net.br/>>. Citado na página 27.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6028: Resumo - apresentação*. Rio de Janeiro, 2003. 2 p. Citado na página 11.

BARSE, E. L.; KVARNSTROM, H.; JONSSON, E. Synthesizing test data for fraud detection systems. In: IEEE. *19th Annual Computer Security Applications Conference, 2003. Proceedings*. [S.l.], 2003. p. 384–394. Citado na página 29.

BERGEAT, M. et al. A french anonymization experiment with health data. In: . [S.l.: s.n.], 2014. Citado na página 29.

BRAY, T. *The JavaScript Object Notation (JSON) Data Interchange Format*. 2017. Internet Engineering Task Force (IETF). Disponível em: <<https://tools.ietf.org/html/rfc8259>>. Acesso em: 31 jul 2019. Citado na página 30.

CROCKFORD, D. *ECMA-404 The JSON Data Interchange Standard*. 2003. Json.org. Disponível em: <<https://json.org/json-pt.html>>. Acesso em: 31 jul 2019. Citado na página 30.

DEAN, S.; ILLOWSKY, B. Descriptive statistics: Histogram. *Retrieved from the Connexions Web site: <http://cnx.org/content/m16298/1.11>*, 2009. Citado na página 38.

DEVART. *Data Generator for SQL Server*. 2018. <https://www.devart.com>. Disponível em: <<https://docs.devart.com/data-generator-for-sql-server/>>. Acesso em: 21 ago 2019. Citado na página 43.

EDUCATION, M.-H. *The McGraw-Hill Dictionary of Scientific and Technical Terms, Seventh Edition (McGraw-Hill Dictionary of Scientific & Technical Terms)*. McGraw-Hill Professional, 2016. ISBN 0071608990. Disponível em: <<https://www.amazon.com/McGraw-Hill-Dictionary-Scientific-Technical-Seventh/dp/0071608990?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0071608990>>. Citado na página 29.

FREED J. KLENSIN, J. P. N. *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*. 1996. Internet Engineering Task Force (IETF). Disponível em: <<https://tools.ietf.org/html/rfc2048>>. Acesso em: 31 jul 2019. Citado na página 30.

GARCIA, D.; MILLAN, M. A prototype of synthetic data generator. In: *2011 6th Colombian Computing Congress (CCC)*. IEEE, 2011. Disponível em: <<https://doi.org/10.1109/colomcc.2011.5936311>>. Citado na página 36.

GROUP, W. W. *Web Services Architecture*. 2004. Wwww.w3.org. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso em: 02 ago 2019. Citado na página 30.

HANDBOOK of Missing Data Methodology. Chapman and Hall/CRC, 2014. ISBN 1439854610. Disponível em: <<https://www.xarg.org/ref/a/1439854610/>>. Citado 2 vezes nas páginas 31 e 32.

HAUSENBLAS E. WILDE, J. T. M. *ECMA-404 The JSON Data Interchange Standard*. 2014. Internet Engineering Task Force (IETF). Disponível em: <<https://tools.ietf.org/html/rfc7111#page-3>>. Acesso em: 31 jul 2019. Citado na página 30.

INC., L. S. *UML Use Case Diagram Tutorial*. 2019. <https://www.lucidchart.com/>. Disponível em: <<https://www.lucidchart.com/pages/uml-use-case-diagram>>. Acesso em: 09 sep 2019. Citado na página 48.

KOFINAS, D. T.; SPYROPOULOU, A.; LASPIDOU, C. S. A methodology for synthetic household water consumption data generation. *Environmental Modelling & Software*, Elsevier BV, v. 100, p. 48–66, fev. 2018. Disponível em: <<https://doi.org/10.1016/j.envsoft.2017.11.021>>. Citado na página 36.

KORPELA, J. *Tab Separated Values (TSV): a format for tabular data exchange*. 2000. <http://jkorpela.fi>. Disponível em: <<http://jkorpela.fi/TSV.html>>. Acesso em: 31 jul 2019. Citado na página 30.

KUMAR, V. *15 Best Test Data Generation Tools In 2019*. 2019. <https://www.rankred.com>. Disponível em: <<https://www.rankred.com/test-data-generation-tools/>>. Acesso em: 17 ago 2019. Citado na página 29.

LARSEN, M. D.; HUCKETT, J. C. Multimethod synthetic data generation for confidentiality and measurement of disclosure risk. *International Journal of Information Privacy, Security and Integrity*, Inderscience Publishers, v. 1, n. 2/3, p. 184, 2012. Disponível em: <<https://doi.org/10.1504/ijipsi.2012.046132>>. Citado na página 39.

LITTLE, T. D. et al. Missing data. *Developmental psychopathology*, Wiley Online Library, p. 1–37, 2016. Citado 2 vezes nas páginas 31 e 32.

LIU, R. et al. Synthetic data generator for classification rules learning. In: *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, 2016. Disponível em: <<https://doi.org/10.1109/ccbd.2016.076>>. Citado na página 36.

LOPEZ-ROJAS, E. A.; AXELSSON, S. Money laundering detection using synthetic data. In: LINKÖPING UNIVERSITY ELECTRONIC PRESS. *The 27th annual workshop of the Swedish Artificial Intelligence Society (SAIS); 14-15 May 2012; Örebro; Sweden*. [S.l.], 2012. p. 33–40. Citado na página 29.

LTD, R. G. S. *SQL Data Generator*. 2019. <https://www.red-gate.com/>. Disponível em: <<https://www.red-gate.com/products/sql-development/sql-data-generator/>>. Acesso em: 18 ago 2019. Citado na página 41.

MCKNIGHT, P. E. *Missing Data: A Gentle Introduction (Methodology in the Social Sciences)*. The Guilford Press, 2007. ISBN 9781593853938. Disponível em: <<https://www.xarg.org/ref/a/1593853939/>>. Citado na página 31.

MICROSOFT. *Generating Test Data for Databases by Using Data Generators*. 2019. <https://www.microsoft.com/>. Disponível em: <[https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/dd193262\(v=vs.100\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/dd193262(v=vs.100))>. Acesso em: 19 ago 2019. Citado na página 43.

MOCKAROO. *Mockaroo APIs*. 2019. <https://www.mockaroo.com/>. Disponível em: <<https://www.mockaroo.com/api/docs>>. Acesso em: 23 ago 2019. Citado na página 45.

MOCKAROO. *Mockaroo, realistic data generator*. 2019. <https://www.mockaroo.com/>. Disponível em: <<https://www.mockaroo.com/>>. Acesso em: 23 ago 2019. Citado na página 44.

RATHI, A. *Dealing with Noisy Data in Data Science*. Analytics Vidhya, 2019. Disponível em: <<https://medium.com/analytics-vidhya/dealing-with-noisy-data-in-data-science-e177a4e32621>>. Citado 2 vezes nas páginas 33 e 34.

RAYMOND, E. S. *Data File Metaformats. Chapter 5. Textuality*. 2003. <http://www.catb.org>. Disponível em: <<http://www.catb.org/~esr/writings/taoup/html/ch05s02.html>>. Acesso em: 31 jul 2019. Citado na página 30.

RODRÍGUEZ-HERNÁNDEZ, M. del C. et al. Datagencars: A generator of synthetic data for the evaluation of context-aware recommendation systems. *Pervasive and Mobile Computing*, Elsevier BV, v. 38, p. 516–541, jul. 2017. Disponível em: <<https://doi.org/10.1016/j.pmcj.2016.09.020>>. Citado na página 39.

RUBIN, D. B. Statistical disclosure limitation. *Journal of official Statistics*, v. 9, n. 2, p. 461–468, 1993. Citado na página 29.

SAKSHAUG, J. W.; RAGHUNATHAN, T. E. Nonparametric generation of synthetic data for small geographic areas. In: *Privacy in Statistical Databases*. Springer International Publishing, 2014. p. 213–231. Disponível em: <https://doi.org/10.1007/978-3-319-11257-2_17>. Citado na página 38.

SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. 2005. Internet Engineering Task Force (IETF). Disponível em: <<https://tools.ietf.org/html/rfc4180#page-2>>. Acesso em: 31 jul 2019. Citado na página 30.

SOFT., D. *DTM Database Tools*. 2019. [Http://www.sqledit.com/](http://www.sqledit.com/). Disponível em: <<http://www.sqledit.com/dg/index.html>>. Acesso em: 17 ago 2019. Citado na página 41.

STACKIFY. *SOAP vs. REST: The Differences and Benefits Between the Two Widely-Used Web Service Communication Protocols*. 2017. Stackify.com. Disponível em: <<https://stackify.com/soap-vs-rest/>>. Acesso em: 02 ago 2019. Citado na página 30.

TANENBAUM, A. S.; FILHO, N. M. *Sistemas operacionais modernos*. [S.l.]: Prentice-Hall, 1995. v. 3. Citado na página 30.

WANG, B.; RUCHIKACHORN, P.; MUELLER, K. SketchPadN-d: WYDIWYG sculpting and editing in high-dimensional space. *IEEE Transactions on Visualization and Computer Graphics*, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 12, p. 2060–2069, dez. 2013. Disponível em: <<https://doi.org/10.1109/tvcg.2013.190>>. Citado na página 35.

WHITING, M. A.; HAACK, J.; VARLEY, C. Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software. In: *Proceedings of the 2008 conference on BEyond time and errors novel evaluation methods for Information Visualization - BELIV 08*. ACM Press, 2008. Disponível em: <<https://doi.org/10.1145/1377966.1377977>>. Citado na página 38.

WILSON, P.; MADSEN, L. *The Memoir Class for Configurable Typesetting - User Guide*. Normandy Park, WA, 2010. Disponível em: <<http://mirrors.ctan.org/macros/latex/contrib/memoir/memman.pdf>>. Acesso em: 19 dez. 2012. Citado na página 27.