



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 Exatas e Naturais como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Pará

Orientador: Prof. Dr. Carlos Gustavo Resque dos Santos

Belém
2019

Insira a ficha catalográfica aqui.

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 Exatas e Naturais como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Conceito: _____

Belém, 1 de janeiro de 2019.

BANCA EXAMINADORA

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

Prof. Dr. Bianchi Serique Meiguins - Membro Interno
UFPA

MSc. Diego Hortêncio dos Santos - Membro Externo
UFPA

Agradecimentos

Agradeço a Deus por permitir a oportunidade de desenvolver meu trabalho. Também agradeço a minha família por me dar suporte em todos os aspectos da minha vida. Também agradeço ao meu orientador e todos meus colegas do laboratório que me ajudaram incansavelmente a sanar minhas dúvidas. Aos meus amigos da faculdade estiveram comigo nos melhores e piores momentos, trabalhos e descontrações. Agradeço aos meus amigos fora da faculdade também que, por mais que estivessem distantes, também significam muito para mim.

Resumo

Neste trabalho é apresentado e analisado o sistema de geração e visualização de dados sintéticos chamado de Blocks para geração de dados sintéticos faltantes e discrepantes. Também foi acrescentado ao Sistema Blocks novos geradores de dados faltantes e discrepantes sintéticos. Esse sistema é composto de geradores chamados de blocos, os quais podem ser encadeados - permitindo que seja gerado um comportamento mais complexo. Os blocos possuem categorias que produzem sequência - quer seja de números ou de tempo, que possuem distribuição de probabilidade aleatória. Categoria funcional, a qual utiliza outra dimensão como parâmetro; de acessórios - adicionam a formatação ou valor dos dados e geométrica para geração de dados a partir de formas geométricas. O Blocks também permite que os dados sejam visualizados, para isso ele possui um gráfico do tipo Coordenadas Paralelas na tela inicial e integração com uma ferramenta. Por conseguinte, os objetivos do trabalho são modelar, visualizar e avaliar a geração de dados sintéticos faltantes e discrepantes. A validação dos resultados foi feita com o uso de visualizações e a partir delas, foi analisado o comportamento das dimensões. Foram utilizados vários tipos de dados como numéricos, categóricos, temporais etc. Em geral, o Blocks permitiu a geração desses dados de forma satisfatória.

Palavras-chave: Dados Sintéticos. Dados Faltantes. Dados Discrepantes. Visualização de Dados.

Abstract

This work presents and analyzes the synthetic data generation and visualization system called Blocks for the generation of missing and discrepant synthetic data. Also added to the Blocks System a new synthetic missing and discrepant data generators. This system is made up of generators called blocks, which can be chained together - allowing more complex behavior to be generated. Blocks have categories that produce sequence - whether of numbers or time, another which have random probability distribution. Functional category, which uses another dimension as a parameter; accessories - add the formatting or value of the data and geometric for data generation from geometric shapes. Blocks also allows data to be previewed, so it has a Parallel Coordinates graph on the home screen and integration with one visualization tool. Therefore, the objectives of the work are to model, visualize and evaluate the generation of missing and discrepant synthetic data. The validation of the results was made using visualizations and from them, the behavior of the dimensions was analyzed. Various types of data were used such as numeric, categorical, temporal etc. In general, Blocks has allowed this data to be satisfactorily generated.

Keywords: Synthetic Data. Missing Data. Outliers. Data Visualization.

Lista de ilustrações

Figura 1.	Exemplo de escala de discrepância (Adaptado de Aggarwal)	19
Figura 2.	Exemplo de ruído e de anomalia.	20
Figura 3.	Exemplo das classificações de ruído (adaptado de Rathi)	21
Figura 4.	Visão Geral de geração de dados.	22
Figura 5.	Visão Geral de geração de dados do Sketchpad.	23
Figura 6.	Exemplo de árvore de decisão para jogar tennis criado a partir de regras encontradas em um conjunto de dados.	24
Figura 7.	Exemplo da interface do usuário para configuração do gerador de dados.	24
Figura 8.	Fluxo de passos para geração dos dados sintéticos.	25
Figura 9.	Comparação da média dos dados reais e sintéticos na simulação paramétrica e não paramétrica.	26
Figura 10.	Fluxo de passos para geração dos dados sintéticos.	27
Figura 11.	Fluxo de passos para geração dos dados sintéticos.	28
Figura 12.	Usando o DTM Data Generator. Fonte: DTM Data Generator	30
Figura 13.	Usando o Redgate SQL Data Generator. Fonte: Red Gate SQL Data Generator.	31
Figura 14.	Usando o Microsoft Visual Studio. Fonte: anranik.	32
Figura 15.	Usando o dbForge Test Data Generator. Fonte: anranik.	33
Figura 16.	Usando o Mockaroo. Fonte: anranik.	34
Figura 17.	Diagrama de Caso de uso do Blocks. Fonte: o autor.	35
Figura 18.	Diagrama de Classes dos geradores do Blocks. Fonte: o autor.	37
Figura 19.	Diagrama de Classes (2 ^a Geração) dos geradores do Blocks. Fonte: o autor.	37
Figura 20.	Diagrama de Classes (3 ^a Geração) dos geradores do Blocks. Fonte: o autor.	38
Figura 21.	Diagrama de Sequência para geração de dados em arquivos no Blocks. Fonte: O autor.	38
Figura 22.	Diagrama de Sequência para visualização dos dados no Blocks. Fonte: o autor.	39
Figura 23.	Fluxograma de utilização do Blocks. Fonte: Yvan Brito, 2019.	41
Figura 24.	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.	47
Figura 25.	Conhecendo os elementos da tela principal do Blocks, na sua versão para Windows. Fonte: O Autor.	49

Figura 26.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	50
Figura 27.	Conhecendo os elementos da <i>context menu</i> na aba do modelo. Fonte: O Autor.	51
Figura 28.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	52
Figura 29.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	52
Figura 30.	Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.	53
Figura 31.	Base sintética de avaliação de Carros.	54
Figura 32.	Base sintética de avaliação de Redes Sociais.	55
Figura 33.	Base sintética de avaliação de Convênios Médicos.	55
Figura 34.	Base sintética de avaliação de Estrutura de Conta Bancária.	56
Figura 35.	Base sintética genérica para geração de dados faltantes numéricos.	56
Figura 36.	Visualização Coordenadas Paralelas da base de carros.	57
Figura 37.	Visualização Scatterplot da base de carros.	58
Figura 38.	Visualização Coordenadas Paralelas da base de Redes Sociais.	58
Figura 39.	Visualização Histograma da base de Redes Sociais. A unidade de Curtida é bilhões; Seguidores está em milhões e Postagens está em milhares.	59
Figura 40.	Visualização Dendrograma da base sobre estrutura de conta bancária.	60
Figura 41.	Visualização Gráfico de Colunas da base Convênios Médicos. Eixo X: Planos de Saúde, Eixo Y: Especialidades, Barra: Preço por Consulta	61
Figura 42.	Visualização Gráfico de Colunas da base genérica para visualização de dados faltantes.	62

Lista de tabelas

Tabela 1.	Exemplo de dados ausentes MCAR	18
Tabela 2.	Exemplo de dados ausentes MAR	18
Tabela 3.	Propriedade dos geradores do Blocks	42
Tabela 4.	Resumo dos geradores do Blocks	43

Lista de abreviaturas e siglas

JSON	JavaScript Object Notation
CSV	Comma Separated Values
DSV	Delimiter Separated Values
TSV	Tabular Separated Values
TSG	Threat Streaming Generator
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
HTTP	HiperText Transfer Protocol
MCAR	Missing Completely At Random
MAR	Missing At Random
MNAR	Missing not At Random
SVM	Support Vector Machine
CARS	Context-Aware Recommender Systems
UC	Use Case

Sumário

1	INTRODUÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Dados Sintéticos	15
2.2	Arquivo	15
2.3	Web Service	16
2.4	Dados Ausentes	17
2.4.1	MCAR	17
2.4.2	MAR	18
2.4.3	MNAR	18
2.5	Dados Discrepantes	19
2.5.1	Dados Ruidosos e Anômalos	20
3	TRABALHOS RELACIONADOS	22
3.1	Trabalhos acadêmicos	22
3.2	Aplicações	28
4	ARQUITETURA DO PROJETO	35
4.1	Casos de uso do sistema	35
4.2	Diagrama de Classes	36
4.3	Sequência de atividades no Blocks	36
5	SISTEMA BLOCKS	40
5.1	Tipos de Geradores de Dados	41
5.1.1	Sequencial	42
5.1.2	Aleatório	44
5.1.3	Função	44
5.1.4	Acessórios	45
5.1.5	Geométrico	46
5.1.6	Baseado em dados reais	46
5.2	Modos de Geração de Dados	46
5.2.1	<i>Streaming Data</i>	47
5.2.2	Web Service	47
5.3	Modos para Visualização de Dados	48
5.3.1	Preview	48
5.3.2	Módulo de Visualização Externo e Integralizado	48

5.4	Estrutura da Interface Gráfica do Blocks	48
5.4.1	Mensagens para o usuário	51
5.4.2	Atalhos do Teclado	51
6	VALIDAÇÃO VISUAL	54
6.1	Modelagem dos Dados	54
6.2	Apresentação das Visualizações	56
7	CONSIDERAÇÕES FINAIS	63
	REFERÊNCIAS	64

1 Introdução

Dados sintéticos são dados gerados por uma máquina (MORAES,) a partir de outros dados, geradores, fórmulas matemáticas, funções etc. Esses dados, apesar de não possuirem um comportamento comparável aos da realidade (MORAES,), podem ser aplicados de variadas formas. Quer seja substituindo/simplificando dados reais - dados previsíveis podem ser substituídos por fórmulas ou funções - ou adicionando novos dados a partir de uma instância apenas fazendo algumas alterações.

Ainda, um problema muito complicado aumenta a demanda por dados sintéticos: confidencialidade dos dados. Ninguém quer ter seus dados sendo usados de forma arbitrária por empresas, mas todos querem melhores serviços. Logo, para melhorar a situação dos dois lados, os dados sintéticos podem atingir um grau de realismo - dependendo da ferramenta e da habilidade do operador - sem ter grandes problemas de privacidade.

Também, outro problema é a escassez dos dados. Este cenário é encontrado principalmente em lançamento de novas tecnologias ou tratamento de exceções. Portanto, nada mais esperto do que, a partir da amostra que possui, gerar novas instâncias, determinando novos padrões, logo, novos casos de teste ou de uso.

Os dados podem ser variados, como um número, uma categoria, uma marcação de tempo, uma imagem, um áudio, um vídeo e afins. Portanto, gerar dados sintéticos exigem variados métodos. Entre os mais conhecidos, pode-se citar fórmulas matemáticas, redes neurais, interpolação de frames, embaralhamentos, aleatoriedade etc.

Como dito por Moraes dados sintéticos podem ser muito simples. Sendo assim, uma forma de dar realismo aos dados é permitir que algumas instâncias sejam faltantes. Por exemplo, alguém pode se esquecer ou recusar responder uma informação, pode haver uma falha no disco rígido etc. Logo, dados faltantes são bastante presentes em bases de dados reais.

Dados faltantes podem assumir formas dependendo do seu contexto. Na literatura são encontradas 3 formas: *Missing Completely At Random* (MCAR), *Missing At Random* (MAR) e *Missing Not At Random* (MNAR). Basicamente o MCAR determina que a falta do dado é completamente aleatória. No MAR ainda é aleatório, mas é possível correlacionar em buscar de predizer o valor faltante. O MNAR indica que o dado está faltante e o seu motivo não está dito na base de dados, ou seja, pode haver influência de contexto externo.

Outra característica de dados reais são dados discrepantes. Entenda-se por discrepante aquele dado que sai do padrão, quer seja no valor ou na formatação. Semelhante aos faltantes, dados discrepantes podem existir por exemplo, em uma falha no sensor, um

evento atípico etc.

Dados discrepantes podem assumir 2 formas também: dados ruidosos e dados anômalos. Basicamente um dado ruidoso - ou simplesmente ruído - possui um baixo grau de discrepância. Já o dado anômalo - ou anomalia - possui um alto grau de discrepância, podendo até atrapalhar o entendimento do fenômeno e a visualização dos dados.

Pensando em oferecer acessibilidade aos dados sintéticos foi desenvolvido o software de código aberto e gratuito chamado de Blocks. O Blocks oferece vários geradores de diferentes categorias, bem como encadeá-los para que os dados tenham cada vez mais forma de um dado real. É possível gerar sequências, dados randômicos, correlacionar dimensões e adicionar acessórios como ruídos e dados faltantes.

Por conseguinte, o objetivo é apresentar o Blocks e como gerar dados faltantes e discrepantes nessa aplicação. Também serão apresentados os geradores de dados faltantes desenvolvidos no âmbito deste trabalho. Feito isso, o Blocks permite que os dados sejam visualizados, então é interessante que essa funcionalidade seja analisada. Em suma, este trabalho vai apresentar o funcionamento do Blocks modelando e visualizando dados faltantes e discrepantes.

2 Fundamentação Teórica

Neste capítulo é abordado em mais detalhes sobre a literatura dos dados sintéticos, discrepantes, 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) Como exemplo, Rubin (RUBIN, 1993) a introduziu um conjunto de dados completamente sintético. Com o objetivo de 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 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.

Os arquivos são amplamente utilizados na computação, e por conta da larga escala de uso, foram criadas várias extensões de arquivos. Toda linguagem de programação possui seu arquivo como .java para Java, .js para JavaScript, .py para Python e muitos outros.

Também existem os arquivos cujos dados são acessados para várias plataformas. O XML é muito utilizado na Web e pela linguagem Java, o JSON (*Javascript Object Notation*, ou em português Notação de Objeto Javascript) tomou uma parcela da web com o uso, principalmente, de APIs, e o CSV (*Comma Separated Values*) é largamente utilizado em aplicações de ciência de dados, por exemplo.

E quanto ao JSON, (BRAY, 2017) (CROCKFORD, 2003) este 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 na notação de objeto 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 codificação 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 demorou 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.

2.3 Web Service

Um *Web Service* (GROUP, 2004) é definido como um software criado para suportar interoperabilidade entre máquinas através da rede computadores. Na história do

Web Service surgiram o Protocolo SOAP (Simple Object Access Protocol) que utiliza o WSDL (Web Services Description Language) como uma interface descrita em um formato processável por máquinas (GROUP, 2004) e o modelo de arquitetura REST (Representational State Transfer) proposto por (Por autor) que não utiliza o WSDL. 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) O REST foi criado em conjunto ao HTTP 1.1 e a sua principal característica é a utilização simplificada dos verbos do protocolo HTTP (GET, POST, PUT, HEAD, OPTIONS e DELETE) (citar esse link(<https://www.devmedia.com.br/web-services-rest-versus-soap/32451>))

2.4 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 obtenção de dados em uma situação real como um participante desistindo ou não respondendo parte da pesquisa, o pesquisador perdendo seu dispositivo de anotação, má operação ao salvar em dispositivos eletrônicos etc. (MCKNIGHT, 2007)

O grande impacto dos dados ausentes está quando faltam dados a ponto da pesquisa se tornar tendenciosa, inconclusiva ou inconsistente. (MCKNIGHT, 2007) Um exemplo seria uma pesquisa de avaliação de motos. Características como modelo, ano, preço, número de vendas e média de notas seriam coletadas. Contudo, ao salvar a pesquisa no mesmo disco rígido, a parte referida a um dos integrantes foi corrompida e, portanto, há uma ausência de dados.

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 - *MCAR* (*Missing completely at random*); faltando de forma aleatória - *MAR* (*Missing At Random*); faltando de forma não aleatória - *MNAR* (*Missing Not At random*). (HANDBOOK..., 2014)

2.4.1 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 mecanismo 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.

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

2.4.2 MAR

Quanto ao MAR, este é definido como a probabilidade da resposta está faltando depende dos dados obtidos, mas não está relacionado com dados fora da amostra coletada. Este é o mecanismo mais fácil de utilizar técnicas de imputação de dados, 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 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	

2.4.3 MNAR

Quanto ao MNAR, este é definido como a probabilidade da resposta está relacionada com os dados que não estão presentes na amostra. (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 ausência 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.

2.5 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 ou dados desviantes 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 anomalias não são precisamente definidos, mas algoritmos de detecção de discrepância podem dar pontuação de discrepância para cada dado e utilizar este nível. (AGGARWAL, 2012)

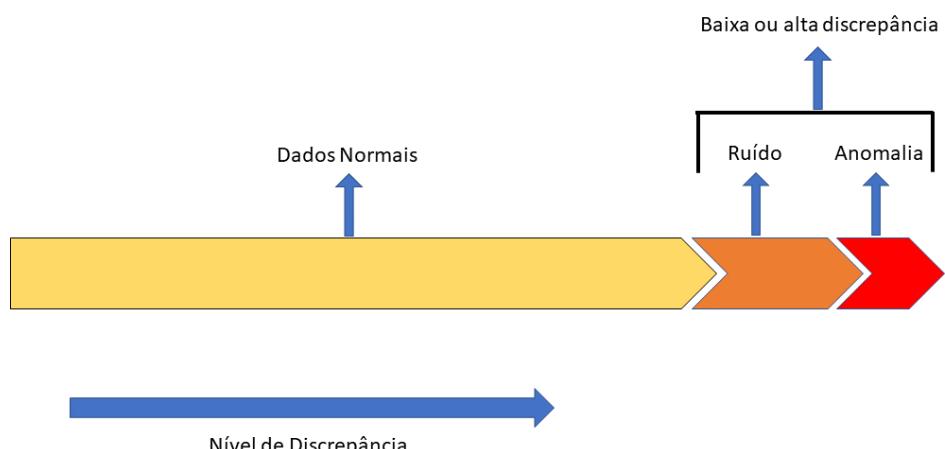


Figura 1. Exemplo de escala de discrepância (Adaptado de Aggarwal)

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 a 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.

2.5.1 Dados Ruidosos e Anômalos

Dados ruidosos são dados indesejaveis, 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.

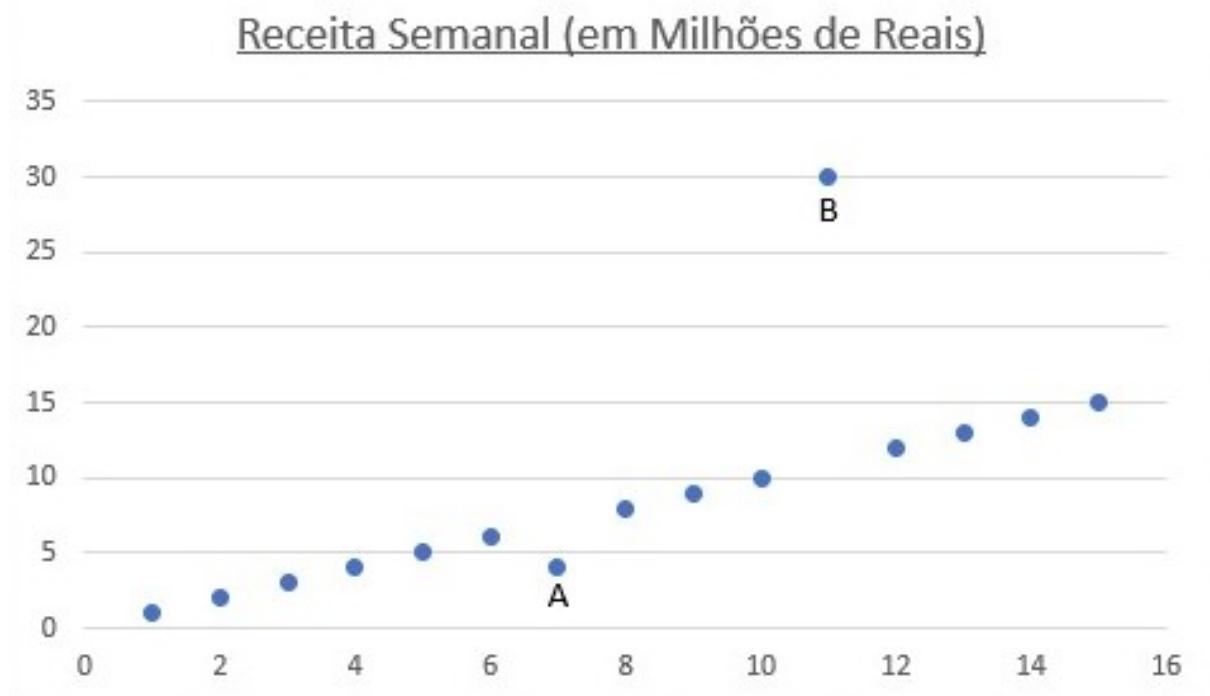


Figura 2. Exemplo de ruído e de anomalia.

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

Índice	Dimensão1	Dimensão2	Dimensão3	Dimensão4	Dimensão..	DimensãoN-1	DimensãoN	Evento		
Registro1										
Registro2										
Registro3									Ruído1	
Registro4									Ruído2	
Registro5									Ruído3	
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

Albuquerque et al. (ALBUQUERQUE; LOWE; MAGNOR, 2011) descreveu um *framework* capaz de gerar dados sintéticos multidimensionais. O sistema como mostra a 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. No texto, nada foi citado sobre dados faltantes ou discrepantes.

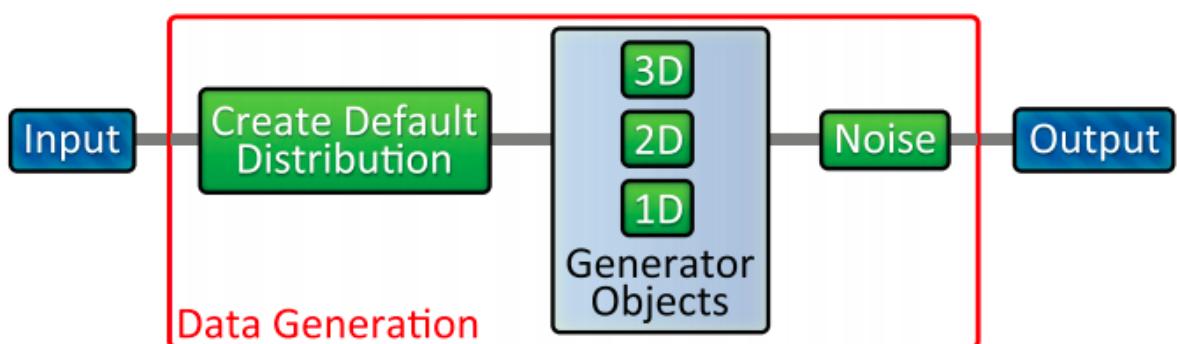


Figura 4. Visão Geral de geração de dados.

Wang et al. (WANG; RUCHIKACHORN; MUELLER, 2013) apresentou uma aplicação chamada SketchPad 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 como a introdução de dados, visualização e modelação utilizando o gráfico Coordenadas Paralelas e o espaço para desenho chamado de *Touchpad^{n-d}* e a criação ou modificação de um conjunto de dados.

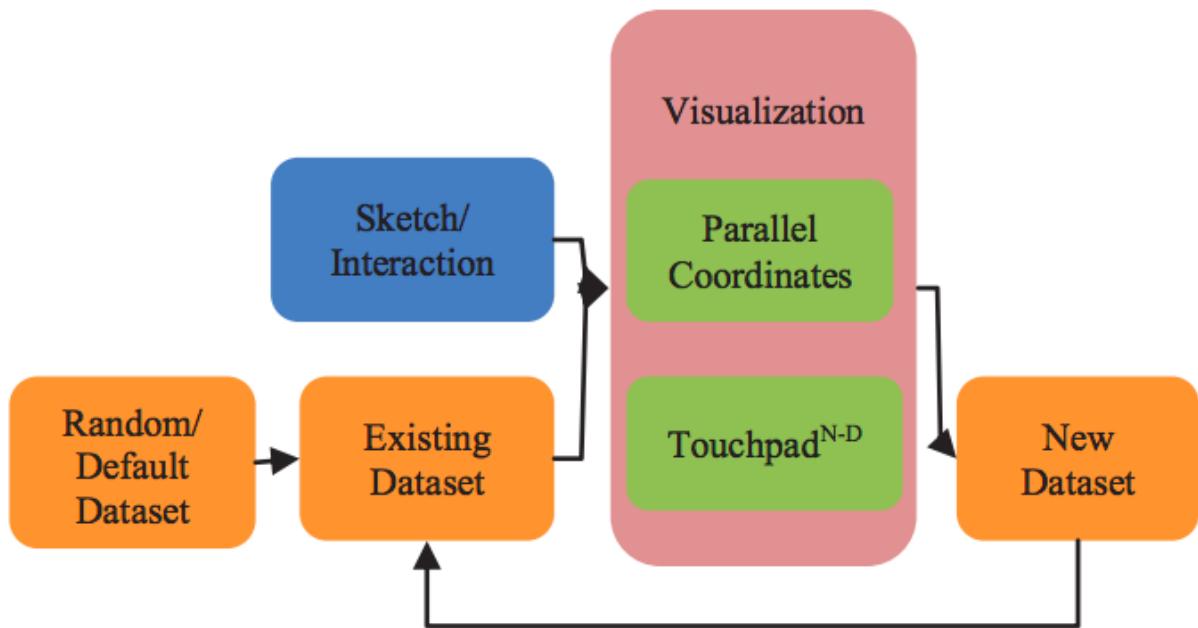


Figura 5. Visão Geral de geração de dados do Sketchpad.

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 construindo 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. As regras são interessantes tanto para reutilizar os dados quanto para facilitar em determinar o comportamento dos dados.

Garcia e Millán (GARCIA; MILLAN, 2011) criaram um sistema de geração de dados sintéticos pensado para desenvolvedores que buscam testar de forma eficiente e exaustiva a sua aplicação. Esses dados podem 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,

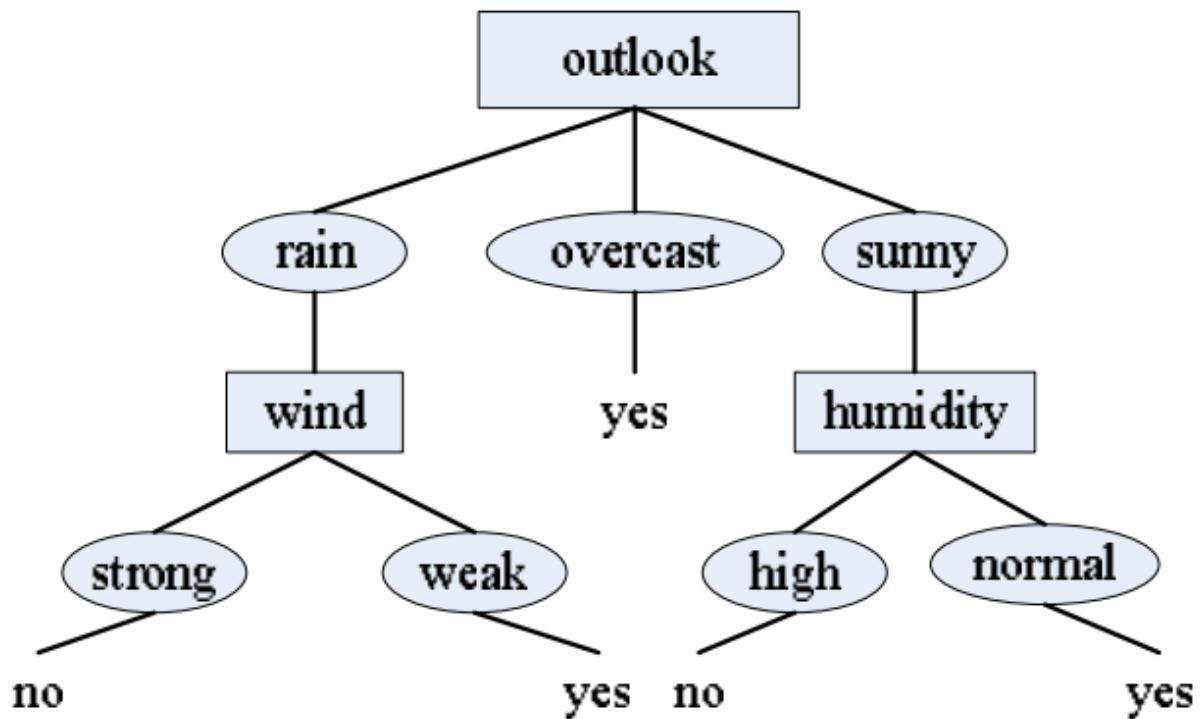


Figura 6. Exemplo de árvore de decisão para jogar tennis criado a partir de regras encontradas em um conjunto de dados.

Intervalo ou Lista de valores. Contudo, os geradores, para alterar o comportamento dos dados, depende apenas dos parâmetros e não é possível misturar os geradores.

The interface consists of two main sections: 'Estructura de la BDO' (left) and 'Detalle' (right).

Estructura de la BDO:

- afiliado
 - id
 - nombre
 - apellido** (highlighted)
 - sexo
 - fecha_nacimiento
 - tipo_afiliacion
 - ingreso_mensual
 - cita_medica
 - medico
 - vacunacion

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: 0
- Fuente de datos: Archivo

Configuración de fuente de datos

- Conexión archivo: Apellidos
- Tipo de acceso: Aleatorio

Botones:

- Botón '?': Ayuda
- Botón 'Establecer': Aplicar cambios

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

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.

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 objetivo de encontrar a que mais se aproxima.

A fase 2 cuida da geração de dados sintéticos utilizando a distribuição criada na fase 1 para gerar os dados para 24h, respeitando as características diferenciadas para dias de semana e finais de semana.

Apesar de o trabalho ter como principal a calibração, modelagem e geração dos dados, não foi disponibilizada uma interface gráfica do usuário para utilização dos dados, nem visualização dos mesmos nem possibilidade de gerar dados faltantes e discrepantes.

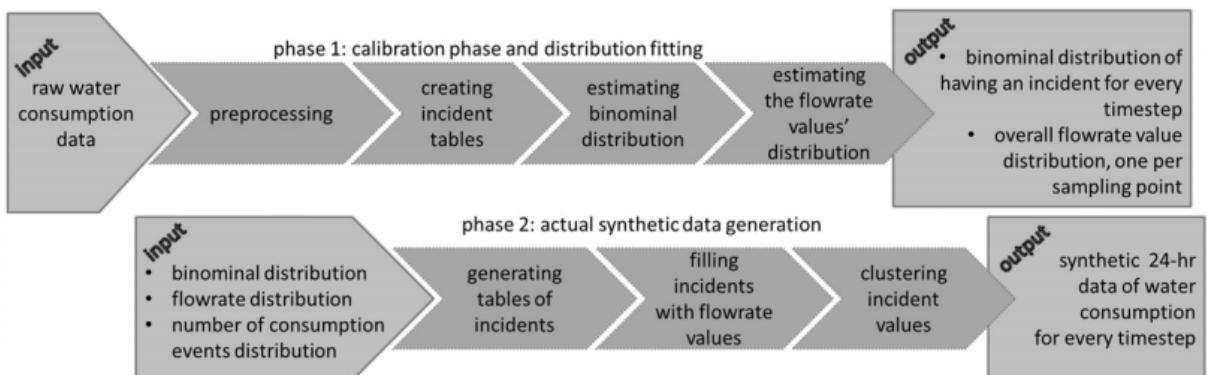


Figura 8. Fluxo de passos para geração dos dados sintéticos.

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 geográfica. 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.

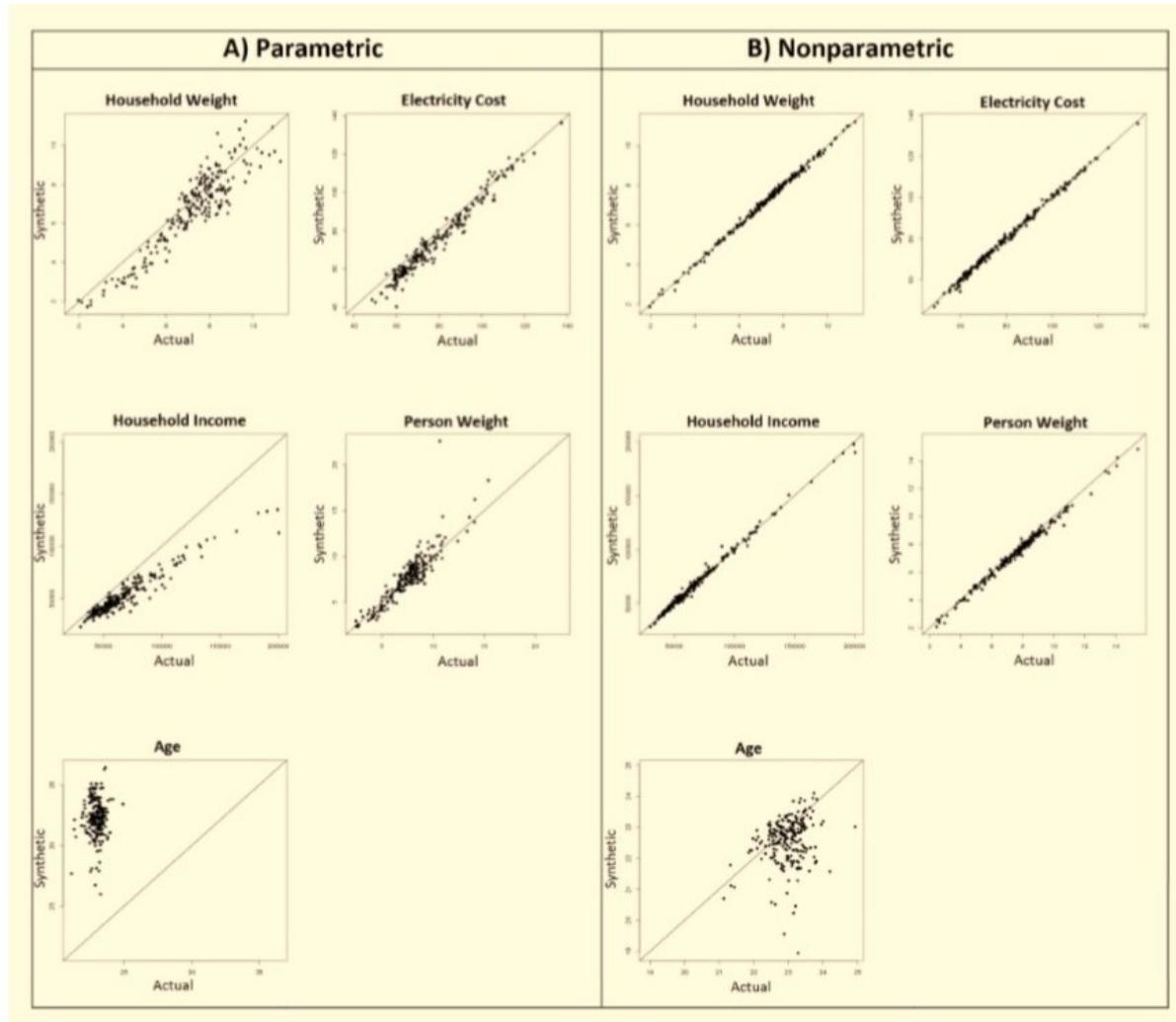


Figura 9. Comparação da média dos dados reais e sintéticos na simulação paramétrica e não paramétrica.

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.

O projeto é bem completo para desenvolvimento de dados sintéticos, inclusive com a validação pelos especialistas, o qual pode elevar o preço para geração dos dados.

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.

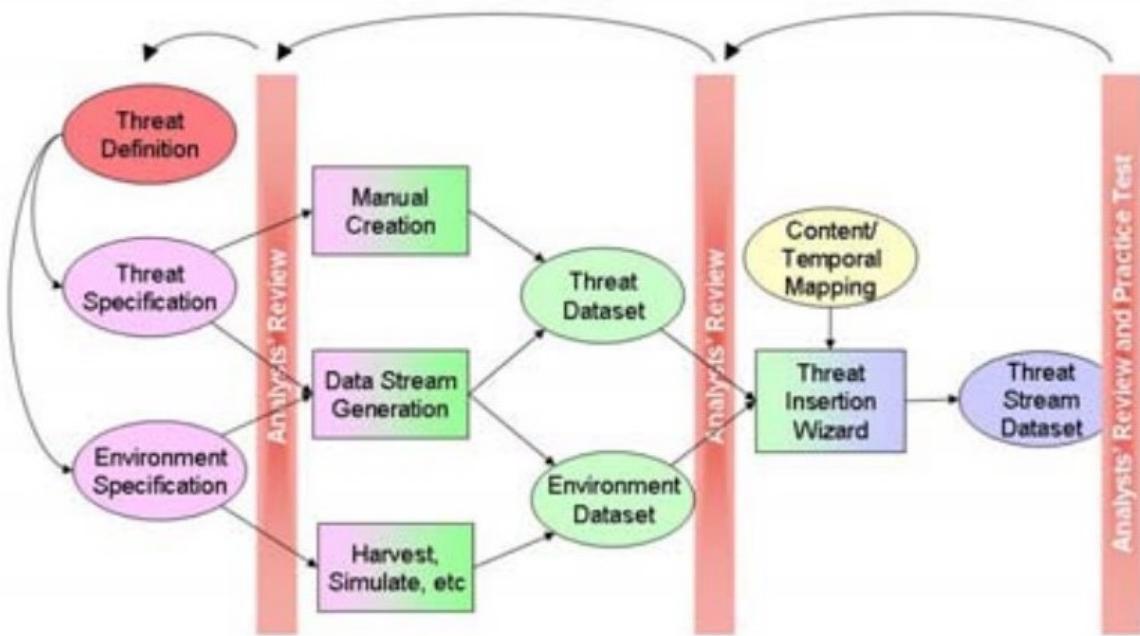


Figura 10. Fluxo de passos para geração dos dados sintéticos.

Permitindo que o usuário possa inserir os tipos de 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.

Dentro do tema que foi especificado (CARS) o DataGenCARS permite gerar dados de forma satisfatória bem como validar por meio de visualizações, mas não foi citado um tratamento para geração de dados faltantes e discrepantes.

Pensando em oferecer confidencialidade dos dados governamentais, Larsen and Huckett (LARSEN; HUCKETT, 2012) desenvolveram um gerador de dados sintéticos que alia regressão de quartis 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, 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. Contudo, não foi citado sobre a geração de dados sintéticos ou faltantes.

Dos trabalhos realizados, nem todos apresentaram uma interface gráfica ao usuário,

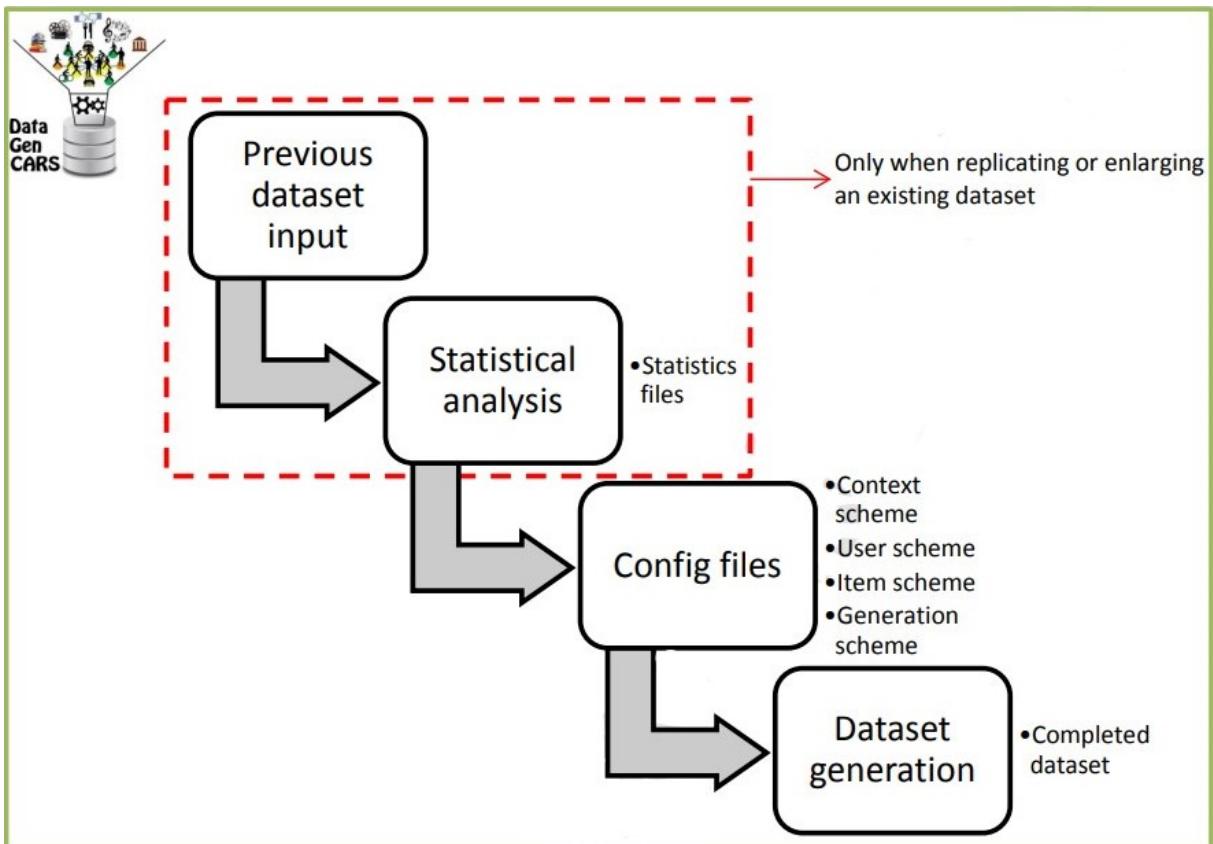


Figura 11. Fluxo de passos para geração dos dados sintéticos.

ou variedade de geradores, com tratamento de dados faltantes ou discrepantes. Também sobre a validação por visualização poucos trabalhos apresentam integração dos dados gerados com uma ferramenta de visualização em tempo real.

3.2 Aplicações

DTM Data Generator (SOFT., 2019) é uma plataforma de geração de dados sintéticos que existe desde 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 pago para utilização oferecendo uma demonstração por tempo limitado. 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 que 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. Não foi encontrado uma documentação com mais detalhes dos geradores e se há suporte para geração de dados faltantes, bem como discrepantes.

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.

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*.

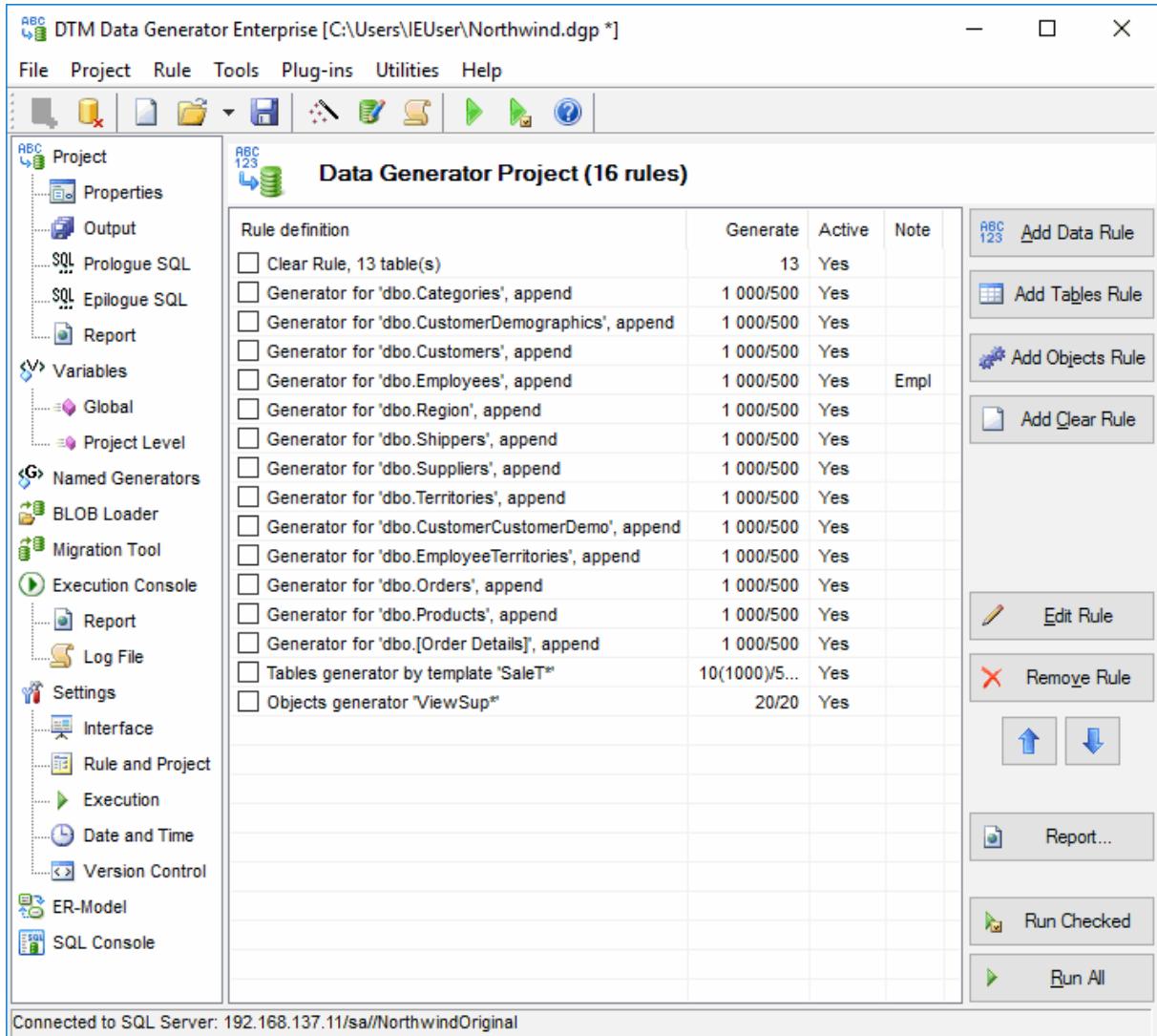


Figura 12. Usando o DTM Data Generator. Fonte: DTM Data Generator

Na documentação do SQL (LTD, b) é possível encontrar os geradores de dados sintéticos. Entre os exemplos estão gerador por regex, importação de arquivo, lista ponderada etc. Não houve uma citação por um gerador de dados discrepantes ou faltantes, contudo é possível criar geradores de dados manualmente utilizando XML em conjunto com classes pré-definidas. Também o uso de embaralhador de texto, bem como a lista ponderada seja possível gerar dados faltantes e discrepantes.

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-lo 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

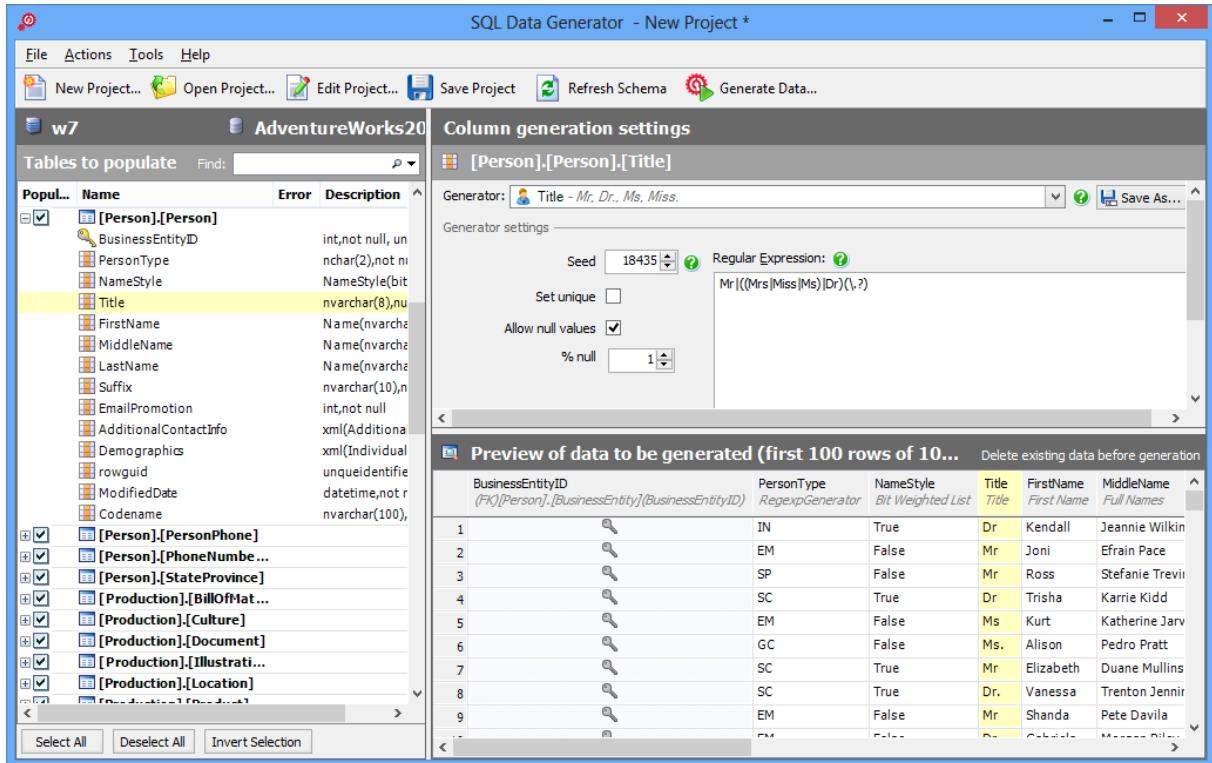


Figura 13. Usando o Redgate SQL Data Generator. Fonte: Red Gate SQL Data Generator.

mais primitivos (Binários, Inteiros, Data, *Float*), como de Imagem, Dinheiro, Expressão Regular, Categórico entre outros. Entre os geradores genéricos pré-definidos, não foi encontrado suporte para dados faltantes ou dados discrepantes, mas há a possibilidade de criar geradores personalizados. 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 predifinidos 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 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

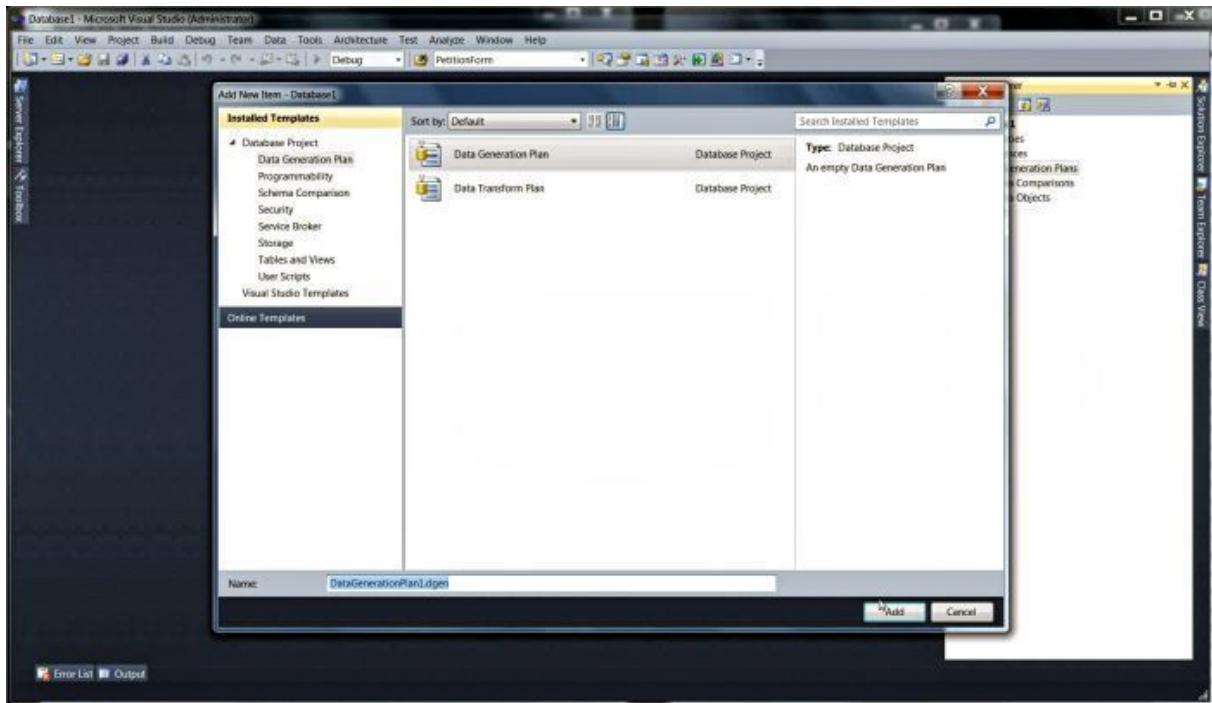


Figura 14. Usando o Microsoft Visual Studio. Fonte: anranik.

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. Ao acessar a documentação (LTD, a) dos geradores, elas não estavam disponíveis (erro 404), logo não foi encontrado mais informações inclusive sobre dados faltantes ou discrepantes.

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 exemplo de 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, 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. O campo *blank* é possível determinar uma porcentagem de dados que ficarão em branco em ordem aleatória. Por conta da característica aleatória, o mecanismo de dados faltantes utilizado é o MCAR.

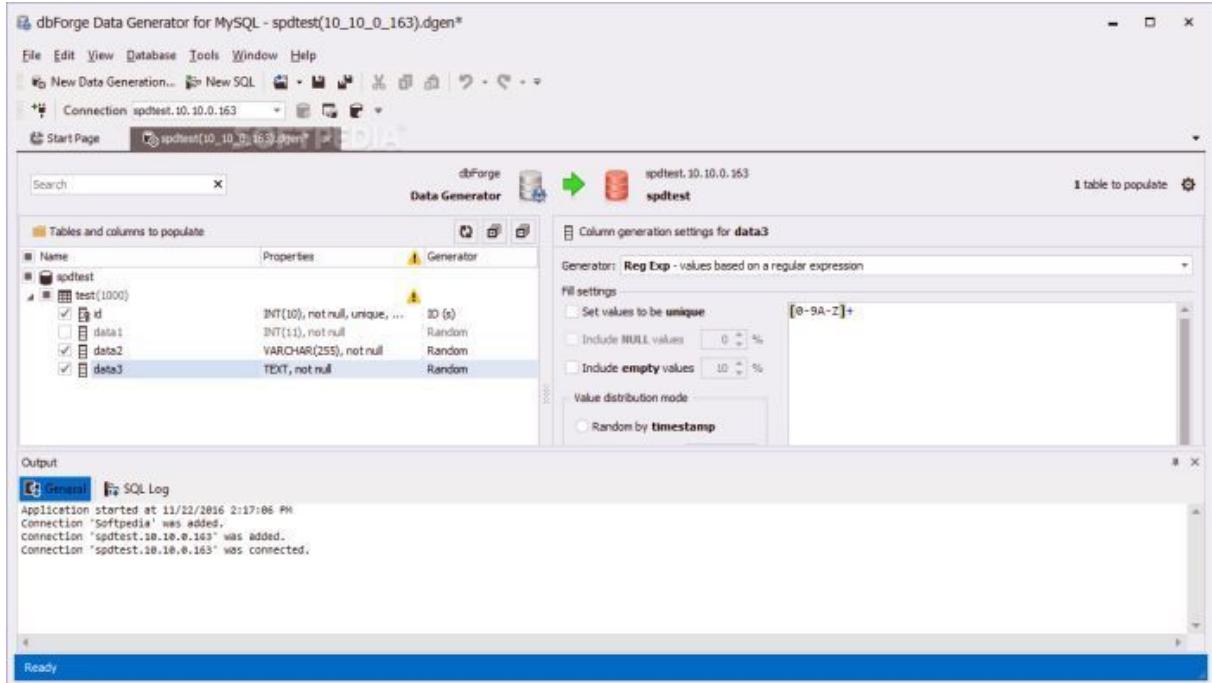


Figura 15. Usando o dbForge Test Data Generator. Fonte: anranik.

Ao lado de *blank* há o botão de fórmula, o qual é possível operar uma função com o valor gerado. Essas funções podem ser matemáticas ou outras disponibilizadas de acordo com a sintaxe de fórmula do Mockaroo. A partir dessa funcionalidade, é possível gerar dados discrepantes tanto ruidosos como anômalos, dependendo da função e parâmetros inseridos.

Ainda na tela inicial encontra-se o botão para *download* dos dados, pré-visualização dos mesmo, - sem gráficos, apenas tabular e CSV - 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
id	Row Number	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
first_name	First Name	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
last_name	Last Name	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
email	Email Address	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
gender	Gender	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
ip_address	IP Address v4	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>

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

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

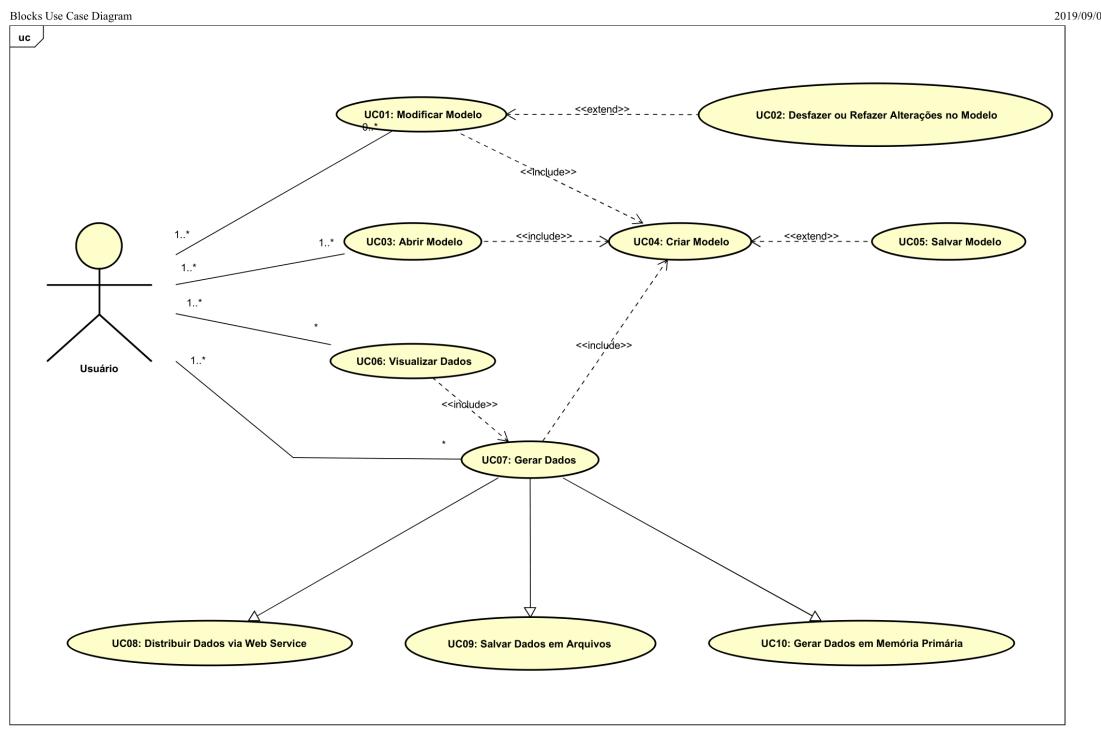


Figura 17. Diagrama de Caso de uso do Blocks. Fonte: o autor.

O diagrama de casos de uso visa demonstrar as diferentes formas de um usuário

pode utilizar um sistema. (INC., 2019) Por conta disso, é utilizado o Diagrama de Casos de Uso para demonstrar as principais funcionalidades do sistema Blocks. Na figura 17 são encontrados um total de 10 casos de uso, nomeados de UC01 a UC10.

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 principal caso de uso do Blocks, 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 VisApplication) - tendo um panorama do comportamento dos dados. Com isso, pode-se validar os dados, ter insights para aprimorar o modelo e afins.

4.2 Diagrama de Classes

No diagrama de classe visto na figura 18 é possível encontrar a estrutura de classe geradores de forma resumida. Há uma classe abstrata chamada *Generator* que possui a maioria dos métodos e atributos, sendo que alguns deles são sobreescritos nas subclasses. Na segunda geração de classes (ver figura 19), elas basicamente existem para identificar as categorias - exceto *Real Data Wrapper* e não adicionam novas propriedades - com exceção da classe *Sequence*, por exemplo. Então a terceira geração de classes (ver figura 20) são os geradores em si, que herdam das suas respectivas categorias, mas uma exceção é a classe *SwitchCaseFunction* a qual possui a propriedade de permitir geradores distintos de acordo com o valor.

4.3 Sequência de atividades no Blocks

Para gerar os diagramas de sequência do Blocks, foram utilizados os casos de uso UC06 e UC09, vistos na figura 17. No diagrama 21 percebe-se que há a introdução das configurações do modelo pelo usuário, então o sistema cria/modifica o objeto modelo (UC01). Mais abaixo, há 2 loops para geração dos dados, o primeiro é a geração pelos

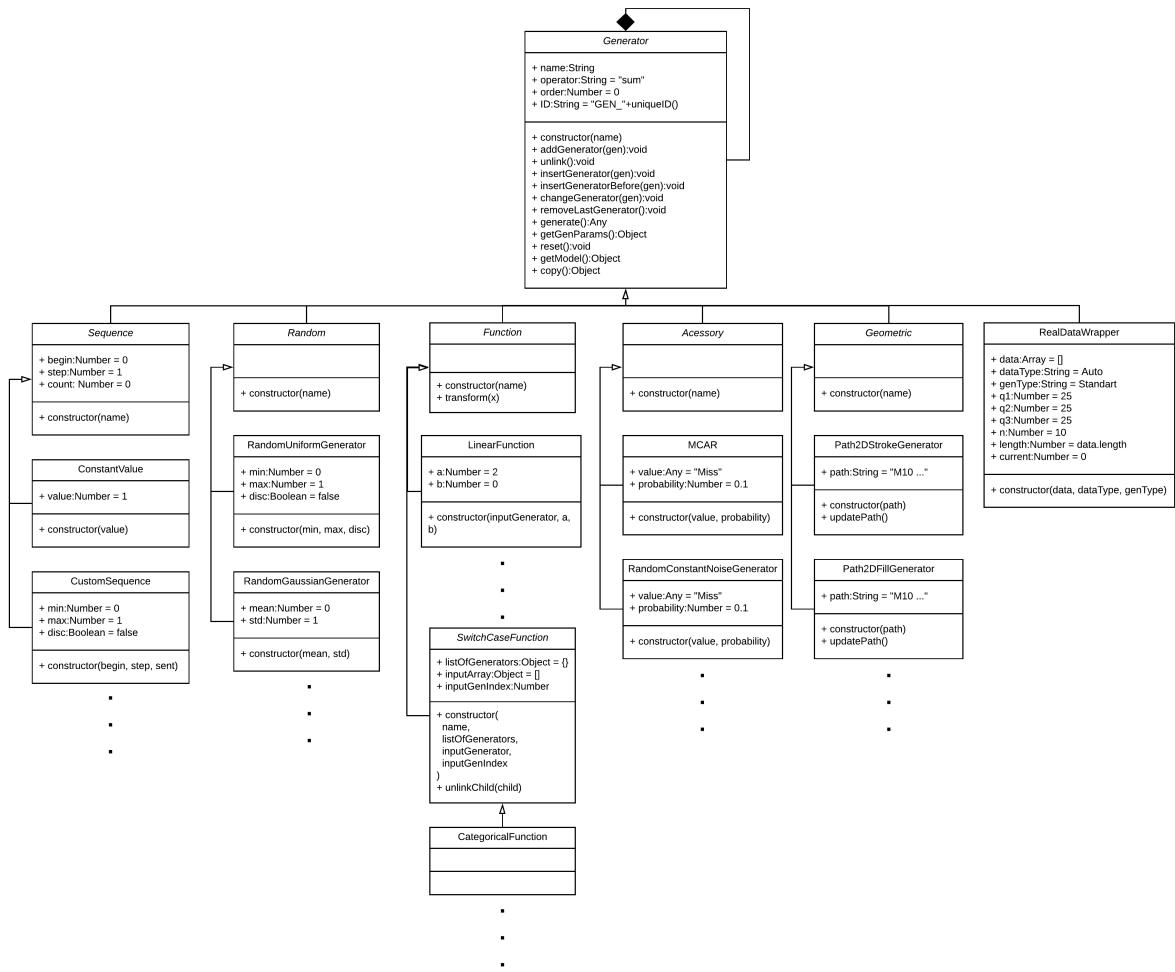


Figura 18. Diagrama de Classes dos geradores do Blocks. Fonte: o autor.

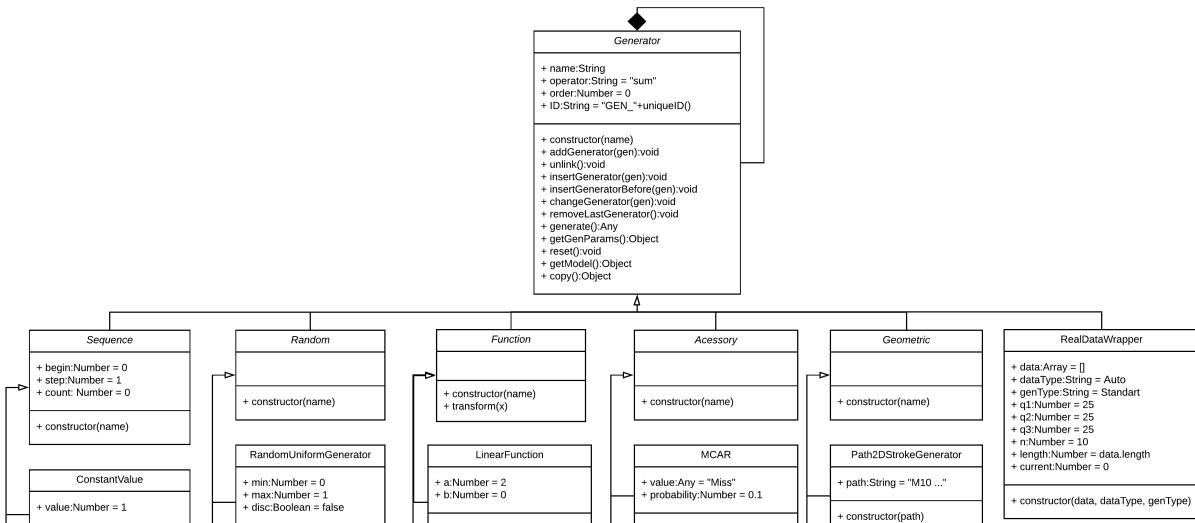


Figura 19. Diagrama de Classes (2^a Geração) dos geradores do Blocks. Fonte: o autor.

geradores presentes no modelo até um certo limite, então esse conjunto de dados é salvo no arquivo, por conseguinte há uma resposta ao usuário de que houve progresso.

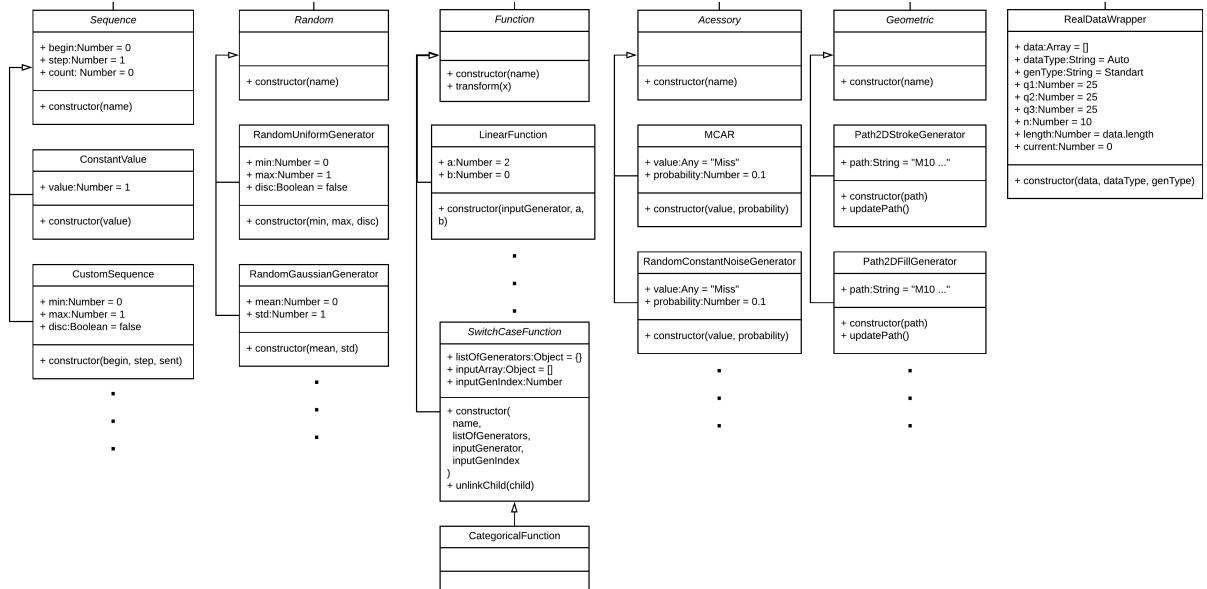


Figura 20. Diagrama de Classes (3^a Geração) dos geradores do Blocks. Fonte: o autor.

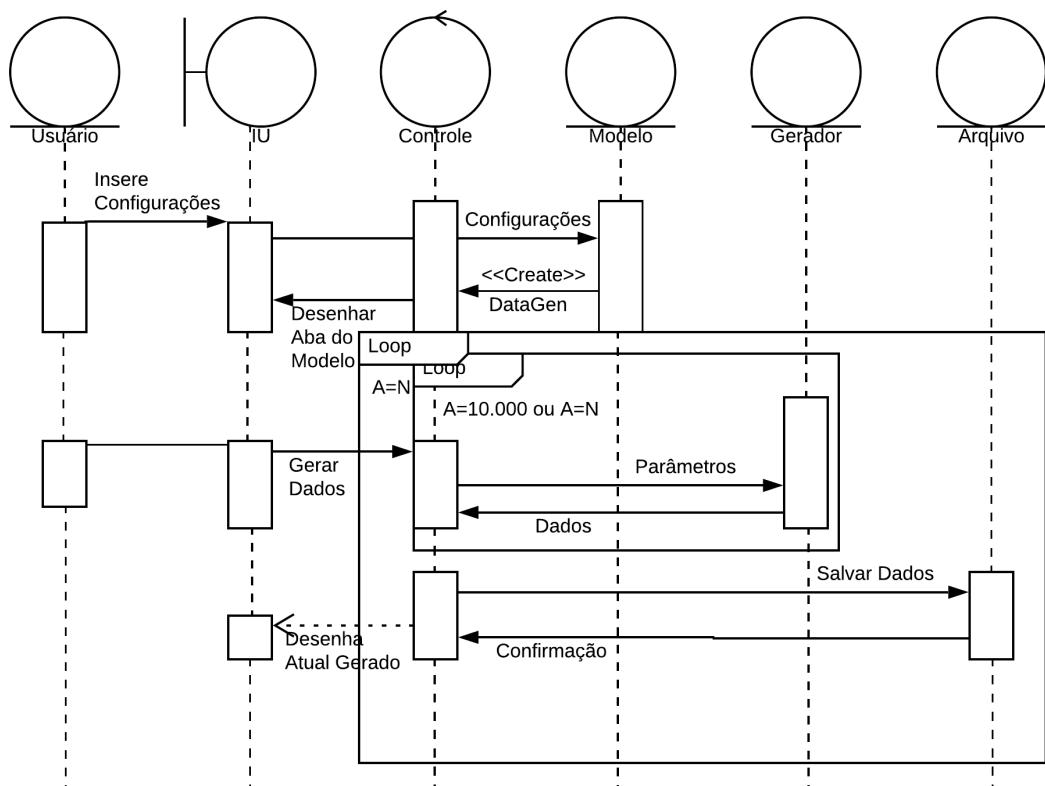


Figura 21. Diagrama de Sequência para geração de dados em arquivos no Blocks. Fonte: O autor.

No diagrama de sequência visto na figura 22 é em relação ao UC06. Primeiramente, assim como na geração do arquivo, há o uso do UC01. Em seguida há geração dos dados em memória referente ao UC10. Com esses dados em memória, o *preview* é desenhado automaticamente. A partir dos dados, há uma comunicação via *WebSocket* com o VisApplication enviando os dados. O VisApplication retorna as opções de visualização, o usuário escolhe uma e então é retornado a visualização para que esta seja renderizada.

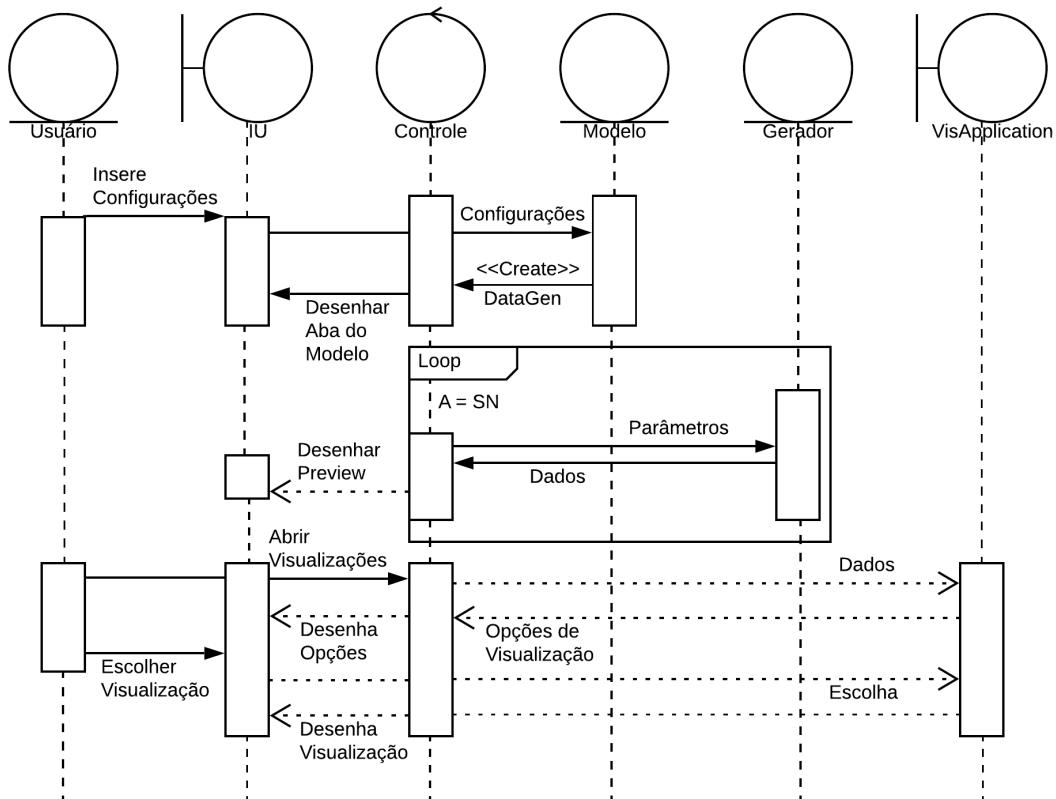


Figura 22. Diagrama de Sequência para visualização dos dados no Blocks. Fonte: o autor.

5 Sistema Blocks

Este capítulo é dedicado em explicar mais sobre o sistema, seu fluxo de funcionamento, funcionalidades, detalhes sobre a interface do usuário entre outros. De modo geral, o sistema é chamado de Blocks e visa ser um gerador de dados sintéticos baseado na combinação de 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.

Também é disponibilizado um pré-visualizador de dados com o gráfico 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 *VisApplication*.

O Blocks permite gerar os dados em arquivos JSON, CSV, TSV ou por de requisições HTTP do tipo GET (*Web Service*). Vale ressaltar que é possível configurar a quantidade de dados gerados, pré-visualizados, formato dos dados gerados, se contém o cabeçalho dos dados no arquivo final e as propriedades do *Web Service*.

Na figura 23 é possível visualizar como foi pensada a utilização da aplicação. Na aba "Configuração do Modelo de Dados" da figura 23 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.

Em "Usar Dados para Avaliação" demonstra os caminhos para geração de dados. É possível a geração 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.

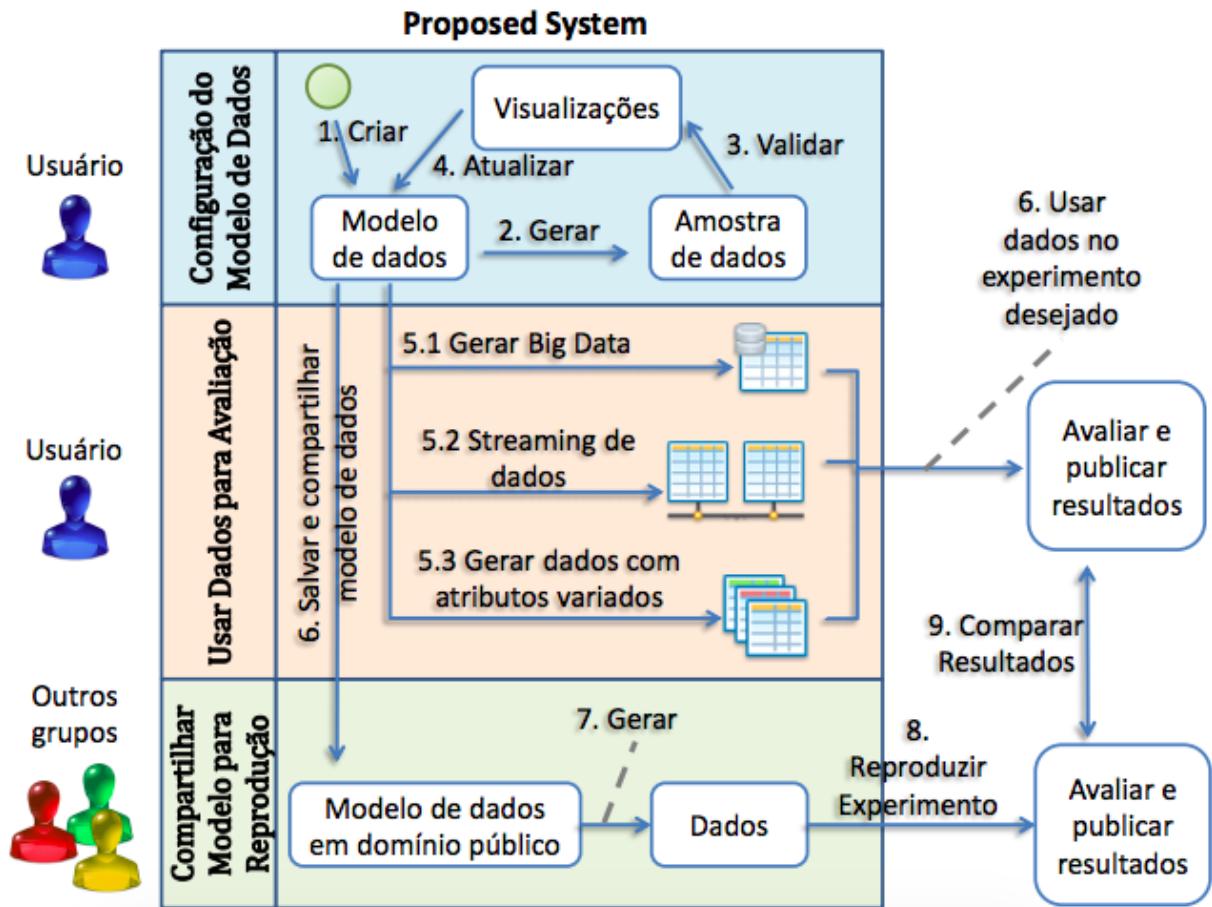


Figura 23. Fluxograma de utilização do Blocks. Fonte: Yvan Brito, 2019.

E em "Compartilhar Modelo para Reprodução" mostra 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 todos os geradores do Blocks até o momento desse trabalho. São apresentados seus nomes, suas categorias, o tipo de valores retornados, os tipos de parâmetros, se possui correlação - isto é, se é baseado no valor de outra dimensão¹, se são dependentes - ou seja, se precisa de um gerador encadeado para 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

Tabela 3. Propriedade dos geradores do Blocks

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

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.

Tabela 4. Resumo dos geradores do Blocks

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	Valores aleatórios temporais em uma distribuição de Poisson.
Uniform	Valores aleatórios númericos em uma distribuição uniforme.
Gaussian	Valores aleatórios númericos em uma distribuição de Gaussiana.
Poisson	Valores aleatórios númericos em uma distribuição de Poisson.
Bernoulli	Valores aleatórios númericos em uma distribuição de Bernoulli.
Cauchy	Valores aleatórios númericos em uma distribuição de Cauchy.
Weighted Categorical	Valores aleatórios categóricos dadas as categorias com probabilidade ponderada.
Categorical	Valores aleatórios categóricos dadas as categorias.
Categorical Quantity	Valores aleatórios categóricos dadas as categorias definindo a quantidade de aparição de categorias.
Linear	Valores a partir de uma dimensão númerica em uma função linear.
Quadratic	Valores a partir de uma dimensão númerica em uma função quadrática.
Polynomial	Valores a partir de uma dimensão númerica em uma função polinomial.
Exponential	Valores a partir de uma dimensão númerica em uma função exponencial.
Logarithm	Valores a partir de uma dimensão númerica em uma função logarítmica.
Sinusoidal	Valores a partir de uma dimensão númerica em uma função senoidal.
Categorical	Valores a partir de uma dimensão categórica .
Piecewise	Valores a partir de uma dimensão númerica com limiar para 2 encadeamentos.
TimeLaps	Valores a partir de uma dimensão temporal que gera dados ao atingir um tempo.
MCAR	Valores faltantes aleatoriamente com auxílio de um ou mais geradores da uma probabilidade.
MNAR	Valores faltantes aleatoriamente com auxílio de um ou mais geradores da uma lógica.
Noise	Valores ruidosos com auxílio de um mais geradores dada uma probabilidade intensidade.
Constant Noise	Dados ruidosos com valor constante com auxílio de um mais geradores da 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	Dados para desenhar uma curva Bezier cúbica.
Path2D Stroke	Dados para desenhar a borda de um polígono.
Path2D Fill	Dados para preencher um polígono.

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 grande número de geradores, pois existe uma diversidade de distribuições de probabilidade comparada às outras categorias.

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 Função

A categoria funções (*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 (*Accessory*) foram pensados especialmente para serem concatenados com outros geradores, com o fim de incrementá-los. Entre os geradores acessórios, pode-se citar o *Missing Value*, o qual foi subdividido em 2 geradores: o *MCAR* e *MNAR*.

O MCAR usa a probabilidade para definir se um dado será faltante; O MNAR trabalha de forma diferente para cada tipo de dado. Os tipos de dado são os Numéricos, os Categóricos e os Temporais.

O MAR é gerado a partir do encadeamento de geradores. Isto é, por conta do MAR levar em consideração a correlação a uma dimensão, este pode utilizar um gerador da categoria função em conjunto do MCAR ou MNAR.

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.

Os geradores MNAR e o MCAR foram desenvolvidos ou modificados no sistema ao longo deste trabalho. Entretanto a implementação foi baseada nos artigos (TWALA, 2009) (XIA et al., 2017) para implementação do MNAR, os trabalhos de (RIEGER; HOTHORN; STROBL, 2010) (XIA et al., 2017) são semelhantes ao gerador MCAR, pois utilizam uma probabilidade para definir aleatoriamente um dado faltante. Os artigos de implementação também podem ser encontrados no artigo (SANTOS et al., 2019) e segundo o mesmo, as implementações são consideradas univariadas, isto é, apenas uma dimensão é afetada.

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* realiza um filtro passa-baixa nos dados. Ou seja, 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 a partir de formas geométricas. Para isso são disponibilizados três geradores. O primeiro deles (*Path2D Stroke*) gera dados bidimensionais presentes nas arestas de um polígono inserido pelo usuário. O segundo (*Path2D Fill*) gera os dados pertencentes do polígono inserido 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 um gerador, chamado como *Real Data Wrapper*. Ele é criado automaticamente quando o usuário importa um conjunto de dados reais através de um CSV. 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 inputação de dados. Esse tratamento é feito através de funções de geração, chamadas de *GenType*.

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

A primeira variação do modo aleatório 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 24. 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

Nessa seção é demonstrado como é feita a concretização do comportamento dos geradores, 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 em dois 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.



Figura 24. 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.

5.2.1 *Streaming Data*

Para otimizar a geração de grande volumes de dados, 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. Esse valor foi definido de forma aleatória, mas que não bloqueasse a aplicação enquanto estava gerando. 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 abanada 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 sistema 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 criando 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 VisApplication 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 Gráfica do Blocks

O sistema possui uma interface gráfica para *Desktop* e segue um modelo de estrutura que possui uma janela principal, mas com janelas auxiliares. Isso significa que há uma tela

com as principais informações (ver figura 25), e outras informações mais raras de serem consumidas aparecem através de *tabs*, *modals*, *alerts* e correlacionados.



Figura 25. Conhecendo os elementos da tela principal do Blocks, na sua versão para Windows. Fonte: O Autor.

Na figura 25, pode-se encontrar os principais elementos para utilizar o Blocks. 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 sistema de gerenciamento de eventos que retira a necessidade de um botão para salvar os dados, toda alteração é salva automaticamente, prevalecendo

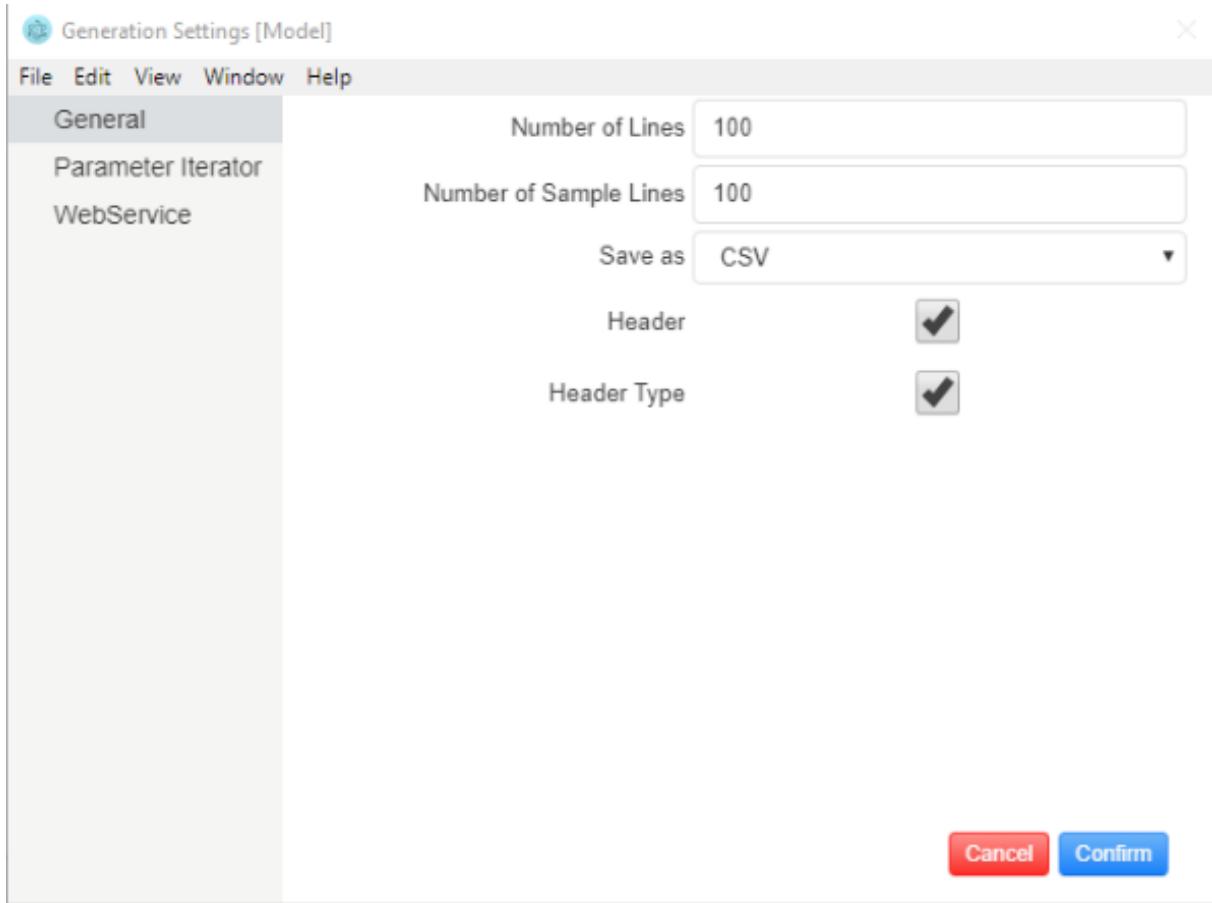


Figura 26. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

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 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.

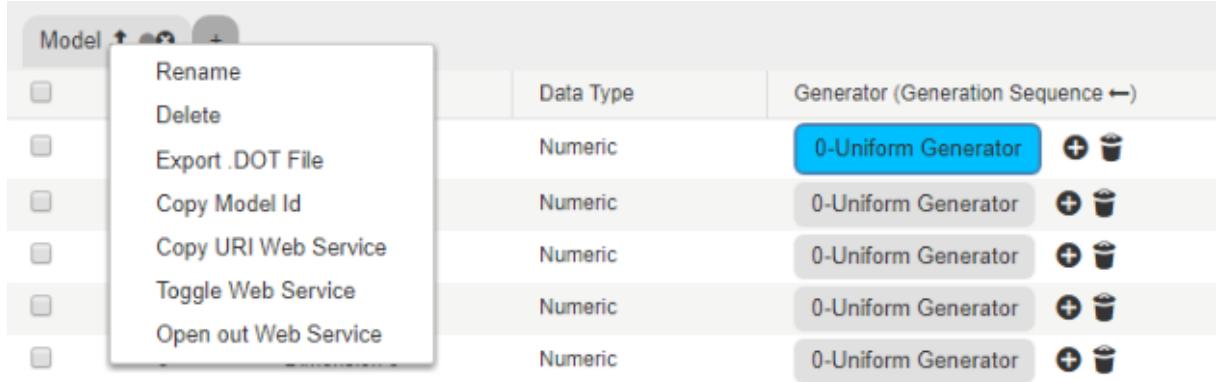


Figura 27. Conhecendo os elementos da *context menu* na aba do modelo. Fonte: O Autor.

Ao lado do botão (*Generate*) para iniciar a geração é encontrado outro botão com o ícone de engrenagem (M21). Na imagem 26, no lado esquerdo, encontra-se 3 seções. A 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 desenvolvida 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 28) 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 29). Para outros avisos mais genéricos como erros ou tarefas bem sucedidas, bem como avisos mais detalhados há o *Modal* (ver figura 30).

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

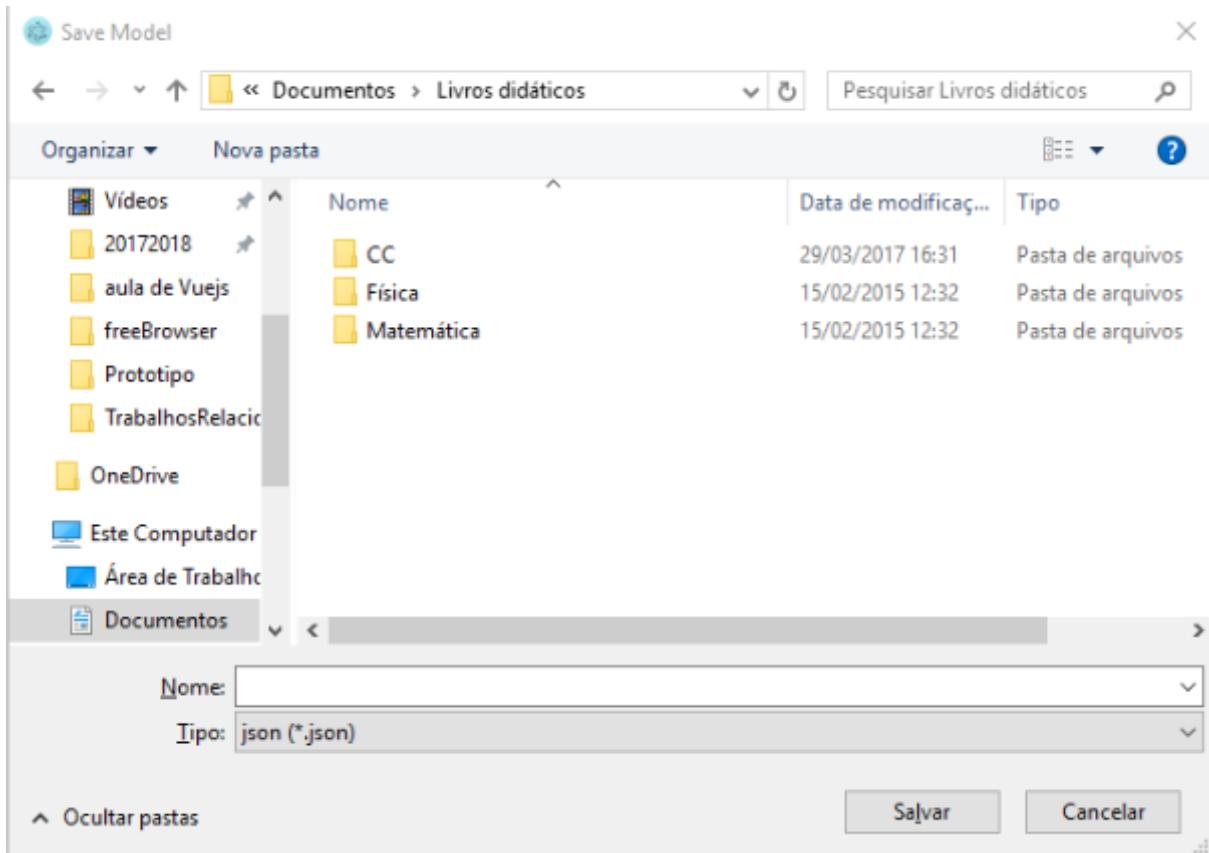


Figura 28. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

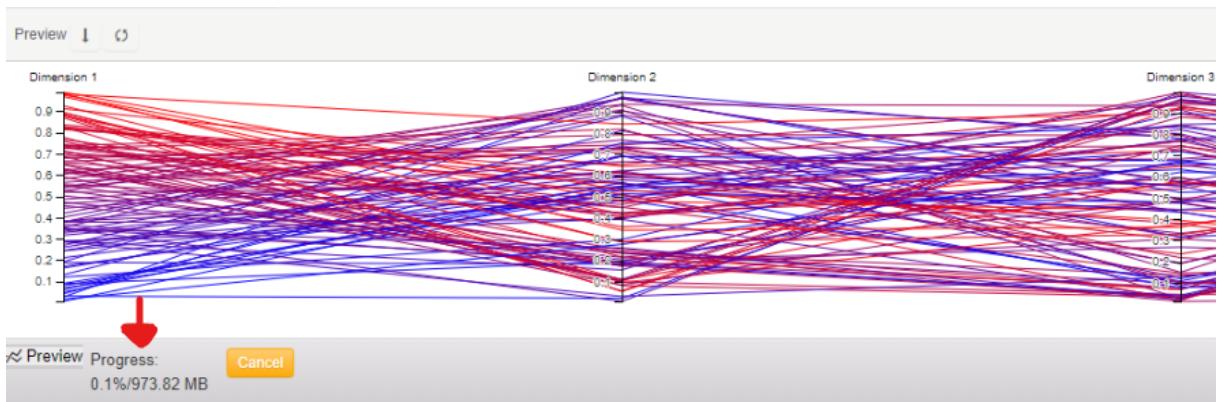


Figura 29. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

acessados pela barra superior da tela inicial (M22). É válido ressaltar, ainda na questão de segurança de uso, que o Blocks salva as mudanças no automaticamente em arquivo separado, o qual pode ser recuperado se não forem salvas/descartadas adequadamente.

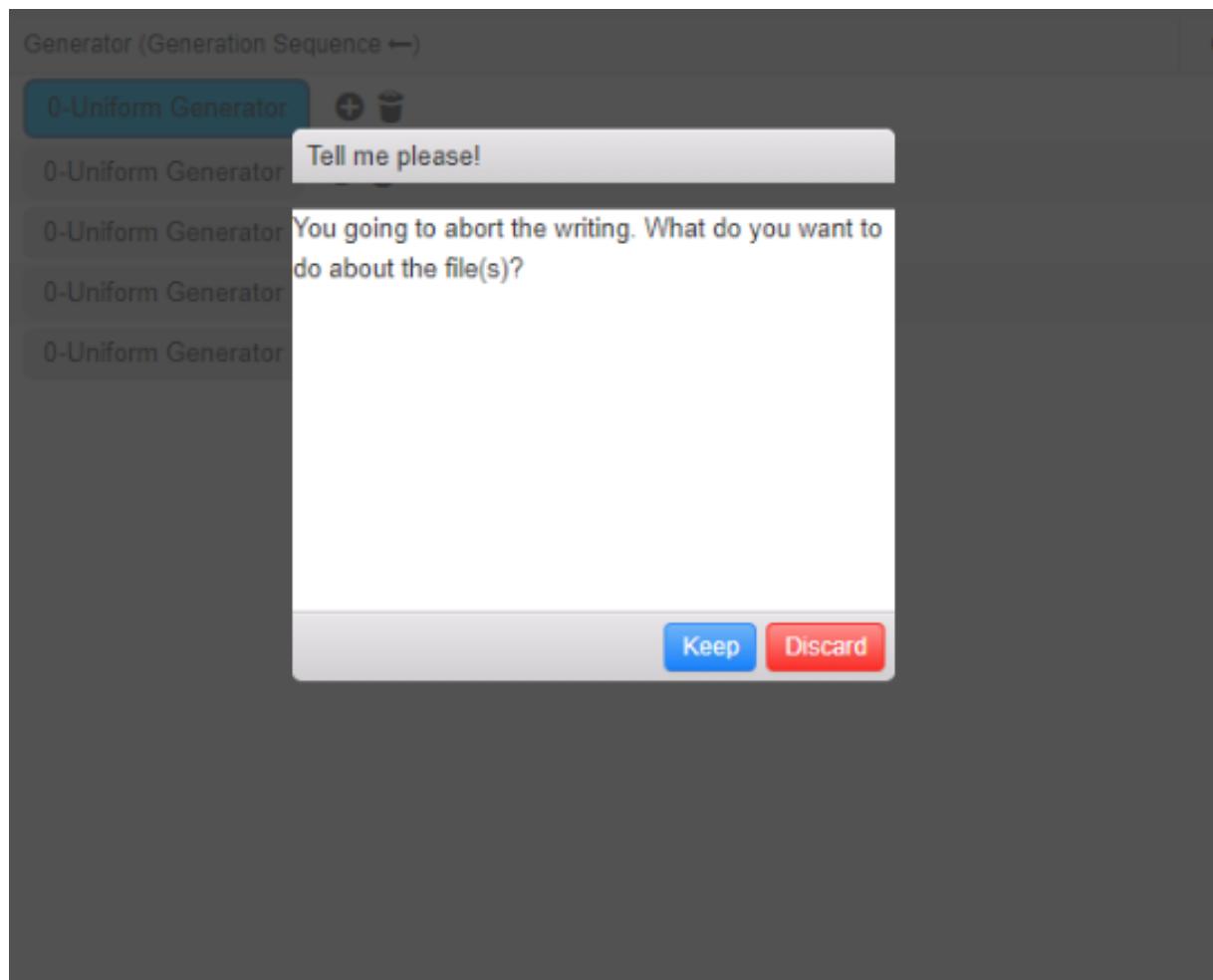


Figura 30. Conhecendo os elementos da tela de configurações para geração de dados. Fonte: O Autor.

6 Validação Visual

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

6.1 Modelagem dos Dados

Foram criadas 5 bases de dados de contextos genéricos, simulando abstratamente contextos reais. Para criação dessas bases com 500 instâncias 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*. Também foram utilizados outros geradores para auxiliar no comportamento das dimensões como os *Linear Function*, *Linear Scale*, *MinMax* e *Piecewise Function*.

A primeira base é sobre uma avaliação de carros, onde há a categoria do carro, o ano, marca, preço, notas dos críticos e dos usuários. Como visto na figura 31, Categoria e Marca são um conjunto de palavras genéricas e uniformemente distribuídas, assim como o Ano do carro, mas com números inteiros. O preço já tem um tratamento de discrepância - os carros podem ficar 30% mais caros ou mais baratos - e uma correlação com o ano - pois carros mais velhos tendem a ficar mais baratos. A nota dos críticos é gerada de forma uniforme e imparcial, para ser um referencial técnico. Contudo, houve uma fraude na modelagem, na qual as notas abaixo de 2 estão faltando, propositalmente, com o fim de reduzir o impacto nas vendas. A nota do público tem dados faltantes com relação ao simples fato de uma pessoa preferir não opinar sobre o assunto e correlação com preço - inversamente proporcional - e nota dos críticos - diretamente proporcional. Quando a nota dos críticos é faltante, o público leva em consideração apenas o preço.

Title	Data Type	Generator (Generation Sequence ←)
Categoria	Categorical	0-Categorical + trash
Ano	Numeric	0-Uniform Generator + trash
Marca	Categorical	0-Categorical + trash
Preço	Numeric	0-Noise Generator 1-Noise Generator 2-Uniform Generator 3-Linear Scale 4-Linear Function + trash
Nota dos críticos	Numeric	0-MNAR 1-Uniform Generator + trash
Nota do público	Numeric	0-Piecewise Function <= Miss 1-MCAR 2-Linear Scale 3-Linear Function + trash > Miss 1-MCAR 2-Linear Function 3-Linear Scale 4-Linear Function + trash +

Figura 31. Base sintética de avaliação de Carros.

A segunda base (ver figura 32) diz respeito a uma avaliação de Redes Sociais. As

dimensões presentes são o nome que é uma categoria genérica; idade que varia de 18 a 65 anos - mas possui dados faltantes acima de 40 anos; postagens que possui uma discrepância fraca - um ruído - para valores altos, visando simular pessoas que postam vários vídeos diariamente. Também uma discrepância para valores baixos, simulando aquelas pessoas que postam esporadicamente; as dimensões Postagens, Seguidores e Curtidas possuem um limite de valores, para manter os dados aproximados da realidade.

Title	Data Type	Generator (Generation Sequence →)						
Nome	Categorical	0-Categorical						
Idade	Numeric	0-MNAR	1-Uniform Generator					
Postagens	Numeric	0-MinMax	1-Noise Generator	2-Noise Generator	3-Uniform Generator	4-Gaussian Generator		
Seguidores	Numeric	0-MinMax	1-Noise Generator	2-Noise Generator	3-MinMax	4-Uniform Generator	5-Gaussian Generator	
Curtidas	Numeric	0-MinMax	1-Noise Generator	2-Noise Generator	3-Linear Function	4-Linear Function		

Figura 32. Base sintética de avaliação de Redes Sociais.

Mais duas dimensões são disponibilizada e também complexas. Entre elas estão a dimensão Seguidores que possui uma forte discrepância, para simular grandes famosos, mas são muito raros. E ainda a dimensão Curtidas, que possui correlação de Seguidores - cerca de 60% das curtidas são de seguidores - e postagens - quanto mais postagens, mais curtidas acumuladas. Além da correlação, possui a discrepância fraca e muito forte, para simular boas postagens e postagens virais.

A terceira base como visto na figura 33 mostra um esquema de convênios médicos. Primeiramente há o profissional e sua especialidade - ambos dados categóricos uniformes; Quanto ao Plano de saúde - é referente aos que atende - e esta dimensão possui um peso, devido à popularidade e/ou acessibilidade dos planos; E o Preço da Consulta varia de acordo com o plano de saúde.

	Order	Title	Data Type	Generator (Generation Sequence ←)						
	1	Profissional	Categorical	0-MCAR	1-Categorical					
	2	Especialidades	Categorical	0-MCAR	1-Categorical					
	3	Planos de Saúde	Categorical	0-Weighted Categorical						
	4	Preço por Consulta	Numeric	0-Categorical Function	A1	0-Gaussian Generator				
					U1	0-Uniform Generator				
					H1	0-Uniform Generator				
					BS1	0-MinMax	1-Cauchy Generator			
					SS2	0-Poisson Generator				
					SYS1	0-Uniform Generator				

Figura 33. Base sintética de avaliação de Convênios Médicos.

A base que simula uma estrutura de conta bancária (ver figura 34) foi feita com a intenção de mostrar dados hierárquicos. A identificação de uma conta bancária é composta

por um banco composto por 3 dígitos, uma agência de 4 dígitos e uma conta composta por 8 dígitos. Foram acrescentados umas opções de dados faltantes.

	Order	Title	Data Type	Generator (Generation Sequence ←)	Options
	1	Banco	Numeric	0-Uniform Generator	+ ⚡
	2	Agência	Numeric	0-MCAR 2-Uniform Generator	+ ⚡
	3	Conta	Numeric	0-MNAR 1-No Repeat 2-Uniform Generator	+ ⚡

Figura 34. Base sintética de avaliação de Estrutura de Conta Bancária.

Há também três bases genéricas unificadas em uma base como vista na figura 35 com o intuito de mostrar os dados faltantes de uma forma mais simples. A primeira dimensão é chamada de dimensão observada, que é a dimensão utilizada para predizer dados faltantes no mecanismo MAR. A segunda dimensão possui dados faltantes do mecanismo MAR. A primeira com a segunda dimensão compõem a primeira base. A terceira dimensão possui do mecanismo MCAR, que também é a segunda base. E a quarta coluna possui valores faltantes do tipo MNAR, a qual é equivalente à terceira base. As bases foram unificadas para termos de comparação, mas as gerações dos dados foram feitas de forma separadas. Utilizando o filtro de dimensões do Blocks.

	Order	Title	Data Type	Generator (Generation Sequence ←)	Options
	1	Observada	Numeric	0-Uniform Generator	+ ⚡
	2	MAR	Numeric	0-Piecewise Function ≤ 50 0-Uniform Generator ≥ 50 0-MCAR 1-Uniform Generator	+ ⚡
	3	MCAR	Numeric	0-MCAR 1-Uniform Generator	+ ⚡
	4	MNAR	Numeric	0-MNAR 1-Uniform Generator	+ ⚡

Figura 35. Base sintética genérica para geração de dados faltantes numéricos.

6.2 Apresentação das Visualizações

Uma vez modelo é preciso visualizar os dados e verificar os resultados da modelagem. O Blocks possui visualizações integradas e elas são utilizadas como exemplos. Entretanto, outras aplicações foram utilizadas para visualização de dados como o RawGraphs <<https://app.rawgraphs.io>>, o Tableau Desktop, e o Excel - Programa de computador que faz parte do pacote *Office* da Microsoft.

Para a base de Carros foram utilizadas 2 visualizações de dados, Coordenadas Paralelas e Scatterplot. Na figura 36, a qual possui 3 imagens de coordenadas paralelas agrupadas. De cima para baixo, a origem coloração dos dados é Preço, Nota dos críticos e

Nota do público. Tom avermelhado significa um valor alto e tom azulado significa valores baixos.

Nessa visualização é possível perceber a grande quantidade de carros baratos, e há uma forte correlação com o ano - são mais antigos - e possuem, em geral, uma alta nota dos críticos e do público. Pode-se perceber um vácuo nas notas 1 e 2 na dimensão das Notas dos críticos, o que é resultado do gerador MNAR. Também é observável a correlação entre as notas dos críticos e nota do público.

Na figura 37 mais detalhes são perceptíveis por conta da dispersão. Primeiramente, os valores abaixo de 0 são os dados faltantes. Esse padrão é adotado não só nesse modelo de dados.

E ainda na figura 37, percebe-se esses dados nas notas do público, e por comparar com outras dimensões, observa-se que é uniforme, logo, caracteriza-se um MCAR. No preço, há um vácuo na faixa de 80.000, apesar de não ter sido intencional, caracteriza-se um MNAR, pois a resposta não está na base de dados. Sobre a correlação, agora é possível perceber com mais clareza a correlação diretamente proporcional entre nota dos críticos e do público, e uma leve correlação inversamente proporcional entre preço e nota do público.

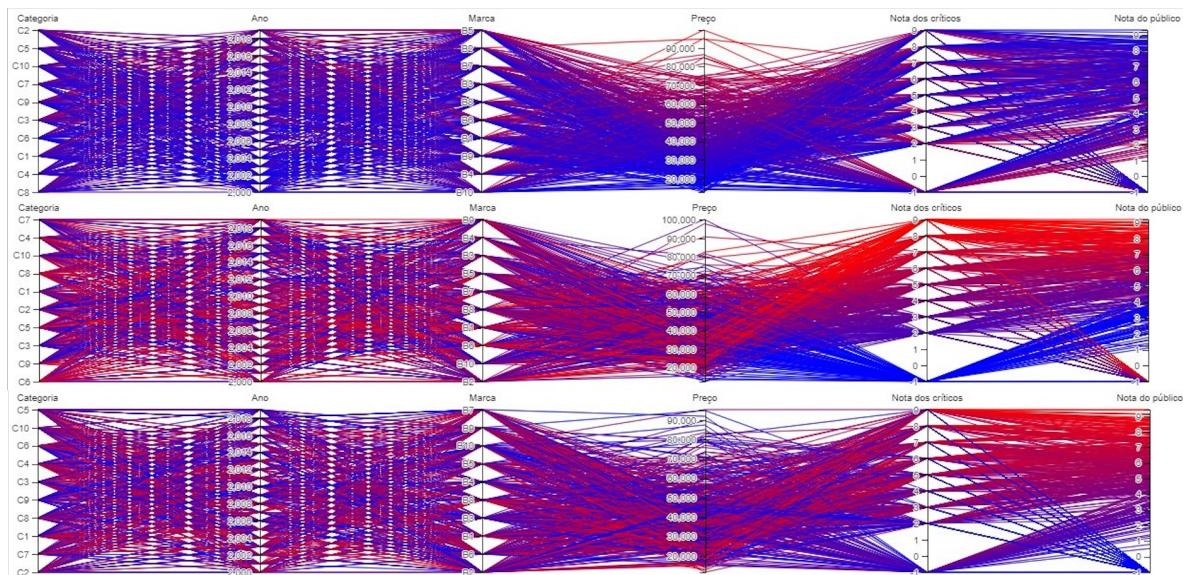


Figura 36. Visualização Coordenadas Paralelas da base de carros.

A figura 38 é capaz de mostrar as principais características do modelo de dados sobre Redes Sociais. Na dimensão Idade é possível encontrar dados faltantes a partir dos 40 anos, logo, caracteriza-se um MNAR, pois não está na base de dados a explicação. E os dados discrepantes nas dimensões de seguidores, curtidas e postagens. O padrão encontrado é que as pessoas possuem uma faixa de 2 mil postagens, menos de 5 milhões

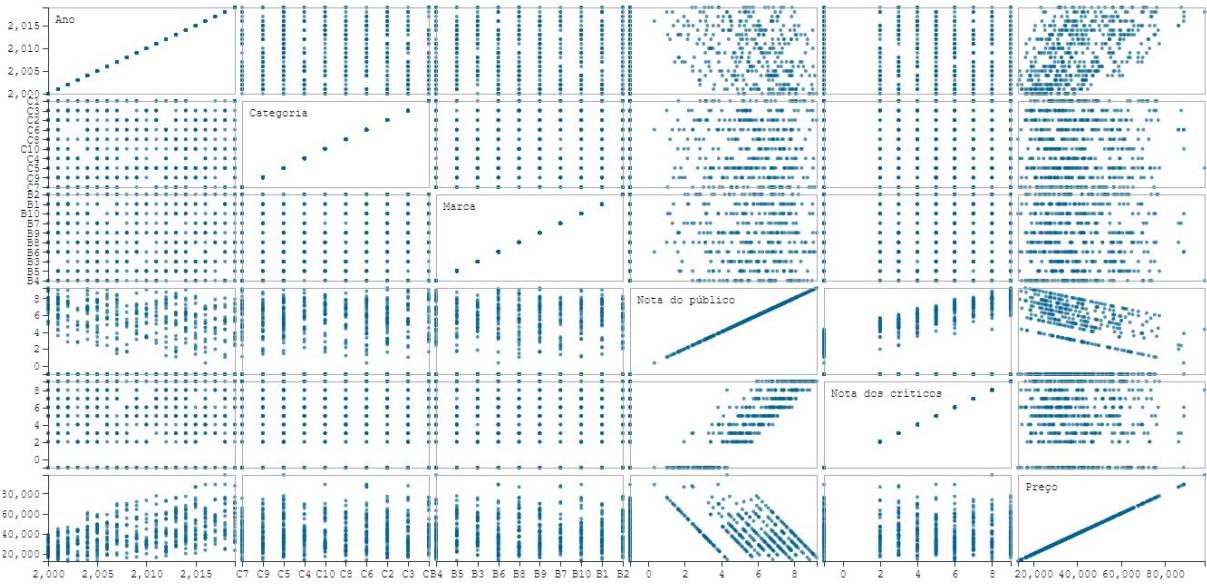


Figura 37. Visualização Scatterplot da base de carros.

de seguidores e menos de 500 mil curtidas no total. E por conta dos dados discrepantes, a própria escala é prejudicada.

No histograma (ver figura 39) é possível visualizar um pouco melhor a questão da escala dos dados. A frequência dos dados numa mesma coluna é muito pouco ou nada nas outras ratifica o problema do alto grau de discrepância da base.

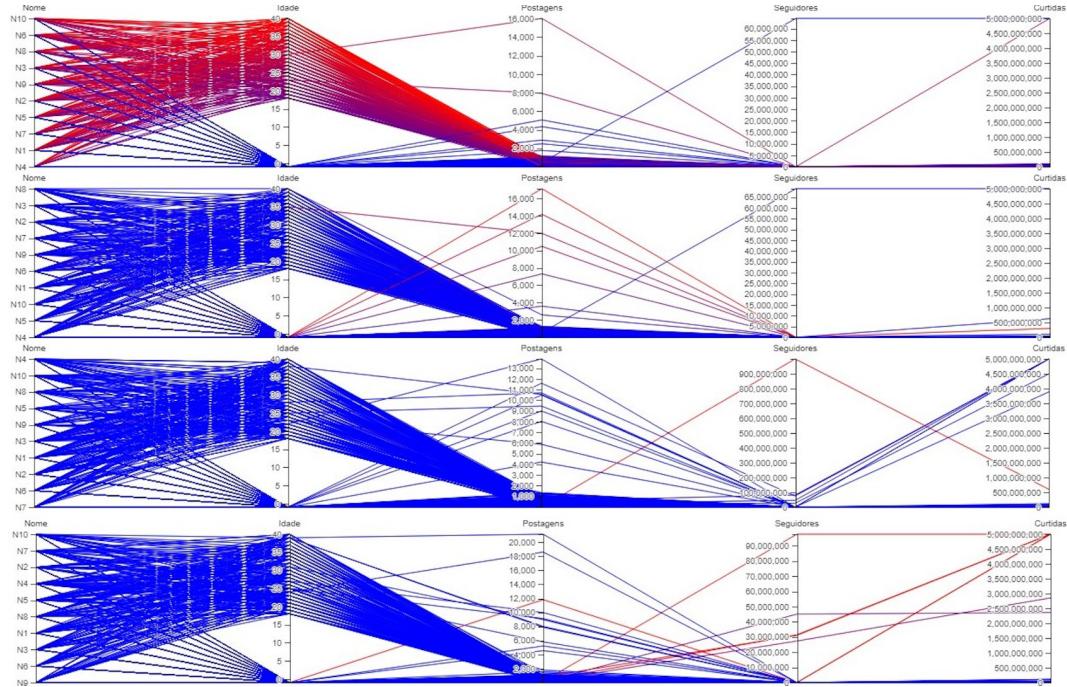


Figura 38. Visualização Coordenadas Paralelas da base de Redes Sociais.

As bases anteriores foram geradas pelo VisApplication e o Preview do Blocks. As visualizações do modelo de base genérica foram geradas no Excel, utilizando os gráficos de

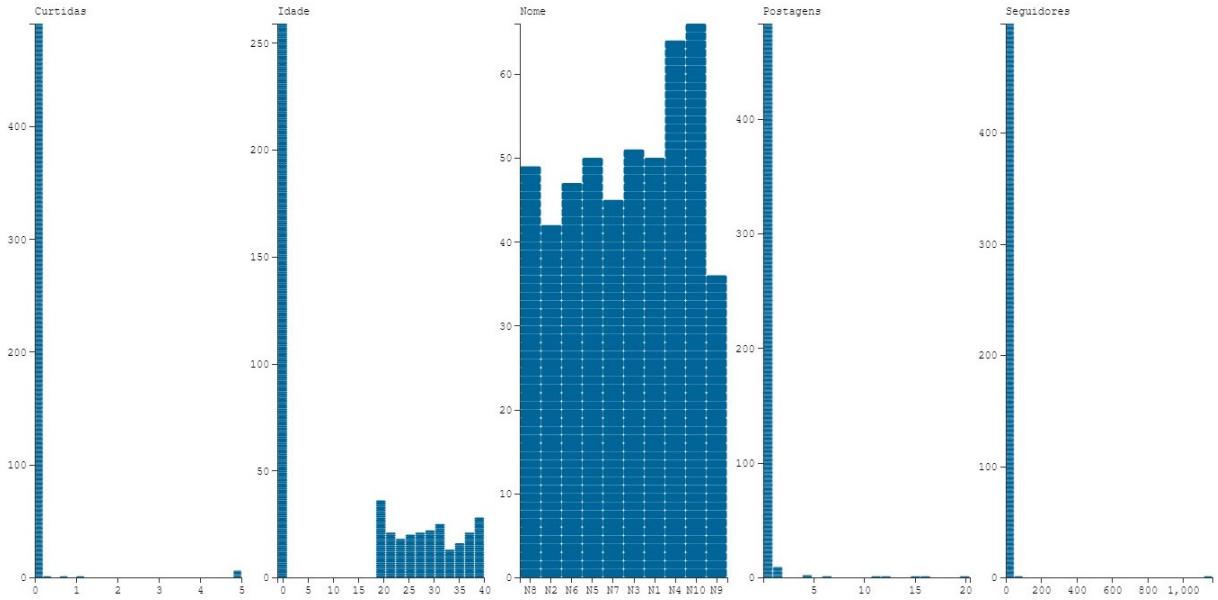


Figura 39. Visualização Histograma da base de Redes Sociais. A unidade de Curtida é bilhões; Seguidores está em milhões e Postagens está em milhares.

linha, pontos e colunas agrupadas.

O modelo de dados sobre estruturas de conta bancária possui uma arquitetura hierárquica e pode ser vista através de um Dendrograma (ver figura 40) A priori, para esta visualização foi necessário reduzir a 10% - 50 instâncias - o volume de dados, para que as propriedades hierárquicas sejam perceptíveis. Na visualização é possível identificar dados faltantes do tipo MCAR, visto que foram perdida de forma aleatória, mas não foi imaginada uma forma de gerar dados discrepantes. Também percebe-se que os valores faltantes foram repetidos para que não fira a propriedade dos dados hierárquicos.

Na figura 41 é possível ver um gráfico de barras subagrupado por especialidades médicas. Neste gráfico pode-se comparar os valores de cada especialidade por plano de saúde. A priori, o SYS1 possui uma discrepância anômala, pois os dados são próximos de 0 - fora adicionados valores ínfimos para que não seja confundido com um dado faltante. Os dados faltantes são representados por uma barra sem tamanho. É possível identificar que há discrepância nos dados, mas não foi utilizado um gerador específico, apenas uma função categórica e diferentes geradores de dados, os quais foram escolhidos aleatoriamente.

Também foi gerada uma base genérica para visualização dos mecanismo de dados faltantes e essas podem ser vistas na figura 42. As duas dimensões à esquerda foram ordenadas juntas, as outras foram dissociadas para que sejam mantidas as propriedades dos dados faltantes. Na figura percebe-se que o a dimensão MAR podem ser preditos a partir da dimensão Observada. A dimensão MCAR possui dados faltantes espalhados, caracterizando completamente aleatório. E a coluna MNAR possui somente seus valores mais baixos como dados faltantes, o que pode ter uma explicação fora da base de

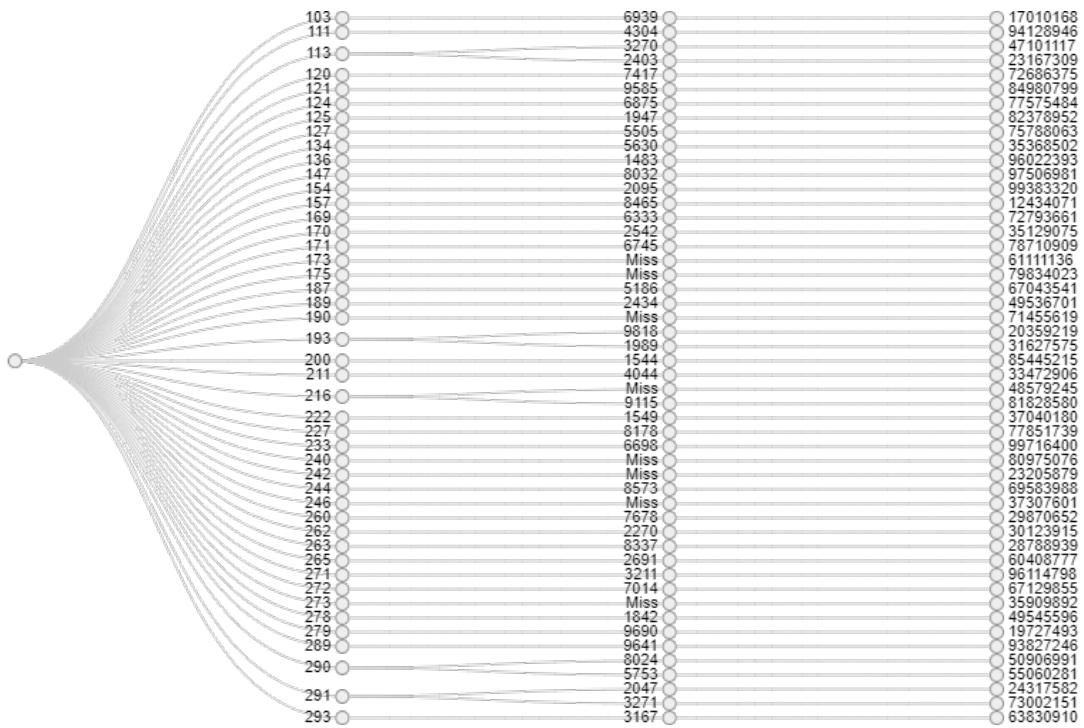


Figura 40. Visualização Dendrograma da base sobre estrutura de conta bancária.

dados.

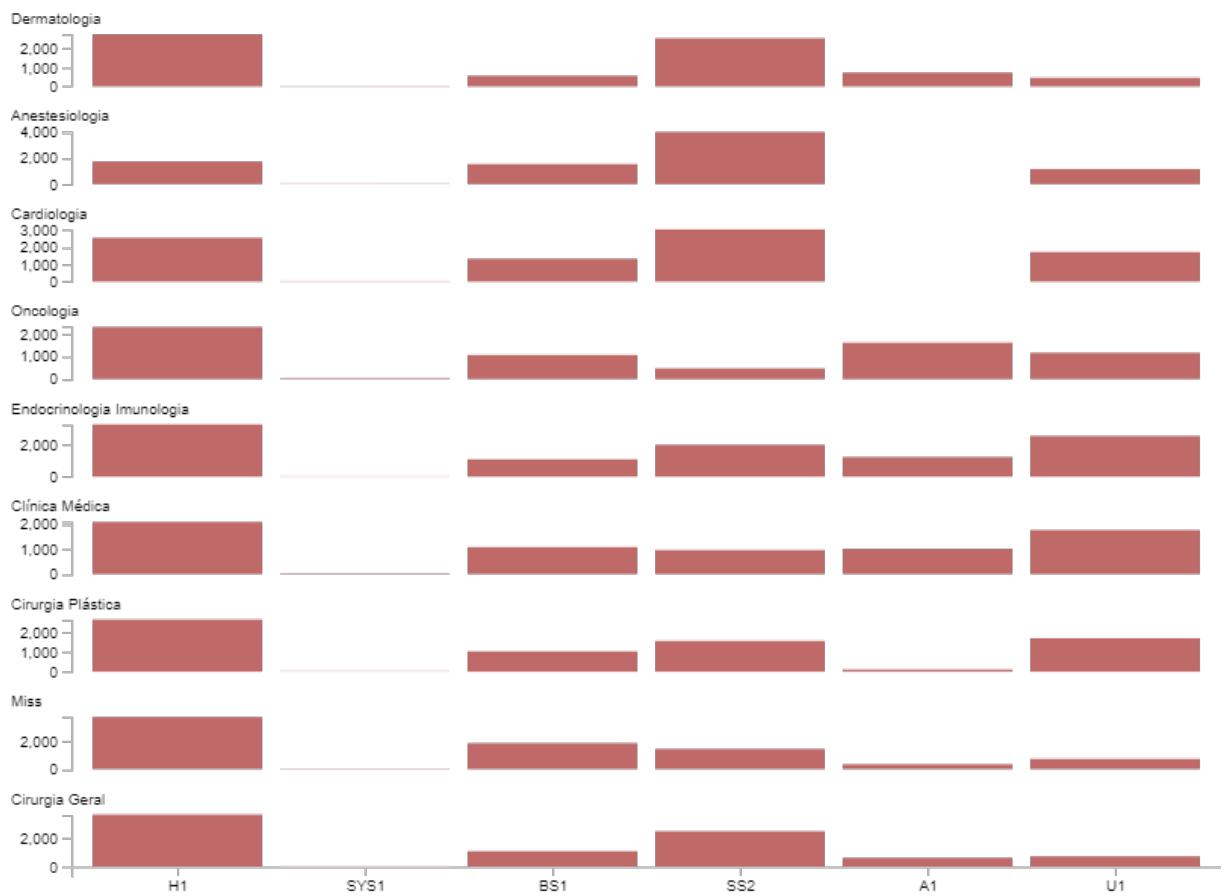


Figura 41. Visualização Gráfica de Colunas da base Convênios Médicos. Eixo X: Planos de Saúde, Eixo Y: Especialidades, Barra: Preço por Consulta

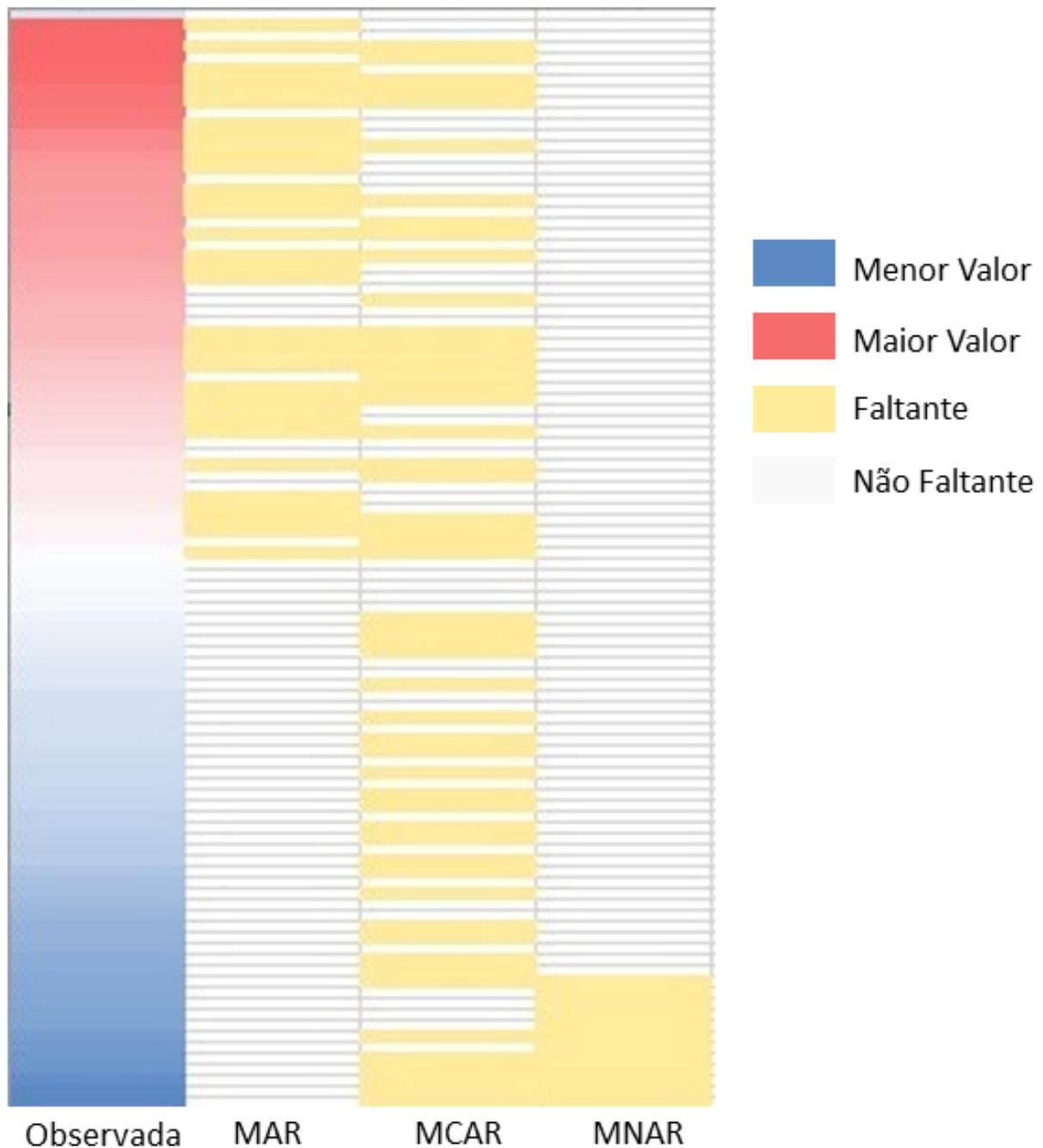


Figura 42. Visualização Gráfica de Colunas da base genérica para visualização de dados faltantes.

7 Considerações Finais

O desenvolvimento do presente trabalho proporcionou a análise da geração de dados sintéticos discrepantes e faltantes na aplicação Blocks. O sistema é *Open Source* e permite que sejam gerados dados sintéticos e visualizá-los.

Os dados sintéticos são importantes para mais variadas aplicações, por conta da seu desvinculo com a confidencialidade dos dados. O fato de geração basear-se em modelos permite que o compartilhamento dos dados seja mais leve, mas exigindo do processamento em contrapeso. A gratuidade para consumir os dados sintéticos permite que a pesquisa e teste de aplicações seja fomentado e democratizado, pois é acessível para todos.

O Blocks é um sistema bastante completo. Sua geração de dados é baseada em modelos, os quais por sua vez possuem blocos de geradores que podem ser encadeados. Esse encadeamento permite que o comportamento dos dados sejam mais complexos, o que distancia de dados mais previsíveis e comuns. Além do encadeamento, o sistema possui uma vasta gama de geradores, desde sequenciais, aleatórios, funcionais, acessórios e geométricos, os quais permitem que o usuário crie variados cenários e níveis de abstração.

O Blocks também permite que esses dados sejam distribuídos via *Web Service*; salvos em arquivos - CSV, JSON, TSV; O sistema permite visualizar os modelos de dados de forma rápida e prática ou de forma mais detalhada e diversa com a integração com o *VisApplication*.

Nos resultados, os dados faltantes do mecanismo MCAR apresentaram a característica aleatória dada uma taxa de ausência. No MAR, os dados também foram faltantes, mas foi possível enxergar correlação entre as dimensões que possui a ausência e a dimensão observada. O gerador do mecanismo MNAR permite que os dados faltem dada uma lógica - para simular um motivo fora da base de dados.

Portanto, de forma geral, a aplicação permitiu que as características - dados faltantes e discrepantes - fossem percebidas no conjunto de dados de forma satisfatória, como vista na seção de Resultados, pois além da discrepância e falta de dados, também foi possível fazer correlações. Para propor soluções para os problemas vistos, é interessante que o Blocks possua mais geradores categóricos, como gerar a partir de uma expressão regular, a partir de um arquivo, geração de anagramas datas as letras ou uma categoria, permitir embaralhar categorias etc. Um gerador temporal sazonal, que possui a estrutura de encadeamento similar ao *TimeLaps Function*. Alguns acessórios também são interessantes como formatar números de diferentes geradores - como ter números discretos ou com casas decimais fixas.

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 na página 19.
- ALBUQUERQUE, G.; LOWE, T.; MAGNOR, M. Synthetic generation of high-dimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, Institute of Electrical and Electronics Engineers (IEEE), v. 17, n. 12, p. 2317–2324, dez. 2011. Disponível em: <<https://doi.org/10.1109/tvcg.2011.237>>. Citado na página 22.
- 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 15.
- BERGEAT, M. et al. A french anonymization experiment with health data. In: . [S.l.: s.n.], 2014. Citado na página 15.
- 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 16.
- 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 16.
- DEAN, S.; ILLOWSKY, B. Descriptive statistics: Histogram. *Retrieved from the Connexions Web site: http://cnx.org/content/m16298/1.11*, 2009. Citado na página 25.
- DEVART. *Data Generator for SQL Server*. 2018. [Https://www.devart.com](https://www.devart.com). Disponível em: <<https://docs.devart.com/data-generator-for-sql-server/>>. Acesso em: 21 ago 2019. Citado na página 31.
- 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 15.
- 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 16.
- 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 23.

GROUP, W. W. *Web Services Architecture*. 2004. Www.w3.org. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso em: 02 ago 2019. Citado 2 vezes nas páginas 16 e 17.

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 17 e 18.

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 16.

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 36.

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 25.

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 16.

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 15.

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 27.

LITTLE, T. D. et al. Missing data. *Developmental psychopathology*, Wiley Online Library, p. 1–37, 2016. Citado 2 vezes nas páginas 17 e 18.

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 23.

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.I.], 2012. p. 33–40. Citado na página 15.

LTD, R. G. S. *Data Generator for SQL Server Documentation*. Disponível em: <<Https://docs.devart.com/data-generator-for-sql-server/basic-generators/check-constraint-generator>>. Acesso em: 10 dec 2019. Citado na página 32.

LTD, R. G. S. *SQL Data Generator Documentation*. Disponível em: <Https://documentation.red-gate.com/xx/files/7471185/15925341/1/1370009573853/SQL+Data+Generator+1_2.pdf>. Acesso em: 10 dec 2019. Citado na página 30.

LTD, R. G. S. *SQL Data Generator*. 2019. [Https://www.red-gate.com/](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 29.

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 17.

MICROSOFT. *Generating Test Data for Databases by Using Data Generators*. 2019. [Https://www.microsoft.com/](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 30.

MOCKAROO. *Mockaroo APIs*. 2019. [Https://www.mockaroo.com/](https://www.mockaroo.com/). Disponível em: <<https://www.mockaroo.com/api/docs>>. Acesso em: 23 ago 2019. Citado na página 33.

MOCKAROO. *Mockaroo, realistic data generator*. 2019. [Https://www.mockaroo.com/](https://www.mockaroo.com/). Disponível em: <<https://www.mockaroo.com/>>. Acesso em: 23 ago 2019. Citado na página 32.

MORAES, D. *Um pensamento sobre os Dados Sintéticos*. Update or Die. Disponível em: <<https://www.updateordie.com/2019/09/25/um-pensamento-sobre-os-dados-sinteticos/>>. Acesso em: 09 dec 2019. Citado na página 13.

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 19 e 20.

RAYMOND, E. S. *Data File Metaformats. Chapter 5. Textuality*. 2003. [Http://www.catb.org](http://www.catb.org). Disponível em: <<http://www.catb.org/~esr/writings/taoup/html/ch05s02.html>>. Acesso em: 31 jul 2019. Citado na página 16.

RIEGER, A.; HOTHORN, T.; STROBL, C. Random forests with missing values in the covariates. 2010. Citado na página 45.

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 26.

RUBIN, D. B. Statistical disclosure limitation. *Journal of official Statistics*, v. 9, n. 2, p. 461–468, 1993. Citado na página 15.

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 25.

SANTOS, M. S. et al. Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, IEEE, v. 7, p. 11651–11667, 2019. Citado na página 45.

- 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 16.
- 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 28.
- 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 17.
- TANENBAUM, A. S.; FILHO, N. M. *Sistemas operacionais modernos*. [S.l.]: Prentice-Hall, 1995. v. 3. Citado 2 vezes nas páginas 15 e 16.
- TWALA, B. An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, Taylor & Francis, v. 23, n. 5, p. 373–405, 2009. Citado na página 45.
- 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 22.
- 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 26.
- XIA, J. et al. Adjusted weight voting algorithm for random forests in handling missing values. *Pattern Recognition*, Elsevier, v. 69, p. 52–60, 2017. Citado na página 45.