Explorando o Jogo da Vida: Uma Abordagem Prática para Programação e Sistemas Distribuídos

Ana Beatriz M. Soares[1], Camille Vitoria S. Andrade[1], Eduardo Victor de O. Gonçalves[1]

[1]Instituto de Informática –Instituto Federal do Pará(IFPA)
Caixa Postal 15.064 –66093-020 – Belém – PA – Brazil

eduardo.vic.goncalves@gmail.com, camilleandrade125@gmail.com, anamotasoa@gmail.com

Abstract. *This article explores the Game of Life, a cellular automaton created by John Conway, as a tool to demonstrate and evaluate concepts of parallel programming and distributed systems. The game is characterized by simple rules that determine the evolution of cells in a two-dimensional grid, resulting in complex patterns from initial configurations. A Python implementation was developed, optimizing performance using parallel and distributed techniques, essential for dealing with the large volume of simultaneous operations required by the game on extensive grids. This work analyzes the impact of distributing tasks between multiple cores and machines, highlighting how the efficient division of processing can improve system performance. Additionally, the educational and scientific implications of using the Game of Life as a model for the study of emergent behavior and practical application in modern computing systems.*

Resumo. *O presente artigo explora o Jogo da Vida, um autômato celular criado por John Conway, como ferramenta para demonstrar e avaliar conceitos de programação paralela e sistemas distribuídos. O jogo é caracterizado por regras simples que determinam a evolução de células em uma grade bidimensional, resultando em padrões complexos a partir de configurações iniciais. Uma implementação em Python foi desenvolvida, otimizando o desempenho com o uso de técnicas paralelas e distribuídas, essenciais para lidar com o grande volume de operações simultâneas exigidas pelo jogo em grades extensas. Este trabalho analisa o impacto da distribuição das tarefas entre múltiplos núcleos e máquinas, evidenciando como a divisão eficiente do processamento pode melhorar a performance do sistema. Além disso, as implicações educacionais e científicas do uso do Jogo da Vida como um modelo para o estudo de comportamento emergente e a aplicação prática em sistemas computacionais modernos.*

1. Introdução

O *Jogo da Vida*, criado por John Conway, é um dos autômatos celulares mais conhecidos e serve como uma fascinante representação de como padrões complexos podem surgir a partir de regras simples. Este artigo propõe uma análise aprofundada do *Jogo da Vida* como uma ferramenta pedagógica e experimental para a compreensão de conceitos fundamentais em programação paralela e sistemas distribuídos. Por meio da evolução das células em uma grade bidimensional, o jogo oferece um ambiente ideal para estudar o comportamento emergente em sistemas computacionais. Dada a sua natureza de alto custo computacional em grandes escalas, uma implementação otimizada em Python foi desenvolvida, integrando técnicas paralelas e distribuídas para gerenciar eficientemente o grande volume de operações simultâneas necessárias. A

investigação aborda como a distribuição das tarefas entre múltiplos núcleos e máquinas impacta o desempenho do sistema, destacando a importância da divisão eficiente do processamento para alcançar um desempenho aprimorado. Além disso, o artigo explora as implicações educacionais e científicas do *Jogo da Vida*, ressaltando sua relevância como modelo para o estudo de sistemas dinâmicos complexos e sua aplicação prática no contexto de tecnologias de computação modernas.

2. Metodologia

A metodologia deste artigo é estruturada para investigar a aplicação do *Jogo da Vida* como uma ferramenta para demonstrar e avaliar conceitos de programação paralela e sistemas distribuídos. A seguir, são detalhados os principais passos e abordagens adotadas na realização deste estudo:

1. Pesquisas e literaturas de materiais teóricos e práticos relacionados aos conceitos de **Programação Paralela**, **Sistemas Distribuídos** e o **Jogo da Vida de Conway.**
2. O desenvolvimento foi conduzido utilizando a **Linguagem Python**, com implementação e testes realizados no ambiente de desenvolvimento **Visual Studio Code (VSCode)**.

Figura 1. Código em partes e Tela do Jogo

2.1. Regras do Jogo da Vida

Autômatos celulares podem ser implementados em uma, duas ou três dimensões, desde que as regras que governam seu comportamento sejam cuidadosamente definidas com base em experimentos detalhados. Essas regras determinam o estado futuro de uma

célula com base no estado atual de suas células vizinhas. No caso específico do "Jogo da Vida" de Conway, os vizinhos de uma célula são as oito células adjacentes a ela. Embora o tabuleiro do jogo seja finito na prática, ele é considerado teoricamente infinito, e as regras são aplicadas de forma análoga às células localizadas nas bordas, conforme ilustrado na Figura 1[parte 1].

Algumas configurações de regras podem levar à morte prematura de células, enquanto outras podem resultar no nascimento excessivo delas. O "Jogo da Vida" busca alcançar um equilíbrio entre essas duas tendências, criando uma dinâmica complexa e imprevisível. Por conta disso, é desafiador prever se uma determinada configuração inicial desaparecerá completamente ou crescerá de forma indefinida. As regras que regem o sistema foram definidas da seguinte maneira:

1. Uma célula morta com exatamente três vizinhos vivos torna-se viva (nasce Figura 1[parte 2]).
2. Uma célula viva permanece viva se possuir dois ou três vizinhos vivos (sobrevive – Figura 1[parte 3 e parte 4]).
3. Em todos os outros casos, as células morrem ou permanecem mortas, seja por superpopulação, apenas um vizinho ou solidão (Figura 1[parte 5]).

O jogo começa a partir de uma configuração inicial de células, onde as regras acima são aplicadas em ciclos chamados de "gerações" ou iterações. Cada célula viva no tabuleiro representa uma unidade da população, e o tempo avança discretamente à medida que essas interações ocorrem. O "Jogo da Vida" de Conway é considerado um modelo ecológico, pois a sobrevivência e a reprodução de cada célula dependem da distribuição espacial e da frequência das células vizinhas, obedecendo a regras claramente definidas. Embora também seja classificado como um modelo evolutivo, o termo deve ser usado com cautela, já que, em teoria, podem surgir padrões ou estruturas inesperadas e totalmente diferentes durante a simulação.

Essa abordagem destaca como padrões, estruturas e comportamentos complexos podem emergir a partir de regras simples. Um dos aspectos mais fascinantes do *Jogo da Vida* é que o estado de cada célula depende não apenas de suas próprias propriedades, mas também de um conjunto de outras células ao seu redor. Esse conceito se assemelha à Teoria dos Jogos, em que as ações de um indivíduo são influenciadas por decisões de outros, gerando padrões que emergem no nível macro.

Além disso, os processos observados no *Jogo da Vida* possuem uma estreita relação com os conceitos de Sistemas Distribuídos e Programação Paralela. Cada célula pode ser vista como um "agente" que age localmente, baseado apenas em informações de seus vizinhos imediatos, enquanto o comportamento global do sistema emerge da interação coletiva entre essas células. Essa característica reflete princípios fundamentais de sistemas distribuídos, onde diferentes partes de um sistema trabalham de forma independente e coordenada. Na programação paralela, esse modelo também é relevante, pois o cálculo do estado de cada célula em uma geração pode ser realizado simultaneamente, maximizando a eficiência computacional. Assim, o *Jogo da Vida*

exemplifica como sistemas baseados em regras locais podem produzir dinâmicas globais surpreendentemente complexas.

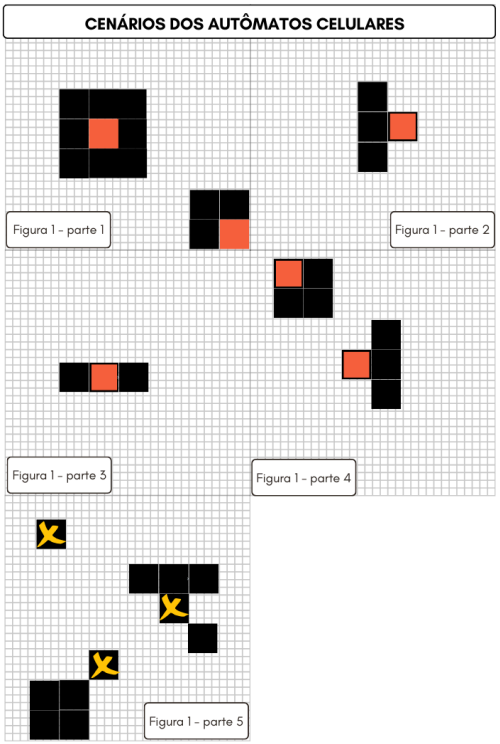

Figura 2. Cenários dos Autômatos Celulares

2.2 Paralelismo e Sistemas Distribuídos

Como foi descrito anteriormente, o *jogo da vida* de John Conway, seria um autômato celular que simula a evolução de uma grande bidimensional composta por células vivas e mortas, baseado em conjuntos simples de regras. Apesar de ser um jogo simples de se desenvolver, com o passar do tempo do desenvolvimento o jogo pode produzir padrões complexos e fascinantes. Com isso, à medida que o tamanho da grade aumenta, o número de cálculos necessários para determinar o próximo estado de cada célula cresce exponencialmente, tornando o processamento em um único núcleo de CPU lento e ineficiente.

O objetivo do paralelismo é aproveitar os recursos modernos de hardware, como processadores multicore, GPUs e clusters distribuídos, para acelerar os cálculos e permitir a simulação de grades maiores em menos tempo. Para conseguir implementar o paralelismo no jogo, requer abordar aspectos como sincronização, comunicação entre subgrids e particionamento eficiente do tabuleiro.

De acordo com o trabalho de Silva e Bittencourt (2020), a gerência de recursos em sistemas distribuídos é fundamental para aplicações que demandam alto desempenho, como jogos em tempo real e simulações. No contexto do *Jogo da Vida*, os princípios descritos para a gestão eficiente de recursos em sistemas distribuídos são:

[Particionamento da Grade] onde subgrades menores, atribuindo cada uma a um nó do sistema distribuído, permitindo que cada nó processe um parte da simulação de forma independente, reduzindo a carga em um único recurso computacional. [Sincronização entre Nós] os nós devem compartilhar as informações das células que estão nas bordas das subgrades. Isso garante a consciência global da simulação e a continuidade dos padrões gerados. [Balanceamento de Carga] onde o ambiente distribuído é essencial garantir que a carga de trabalho seja distribuída de maneira equilibrada entre os nós, evitando que alguns nós fiquem sobrecarregados enquanto outros permanecem subutilizados. [GPUs] processadores multicore, onde o uso pode acelerar o processamento, devido à capacidade de lidar com um grande número de operações em paralelo. Isso é especialmente relevante em simulações que envolvem milhares ou milhões de células.

Essas técnicas não apenas aceleram o processamento do *Jogo da Vida*, mas também permitem a simulação de padrões em escalas que seriam impraticáveis em sistemas centralizados. Ademais, os conceitos de comunicação eficiente e tolerância a falhas, também destacados no artigo de Silva e Bittencourt, são essenciais para garantir que a simulação seja robusta e escalável em sistemas distribuídos.

2.3 Interface Gráfica

A interface gráfica do *Jogo da Vida* foi desenvolvida utilizando a biblioteca **Tkinter**, proporcionando uma visualização interativa e intuitiva do comportamento das células ao longo das iterações. A seguir, detalhamos os principais componentes e funcionalidades dessa interface, além de sua interação com a simulação paralela do jogo.

Estrutura Básica da Interface

A interface é dividida em três áreas principais:

1. **Informações e Regras do Jogo**: Um painel superior exibe as regras do *Jogo da Vida*, incluindo as condições de sobrevivência, nascimento e morte das células. Essa explicação é exibida em formato de texto simples para garantir que o usuário entenda como o jogo funciona.
2. **Área de Exibição do Grid**: No centro da interface, um gráfico gerado pela biblioteca **Matplotlib** exibe a grade do *Jogo da Vida*. Cada célula da grade é representada por um pixel, onde células vivas são mostradas em preto e células mortas são transparentes. A visualização também inclui uma sobreposição de cores representando as regiões processadas por diferentes threads, oferecendo uma visão detalhada da execução paralela.
3. Botões que contém as ações necessárias para o usuário jogar.

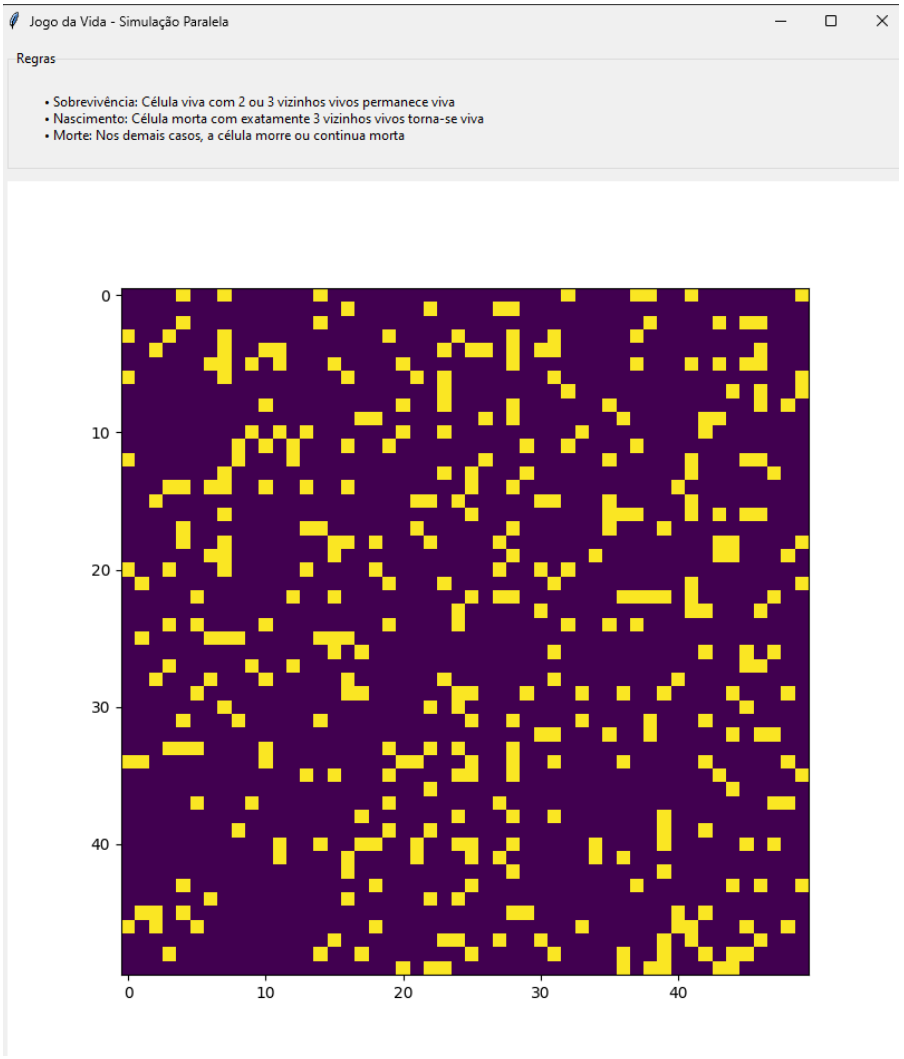

Figura 3 . Interface Gráfica - Inicialização

4.1 Resultados

A implementação do **Jogo da Vida** com a utilização de múltiplas threads mostrou-se eficaz na simulação do comportamento das células ao longo das gerações. O sistema foi projetado para dividir a carga de trabalho entre várias threads, o que resultou em um desempenho significativamente melhor em comparação a uma implementação sequencial, especialmente quando a grade é de tamanho grande.

A interface gráfica, construída com **Tkinter** e **Matplotlib**, apresentou uma visualização clara e interativa do estado das células, permitindo ao usuário acompanhar as mudanças em tempo real. As cores utilizadas para representar as regiões processadas por diferentes threads facilitam a compreensão do paralelismo e ajudam a monitorar o progresso das operações. O uso de cores distintas para cada thread também contribuiu para uma melhor visualização das divisões de carga, o que pode ser útil para fins de diagnóstico e otimização.

A implementação do controle de velocidade, por meio de um slider, permitiu ajustar a frequência de atualização da simulação, proporcionando flexibilidade ao usuário para observar o comportamento das células em diferentes ritmos. Os botões para iniciar, pausar, limpar e resetar a simulação também garantiram uma interface intuitiva e fácil de usar.

4.2 Discussão

A escolha de utilizar **ThreadPoolExecutor** para gerenciar as threads foi fundamental para garantir um controle eficiente sobre os recursos do sistema e evitar sobrecarga de threads. A divisão da grade em segmentos para cada thread foi uma estratégia bem-sucedida, pois permite que as operações sejam realizadas de maneira paralela, aproveitando melhor os múltiplos núcleos do processador. No entanto, é importante destacar que, à medida que o número de threads aumenta, o custo de sincronização também aumenta, o que pode diminuir a eficiência do sistema em grades muito grandes ou com um número de threads excessivo.

A implementação das bordas toroidais, que fazem com que as células da grade se conectem de forma cíclica, foi um desafio interessante, pois exigiu um cuidado especial na atualização das células nas bordas da grade. No entanto, isso foi realizado com sucesso ao considerar as bordas durante a atualização de cada segmento processado pelas threads. Esse cuidado garantiu a consistência da simulação, sem que células fossem negligenciadas devido à sua posição nas extremidades.

Apesar do bom desempenho da simulação, o uso de múltiplas threads também apresentou desafios. A coordenação entre as threads, especialmente ao lidar com a atualização das células que têm vizinhos em diferentes segmentos, exigiu uma abordagem cuidadosa para garantir que não houvesse falhas de sincronização. Felizmente, o uso do **ThreadPoolExecutor** e a técnica de atribuição de diferentes partes da grade a cada thread ajudaram a minimizar esses problemas.

4.3 Conclusão

O projeto do **Jogo da Vida** com simulação paralela e interface gráfica foi bem-sucedido em seus objetivos, demonstrando uma aplicação prática do paralelismo para a otimização de simulações em larga escala. A utilização de múltiplas threads proporcionou uma redução significativa no tempo de processamento da simulação, especialmente em grades maiores, tornando o sistema escalável e eficiente.

A interface gráfica forneceu uma visualização clara e interativa da evolução do jogo, com controle total sobre a execução da simulação. A implementação também mostrou que é possível combinar técnicas de paralelismo com uma interface de usuário intuitiva, permitindo uma experiência mais rica e interativa.

No futuro, uma possível melhoria seria a implementação de algoritmos de balanceamento dinâmico de carga, para ajustar automaticamente o número de threads ou redistribuir o trabalho entre elas, dependendo da complexidade da simulação. Outra

linha de pesquisa interessante seria a aplicação de técnicas de otimização de desempenho, como o uso de GPUs para acelerar a simulação de grandes grades.

Em resumo, este trabalho não apenas apresentou uma implementação sólida do **Jogo da Vida** com execução paralela, mas também abriu caminho para futuras otimizações e melhorias na área de simulações computacionais.

Referências

LIMA, L. Z.; OLIVEIRA, P. P. M.; SANTORO, F. M. *Jogo da Vida: Conceitos e Aplicações*. Rio de Janeiro, RJ, Brasil, Junho de 2014.

Martins, M. D. C. (2015). John Conway e o seu Jogo da Vida. *Correio dos Açores*, 9-9.

Silva, V. V. F., de Aquino Sales, J. M., Gonçalves, T. M. P., da Silva Menezes, M., & Mezzomo, I. (2018). Jogo da Vida com Probabilidades. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, *6*(1).

SILVA, L. P. O.; BITTENCOURT, L. F. **Gerência de Recursos em Sistemas Distribuídos para Cloud Gaming**. Relatório Técnico – IC-PFG-20-22, Projeto Final de Graduação, Universidade Estadual de Campinas, Instituto de Computação, Dezembro 2020.

UNIVERSIDADE DE SÃO PAULO. **Jogo da Vida**. Acervo da Matemática. Disponível em: https://matemateca.ime.usp.br/acervo/jogo_vida.html. Acesso em: 28 jan. 2025.