XII SINGEP
Simpósio Internacional de Gestão, Projetos, Inovação e Sustentabilidade
International Symposium on Management, Project, Innovation and Sustainability
ISSN: 2317-8302
CYRUS Institute of Knowledge
MAKE A DIFFERENCE
CIK 12th INTERNATIONAL CONFERENCE

CODIFICAÇÃO PARA SIMULAÇÃO DE MONTE CARLO EM PLANILHAS ELETRÔNICAS

CODING FOR MONTE CARLO SIMULATION IN SPREADSHEETS

WANDERSON DUTRA GRESELE
UNIOESTE - UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ

Comunicação:
O XII SINGEP foi realizado em conjunto com a 12th Conferência Internacional do CIK (CYRUS Institute of Knowledge) e com o Casablanca Climate Leadership Forum (CCLF 2024), em formato híbrido, com sede presencial na ESCA Ecole de Management, no Marrocos.

Agradecimento à orgão de fomento:
Agradecimentos ao Fundo de Apoio à Pesquisa - FAP UNINOVE e destacamos que o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001

CODIFICAÇÃO PARA SIMULAÇÃO DE MONTE CARLO EM PLANILHAS ELETRÔNICAS

Objetivo do estudo
Desenvolver e apresentar uma solução de codificação para implementar o Método de Monte Carlo em planilhas eletrônicas, utilizando o Visual Basic for Applications (VBA), facilitando a modelagem e análise de sistemas complexos em ambientes organizacionais.

Relevância/originalidade
O estudo oferece uma abordagem prática e acessível para a aplicação do Método de Monte Carlo em contextos de gestão, abordando a crescente necessidade de modelagem diante da complexidade dos sistemas do mundo real.

Metodologia/abordagem
Foi utilizado o Visual Basic for Applications (VBA) para codificar a geração de números aleatórios baseados em diferentes distribuições de probabilidade, seguindo os processos descritos por Thomopoulos (2013), e testando a codificação conforme as características de cada distribuição.

Principais resultados
A ferramenta demonstrou alta aplicabilidade e replicabilidade, confirmando sua utilidade na modelagem de sistemas complexos. Facilita o processo de ensino/aprendizagem em contextos acadêmicos e possui potencial significativo para uso em ambientes organizacionais, impactando positivamente a tomada de decisão.

Contribuições teóricas/metodológicas
Ajuda na a compreensão sobre a aplicação do Método de Monte Carlo e fornece uma ferramenta replicável para profissionais e acadêmicos, contribuindo para a disseminação prática e teórica da SMC.

Contribuições sociais/para a gestão
A ferramenta pode melhorar a tomada de decisões em ambientes organizacionais complexos, permitindo uma análise mais precisa de riscos, otimização de processos e suporte à gestão em diversos setores.

Palavras-chave: Geração de números aleatórios, Tomada de decisão, Análise de riscos, Sistemas complexos, Simulação

CODING FOR MONTE CARLO SIMULATION IN SPREADSHEETS

Study purpose

To develop and present a coding solution for implementing the Monte Carlo Method in spreadsheets using Visual Basic for Applications (VBA), facilitating the modeling and analysis of complex systems in organizational environments.

Relevance / originality

The study offers a practical and accessible approach to applying the Monte Carlo Method in management contexts, addressing the growing need for modeling in light of the complexity of real-world systems.

Methodology / approach

Visual Basic for Applications (VBA) was used to code the generation of random numbers based on different probability distributions, following the processes described by Thomopoulos (2013), and testing the coding according to the characteristics of each distribution.

Main results

The tool demonstrated high applicability and replicability, confirming its usefulness in modeling complex systems. It facilitates the teaching/learning process in academic contexts and has significant potential for use in organizational environments, positively impacting decision-making.

Theoretical / methodological contributions

It enhances the understanding of the application of the Monte Carlo Method and offers a replicable tool for professionals and academics, contributing to the practical and theoretical dissemination of Monte Carlo Simulation (MCS).

Social / management contributions

The tool can improve decision-making in complex organizational environments by enabling more accurate risk analysis, process optimization, and management support across various sectors.

Keywords: Random number generation, Decision-making, Risk analysis, Complex systems, Simulation

CODIFICAÇÃO PARA SIMULAÇÃO DE MONTE CARLO EM PLANILHAS ELETRÔNICAS

1 APRESENTAÇÃO E CONTEXTO DO PROBLEMA

As decisões que que os gestores tomam nas organizações sempre estão pautadas no modelo de mundo construídos por eles (WEICK, 1995). Se as relações que compõem este modelo forem simples é possível usar métodos matemáticos (LAW, 2015). Entretanto, quando se busca compreender sistemas complexos alguns métodos não são efetivos, como os sistemas do mundo real. Logo, os modelos devem comtemplar a complexidade do mundo real, que pode ser desenvolvido por meio de simulação, a fim de estimar as verdadeiras características desejadas do modelo. (LAW, 2015).

Foi na década de 50, no ambiente do Projeto Manhattan e do primeiro computador eletrônico, conhecido como ENIAC, houve a renascença de uma técnica matemática, tradicionalmente conhecida como amostragem estatística, que em seu novo ambiente e devido à sua natureza, adveio com o nome de Método de Monte Carlo (METROPOLIS, 1987).

Atualmente, o Método de Monte Carlo tem sido extensivamente usado em vários segmentos para além da área da física e estatística, em empresas privadas e governos, sempre na busca de estudar o comportamento de sistemas, consequentemente, como uma ferramenta para o processo de tomada de decisões, obtendo resultados mais assertivos em face da complexidade (BRANDIMARTE, 2014).

Por exemplo, Xei (2020) desenvolveu um modelo de simulação de Monte Carlo para representar a dinâmica de propagação do COVID-19. Primeiramente examinou vários desempenhos esperados do modelo de simulação, assumindo uma série de cenários definidos arbitrariamente e, então, estudos de simulação foram realizados nos dados reais do COVID-19 relatados na Austrália e Reino Unido. Os resultados dos números de reprodução estimados do COVID-19 foram consistentes, mostrando que o modelo de simulação foi considerado uma ferramenta de tomada de decisão eficaz e adaptável na luta contra o COVID-19 na necessidade imediata e para modelar quaisquer outras doenças infecciosas no futuro.

Claro que atualmente o método também está presente na área de gestão, como exemplo, considere uma empresa de manufatura que está pensando em expandir a produção uma de suas fábricas, mas não tem certeza se o potencial ganho de produtividade justificaria o gasto da construção (Law, 2015); quando uma empresa se depara com a situação de pedido extra, como estabelecer métricas adequadas para a decisão econômica de atender ou não a pedidos extras cujos produtos têm grande variabilidade de custos variáveis (Saraiva Jr., Tabosa e Costa, 2011); ou ainda a necessidade de analisar a viabilidade econômica de um projeto de renovação de equipamentos operacionais numa avaliação comparativa dentro do portfólio de projetos de uma empresa (De Lucas *et al.*, 2022).

Dado sua importância, e na busca de difundir a prática de simulação junto ao campo, tanto no processo de ensino quanto no processo de pesquisa, este trabalho tem como objetivo **apresentar uma solução de codificação para utilização do Método de Monte Carlo junto à softwares de planilhas utilizando-se do *Visual Basic for Aplications***. Para tal, buscou-se apresentar as distribuições de frequências e a formulação para a geração de números aleatórios; codificação da técnica de geração de números aleatórios em linguagem computacional, neste caso em VBA (*Visual Basic for Aplications*); apresentação de dados com base na geração de dados aleatórios conforme as distribuições e codificações específicas.

No geral, este material se destina a basicamente dois tipos de usuários do método. Um usuário casual, em que tudo o que necessita é executar o método em um software de planilha e tomar decisões, assim esse pode não estar completamente ciente de como o modelo funciona internamente e pode não ter necessidade de saber. Outros podem querer, ou ainda, dado uma escolha metodológica, necessitam saber mais sobre como o modelo faz o que faz, precisando compreender como é codificado seus próprios modelos de simulação, os fundamentos, a base matemática dos modelos de simulação, o modo como os números pseudo-aleatórios são gerados, e ainda, como eles são usados para gerar variáveis aleatórias que vêm de distribuições de probabilidade, discretas ou contínuas.

1.1 BREVE HISTÓRICO SOBRE SMC

A Simulação de Monte Carlo (SMC) é um processo estatístico com base em amostragens aleatórias na geração de resultados, onde o analista constrói um modelo matemático que simula um sistema real. No geral, um número de amostragens aleatórias é aplicado ao modelo, produzindo um significativo número de resultados. (THOMOPOULOS, 2013; BRANDIMARTE, 2014)

De modo simplista, o método se baseia na execução do modelo do sistema muitas vezes, por meio de amostragem aleatória. Para cada amostra, variáveis aleatórias são geradas em cada variável de entrada; os cálculos são executados através do modelo, produzindo resultados aleatórios para cada variável de saída. Uma vez que cada entrada é aleatória, os resultados são aleatórios. Do mesmo jeito, eles geraram milhares dessas amostras e alcançaram milhares de resultados para cada variável de saída. (THOMOPOULOS, 2013).

Uma das primeiras utilizações do método está relacionada com técnica utilizada por Buffon para estimar o valor de Pi (π). Neste processo uma agulha é lançada várias vezes em um piso consistindo em tiras de madeira e se observa o número de vezes que a agulha atravessa a borda. No processo desenvolvido por Buffon percebe-se que um problema puramente determinístico, pode ser resolvido usando o MMC como uma ferramenta de estimação.

Ademais, na década de 1930, Enrico Fermi experimentou pela primeira vez o método com o estudo da a difusão de nêutrons, mas não publicou o trabalho. Alguns anos depois, grande impulso do método ocorreu no projeto Manhatan, período entre 1930 e 1950, no Laboratório Nacional de Los Alamos, onde Stanislaw Ulam e John Von Neumann utilizaram do método no desenvolvimento da bomba nuclear, mais propriamente para resolver problemas relacionados à física[1] (Thomopoulos, 2013; Brandimarte, 2014).

Stanislaw Ulam teve a idéia de utilização do método no processo de cálculo da probabilidade de ganhar um jogo de cartas de paciência, que ocorreu durante a sua recuperação de uma cirurgia, que para ele era considerado um combinatório intratável. Ulam pensou em jogar centenas de jogos para estimar estatisticamente a probabilidade de um resultado bem-sucedido. Conforme ele mesmo menciona, relatado por Eckhardt (1987, p. 131):

> Os primeiros pensamentos e tentativas que fiz para praticar [o Método de Monte Carlo] foram sugeridos por uma pergunta que me ocorreu em 1946 quando eu estava convalescendo de uma doença e jogando paciência. A questão era quais são as chances de que um jogo de paciência com 52 cartas obtivesse sucesso? Depois de passar muito tempo tentando estimá-los por meio de cálculos combinatórios puros, me perguntei se um método mais prático do que o "pensamento abstrato" não seria expô-lo, digamos, cem vezes e simplesmente observar e contar o número de jogadas bem-sucedidas. Isso já era possível imaginar com o início da nova era de computadores rápidos, e eu imediatamente pensei em problemas de difusão de nêutrons e outras questões de física matemática e, de forma mais geral, como mudar os processos descritos por certas equações diferenciais em uma forma equivalente interpretável como uma sucessão de operações aleatórias. Mais tarde [em 1946], descrevi a ideia para John von Neumann e começamos a planejar cálculos reais.

[1] O filme O Matemático, no original *Adventures of a Mathematician*, apresenta o ponto de vista de Stanislaw Ulam, matemático polonês que participou do desenvolvimento das primeiras bombas nucleares para os Estados Unidos durante a Segunda Guerra Mundial, enquanto trabalhava na base de Los Alamos no Projeto Manhattan.

A solução proposta foi adotada por John Von Neumann para implementação com aplicações diretas à difusão do nêutron em certos materiais, em comunhão com o advento do primeiro computador digital, o ENIAC (*Electronic Numerical Integrator and Computer*), em fevereiro de 1946.

Por questões de sigilo, o projeto de simulação de sistemas complexos necessitava de um codinome, assim o termo Método de Monte Carlo foi cunhado por Nicholas Metropolis, em referência aos Casinos de Monte Carlo, em Mônaco, dado a característica de aleatoriedade dos resultados que é atribuída aos tipos de jogos de azar presentes em casinos e onde o tio de Ulam (Michał Ulam) costumava jogar.

Em 1947 Ulam e Von Neumann desenvolveram técnicas de inversão e aceitação-rejeição, para simular amostras de distribuições não uniformes. Em 1949, um simpósio sobre Monte Carlo foi apoiado pela empresa *Rand Organization*, *National Bureau of Standards*, e o laboratório de *Oak Ridge*, onde Metropolis e Ulam, publicaram o primeiro artigo sobre o método de Monte Carlo. (Chuanhai e Carroll, 2010)

Durante e após a Segunda Guerra Mundial, o método foi continuamente aprimorado e se tornou uma ferramenta proeminente no desenvolvimento do hidrogênio nuclear. A Rand Corporation e a Força Aérea dos EUA foram duas das principais organizações que financiavam e divulgavam informações sobre o uso do Método de Monte Carlo (Thomopoulos, 2013).

No processo de aplicação do método uma significativa quantidade de números aleatórios é necessária. No início, muitas maneiras foram utilizadas para a geração de números aleatórios, como lançamento de moedas, roletas, cartas embaralhadas. Entretanto, métodos físicos não são práticos quanto necessita-se uma quantidade significativa de números, como por exemplo 50.000 (cinquenta mil) números aleatórios.

Em 1946, a Rand Corporation gerou números aleatórios usando um dispositivo do tipo roleta eletrônica conectado a um computador, qual deu origem ao livro *A Million Random Digits with 100,000 Normal Deviates,* publicado em 1955, que foi considerado por muito tempo a "bíblia" de analistas (caso tenha se interessado em saber mais sobre este livro, sugiro um link[2]). (Thomopoulos, 2013)

Von Neumann também desenvolveu uma maneira de calcular números pseudo-aleatório, usando-se do método do quadrado do meio (?). Mas com o advento dos computadores de alta velocidade, algoritmos matemáticos tornaram-se práticos, novas técnicas levaram a melhoria no processo de gerar um grande fluxo de números *pseudo*-aleatórios. (Thomopoulos, 2013)

Após o período da Segunda Guerra Mundial, o processo de simulação adentrou em outros domínios, incluindo pesquisa operacional e economia. Atualmente, o método de Monte Carlo é extremamente difundido em vários tipos de situações, incluindo negócios, engenharias, ciências e finanças (Brandimarte, 2014). Claro que o desenvolvimento de computadores foram fundamentais para sua propagação, não somente pela necessidade de geração de número aleatórios, mas pela simulação do modelo como um todo. (Thomopoulos, 2013)

As linguagens de computador, como COBOL, FORTRAN, Basic, Visual Basic, JAVA e C ++, permitiram que programadores escrevam códigos para inúmeras aplicações e que viabilizaram o desenvolvimento de modelos de simulação de computador junto à prática gerencial. À medida que a simulação se tornou mais popular na solução de problemas complexos nos negócios, um novo conjunto de soluções começou a ser apresentados, como SAS, SPSS, R, @Risk, Cristal Ball, entre outros; permitindo que o usuário construa o processo e desenvolva um resumo estatístico, executando análises de dados. (Thomopoulos, 2013)

---

[2] https://www.wsj.com/articles/rand-million-random-digits-numbers-book-error-11600893049

1.2 DESENVOLVIMENTO DO TRABALHO

Na busca de apresentar uma solução de codificação para utilização do Método de Monte Carlo junto à softwares de planilhas, primeiramente, foi realizado um estudo sobre o processo de geração de números aleatórios para cada distribuição. Para isso, foram seguidos os métodos delineados por Thomopoulos (2013). Por exemplo, o material descreve o seguinte procedimento para gerar uma variável aleatória com Distribuição Bernoulli (x):

1. Gere uma variável aleatória uniforme $u \sim U(0; 1)$;
2. Se $u < p$ p, a variável aleatória (x) com Distribuição Bernoulli será $x = 1$; se não, $x = 0$.

Assim, com base neste procedimento, foi desenvolvido uma codificação para cada distribuição abordada neste trabalho.

A geração de números aleatórios foi testada em um software de planilhas, onde foram gerados cinquenta mil números aleatórios para cada distribuição, e sua adequação foi analisada com base nas propriedades esperadas de cada distribuição. Por exemplo, para uma Distribuição Triangular com parâmetros mínimo de 20, mais provável de 40, e máximo de 60 (Distribuição_Triangular(20; 40; 60)), foram gerados 50.000 valores aleatórios. O valor esperado para essa distribuição é 40, com uma variância de 66,67, resultados que foram confirmados pela simulação. Todas as codificações desenvolvidas neste trabalho mostraram-se consistentes e adequadas para a aplicação proposta.

2 PROCESSO DE APLICAÇÃO DO MMC

A aplicação do Monte Carlo trabalha com variáveis de entrada, variáveis de saída e um modelo matemático, onde o sistema de computador alimenta variáveis independentes no modelo matemático, simula-as e produz variáveis dependentes. Em suma, a simulação opera com variáveis de entrada, saída e um modelo de compreensão da realidade.

As variáveis de entrada são valores que exercem impacto sobre os resultados. Estas variáveis de entradas devem representar um dado tipo de distribuição probabilística. Por exemplo, na durabilidade de um smartphone, fatores como qualidade de fabricação e temperatura são variáveis de entrada influenciando diretamente esse resultado; na lucratividade de um segmento, mudanças nas receitas, e/ou custos, despesas são aspectos que impactam na lucratividade.

A variável de saída, ou ainda, variáveis, são resultados derivado da análise de Monte Carlo. Por exemplo, a expectativa de vida de um dispositivo eletrônico representa uma variável de saída, tendo como valor um intervalo temporal, como seis meses ou dois anos. O modelo trada-se de uma forma de representar a relação entre as variáveis de saída e de entrada de forma matemática. Por exemplo, o modelo matemático para lucro é expresso como $lucro = receita - despesas$.

Assim, o processo de simulação de Monte Carlo consiste na aplicação de distribuições de probabilidades das variáveis consideradas incertas (variáveis de entrada), sendo que cada uma dessas variáveis tem um valor aleatório dentro de sua distribuição de probabilidade, gerando, assim, combinações que levam a resultados (variáveis de saída), dado o modelo a ser considerado, permitindo assim determinar o risco associado a certa escolha. (MENDONÇA, 2009).

De forma mais sistemática, alguns autores oferecem uma detalhada descrição do processo de avaliação de riscos em várias etapas. Seguindo a abordagem de Rees (2015), é possível visualizar um caminho claro. Primeiramente, a identificação dos principais riscos e incertezas é essencial para orientar as próximas atividades. Em seguida, o mapeamento dos riscos e seus impactos é crucial para refleti-los em modelos quantitativos, particularmente em abordagens de avaliação de riscos quantitativos. Destacar os riscos mais significativos facilita decisões e comunicação, considerando trocas entre custo e benefício das medidas de mitigação ou devido a limitações de recursos. Na fase de mitigação e resposta, os planos originais são adaptados para gerenciar, mitigar ou explorar riscos, levando em conta o surgimento de novos riscos como resultado das ações de mitigação. Em seguida, uma análise dos riscos e incertezas remanescentes é realizada para escolher um plano final ou uma opção de decisão modificada, alinhada às tolerâncias de risco dos tomadores de decisão. Finalmente, a atualização contínua da avaliação de riscos, gerenciamento e controle da execução do projeto é crucial para monitorar e controlar os riscos envolvidos.

Para Rogers, Santos e Lemes (2008), o processo de simulação pelo método de Monte Carlo é conduzido definindo-se as variáveis onde ocorrerão os choques de aleatoriedade e atribuindo a cada uma uma distribuição de probabilidade. Após a identificação de cada variável, inicia-se o processo de geração de números aleatórios, com valores para as variáveis de entrada (inputs), o que permite calcular automaticamente os valores das variáveis de saída (outputs), representando cenários prováveis de ocorrência.

Segundo Noronha (1987), citado por Mendonça *et al.* (2009), o método de simulação de Monte Carlo pode ser dividido em cinco etapas. Primeiramente, a análise de sensibilidade verifica o efeito de variações em cada variável sobre os principais indicadores de viabilidade, visando identificar as variáveis mais relevantes do projeto. Em seguida, busca-se determinar a distribuição de probabilidades de cada variável relevante do fluxo de caixa do projeto. Posteriormente, ocorre a seleção aleatória de um valor para cada variável, seguida pelo cálculo dos indicadores de viabilidade. Por fim, o processo é repetido até que se obtenha uma confirmação adequada das distribuições de probabilidade dos indicadores.

No geral, podemos definir que o processo de aplicação do SMC deve contemplar:

1. Definição do problema e modelo de análise: onde é crucial entender o problema em questão e desenvolver um modelo analítico que represente adequadamente o sistema ou processo a ser estudado;
2. Identificação das variáveis de entrada e suas distribuições de probabilidade: as variáveis de entrada, que influenciam o resultado do modelo, são identificadas e suas distribuições de probabilidade são determinadas com base em dados históricos, experiência especializada ou outros métodos, como em testes de aderência;
3. Configurar e testar o sistema/software para o processo de simulação: o sistema ou software utilizado para realizar a simulação Monte Carlo é configurado de acordo com o modelo desenvolvido, e é importante realizar testes para garantir que esteja funcionando corretamente;
4. Criar um conjunto de dados de amostra: um grande número de dados de entrada é gerado aleatoriamente, seguindo as distribuições de probabilidade especificadas para cada variável de entrada;
5. Analisar os resultados: os resultados são analisados para entender a variabilidade e as características do sistema em estudo, que pode incluir a identificação de tendências, padrões ou eventos raros;
6. Interação e ajustes de mitigação: podem ser feitos ajustes no modelo ou no processo real para mitigar riscos identificados ou melhorar o desempenho, onde a interação pode envolver revisão do modelo, refinamento das distribuições de probabilidade ou implementação de estratégias de mitigação de risco.

Essas etapas fornecem uma estrutura geral para aplicar o Método de Monte Carlo de forma eficaz em uma variedade de contextos

3 DISTRIBUIÇÕES DE PROBABILIDADE E SMC

A distribuição de frequência pode ser pensada como o agrupamento de dados em classes, onde se contabiliza o número de ocorrência dos dados e busca a apresentação de dados de uma maneira mais concisa, permitindo explorar informações acerta de seu comportamento.

Devido ao tipo de dado, as distribuições de probabilidade podem ser classificadas em dois grandes grupos: distribuições contínuas e distribuições discretas. As Distribuições Contínuas ocorrem quando a variável que está sendo medida é expressa em uma escala contínua, como no caso de uma característica dimensional. As Distribuições Discretas se dão quando a variável que está sendo medida só pode assumir certos valores, como por exemplo os valores inteiros: 0, 1, 2, etc.

Neste trabalho aborda-se as seguintes Distribuições Discretas: Distribuição Discreta Uniforme, Distribuição Bernoulli, Distribuição Binomial, Distribuição Hipergeométrica, Distribuição Geométrica e Distribuição Poisson; e as seguintes Distribuições Contínuas: Distribuição Uniforme Contínua, Distribuição Exponencial, Distribuição Erlang, Distribuição Gama, Distribuição Beta, Distribuição Weibull, Distribuição Normal, Distribuição LogNormal, Distribuição Qui-quadrado, Distribuição Tstudent, e Distribuição Triangular. A Figura 3 apresenta um resumo de cada distribuição codificada.

Figura 1: Descrição dos tipos de distribuição codificadas

Distribuição	Tipo	Descrição	Parâmetro
Triangular	Contínua	Utilizada quando a distribuição verdadeira é desconhecida, mas se tem uma estimativa dos valores mínimo, máximo e mais provável. Tem formato triangular.	(mínimo, m provável, máximo)
Uniforme Contínua	Contínua	Utilizada quando todos os valores dentro de um intervalo possuem igual probabilidade de ocorrer.	(mínimo, máximo)
Exponencial	Contínua	Representa o tempo de espera até a primeira ocorrência de um evento em um processo com taxa constante. É amplamente usada para modelar tempos de espera entre eventos independentes e aleatórios.	(taxa λ)
Erlang	Contínua	É uma generalização da distribuição exponencial, usada para modelar o tempo de espera até a ocorrência de um número fixo de eventos. Comumente utilizada em filas e processos de atendimento.	(taxa λ, núr de eventos
Gamma	Contínua	Generalização da distribuição Erlang, utilizada para modelar tempos de espera ou tempos de vida, com parâmetros que permitem ajustar a forma da distribuição.	(forma k, es θ)
Beta	Contínua	Usada para modelar variáveis limitadas a um intervalo específico, como proporções. Frequente em modelagem de probabilidades e percentuais.	(alfa α, beta
Weibull	Contínua	Usada para modelar tempos de vida ou tempos até a ocorrência de um evento, com flexibilidade para representar diferentes taxas de falha (crescentes, decrescentes ou constantes) ao longo do tempo.	(escala λ, fo k)
Normal	Contínua	Distribuição amplamente utilizada que surge naturalmente em muitos fenômenos devido ao Teorema Central do Limite. Modela variáveis cujos valores tendem a se agrupar em torno de uma média.	(média μ, desvio padr σ)
Log-Normal	Contínua	Modela uma variável cujo logaritmo tem uma distribuição normal. É usada quando os valores da variável são produtos de muitos fatores pequenos independentes.	(média μ, desvio padr da log-norm
Chi-quadrado	Contínua	Distribuição derivada da soma dos quadrados de variáveis normais independentes. Usada principalmente em testes de hipóteses, como o teste de independência e o teste de adequação.	(graus de liberdade k
T-Student	Contínua	Usada para estimar a média de uma população normalmente distribuída quando o tamanho da amostra é pequeno e a variância populacional é desconhecida. A distribuição é semelhante à normal, mas com caudas mais largas.	(graus de liberdade ν

Uniforme Discreta	Discreta	Modela variáveis inteiras onde todos os valores entre um mínimo e um máximo têm igual probabilidade de ocorrer.	(mínimo, máximo)
Bernoulli	Discreta	Modela um experimento binário de único ensaio, com apenas dois resultados possíveis: sucesso (1) ou fracasso (0). É a base para outras distribuições discretas, como a Binomial.	(probabilida de sucesso
Binomial	Discreta	Modela o número de sucessos em uma sequência de ensaios de Bernoulli independentes, todos com a mesma probabilidade de sucesso.	(número de ensaios n, probabilida de sucesso
Hipergeométrica	Discreta	Modela o número de sucessos em amostras retiradas sem reposição de uma população finita. É útil quando a amostra representa uma porção significativa da população total.	(tamanho da população N número de sucessos na população K tamanho da amostra n)
Geométrica	Discreta	Modela o número de tentativas até o primeiro sucesso em uma série de ensaios de Bernoulli independentes. É frequentemente usada para modelar cenários de "até que" um evento ocorra.	(probabilida de sucesso
Poisson	Discreta	Modela o número de eventos que ocorrem em um intervalo fixo de tempo ou espaço, quando os eventos ocorrem com uma taxa constante e independentemente uns dos outros. É uma extensão da Binomial para grandes n e pequenas p.	(taxa de ocorrência

Fonte: elaborado pelos autores.

3.1 CODIFICAÇÕES PARA SMC

A figura 4 traz códigos para funções em VBA (Virtual Basic for Applications) para a geração de uma variável discreta uniforme em software de planilhas. O código gera uma função pública (Public Function), por exemplo, definida por "Distribuição_Uniforme_Discreta(Menor, Maior)", acessível a todos os outros procedimentos nos módulos do software de planilhas. As codificações seguiram os delineamentos de Thomopoulos (2013) para a geração de números aleatórios em cada distribuição.

As codificações criam *public functions*. No geral as *function* em VBA podem ser usadas nas células da planilha para fazer cálculos especializados não programados nas funções existentes no Microsoft Excel, caso esse aplicado no presente trabalho. Basicamente criam uma nova "formula". Ou seja, utilizou-se da programação em VBA para construção de Funções de Simulação de Números aleatórios de vários tipos de distribuição de frequência

Figura 2: Descrição dos tipos de distribuição codificadas

Distribuição Uniforme Discreta

```
Public Function Distribuição_Uniforme_Discreta(Menor As Double, Maior As Double) As Integer
   Application.Volatile
      If (Maior < Menor) Then
      Distribuição_Uniforme_Discreta = "Erro"
   Else
      Distribuição_Uniforme_Discreta = Application.WorksheetFunction.RoundUp((Menor - 1) + Rnd *
(Maior - Menor + 1), 0)
   End If
End Function
```

Distribuição Bernoulli

```
Public Function Distribuição_Bernoulli(Probabilidade As Double) As Double
   Dim Alea As Double
   Application.Volatile
   If (Probabilidade < 1) Then
      Alea = Rnd
      If Alea < Probabilidade Then
         Distribuição_Bernoulli = 1
      Else
         Distribuição_Bernoulli = 0
      End If
   Else
      Distribuição_Bernoulli = "Erro"
   End If
End Function
```

Distribuição Binomial

```
Public Function Distribuição_Binomial(N_sorteio As Integer, P_sucesso As Double) As Integer
   Dim Counter As Integer
   Dim Xset As Integer
   Application.Volatile
   If (N_sorteio <= 10 And P_sucesso <= 0.5) Then
      Counter = 0
      Xset = 0
      Do Until Counter = N_sorteio
         Counter = Counter + 1
         If Rnd < P_sucesso Then
            Xset = Xset + 1
         Else
            Xset = Xset + 0
         End If
      Loop
      Distribuição_Binomial = Xset
```

```
    ElseIf (P_sucesso <= 0.5 And (N_sorteio * P_sucesso) > 5) Or (P_sucesso > 0.5 And (N_sorteio * (1 -
P_sucesso)) > 5) Then
        Distribuição_Binomial = Int(Abs((N_sorteio * P_sucesso) + (Sqr((-2 * (Log(Rnd)))) * Cos(2 *
Application.WorksheetFunction.Pi * Rnd)) * Sqr(((N_sorteio * P_sucesso) * (1 - P_sucesso))) + 0.5))
    Else
        Counter = 0
        Xbset = 0
        Do While Xbset <= 1
            Counter = Counter + 1
            Xbset = Xbset + (-1 / (N_sorteio * P_sucesso)) * (Log(1 - Rnd))
        Loop
        Distribuição_Binomial = Counter - 1
    End If
End Function
```

Distribuição Hipergeométrica

```
Public Function Distribuição_HiperGeometrica(Pop As Integer, n As Integer, d As Integer) As Integer
Application.Volatile
Do Until Counter = n
    Dim Xset As Integer
    Dim p As Double
    p = d / (Pop - Counter)
    Counter = Counter + 1
    If (Rnd >= p) Then
        Xset = Xset + 0
        Else
        Xset = Xset + 1
        d = d - 1
    End If
Loop
Distribuição_HiperGeometrica = Xset
End Function
```

Distribuição Geométrica

```
Function Distribuição_Geométrica_Suc(Probabilidade As Double) As Double
    Application.Volatile
    Distribuição_Geométrica_Suc = Int(((Log((1 - Rnd)) / Log((1 - Probabilidade))))) + 1
End Function
```

Obs.: Quando a variável é definida em número de falhas até obter um sucesso, a variável é $x' = x - 1$

```
Function Distribuição_Geométrica_Fal(Probabilidade As Double) As Double
    Application.Volatile
    Distribuição_Geométrica_Fal = Int(((Log((1 - Rnd)) / Log((1 - Probabilidade))))) - 1
End Function
```

Distribuição de Poisson

```
Public Function Distribuição_Poisson(Lambda As Double) As Integer
    Dim Counter As Integer
    Dim Xbset As Double
    Application.Volatile
    If (Lambda <= 0) Then
        Distribuição_Poisson = "Erro"
    Else
        Counter = 0
        Xbset = 0
        Do While Xbset <= 1
            Counter = Counter + 1
            Xbset = Xbset + (-1 / Lambda) * (Application.WorksheetFunction.Ln(1 - Rnd))
        Loop
        Distribuição_Poisson = Counter - 1
    End If
End Function
```

Distribuição Uniforme Contínua

```
Public Function Distribuição_Uniforme_Continua(Mínimo As Double, Máximo As Double) As Double
    Dim Alea As Double
    Application.Volatile
    If (Máximo > Mínimo) Then
        Alea = Rnd
        Distribuição_Uniforme_Continua = Mínimo + Alea * (Máximo - Mínimo)
    Else
        Distribuição_Uniforme_Continua = "Erro"
    End If
End Function
```

Distribuição Exponencial

```
Public Function Distribuição_Exponencial(beta As Double) As Double
    Dim Alea As Double
    Application.Volatile
    If (beta > 0) Then
        Alea = Rnd
        Distribuição_Exponencial = -beta * (Application.WorksheetFunction.Ln(1 - Alea))
    Else
        Distribuição_Exponencial = "Erro"
    End If
End Function
```

Distribuição Erlang

```
Public Function Distribuição_Erlang(K As Double, Média As Double)
    Dim Counter As Integer
    Dim Xbset As Double
    Application.Volatile
    If (K <= 0) Then
        Distribuição_Erlang = "Erro"
    Else
        Counter = 0
        Xbset = 0
        Do While Counter < K
            Counter = Counter + 1
            Xbset = Xbset + (-1 / (K / Média)) * (Log(1 - Rnd))
        Loop
        Distribuição_Erlang = Xbset
    End If
End Function
```

Distribuição Gama

```
Public Function Distribuição_Gama(Media As Double, Variancia As Double) As Double
    Application.Volatile
    Tetha = Media / Variancia
    K = Media * Tetha
    If (K < 1) Then
        Line8:
        Alea1 = Rnd
        Alea2 = Rnd
        e = (2.718282 + K) / 2.71828
        p = e * Alea1
        y = p ^ (1 / K)
        y2 = -Log(((e - p) / K))
        If (p <= 1 And Alea1 <= e ^ (-y)) Then
            Distribuição_Gama = y / Tetha
        ElseIf (p > 1 And Alea2 <= y2 ^ (K - 1)) Then
            Distribuição_Gama = y2 / Tetha
        Else: GoTo Line8
        End If
```

```
    ElseIf (K > 1) Then
        a = 1 / Sqr((2 * K - 1))
        b = K - Log(4)
        q = K + 1 / a
        c = 4.5
        d = 1 + Log(4.5)
        LineGama2:
        Alea1 = Rnd
        Alea2 = Rnd
        v = a * Log((Alea1 / (1 - Alea1)))
        y = K * 2.71828 ^ v
        z = Alea1 ^ (2) * Alea2
        w = b + q * v - y
        If ((w + d - c * z) >= 0) Then
            Distribuição_Gama = y / Tetha
        ElseIf ((w + d - c * z) < 0 And w >= Log(z)) Then
            Distribuição_Gama = y / Tetha
        Else: GoTo LineGama2
        End If
    ElseIf (K = 1) Then
        Distribuição_Gama = -Media * (Application.WorksheetFunction.Ln(1 - Alea))
    End If
End Function
```

Distribuição Beta

```
Public Function Distribuição_Beta(k1 As Double, k2 As Double, minimo As Double, maximo As Double) As Double
Application.Volatile
If (k1 < 1) Then
Line81:
        Alea1 = Rnd
        Alea2 = Rnd
        e = (2.718282 + k1) / 2.71828
        p = e * Alea1
        y = p ^ (1 / k1)
        y2 = -Log(((e - p) / k1))
        If (p <= 1 And Alea1 <= e ^ (-y)) Then
        Gama1 = y / 1
        ElseIf (p > 1 And Alea2 <= y2 ^ (k1 - 1)) Then
        Gama1 = y2 / 1
        Else: GoTo Line81
        End If
ElseIf (k1 > 1) Then
        a = 1 / Sqr((2 * k1 - 1))
        b = k1 - Log(4)
        q = k1 + 1 / a
        c = 4.5
        d = 1 + Log(4.5)
LineGama21:
        Alea1 = Rnd
        Alea2 = Rnd
        v = a * Log((Alea1 / (1 - Alea1)))
        y = k1 * 2.71828 ^ v
        Z = Alea1 ^ (2) * Alea2
        w = b + q * v - y
        If ((w + d - c * Z) >= 0) Then
        Gama1 = y / 1
        ElseIf ((w + d - c * Z) < 0 And w >= Log(Z)) Then
        Gama1 = y / 1
```

```
        Else: GoTo LineGama21
        End If
ElseIf (k1 = 1) Then
        Gama1 = -k1 * (Log(1 - Rnd))
End If
If (k2 < 1) Then
Line82:
        Alea1 = Rnd
        Alea2 = Rnd
        e = (2.718282 + K) / 2.71828
        p = e * Alea1
        y = p ^ (1 / k2)
        y2 = -Log(((e - p) / k2))
        If (p <= 1 And Alea1 <= e ^ (-y)) Then
        Gama2 = y / 1
        ElseIf (p > 1 And Alea2 <= y2 ^ (k2 - 1)) Then
        Gama2 = y2 / 1
        Else: GoTo Line82
        End If
ElseIf (k2 > 1) Then
        a = 1 / Sqr((2 * k2 - 1))
        b = k2 - Log(4)
        q = k2 + 1 / a
        c = 4.5
        d = 1 + Log(4.5)
LineGama22:
        Alea1 = Rnd
        Alea2 = Rnd
        v = a * Log((Alea1 / (1 - Alea1)))
        y = k2 * 2.71828 ^ v
        Z = Alea1 ^ (2) * Alea2
        w = b + q * v - y
        If ((w + d - c * Z) >= 0) Then
        Gama2 = y / 1
        ElseIf ((w + d - c * Z) < 0 And w >= Log(Z)) Then
        Gama2 = y / 1
        Else: GoTo LineGama22
        End If
ElseIf (k1 = 1) Then
        Gama2 = -k2 * (Log(1 - Rnd))
End If
Beta_padrao = Gama1 / (Gama1 + Gama2)
Distribuição_Beta = minimo + Beta_padrao * (maximo - minimo)
End Function
```

Distribuição Weibull

```
Public Function Distribuição_Weibull(K1 As Double, K2 As Double) As Double
    Application.Volatile
    If (K1 < 0 And K2 < 0) Then
        Distribuição_Weibull = "Erro"
    Else
        Distribuição_Weibull = K2 * ((-1 * (Application.WorksheetFunction.Ln(1 - Rnd))) ^ (1 / K1))
    End If
End Function
```

Distribuição Normal

```
Public Function Distribuição_Normal(Média As Double, Desvio_Padrão As Double) As Double
    Dim Alea1 As Double
    Dim Alea2 As Double
    Application.Volatile
```

```
    If (Média > 0 And Desvio_Padrão > 0) Then
        Alea1 = Rnd
        Alea2 = Rnd
        Distribuição_Normal = Média + (((Sqr(-2 * (Log(Alea1)))) * (Cos(2 *
Application.WorksheetFunction.Pi() * (Alea2)))) * Desvio_Padrão)
    Else
        Distribuição_Normal = "Erro"
    End If
End Function
```

Distribuição LogNormal

```
Public Function Distribuição_LogNormal(Média As Double, Variância As Double) As Double
    Dim Alea1 As Double
    Dim Alea2 As Double
    Application.Volatile
    If (Média < 0 And Variância < 0) Then
        Distribuição_LogNormal = "Erro"
    Else
        Alea1 = Rnd
        Alea2 = Rnd
        Med_y = Application.WorksheetFunction.Ln(((Média ^ 2) / (Sqr(Média ^ 2 + Variância))))
        Var_y = Application.WorksheetFunction.Ln(1 + (Variância / (Média ^ 2)))
        Des_Pad_y = Sqr(Var_y)
        Z_Normal = ((Sqr(-2 * (Log(Alea1)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Alea2))))
        Distribuição_LogNormal = 2.718281828 ^ (Med_y + (Z_Normal * Des_Pad_y))
    End If
End Function
```

Distribuição Qui-quadrado

```
Public Function Distribuição_Chiquadrado(Graus_de_Liberdade) As Double
    Application.Volatile
    If (Graus_de_Liberdade > 30) Then
        Dim Alea1   As Double
        Dim Alea2   As Double
        Alea1 = Rnd
        Alea2 = Rnd
        Z_normal = ((Sqr(-2 * (Log(Alea1)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Alea2))))
        Distribuição_Chiquadrado = Int((Graus_de_Liberdade + (Z_normal * Sqr(2 * 239)) + 0.5))
    Else
        Dim Counter As Integer
        Dim Xset    As Double
        Counter = 0
        Xset = 0
        Do Until Counter = Graus_de_Liberdade
            Counter = Counter + 1
            Xset = Xset + (((Sqr(-2 * (Log(Rnd)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Rnd)))) ^ 2)
        Loop
        Distribuição_Chiquadrado = Xset
    End If
End Function
```

Distribuição T de Student

```
Public Function Distribuição_Tstudent(Graus_de_Liberdade) As Double
    Application.Volatile
    If (Graus_de_Liberdade > 30) Then
        Z_normal = ((Sqr(-2 * (Log(Rnd)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Rnd))))
        X2 = Int((Graus_de_Liberdade + (Z_normal * Sqr(2 * 239)) + 0.5))
    Else
        Dim Counter As Integer
        Dim Xset    As Integer
        Counter = 0
```

```
        Xset = 0
        Do Until Counter = Graus_de_Liberdade
            Counter = Counter + 1
            Xset = Xset + (((Sqr(-2 * (Log(Rnd)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Rnd)))) ^ 2)
        Loop
        X2 = Xset
    End If
    Z_normal = ((Sqr(-2 * (Log(Rnd)))) * (Cos(2 * Application.WorksheetFunction.Pi() * (Rnd))))
    Distribuição_Tstudent = Z_normal / (Sqr((X2 / Graus_de_Liberdade)))
End Function
```

Distribuição Triangular

```
Public Function Distribuição_Triangular(Mínimo As Double, Mais_Provável As Double, Máximo As
Double) As Double
    Dim Alea As Double
    Application.Volatile
    If (Mínimo < Mais_Provável And Mais_Provável < Máximo) Then
        Alea = Rnd
        If Alea <= ((Mais_Provável - Mínimo) / (Máximo - Mínimo)) Then
            Distribuição_Triangular = Mínimo + ((Sqr(Alea * ((Mais_Provável - Mínimo) / (Máximo -
Mínimo)))) * (Máximo - Mínimo))
        Else
            Distribuição_Triangular = Mínimo + (1 - Sqr((1 - (Mais_Provável - Mínimo) / (Máximo - Mínimo)) *
(1 - Alea))) * (Máximo - Mínimo)
        End If
    Else
        Distribuição_Triangular = "Erro"
    End If
End Function
```

Fonte: Elaborado pelos autores

5 CONSIDERAÇÕES E CONTRIBUIÇÕES DO TRABALHO

Este trabalho oferece uma contribuição para a prática da simulação Monte Carlo, apresentando uma solução de codificação para sua implementação em softwares de planilhas, utilizando o Visual Basic for Applications (VBA). Primeiramente destacamos a aderência à área de concentração do programa de pós-graduação no qual os autores estão vinculados. Pois, no geral, o estudo proporciona uma ferramenta para a modelagem e análise de sistemas complexos, alinhada com as necessidades de gestão e tomada de decisão em ambientes organizacionais.

Ao seguir os processos detalhados por Thomopoulos (2013) para a geração de números aleatórios em diversas distribuições, este trabalho não apenas amplia a compreensão sobre a aplicação do Método de Monte Carlo, também oferece um recurso acessível e replicável para profissionais e acadêmicos. A abordagem abrangente de diferentes distribuições, tanto discretas quanto contínuas proporciona uma base robusta para a utilização SMC em cenários reais, onde a complexidade dos sistemas demanda abordagens de modelagem mais sofisticadas do que aquelas oferecidas por métodos analíticos tradicionais.

O impacto deste trabalho se destaca não somente no ambiente acadêmico, mas também no organizacional. Em sala de aula, a aplicabilidade realizada já foi testada, permitindo que estudantes e professores utilizem a ferramenta para simular cenários e compreender melhor os processos de tomada de decisão. No ambiente organizacional, o impacto potencial é igualmente significativo, especialmente em áreas onde a simulação pode auxiliar na previsão de riscos, otimização de processos e análise de incertezas.

A aplicabilidade do produto é alta, com um grau de facilidade considerável na sua implementação e uso. A ferramenta desenvolvida pode ser facilmente replicada em diferentes contextos, com a complexidade média necessária para a adaptação e personalização do modelo conforme as especificidades de cada organização ou projeto. A replicabilidade irrestrita deste trabalho garante que sua utilização possa ser amplamente difundida, beneficiando uma variedade de setores e contribuindo para o aprimoramento das práticas de modelagem e tomada de decisão.

Por fim, o trabalho atende às diversas necessidades e níveis de conhecimento dos usuários, desde aqueles que desejam utilizar a ferramenta de simulação de forma prática, sem aprofundar-se na codificação, até aqueles que buscam compreender os detalhes técnicos. Tornando-a acessível e compreensível, e, assim, contribui para o avanço do conhecimento e aprimoramento das práticas de gestão em diferentes áreas de aplicação.

REFERÊNCIAS

Brandimarte, P. (2014). *Handbook in Monte Carlo Simulation: Applications in Financial Engineering, Risk Management, and Economics.* Wiley.

De Lucas, P. I. J., Santos, D. F. L., Sabbag, O. J., & Queiroz, T. R. (2022). Análise de viabilidade econômica para a renovação de equipamentos na produção da cana-de-açúcar: um estudo de caso nas operações de tratos culturais. *Revista Capital Científico – Eletrônica (RCCe)*, 20(1), 1-19. https://doi.org/10.5935/2177-4153.20220005

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo Method. *Los Alamos Science Special Issue*, 131-143.

Law, A. M. (2015). *Simulation Modeling and Analysis (15th ed.)*. McGraw-Hill Education.

Mendonça, T. G., Lírio, V. S., Moura, A. D., Reis, B. S., & Silveira, S. F. R. (2008). Avaliação da viabilidade econômica da produção de mamão em sistema convencional e de produção integrada de frutas. *Organizações Rurais & Agroindustriais*, 10(3), 405-419.

Metropolis, N. (1987). The Beginning of the Monte Carlo Method. *Los Alamos Science Special Issue*.

Rees, M. (2015). *Business Risk and Simulation Modelling in Practice: Using Excel, VBA and @RISK*. Wiley.

Rogers, P., Santos, E. J., & Lemes, S. (2008). Precificação em empresas comerciais: um estudo de caso aplicando o custeio variável através do método de monte carlo. *Revista da FAE*, 11(1), 55-67.

Saraiva Junior, A. F., Tabosa, C. M., & Costa, R. P. (2011). Simulação de Monte Carlo aplicada à análise econômica de pedido. *Produção*, 21(1), 149-164. [URL inválido removido]

Thomopoulos, N. T. (2013). *Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models*. Springer. https://doi.org/10.1007/978-1-4614-6022-0

Weick, K. E. (1995). *Sensemaking in organizations*. Thousand Oaks: Sage Publications.

Xei, G. (2020). A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time. *Scientific Reports*, 10, 13120. https://doi.org/10.1038/s41598-020-70091-1