

Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Trabalho de Conclusão de Curso – 2020/2

SMALG: SISTEMA PARA MODELAGEM E APRENDIZADO DE ALGORITMOS

Aluno: Adriner Maranhão de Andrade

Orientador: Dalton Solano dos Reis

Introdução

- A ciência da computação pode se demonstrar desafiadora para ensinar.
- Problemas na compreensão conceitual do estudante podem levar a confusões e equívocos em seu modelo mental.
- A utilização de ferramentas para auxiliar neste processo são bem vindas.

Objetivos

- Possibilitar o cadastro de cenários de programação, sendo estes baseados em contratos e assertivas para validação de execução, com uma documentação para descrição do cenário;
- Disponibilizar um ambiente de interação do estudante com o cenário criado, permitindo a implementação desses contratos e execução do código;

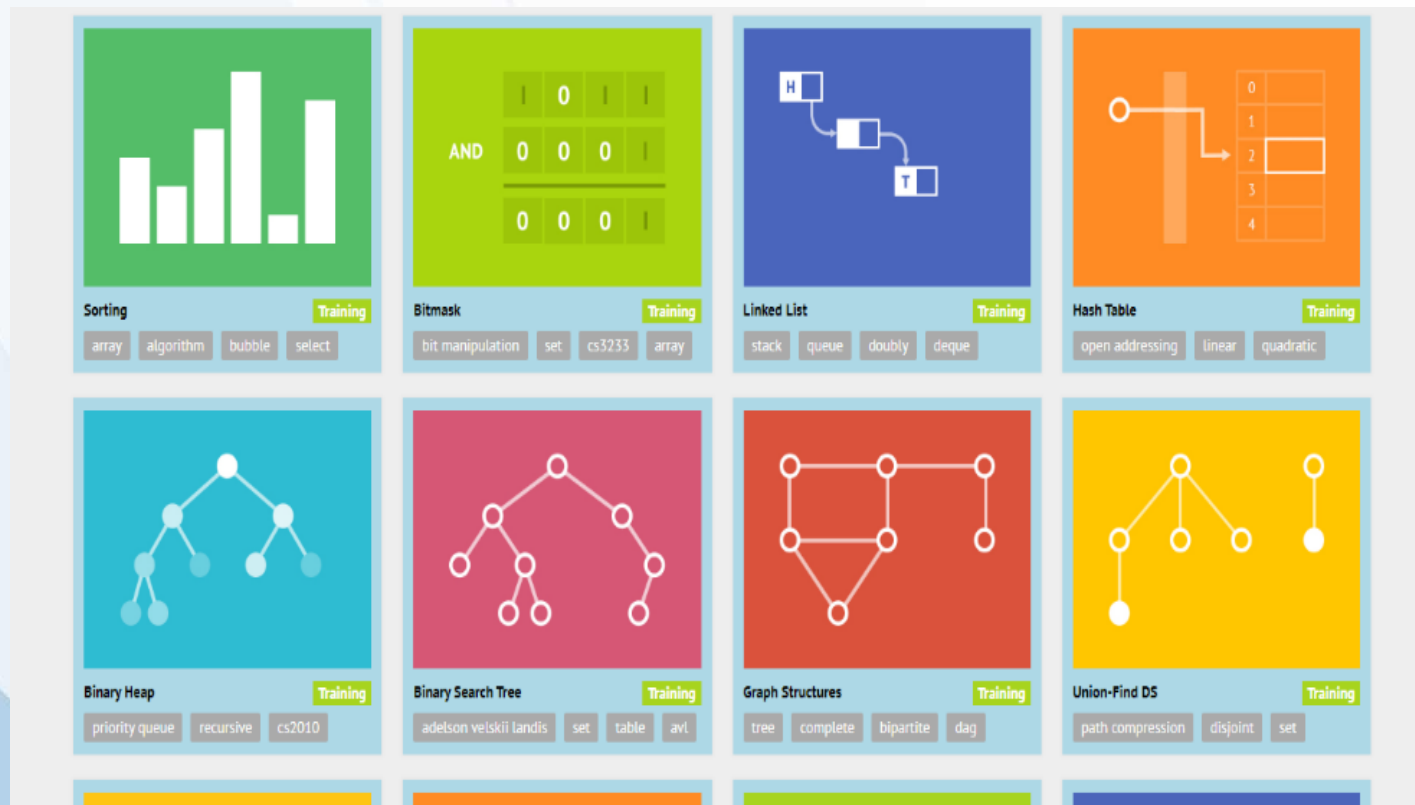
Objetivos

- Apresentar uma representação visual da execução do código;
- Permitir o acompanhamento da execução do código;
- Implementar por parte do autor, os algoritmos de Array List, Linked List, Hash Map e Bubble Sort em cima da plataforma construída como exemplos e para validação da mesma.

VISUALGO (HALIM et al., 2012)

- Ambiente unificado e interativo para visualização de algoritmos. Tem como objetivo o aprendizado autodidático do aluno de vários algoritmos clássicos e não clássicos da área.

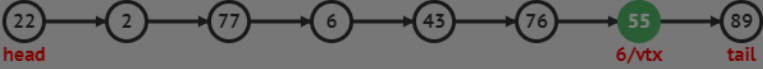
VISUALGO (HALIM et al., 2012)



VISUALGO (HALIM et al., 2012)

7 VISUALGO.NET / en /list LINKED LIST STACK QUEUE DLL DEQUE e-Lecture Mode Login

slide 3-11 (35%)



Prev PgUp 3-11. Insert(i, v) - In Between, $i \in [1..N-1]$ Next PgDn

With the Linked List traversal `Get(i)` sub-routine, we can now implement insertion in the middle of the Linked List as follows (in C++):

```
Vertex* pre = Get(i-1); // traverse to (i-1)-th vertex, O(N)
aft = pre->next; // aft cannot be null, think about it
Vertex* vtx = new Vertex(); // create new vertex
vtx->item = v;
vtx->next = aft; // link this
pre->next = vtx; // and this
```

Try `Insert(3, 44)` on the example Linked List [22 (head)->2->77->6->43->76->89 (tail)].

Also try `Insert(6, 55)` on the same example Linked List. This is a corner case: Insert at the position of tail item, shifting the tail to one position to its right.

This operation is slow, $O(N)$, due to the need for traversing the list (e.g. if i close to $N-1$).

X Esc

pre = head
= 0; k < i-1; k++)
pre.next
aft = pre.next
vtx = new Vertex(v)
= aft
pre.next = vtx

-Lecture Example (auto play until done)
Insert 55 at index 6

the Linked List for visualization.
e process is $O(N)$.

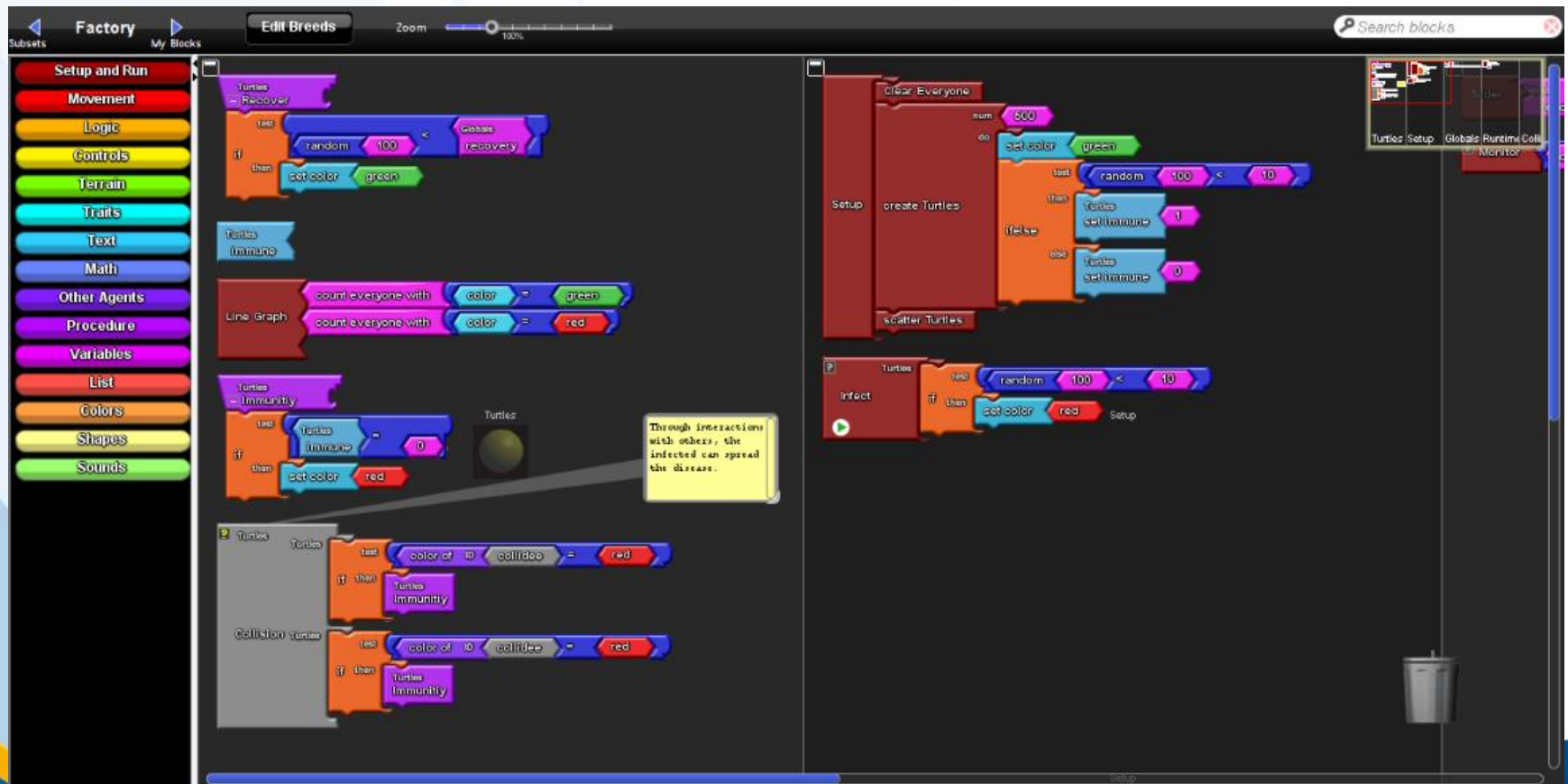
slow fast

About Team Terms of use

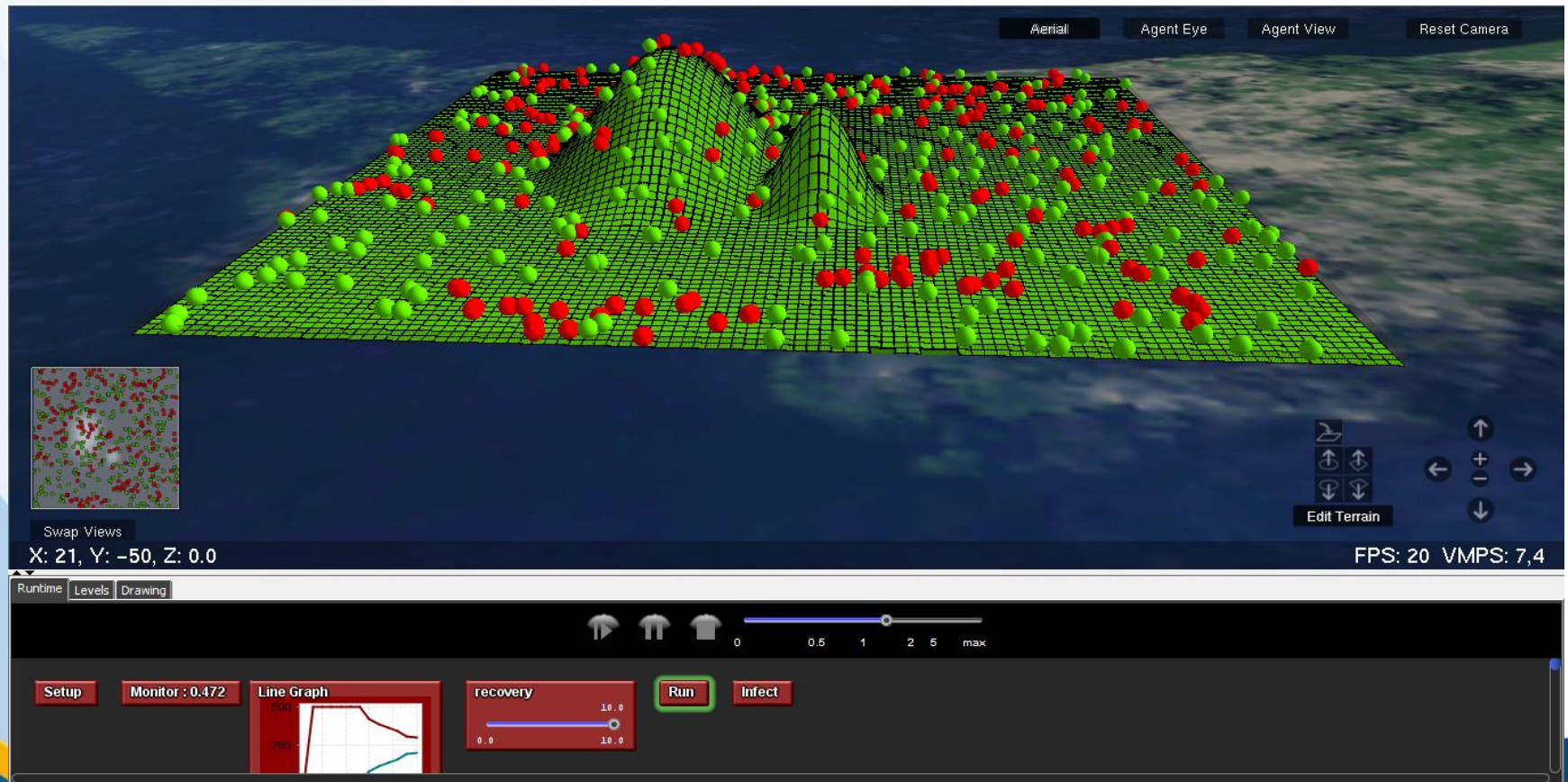
STARLOGO TNG (KLOPER et al., 2009)

- A ferramenta possibilita a criação de cenários virtuais em que é possível interagir construindo sistemas baseados em agentes através de blocos lógicos e visuais.

STARLOGO TNG (KLOPER et al., 2009)



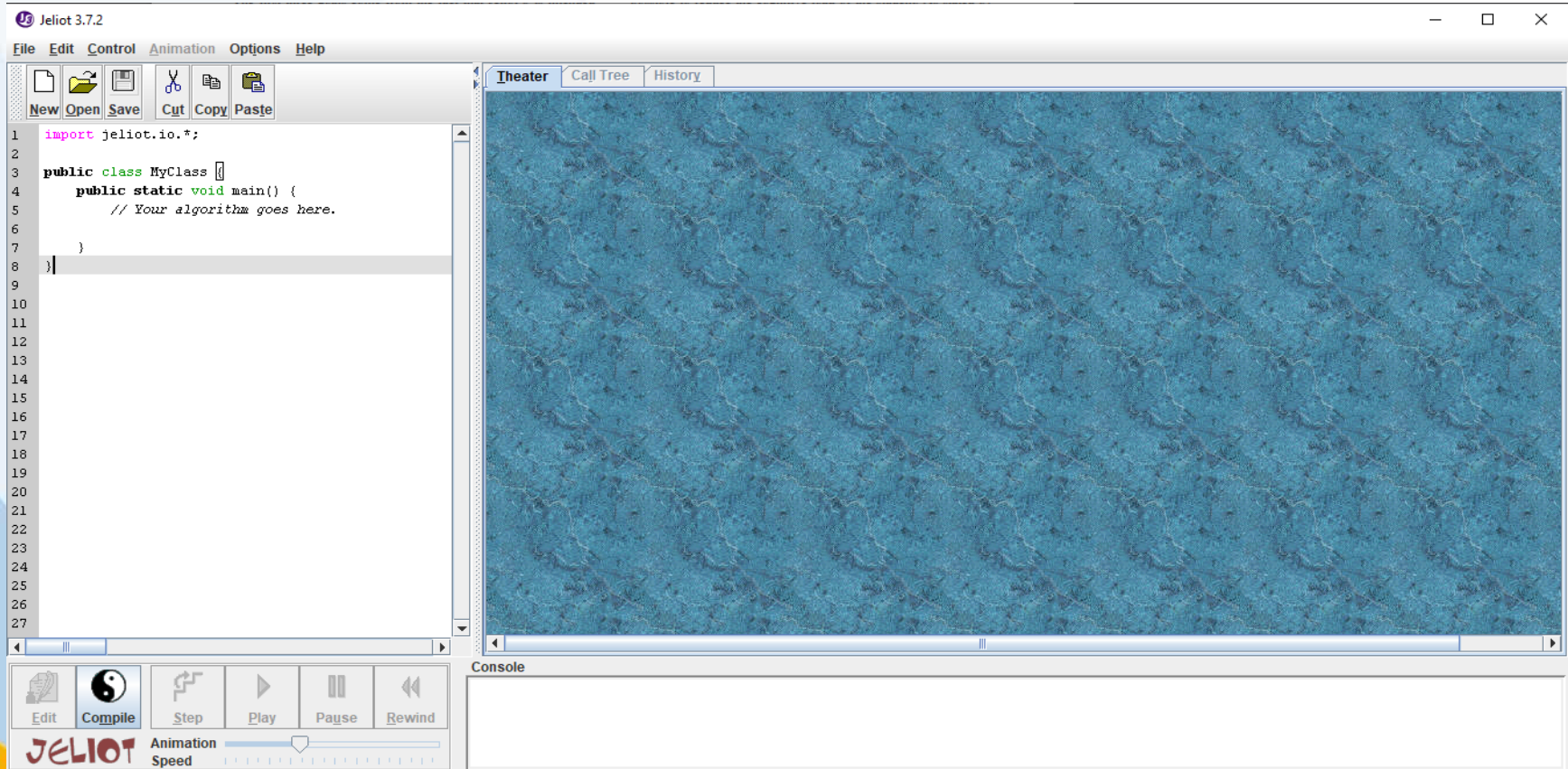
STARLOGO TNG (KLOPER et al., 2009)



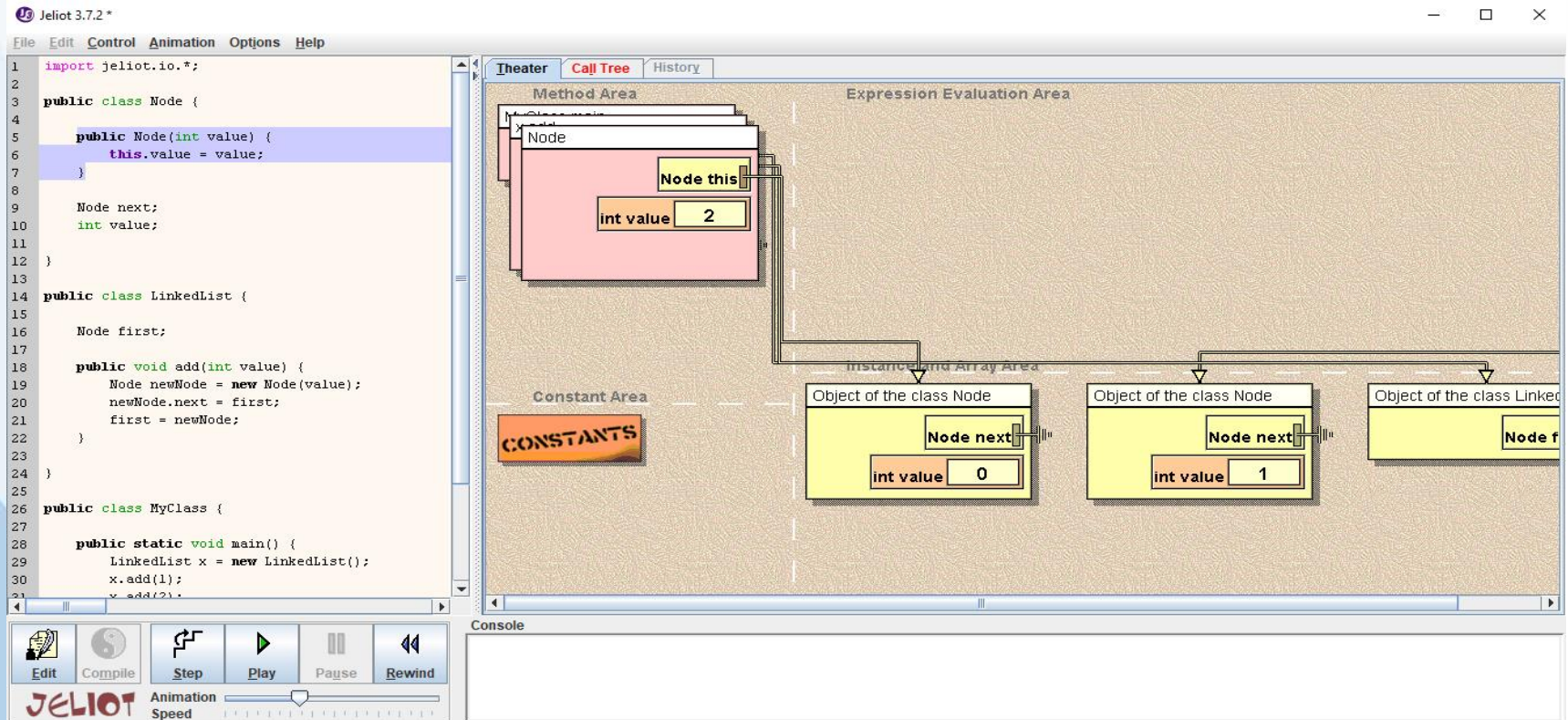
JELIOT 3 (MORENO et al., 2004)

- Tem o objetivo de ajudar novos estudantes a aprender programação procedural e programação orientada a objetos.
- Consiste na programação de um código JAVA, que após ser compilado, possui uma representação visual passo a passo da execução do código.

JELIOT 3 (MORENO et al., 2004)



JELIOT 3 (MORENO et al., 2004)



Proposta

Quadro 1 – Comparativo entre os trabalhos correlatos

Trabalhos Características	<u>Halim</u> et al. (2012)	<u>Klopfer</u> et al. (2009)	Moreno et al. (2004)
Plataforma	Web	Desktop	Desktop
Execução passo a passo	Sim	Sim	Sim
Representação visual	Sim	Sim	Sim
Estilos dos cenários	Pré-definidos	Dinâmicos	Dinâmicos
Tipo de interação	Acompanhamento de código pré-definido	Codificação	Codificação
Tipo de código	Pseudocódigo	Blocos	Código JAVA

Proposta

- O trabalho proposto é uma oportunidade de apresentar uma nova estratégia baseada na combinação de determinadas características já encontradas nos trabalhos apresentados.

Requisitos Funcionais (RF) e não funcionais (RNF)

- Permitir o cadastro de usuários (RF);
- Manter cenários de programação criados pelos usuários (RF);
- Disponibilizar no gerenciamento do cenário a definição do contrato que o estudante deverá seguir (RF);
- Disponibilizar no gerenciamento do cenário a definição de assertivas que validarão a execução do código (RF);

Requisitos Funcionais (RF) e não funcionais (RNF)

- Disponibilizar no gerenciamento do cenário a definição da documentação que irá instruir o estudante sobre o problema (RF);
- Possibilitar no gerenciamento do cenário a definição da implementação correta (RF);
- Permitir a implementação e execução dos cenários de programação por parte do estudante (RF);

Requisitos Funcionais (RF) e não funcionais (RNF)

- Apresentar uma representação visual da execução do código implementado para o estudante (RF);
- Permitir ao estudante o acompanhamento da execução do código (RF);
- Permitir ao estudante consultar a implementação correta para fins de correção (RF);

Requisitos Funcionais (RF) e não funcionais (RNF)

- Possuir modelos de cenários prontos com a implementação dos algoritmos de Array List, Linked List, Hash Map e Bubble Sort (RF);
- Ser construído em cima da arquitetura Web (RNF);
- Utilizar a base PostgreSQL para armazenamento dos dados (RNF);
- Ter a interface web construída com o framework Angular (RNF).

Metodologia

- Levantamento bibliográfico: Realizar o levantamento bibliográfico da utilização de ferramentas como auxiliares no processo educacional de computação, tal como os seus benefícios;

Metodologia

- Elicitação de requisitos: detalhar os requisitos e reavaliá-los, se preciso for, com base nas informações levantadas na etapa de revisão bibliográfica;

Metodologia

- Implementação: realizar o desenvolvimento do sistema com base nos requisitos levantados. A plataforma de desenvolvimento será Web, sendo o backend desenvolvido em Java, e o frontend em javascript com o framework Angular.
 - Para a parte de edição de código dentro do browser será utilizado a ferramenta Monaco Editor e para a representação visual da execução do código a ferramenta Cytoscape;

Metodologia

- Testes: validar o funcionamento da ferramenta como um todo, como a criação de cenários de programação, o acompanhamento de execução e a representação visual do mesmo. Nesta etapa é incluída também a validação dos cenários modelos de Array List, Linked List, Hash Map e Bubble Sort criados.

Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Trabalho de Conclusão de Curso – 2020/2

SMALG: SISTEMA PARA MODELAGEM E APRENDIZADO DE ALGORITMOS

Aluno: Adriner Maranhão de Andrade

Orientador: Dalton Solano dos Reis