



# **CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

*Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016*  
*AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.*

João Paulo Oliveira e Silva

APLICAÇÃO WEB PARA A VISUALIZAÇÃO DA CRITICALIDADE  
AUTO-ORGANIZADA NO MODELO DA PILHA DE AREIA

Palmas – TO

2021

João Paulo Oliveira e Silva

APLICAÇÃO WEB PARA A VISUALIZAÇÃO DA CRITICALIDADE  
AUTO-ORGANIZADA NO MODELO DA PILHA DE AREIA

Monografia elaborada e apresentada como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Engenharia de Software pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: M.e. Fabiano Fagundes.

Palmas – TO

2021

João Paulo Oliveira e Silva  
APLICAÇÃO WEB PARA A VISUALIZAÇÃO DA CRITICALIDADE  
AUTO-ORGANIZADA MODELO DA PILHA DE AREIA

Monografia elaborada e apresentada como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Engenharia de Software pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e. Fabiano Fagundes.

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Prof. M.e Fabiano Fagundes

Orientador

Centro Universitário Luterano de Palmas – CEULP

---

Prof. M.e. Jackson Gomes de Souza

Centro Universitário Luterano de Palmas – CEULP

---

Prof. Esp. Fábio Castro Araújo

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2021

## RESUMO

SILVA, João Paulo Oliveira e. **APLICAÇÃO WEB PARA A VISUALIZAÇÃO DA CRITICALIDADE AUTO-ORGANIZADA MODELO DA PILHA DE AREIA**. 2021. 34 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Engenharia de Software, Centro Universitário Luterano de Palmas, Palmas/TO, 2021.

O modelo da pilha de areia é um modelo que simula a execução de um processo de um sistema complexo que é composto de várias unidades que evoluem com o tempo e é representado por uma matriz onde cada célula representa um sítio e para cada grão adicionado em um sítio a sua cor é alterada. Quando atingida a altura máxima é iniciado o processo de avalanche, onde os grãos do sítio são distribuídos para os seus vizinhos, estabilizando a pilha. Entender esse processo não é simples, existem alguns simuladores web, como o *NetLogo Web*, que simula o processo de execução do modelo da pilha de areia, mas nas pesquisas sobre os simuladores, constatou-se um problema em relação a dificuldade de entender e de utilizar a aplicação. Para solucionar esse problema, foi proposta uma aplicação que facilitasse o entendimento do usuário, permitindo que ele possa interagir com o modelo e também definir alguns parâmetros. Assim, a aplicação foi desenvolvida utilizando o framework *Angular*, *JavaScript*, *three.js*, *CSS* e *HTML*, onde primeiramente foram realizadas pesquisas por projetos relacionados, depois as funcionalidades foram definidas, com as funcionalidades definidas foi iniciado o desenvolvimento da aplicação e a adição de novas funcionalidades e por fim, o *deploy* para o *Heroku*. A aplicação oferece uma simulação da execução em tempo real do modelo da pilha de areia, possibilitando uma interação do usuário com o modelo. A aplicação pode ser usada em ambiente acadêmico para exemplificar os processos do modelo da pilha de areia, dessa forma deixando mais claro o seu funcionamento.

## LISTA DE FIGURAS

Figura 1 - Geração da matriz	12
Figura 2 - Processo de avalanche	13
Figura 3 - Fluxo da Metodologia	18
Figura 4 - Arquitetura da aplicação	22
Figura 5 - Aplicação visão geral	23
Figura 6 - Funcionalidades parte superior	24
Figura 7 - Seleção de matriz	24
Figura 8 - Controles do mouse	25
Figura 9 - Processos da pilha de areia	25
Figura 10 - Controles diversos	26
Figura 11 - Modal cores	26
Figura 12 - Gráficos	27
Figura 13 - Matriz 100X100	28
Figura 14 - Resultado após um período	29

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>7</b>
<b>2 REFERENCIAL TEÓRICO</b>	<b>10</b>
2.1 Sistemas Dinâmicos	10
2.2 Criticalidade auto-organizada	10
2.2.1 Pilha de Areia	11
2.3 Simuladores computacionais	13
2.4 TRABALHOS RELACIONADOS	14
2.4.1 A criticalidade auto-organizada na pilha de areia	14
2.4.2 The Abelian Sandpile and Related Models	14
2.4.3 A Discrete Model for the Dynamics of Sandpiles Surfaces	15
<b>3 METODOLOGIA</b>	<b>17</b>
3.1 MATERIAIS	17
3.2 MÉTODOS	18
<b>4 RESULTADOS</b>	<b>22</b>
4.1 FUNCIONALIDADES DA APLICAÇÃO	22
4.1.1 Funcionalidades	23
4.1.2 Simulação	27
<b>CONSIDERAÇÕES FINAIS</b>	<b>30</b>

## 1 INTRODUÇÃO

Sistemas complexos são sistemas dinâmicos que são compostos por várias unidades que evoluem com o tempo.

São considerados sistemas complexos quando são sistemas abertos, quando interagem com o ambiente em que se situa, quando apresentam comportamento aleatório, quando apresentarem geometria fractal, por apresentarem vários estados de um equilíbrio, denominados de atratores (CARNEIRO et al., 2005).

“O termo criticalidade auto-organizada, do inglês *Self-Organized Criticality*, refere-se a tendência de um sistema dinâmico organizar-se num estado estacionário, sem a necessidade de impulso de qualquer parâmetro externo” (OLIVEIRA, 2012, p. 20). A criticalidade auto-organizada encontra-se em sistemas complexos, pois podem consistir em muitas partes que interagem entre si em vários níveis, onde evolui para um estado crítico de forma desequilibrada, o estado crítico é essa auto-organização (BAK et al., 1987).

Um exemplo de sistema que também serve para observar o aparecimento da criticalidade auto-organizada é o modelo da pilha de areia. A ideia por trás do modelo da pilha de areia foi proposta por Bak (1996), quando, na busca por uma representação menos abstrata sobre a criticalidade auto-organizada, usou o exemplo de uma pilha de areia. Bak (1996, p. 2) considera o cenário em que é despejada uma quantidade de areia em uma superfície plana; inicialmente os grãos despejados permanecem onde caem, mas, conforme o processo continua, a pilha se torna mais íngreme, podendo ocorrer pequenos deslizamentos. Com o passar do tempo, os deslizamentos se tornam cada vez maiores e podem ocorrer por toda a pilha.

Carneiro et al. (2005) afirma que, com o passar do tempo, o sistema alcança um estado estacionário, quando a pilha passa a ter uma inclinação constante chamada de ângulo de estabilidade, e a medida dos grãos que entram no sistema é compatível com a medida dos grãos que o deixam. É nesse estado estacionário que o sistema se encontra no seu estado crítico, ou seja, a criticalidade auto-organizada, onde não é possível prever quando e qual será o tamanho que os deslizamentos irão acontecer.

A plataforma proposta, nomeada SAPIENS - *SAndPile ENvironment Simulator*, ou “simulador de ambiente de pilha de areia”, teve o objetivo de desenvolver uma aplicação web que terá funcionalidades com foco de utilização em ambientes de ensino. A aplicação simula a execução do modelo da pilha de areia utilizando recursos gráficos e interativos que permitiu

uma interação com o usuário que, por sua vez, pode observar o funcionamento do modelo da pilha de areia sendo executado em tempo real e também o surgimento da criticalidade auto-organizada.

Assim, este trabalho apresenta o desenvolvimento de uma aplicação web interativa para a visualização em tempo real da execução do modelo da pilha de areia, em que se objetivou a possibilidade de interação com o usuário; a definição de formas de apresentação e interação do usuário com os gráficos que podem ser gerados e a inserção destes em uma interface web interativa onde serão apresentadas as matrizes e gráficos.

Entender o processo de execução do modelo da pilha de areia e o surgimento da criticalidade auto-organizada não é tão simples, ainda mais quando são utilizados materiais que não são interativos. Quando abordado esse tema no ensino acadêmico, por muitas vezes são utilizados materiais que não permitem uma interação com os alunos, como por exemplo, slides e textos.

Existem diversos simuladores do modelo da pilha de areia, tanto desktop quanto web. Nas pesquisas sobre os simuladores foram encontrados alguns problemas, como por exemplo, não apresentar muitas informações do processo realizado, ou do que se trata, dicas ou um manual de como utilizar a aplicação. Os simuladores foram desenvolvidos para quem já tem um conhecimento sobre o assunto, tornando difícil a utilização da aplicação por quem ainda não tem conhecimento sobre.

Utilizando de recursos de software para visualizar em tempo real da execução do modelo da pilha de areia e do surgimento da criticalidade auto-organizada, permitindo o usuário interagir com a aplicação, podendo definir tamanho ou velocidade, pode trazer um maior entendimento a respeito do modelo, assim podendo exemplificar melhor o seu funcionamento. A aplicação visa ser utilizada em ambientes de ensino acadêmico, sendo ela web que vai permitir ser acessada mesmo fora desse ambiente em diversos dispositivos e que não vai requisitar ser instalada, também vai contar com uma interface mais amigável que conterà dicas da utilização e também da sua execução entre outras funcionalidades que ainda serão definidas, tendo um foco maior no entendimento da execução do modelo e tornando claro mesmo para quem está tendo o primeiro contato com esse tema.

Por fim, o trabalho apresenta etapas desde a concepção da proposta até a apresentação dos resultados obtidos com o desenvolvimento da aplicação. O trabalho foi organizado contendo as seções: seção 1, Introdução, apresenta o problema de pesquisa, hipótese,



objetivos e justificativa; seção 2, Referencial teórico, onde é apresentado o embasamento teórico do trabalho; seção 3, Materiais e Métodos, descreve os materiais e tecnologias utilizadas e apresenta a metodologia utilizada para o desenvolvimento da aplicação; seção 4, Resultados, apresenta os resultados obtidos na execução do trabalho e no desenvolvimento da aplicação; seção 5, Considerações Finais, apresenta a conclusão e propostas sobre os possíveis trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

### 2.1 SISTEMAS DINÂMICOS

“Sistemas dinâmicos são sistemas fora de equilíbrio, que são caracterizados por estados que mudam com o tempo, são sistemas usados para modelar e fazer previsões de sistemas físicos, biológicos, financeiros, etc” (SILVA et al., 2018). São considerados sistemas dinâmicos quando pelo menos uma de suas variáveis de estado depende do tempo, dessa forma os efeitos dos seus processos podem mudar de acordo com o tempo e não necessariamente será proporcional a suas causas, diferentemente dos sistemas lineares.

Souza et al. (2004) afirmam que “sistemas complexos são aqueles compostos de muitos elementos e/ou subsistemas diferentes interagindo espacialmente e temporalmente de forma não linear, gerando padrões emergentes que são observáveis apenas em escalas maiores”. Ou seja, os sistemas complexos consistem de muitas partes, ou muitos elementos que interagem em vários níveis ou escalas diferentes, com todos esses níveis afetando uns aos outros.

Sistemas complexos são sistemas não lineares, onde uma pequena entrada pode acabar em um grande efeito sistêmico podendo ou não afetar todos os níveis que compõem o sistema, efeito esse que pode ocorrer em um breve período de tempo de forma inesperada.

### 2.2 CRITICALIDADE AUTO-ORGANIZADA

O termo criticalidade auto-organizada, refere-se à tendência de um sistema dinâmico complexo de organizar-se num estado estacionário, sem qualquer parâmetro externo de ajuste onde evolui para um estado crítico de forma desequilibrada (CASTRO., 2007). Quando o sistema se encontra em um estado crítico auto-organizado há grandes instabilidades denominadas avalanches.

No estado crítico eventos como uma avalanche podem acontecer do nada e pequenos eventos podem desencadear efeitos enormes, quando uma avalanche ocorre por toda a pilha gerando grandes catástrofes, mas por si só organiza-se em um estado estacionário. Por se tratar de um sistema aberto, onde pode influenciar ou sofrer influências do ambiente externo, a troca de informações com o ambiente pode favorecer o aumento da organização interna onde evolui para um estado crítico. Nesse estado crítico o sistema obedeceria às leis da potência, como por exemplo tem-se o modelo da pilha de areia, onde a adição constante de areia resulta em avalanches.

### 2.2.1 Pilha de Areia

Uma forma de representar a criticalidade auto-organizada é utilizando o modelo da pilha de areia. Para uma melhor compreensão analisando a definição do modelo feita por Bak (1996, p. 52 e 53) onde a areia jogada sobre a mesa é representada por uma matriz bidimensional, em que cada sítio da matriz tem coordenadas  $(x, y)$ , é atribuído um número  $Z(x, y)$  que representa o número de grãos presente naquele sítio. Para uma matriz de tamanho  $L = 100$ , onde o número total de sítios é de  $L \times L$ , as coordenadas de  $x$  e  $y$  estão entre 1 e 100. A adição de um grão de areia em um sítio da matriz é realizada escolhendo um local aleatório e aumentando a sua altura  $Z$  nesse local.  $Z(x, y)$  é definido por:

$$Z(x, y) \rightarrow Z(x, y) + 1 \quad (2.1)$$

O processo é repetido algumas vezes e, para ter uma dinâmica mais interessante, é aplicada uma regra que permite que um grão de areia mude de um sítio para outro, criando uma regra de “tombamento” ou avalanche. É definido que, sempre que uma altura  $Z$  exceder um valor crítico que pode ser definido, ocorre a avalanche. Como por exemplo, define-se que o valor crítico é igual a 3, então se em um sítio a altura  $Z$  for igual a 4 ocorre a avalanche e a altura desse local diminui em 4 unidades,

$$Z(x, y) \rightarrow Z(x, y) - 4 \quad (2.2)$$

sendo dispensada uma unidade para cada sítio vizinho.

$$Z(x \pm 1, y) \rightarrow Z(x \pm 1, y) + 1, Z(x, y \pm 1) \rightarrow Z(x, y \pm 1) + 1 \quad (2.3)$$

O processo de avalanche é ilustrado melhor nas Figuras abaixo, onde tem-se uma matriz de 5 por 5, que tem o seu ponto crítico definido igual a 3. Na Figura 1 temos a criação da matriz, onde inicialmente na primeira matriz todos os seu sítios tem o seu valor igual a zero  $Z = 0$ , ou seja, não tem grãos, posteriormente foi atribuído um grão a um sítio  $Z(5, 1) \rightarrow Z(5, 1) + 1$ , o processo continua até chegarmos na quarta matriz onde um dos sítios excedeu o valor crítico.

Figura 1 - Geração da matriz

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

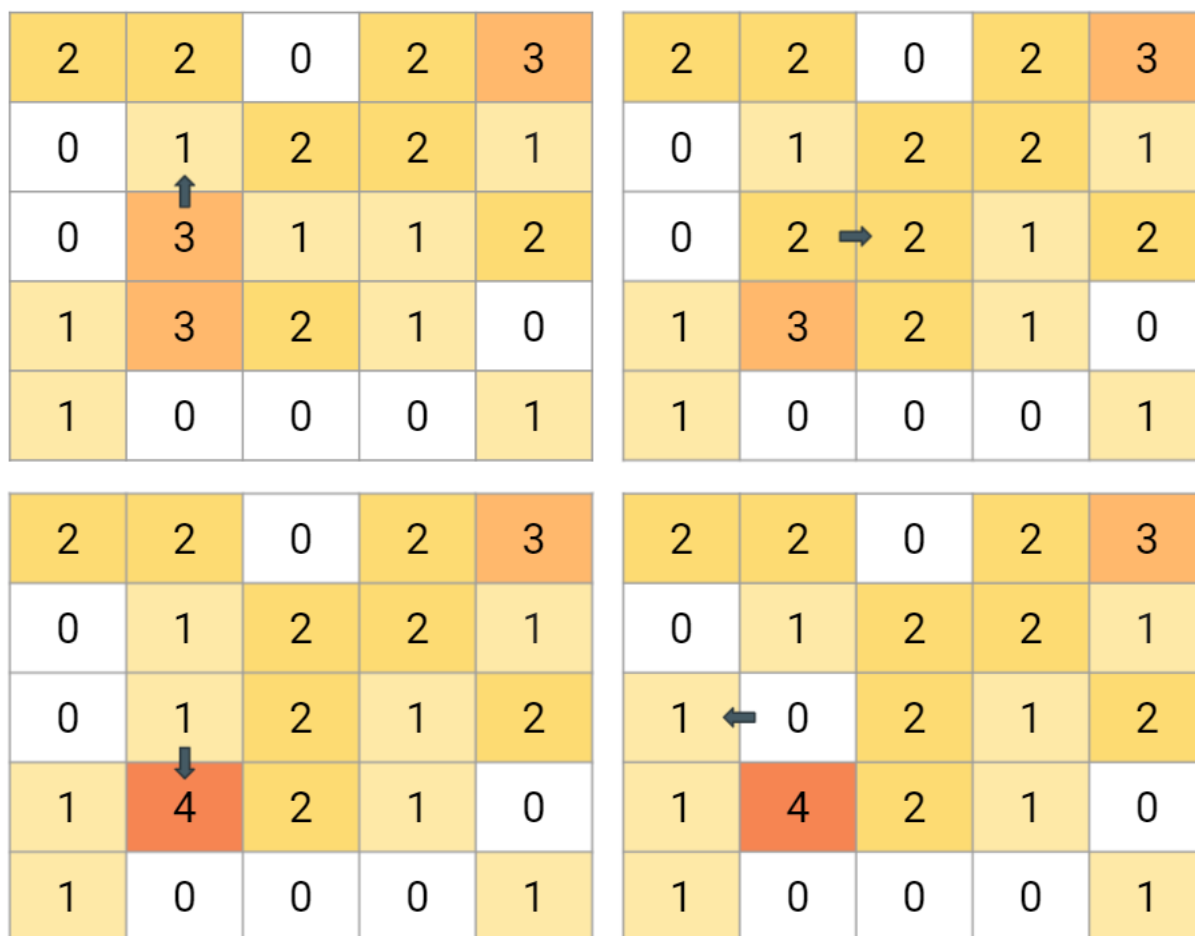
  

2	2	0	2	3
0	0	2	2	1
0	3	1	1	2
1	3	2	1	0
1	0	0	0	1

2	2	0	2	3
0	0	2	2	1
0	4	1	1	2
1	3	2	1	0
1	0	0	0	1

Quando um sítio excede o valor crítico é dado início ao processo de avalanche, onde os grãos do sítio são dispensados para os seus vizinhos, a Figura 2 demonstra como o processo de avalanche ocorre. Primeiro os grãos do sítio onde o valor crítico foi excedido são removidos  $Z(2, 3) \rightarrow Z(2, 3) - 4$  e depois dispensados para os seus vizinhos, conforme demonstrado nas matrizes  $Z(2 \pm 1, 3) \rightarrow Z(2 \pm 1, 3) + 1$ ,  $Z(2, 3 \pm 1) \rightarrow Z(2, 3 \pm 1) + 1$ .

Figura 2 - Processo de avalanche



O processo se repete quantas vezes forem necessárias, até que não exista nenhum sítio que esteja excedendo o valor limite estipulado.

### 2.3 SIMULADORES COMPUTACIONAIS

Simuladores computacionais são recursos tecnológicos que podem auxiliar no entendimento e aprendizagem sobre um determinado assunto. Segundo Martins (2008) “Do ponto de vista pedagógico, simuladores computacionais são softwares úteis e simples, uma vez que levam os alunos a interagirem com os mesmos, o que lhes facilita a compreensão de conceitos abstratos, leis e características do mundo físico”. Para o presente trabalho foi escolhido o modelo do autômato celular. Carneiro et al. (2005) afirmam que “um autômato celular é um sistema dinâmico formado por muitas unidades interagindo umas com as outras”. Um autômato celular pode ser representado por uma matriz, onde pode existir uma regra de interação de um sítio com o outro, como foi exemplificado no item 2.2.3.

## 2.4 TRABALHOS RELACIONADOS

### 2.4.1 A criticalidade auto-organizada na pilha de areia

No trabalho Carneiro e Charret buscaram analisar exemplos simples de sistemas complexos, os autores utilizaram o modelo da pilha de areia, onde também é possível comprovar o surgimento da criticalidade auto-organizada para demonstrar esses exemplos. Para representar o modelo os autores utilizaram de autômatos celulares e desenvolveram uma aplicação em C++ onde realizaram as simulações para o modelo e geraram alguns gráficos.

Como resultado da implementação do modelo da pilha de areia, os autores conseguiram visualizar propriedades comuns aos sistemas complexos como a imprevisibilidade e o aparecimento de estados críticos.

### 2.4.2 The Abelian Sandpile and Related Models

O autor apresentou no trabalho uma breve revisão dos resultados conhecidos sobre o modelo de pilha de areia *Abelian*. Segundo Dhar (1999) a estrutura de grupo *Abelian*, permite um cálculo exato de muitas das suas propriedades, ou seja, quando há uma direção preferencial, é possível calcular os expoentes críticos que caracterizam a distribuição do tamanho da avalanche em todas as direções. O autor também abordou uma generalização do modelo para uma rede de processadores reativos, incluindo modelos de pilha de areia com regras de tombamento probabilístico. Por fim, o autor considerou uma variante probabilística não *Abelian*, relacionada à percolação direcionada.

O autor define o *Abelian Sandpile Model* como uma matriz de  $N$  locais, que são preenchidos por números  $i = 1$  a  $N$ . É definida uma variável de altura inteira não negativa  $Z_i$ , chamada de altura da pilha de areia e também é definido um valor limite  $Z_{ic}$ . A pilha é considerável estável quando para todo  $i$ ,  $Z_i < Z_{ic}$ , ou seja, para cada local da matriz a sua altura deve ser menor que a altura limite. A evolução da pilha é definida por das regras:

1. Na matriz é selecionado um local  $i$  qualquer para que seja adicionado um grão de areia, dessa forma aumentando o seu tamanho em 1.
2. Para quando a altura de um local  $Z_i$  for maior ou igual a altura limite  $Z_{ic}$  é onde o local é considerável instável, então é executada a regra de tombamento, onde os grãos do local são distribuídos para os seus vizinhos. Caso o tombamento torne outro local

instável, a regra é executada novamente até que todos os valores dos locais sejam menores que o valor limite. Pode ocorrer de um grão de areia se perder e sair da matriz, caso o local esteja localizado na extremidade da matriz. Dhar (1999) afirma que, ‘nenhum estado estacionário da pilha de areia é possível, a menos que as partículas possam deixar o sistema’. Então para que seja possível chegar a um estado estacionário onde se encontra a criticalidade auto-organizada, é necessário que os grãos de areia saiam da pilha.

Por fim, Dhar (1999) conclui que “A pilha de areia é, obviamente, um sistema de desequilíbrio impulsionado pela adição lenta de grãos” e que a existência de duas escalas de tempo, não deve ser considerada como condição necessária para a criticalidade auto-organizada.

#### **2.4.3 A Discrete Model for the Dynamics of Sandpiles Surfaces**

O trabalho busca simular fenômenos físicos de forma realista usando autômatos celulares. O modelo proposto pelos autores, usa transições probabilísticas para simular o fluxo de mídia granular através de uma ampulheta. Segundo Devlin et al (2020) modelar um sistema complexo a partir de princípios é ideal para computar fenômenos físicos, pois a precisão é fundamental para esse tipo de fenômeno, mas no que tange videogames e gráficos de computador, geralmente o desempenho supera a qualidade, desde que o resultado seja satisfatório.

O modelo mais comum para a simulação do modelo da pilha de areia bidimensional, que também é usado para o estudo da criticalidade auto-organizada e como a complexidade surge na natureza, é o modelo *Abelian*, ao invés de videogames e computação gráfica. No entanto, para esses modelos se torna mais relevante a utilização de simulações de pilha de areia de perfil lateral. Os autores citam o jogo *Noita*, que utiliza como base para suas simulações de fluido, gás e sólido, um modelo de pilha de areia lateral. Isso leva a uma interessante interação entre o jogo e o jogador, onde, com as ações do jogador é possível manipular o ambiente do jogo. Os autores definiram o domínio do problema como uma coleção de células, onde suas propriedades são atualizadas por meio de um conjunto determinado de regras, para atualizar uma célula são consideradas as células vizinhas.

Os autores tiveram como objetivo, utilizando uma simulação de pilha de areia de perfil lateral padrão, examinar as características computacionais e qualitativas de uma modificação

na pilha, apresentando os resultados e desempenho da execução do modelo. Os autores chegaram à conclusão, que sempre que utilizar a mesma condição inicial, resultará na mesma evolução da simulação, então foi ajustada a probabilidade das transições no comportamento de empilhamento, dessa forma as interações demoraram alguns milissegundos para calcular juntamente com a visualização do modelo. Utilizando a abordagem determinística, o resultado de um tombamento sempre resultará na mesma posição da areia para qualquer número de interações.

O desempenho computacional foi relativo à resolução da tela, conforme a resolução era maior, mais células eram geradas e com isso demorou mais tempo para executar o modelo. Os autores concluem que ao utilizar autômatos celulares para simulações físicas no desenvolvimento de jogos, permite cálculos mais rápidos mantendo uma alta taxa de quadros, além de gerar pouca sobrecarga ao simular outros materiais e suas interações.



### 3 METODOLOGIA

Nesta seção encontram-se os processos metodológicos utilizados para a resolução do problema apresentado.

#### 3.1 MATERIAIS

Para o desenvolvimento da aplicação SAPIENS foram utilizados os seguintes materiais e tecnologias:

- *HTML*: É o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias além do *HTML* geralmente são usadas para descrever a aparência/apresentação (*CSS*) ou a funcionalidade/comportamento (*JavaScript*) de uma página da web.
- *JavaScript*: É uma linguagem leve, interpretada e baseada em objetos com *funções de primeira classe*, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como *node.js*, *Apache CouchDB* e *Adobe Acrobat*. O *JavaScript* é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional). Saiba mais sobre o *JavaScript*.
- *Three.js*: É uma biblioteca que permite a criação de animações 3D usando a linguagem *JavaScript* como parte de um website. *Three.js* usa *WebGL*.
- *CSS*: é uma linguagem de estilo (en-US) usada para descrever a apresentação de um documento escrito em *HTML* ou em *XML* (incluindo várias linguagens em *XML* como *SVG*, *MathML* ou *XHTML*). O *CSS* descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias.
- *Angular*: O *Angular* é um framework feito para desenvolvimento web, mais especificamente de SPA (*Single Page Applications* – Aplicações de página única) tanto para desktop quanto mobile e suas principais características são que ele já traz um conjunto bem completo, com rotas, módulos, pipes, diretivas, entre outras coisas para facilitar o desenvolvimento no dia a dia.
- *Visual Studio Code*: É um editor de código-fonte leve, mas poderoso, que roda em sua área de trabalho e está disponível para *Windows*, *macOS* e *Linux*. Ele vem com suporte integrado para *JavaScript*, *TypeScript* e *Node.js* e tem um rico ecossistema de

extensões para outras linguagens (como *C ++*, *C #*, *Java*, *Python*, *PHP*, *Go*) e runtimes (como *.NET* e *Unity*).

- *Heroku*: *Heroku* é uma plataforma amplamente confiável como uma oferta de serviço que permite aos desenvolvedores realizar implantação, escalonamento e gerenciamento de aplicativos sem complicações. Esta plataforma oferece suporte para uma ampla gama de linguagens de programação, como *Java*, *Ruby*, *PHP*, *Node.js*, *Python*, *Scala* e *Clojure*. O *Heroku* executa aplicativos por meio de contêineres virtuais conhecidos como *Dynos*.

Como base para o desenvolvimento da aplicação, foi utilizado um projeto desenvolvido pelo os autores Valentin Darrigo, Jeremy Fersula e Kevin Perrot, o projeto se trata de uma aplicação chamada JS-Sandpile, que é uma aplicação *open source* desenvolvida com *JavaScript* que simula o funcionamento do modelo da pilha de areia, a aplicação é licenciada pela *Creative Commons Attribution 4.0 International License*, que permite compartilhar e alterar a aplicação, dando as devidas referências.

### 3.2 MÉTODOS

O desenvolvimento dessa aplicação constituiu em 5 etapas (que podem ser vistas na Figura 3), sendo elas: busca por projetos relacionados; definição das funcionalidades; desenvolvimento da aplicação; adição de novas funcionalidades e deploy para o *Heroku*. As etapas serão detalhadas no desenvolvimento desta seção.

Figura 3 - Fluxo da Metodologia



Na primeira etapa foram realizadas pesquisas por projetos relacionados, a fim de extrair referências para o desenvolvimento da aplicação, também foram realizadas pesquisas

sobre frameworks e bibliotecas que poderiam ser utilizadas no desenvolvimento da aplicação. Como resultado da pesquisa, foi encontrada a aplicação JS-Sandpile, que implementa o modelo da pilha de areia.

Na etapa 2 foram definidas as funcionalidades da aplicação, quais funcionalidades seriam necessárias para solucionar o problema proposto no trabalho, sendo elas as listadas a seguir:

- Zoom: Permitir que o usuário possa aplicar zoom na matriz gerada, para melhorar a visualização;
- Tempo entre passos/interações: Ter uma opção para definir o tempo entre as interações da aplicação, como adicionar um grão aleatório ou avalanches;
- Iniciar e pausar processo: Permitir que o usuário possa iniciar ou pausar o processo de simulação da pilha de areia;
- Apresentar número de passos: Apresentar para o usuário o número de passos já realizados na simulação;
- Pausa automática: Ter uma opção de parar o processo de simulação para quando a pilha estiver no estado de estabilidade, ou seja, quando não houver mais nenhuma avalanche para ocorrer;
- Permitir uma melhor visualização dos sítios: Apresentar na matriz linhas que destacam as células apresentando uma melhor visualização;
- Criação de uma matriz: Possibilidade de criar uma matriz no formato de um quadrado, onde o usuário poderia definir o seu tamanho em altura e largura;
- Adicionar grãos: Possibilitar que o usuário possa incluir grãos interagindo através de ações do mouse com a matriz gerada;
- Remover grãos: Remover grãos interagindo através de ações do mouse com a matriz gerada;
- Definir grãos: Diferente da funcionalidade de adicionar, essa define a quantidade de grãos no sítio;
- Definir quantidade de grãos: Um campo onde será definida a quantidade de grãos que serão adicionados, removidos ou definidos, limitando a 4 grãos, que é a altura máxima da pilha;

- Possibilitar realizar um passo/interação da pilha de areia: Um botão que realiza uma interação da pilha de forma manual sem a necessidade de iniciar o processo de simulação;
- Adicionar grão aleatório automaticamente: Essa funcionalidade vai adicionar um grão aleatório de forma automática, mas para isso ocorrer o processo de simulação deve estar iniciada;
- Adicionar grão aleatório: Um botão que quando pressionado vai adicionar um grão aleatório na pilha, sem a necessidade de o processo de simulação estar iniciado;
- Limpar: Limpa todos os grãos da pilha;
- Estabilizar: Estabilizar a pilha;
- Alterar escala de cores: Permite o usuário escolher uma escala de cores dentre as pré-definidas;
- Demonstrar números de sítios: Disponibiliza a quantidade de sítios gerados na matriz;
- Gráficos: Serão disponibilizados dois gráficos, um exibirá a quantidade de avalanches por passos e outro a quantidade de grãos perdidos por passo, ou seja, grãos que saíram da pilha.

Na terceira etapa foi iniciado o desenvolvimento da aplicação, onde foi criado um projeto em Angular e o repositório no GitHub para versionamento, com o projeto criado e configurado foi realizada a integração da aplicação JS-Sandpile com o projeto criado em Angular. Esse processo se fez necessário devido a aplicação JS-Sandpile ter sido desenvolvida sem nenhum framework. Após fazer a integração, foi realizado o refinamento da aplicação, onde foram removidas funcionalidades da aplicação base que não eram necessárias para a solução do problema proposto neste trabalho, e as funcionalidades descritas acima foram desenvolvidas.

Na etapa 4 foram adicionadas novas funcionalidades, sendo elas:

- Adicionar grão aleatório automaticamente: onde foi criado um checkbox que quando ativo, adiciona um grão em um sítio aleatório de forma automática a cada passo realizado quando a simulação está iniciada;
- Gráficos: Foi alterada a implementação do método que executa a distribuição dos grãos de um determinado sítio que excedeu o limite para os vizinhos, para obter a quantidade de avalanches e a quantidade de grãos que saem da pilha;

- Limitar grãos por sítio: foi alterada a implementação do método que adiciona os grãos em um determinado sítio para que fosse limitado para no máximo quatro grãos, que é o mesmo valor de limite definido para que ocorra a avalanche.

Por fim, na etapa 5 foi realizado o *deploy* da aplicação para o *Heroku* e disponibilizada para ser acessada pelos usuários.

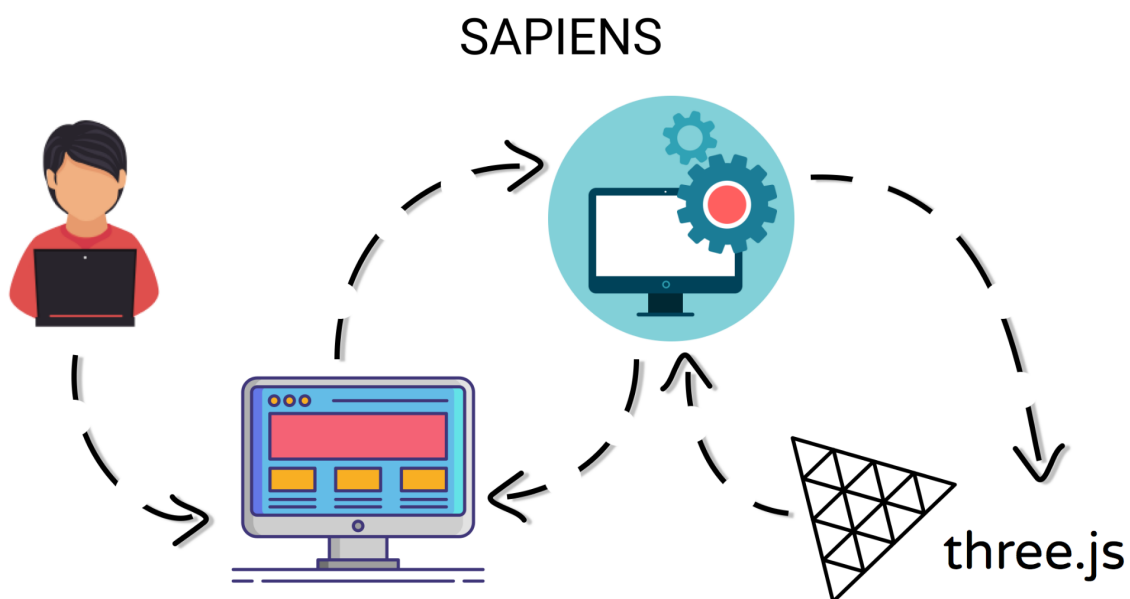
## 4 RESULTADOS

O objetivo do trabalho é uma aplicação web que simula de forma interativa o modelo da pilha de areia e também do surgimento da criticalidade auto-organizada, apresentando informações referentes ao processo em forma de gráfico. Nesta seção serão apresentados os resultados obtidos com o desenvolvimento da aplicação, onde também serão destacadas as suas funcionalidades.

### 4.1 FUNCIONALIDADES DA APLICAÇÃO

Neste tópico serão apresentadas as funcionalidades da aplicação desenvolvida, como também os resultados gerados da execução de uma simulação. A Figura 4, demonstra a arquitetura da aplicação SAPIENS (SAndPIle ENvironment Simulator).

Figura 4 - Arquitetura da aplicação

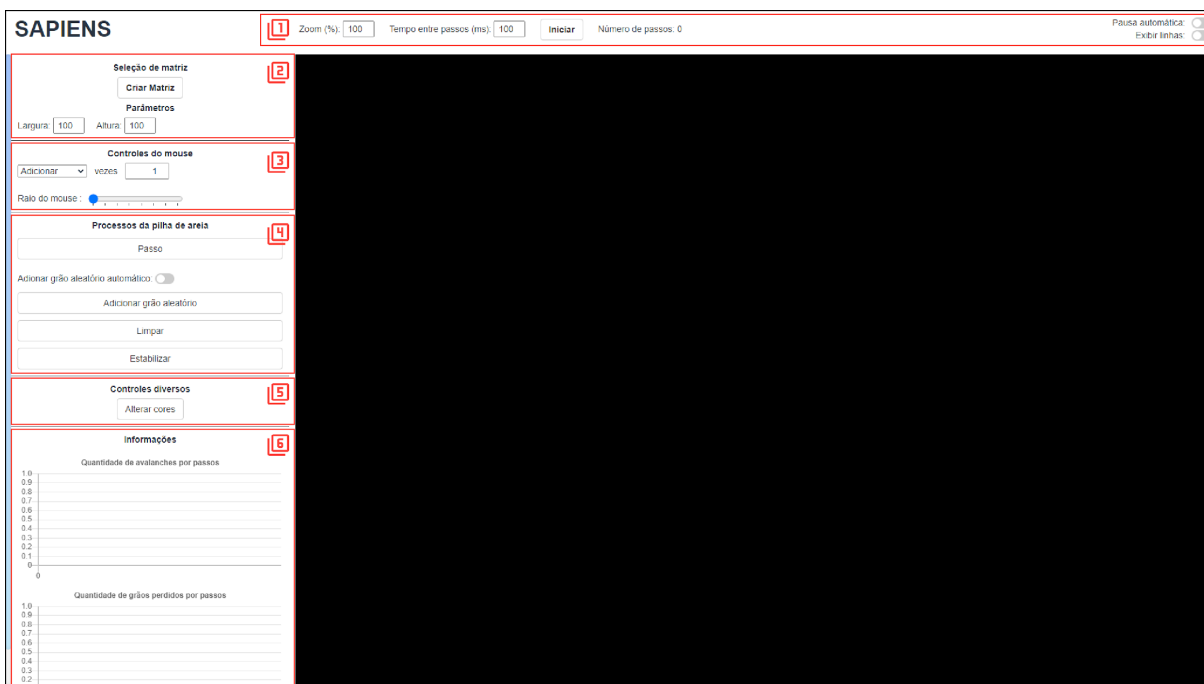


A aplicação é dividida em três partes: interface web, métodos e a biblioteca three.js. O usuário interage com a interface web, onde são dispostas as funcionalidades e também o resultado da execução da simulação; com a ação do usuário, a interface comunica com os arquivos que contêm os métodos que, por sua vez, acessa a biblioteca three.js para executar as

ações referentes à criação e atualização da matriz, retornando os dados para o método que devolve para a interface que por fim, apresenta o resultado para o usuário.

A interface da aplicação é dividida em duas partes com seis blocos, conforme apresentado na Figura 5.

Figura 5 - Aplicação visão geral



Na parte superior e lateral esquerda estão localizadas as funcionalidades para a geração e interação na pilha de areia e na lateral direita é onde a simulação acontece.

#### 4.1.1 Funcionalidades

A aplicação oferece as seguintes funcionalidades para o usuário:

1. Zoom;
2. Tempo entre passos;
3. Iniciar/Pausar;
4. Número de passos;
5. Pausa automática;
6. Exibir linhas;
7. Seleção da matriz;

8. Controles do mouse;
9. Processos da pilha de areia;
10. Controles diversos;
11. Informações.

Na parte superior estão as seguintes funcionalidades apresentadas na Figura 6.

Figura 6 - Funcionalidades parte superior



O *zoom* é onde é controlada o *zoom* da matriz gerada; tempo entre os passos (em milissegundos - ms) é o controle de tempo entre cada passo (situação onde ocorre a avalanche) realizado na pilha de areia em milissegundo; iniciar/pausar é botão que inicia ou pausa a simulação; na pausa automática, caso esteja ativa, a simulação vai pausar todas as vezes que atingir uma estabilidade; exibir linhas, quando selecionado, exibe as linhas entre cada célula da matriz.

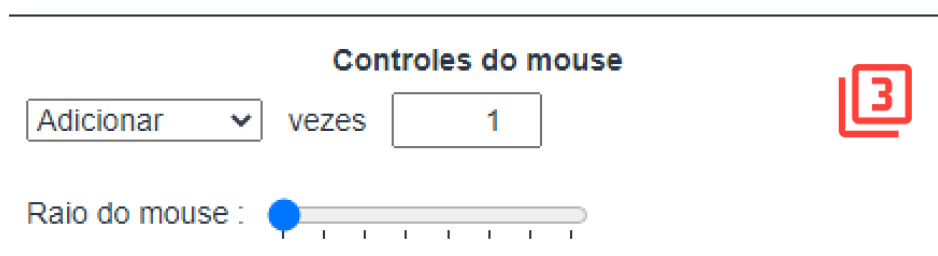
Figura 7 - Seleção de matriz



A funcionalidade representada na Figura 7 é responsável por especificar o tamanho em largura e altura da matriz a ser gerada. Após definir o tamanho, é necessário clicar no botão Criar Matriz para gerar a matriz.

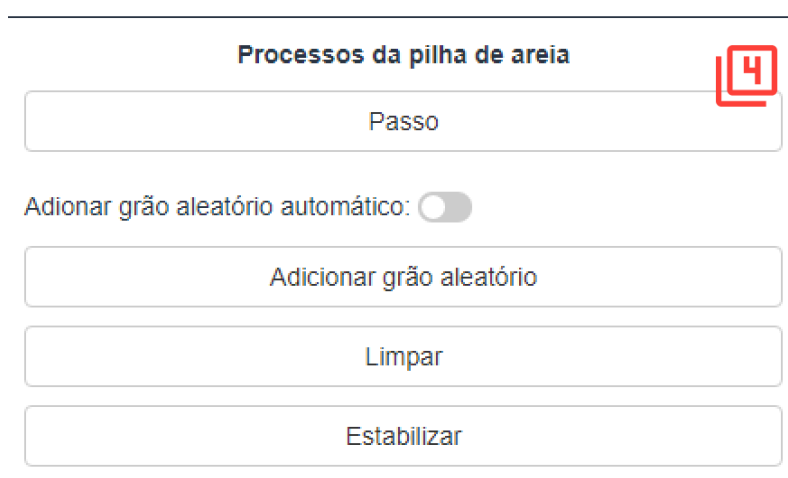


Figura 8 - Controles do mouse



A Figura 8 demonstra a funcionalidade de controles do mouse. Como a aplicação se trata de uma aplicação interativa, é possível interagir com o mouse na matriz gerada, possibilitando adicionar, remover, definir ou selecionar a quantidade de grão em uma ou mais células. Também é possível determinar o tamanho do raio do mouse sobre a matriz como também a quantidade de grãos que serão inseridos, removidos ou definidos na matriz.

Figura 9 - Processos da pilha de areia



A Figura 9 demonstra a forma de acesso às seguintes funcionalidades:

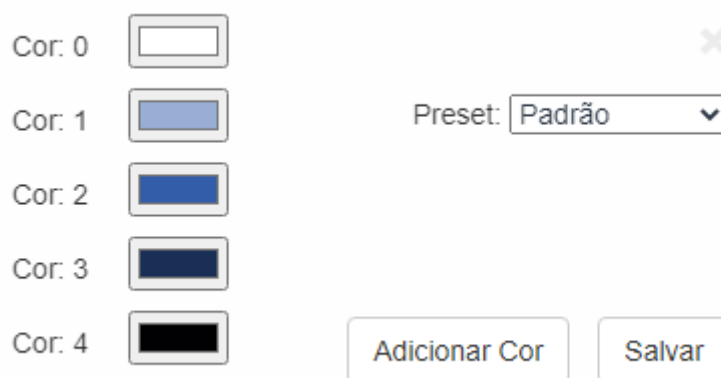
- passo: executa um passo na aplicação, para executar um passo é realizada a chamada de um método *step()* que valida se existe uma matriz criada, realiza a chamada da função *iterate()* que é responsável por executar o processo de avalanche, chama uma outra função que incrementa o número de passos.
- adicionar grão aleatório automático: quando marcado adiciona um grão em um sítio aleatório a cada passo realizado;
- adicionar grão aleatório: adiciona um grão aleatório quando pressionado;

- limpar: limpa todos os grãos da matriz;
- estabilizar: estabiliza a pilha.

Figura 10 - Controles diversos

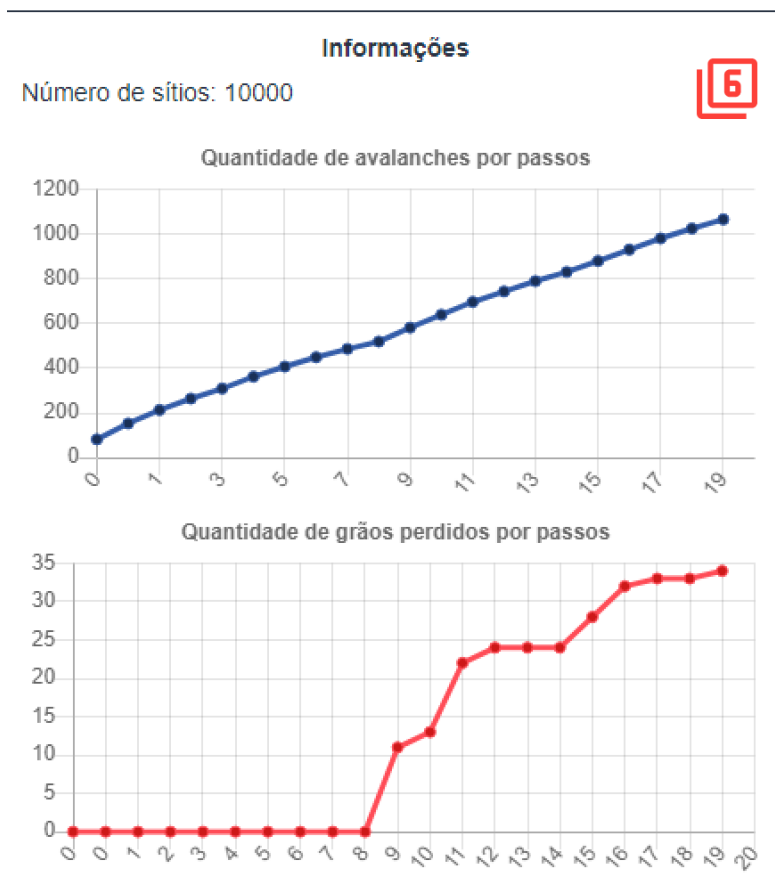


Figura 11 - Modal cores



A funcionalidade da Figura 10 representa os controles diversos que podem ser feitos. Nesta funcionalidade existe uma opção onde é possível configurar as cores dos sítios de acordo com a altura conforme é representado na Figura 11.

Figura 12 - Gráficos

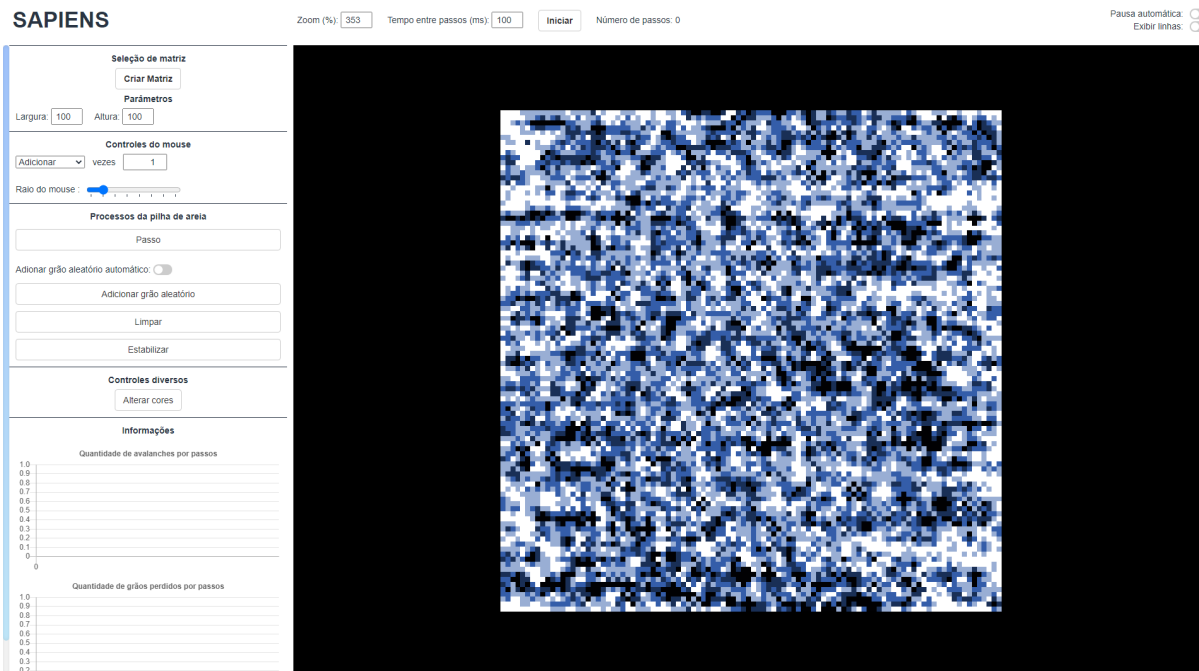


A funcionalidade da Figura 12 apresenta o número de sítios gerados na pilha e também dois gráficos que apresentam a quantidade total de avalanches por passo e a quantidade de grãos que saíram da pilha por passos.

#### 4.1.2 Simulação

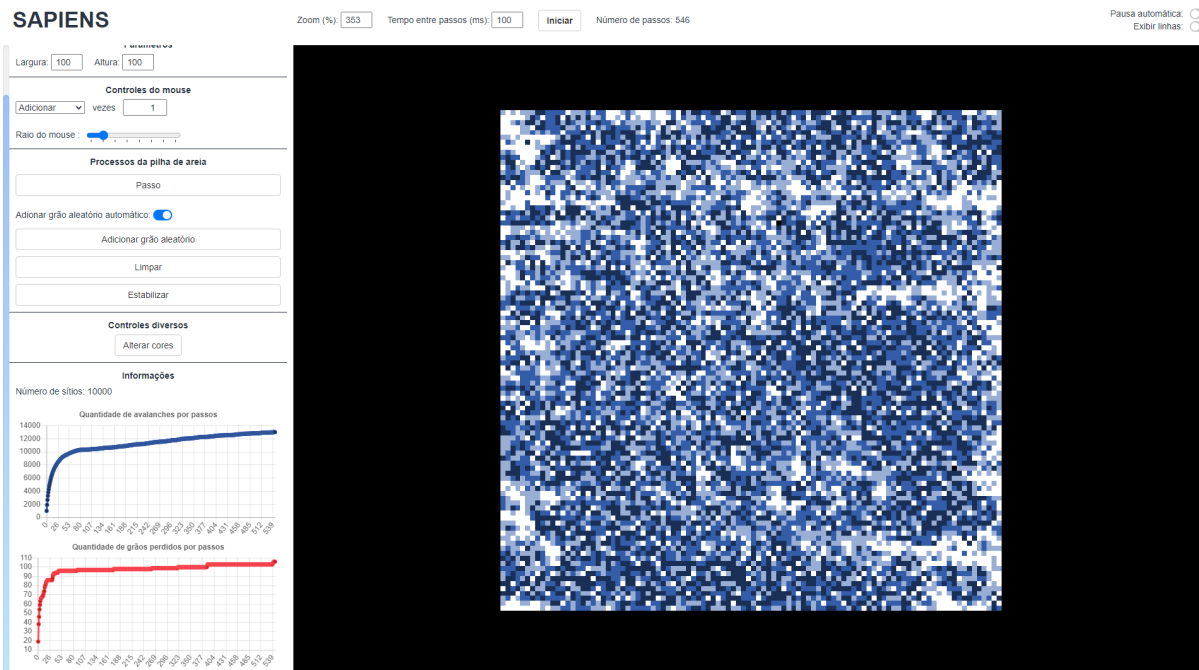
A simulação é demonstrada na área à direita da aplicação. Para representá-la foi gerada uma matriz de 100x100, depois foram adicionados alguns grãos aleatórios representados pela Figura 13, habilitada a opção para adicionar grão aleatório automático e por fim, a simulação foi iniciada.

Figura 13 - Matriz 100X100



Após 546 passos, gerou o seguinte resultado representado na Figura 14.

Figura 14 - Resultado após um período



É possível identificar visualmente a diferença entre a matriz representada na Figura 13 quando a simulação não tinha sido iniciada. Onde é possível visualizar a quantidade de avalanches e grãos perdidos através dos gráficos e quantidade de passos.

## CONSIDERAÇÕES FINAIS

O presente trabalho apresentou uma aplicação que simula a execução do modelo da pilha de areia nomeada de SAPIENS, que permite os usuários visualizarem a execução em tempo real do modelo da pilha de areia.

Foi desenvolvida uma aplicação que simula a execução do modelo da pilha de areia, contendo funcionalidades que permitem ao usuário interagir com o modelo, como definir tamanho da matriz gerada, adicionar, remover ou definir a quantidade de grãos em um determinado sítio, através do controle de mouse. Para o seu desenvolvimento foi necessário ter conhecimento para utilizar recursos gráficos que permitissem criar e interagir com uma matriz.

A simulação do modelo da pilha de areia é uma aplicação web desenvolvida para ser utilizada em ambientes acadêmicos, auxiliando no entendimento. Através desta aplicação os alunos e professores podem acessar e visualizar a execução do modelo em tempo real, também é possível interagir com o modelo permitindo alterar parâmetros para facilitar o entendimento.

Como trabalhos futuros propõe-se a criação de mais tipos de matrizes circular, triangular, hexagonal, entre outros tipos. Também permitir ao usuário definir a altura limite da pilha e adicionar animações para as avalanches. Sugere-se também o incremento de um *tour* para exemplificar como utilizar a aplicação e também, a criação de informações acerca dos processos a cada execução do modelo.

## REFERÊNCIAS

BAK, Per. **How Nature Works: the science of self-organized criticality**. New York: Copernicus (Springer), 1996. 225 p.

BAK, Per; TANG, Chao; WIESENFELD, Kurt. **Self-Organized Criticality: An Explanation of  $1/f$  Noise**. Physical Review Letters, New York, v. 59, n. 4, p. 381-384, 27 jul. 1987.

CARNEIRO, M.V.; CHARRET, I.C.. **A criticalidade auto-organizada na pilha de areia**. Revista Brasileira de Ensino de Física, Lavras, v. 27, n. 4, p. 571-576, dez. 2005. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1806-11172005000400009>.

CASTRO, Paulo Alexandre de. **Rede Complexa e Criticalidade Auto-Organizada: Modelos e Aplicações**. 2007. 175 f. Tese (Doutorado) - Curso de Ciências: Física Básica, Universidade de São Paulo, São Carlos, 2007.

CLARK, Jessica. **O que é o Heroku?** 2021. Disponível em: <https://blog.back4app.com/pt/o-que-e-o-heroku/>. Acesso em: 18 out. 2021.

**CREATIVE Commons Attribution 4.0 International License**. 2021. Disponível em: <https://creativecommons.org/licenses/by/4.0/>. Acesso em: 18 out. 2021.

**CSS**. 2021. Elaborado por MDN Web Docs. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 18 out. 2021.

DEVLIN, Jonathan; SCHUSTER, Micah D. **Probabilistic Cellular Automata for Granular Media in Video Games**. 2020. 5 f. Tese (Doutorado) - Curso de Computer Science, Computer Science And Networking At Wentworth Of Technology, Wentworth Insititue Of Technology, Boston, 2020.

DHAR, Deepak. **The Abelian sandpile and related models**. Physica A: Statistical Mechanics and its Applications, [S.L.], v. 263, n. 1-4, p. 4-25, fev. 1999. Elsevier BV. [http://dx.doi.org/10.1016/s0378-4371\(98\)00493-2](http://dx.doi.org/10.1016/s0378-4371(98)00493-2)

**HTML: Linguagem de Marcação de Hipertexto**. 2021. Elaborado por MDN Web Docs. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 18 out. 2021.

**JAVASCRIPT**. 2021. Elaborado por MDN Web Docs. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 18 out. 2021.

**JS-SANDPILE**. 2020. Disponível em: <https://github.com/huacayacauh/JS-Sandpile>. Acesso em: 18 out. 2021.

MARTINS, Carlos Alberto Gomes. **O USO DE SIMULADORES COMPUTACIONAIS NOS PROCESSOS DE ENSINO- APRENDIZAGEM DE ELETRICIDADE: UM ESTUDO COM ALUNOS DA TERCEIRA SÉRIE DO ENSINO MÉDIO**. 2008. 120 f. Dissertação (Mestrado) - Curso de Física, Concentração em Ensino de Física, Centro Universitário Franciscano, Santa Maria, Rs, 2008.

OLIVEIRA, Guilherme de. **A AUTO-ORGANIZAÇÃO DO PROGRESSO TÉCNICO E DAS MUDANÇAS CLIMÁTICAS NUM MODELO SIMPLES DE EVOLUÇÃO**. 2012. 119 f. Dissertação (Mestrado) - Curso de Economia da Pontifícia, Universidade Católica do Rio Grande do Sul, Porto Alegre, 2012. Disponível em: <http://tede2.pucrs.br/tede2/bitstream/tede/3926/1/438672.pdf>. Acesso em: 22 jun. 2021.

SANTOS, Willian. **Angular 2: o que é e diferenciais**. 2019. Disponível em: <https://www.zup.com.br/blog/angular-2-o-que-e>. Acesso em: 18 out. 2021.

SILVA, Anderson A. A. da; OLIVEIRA, Juliano A. de; HERMES, Joelson D. V. **DINÂMICA CAÓTICA NO MAPA LOGÍSTICO**. 2018. 4 f. Monografia (Especialização) - Curso de Científica e Tecnológica, Instituto Federal do Sul de Minas, Pouso Alegre, 2018.



SOUZA, Gustavo M.; BUCKERIDGE, Marcos S.. **Sistemas complexos: novas formas de ver a Botânica**. Brazilian Journal Of Botany, São Paulo, v. 27, n. 3, p. 407-419, jul. 2004.

**THREE.JS**. 2021. Disponível em: <https://threejs.org/>. Acesso em: 23 nov. 2021.

**VISUAL Studio Code**. 2021. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 18 out. 2021.