

## **ALGORITMO MERGE SORT DESENVOLVIDO EM LINGUAGEM EGUA DE PROGRAMAÇÃO UTILIZADO NO SEQUENCIAMENTO DAS BASES NITROGENADAS**

*Heictor Alves de Oliveira Costa – heictor8@gmail.com*

*Instituto de Estudos Superiores da Amazônia – IESAM/Estácio - Belém  
Av. Gov. José Malcher 1148  
66055-260 – Belém – Pará*

*Denis Carlos Lima Costa – denis.costa@ifpa.edu.br*

*Instituto Federal de Educação, Ciência e Tecnologia do Pará  
Av. Arterial 5-B  
67140-000 – Ananindeua – Pará*

*Lair Aguiar de Meneses – lair.meneses@ifpa.edu.br*

*Instituto Federal de Educação, Ciência e Tecnologia do Pará  
Av. Arterial 5-B  
67140-000 – Ananindeua – Pará*

*Lucas Pompeu Neves – lucaspompeun@gmail.com*

*Instituto de Estudos Superiores da Amazônia – IESAM/Estácio - Belém  
Av. Gov. José Malcher 1148  
66055-260 – Belém – Pará*

*Julio Leite Azancort Neto – julioazancortneto@gmail.com*

*Instituto de Estudos Superiores da Amazônia – IESAM/Estácio - Belém  
Av. Gov. José Malcher 1148  
66055-260 – Belém – Pará*

*Brennus Caio Carvalho da Cruz – brennuscaio@gmail.com*

*Instituto de Estudos Superiores da Amazônia – IESAM/Estácio - Belém  
Av. Gov. José Malcher 1148  
66055-260 – Belém – Pará*

**Resumo:** *O propósito primário desse trabalho é apresentar um algoritmo capaz de realizar uma ordenação, com a finalidade de sequenciar em uma determinada estruturação. O Merge Sort foi desenvolvido na Linguagem Egua de Programação. O projeto foi engendrado por pesquisadores do Instituto Federal de Educação, Ciência e Tecnologia do Pará, com o objetivo de integralizar os conteúdos do Ensino Integrado. O Merge Sort foi aplicado com sucesso no sequenciamento das bases nitrogenadas. Dessa forma, estudantes de Informática e Meio Ambiente, envolveram a Matemática, a Lógica de Programação, a Linguagem de Programação e a Biologia na construção de uma ferramenta que otimizou o processo de ensino-aprendizagem.*

**Palavras-chave:** *Sequenciamento de RNA. Linguagem de Programação. Interdisciplinaridade. Ensino-Aprendizagem. Algoritmo Computacional.*

## 1 INTRODUÇÃO

A possibilidade de mapear e sequenciar o genoma humano e de outras espécies, produziu dados que exigem adequar o conhecimento da Biologia com o conhecimento matemático e representá-los mediante um algoritmo computacional. Esse algoritmo otimizará o processo de sequenciamento, viabilizando estudos mais precisos e antecipando curas e vacinas para eventos como a COVID-19, o coronavírus. A partir de modelos matemáticos foi possível desenvolver um programa de ordenação capaz de agrupar os nucleotídeos de forma a sequenciá-los corretamente.

Segundo Bona (2010), Borba e Penteadó (2011), as escolas que utilizam computadores no processo de ensino-aprendizagem apontam melhorias em estudante com dificuldades de aprendizagem.

Kayama et al. (2014) afirmam, em seu trabalho, que a linguagem dos computadores está adquirindo uma habilidade cada vez mais importante nesse século, e a educação auxiliada pela programação de computadores está se tornando cada vez mais essencial.

Com o argumento de usar a computação como auxílio ao aprendizado Costa et al (2019), apresentam inúmeros scripts, desenvolvidos em linguagem Python de programação, que visam simular os processos matemáticos usados pelas Ciências Naturais.

Por conseguinte, a linguagem utilizada para esses estudos foi construída nos laboratórios do Instituto Federal do Pará, IFPA - Campus Ananindeua, por pesquisadores (professores e alunos) do grupo Gradiente de Modelagem Matemática e Simulação Computacional – GM<sup>2</sup>SC.

Segundo Neves et al. (2020), a linguagem Egua de programação, foi implantada com o objetivo primordial de melhorar o processo de ensino-aprendizagem da Matemática e das Ciências Naturais, sendo aplicada como ferramenta para aglutinar os conhecimentos dessas áreas.

Dessa forma, esse trabalho tem como proposta fundamental apresentar o resultado de um projeto que reúne os fundamentos da Matemática, da Biologia e da Computação em um programa *Merge Sort*, desenvolvido na Linguagem Egua de Programação.

## 2 LINGUAGEM EGUA DE PROGRAMAÇÃO

Com a criação do ProInfo (BRASIL, 2020), foram gerados os NTEs (Núcleos de Tecnologia Educacional), que têm como objetivo auxiliar as escolas em todas as etapas da incorporação das novas tecnologias, desde a sua instalação e manutenção de computadores até a capacitação dos professores

Para Carraher (2001), Moysés (2003) e Piaget (1999), a prática pedagógica sócioconstrutivista simplifica o aprendizado, proporciona mais interesse e principalmente contribui à compreensão dedutiva dos conceitos ensinados na escola. Ou seja, leva-se o aluno a pensar. No entanto um dos principais problemas é o da exclusão digital de diferentes formas: a falta de acesso à tecnologia e a falta de acesso de conhecimento para utilização da tecnologia.

Sendo assim, o IFPA Campus Ananindeua, não se limitou às finalidades dos NTEs, desenvolveu uma linguagem de programação livre e gratuita, na qual as dificuldades do aprendizado de Matemática pudessem ser minimizadas, maximizando o nível de conhecimento adquirido.

Segundo Zanchett et al (2017), a evolução das linguagens permitiu que o objetivo de simplicidade fosse atingido, as linguagens se tornaram mais elegíveis e simples, entretanto o aprendizado de linguagens de programação é uma tarefa não trivial.

A Linguagem Egua de Programação (LEP) tem como missão principal o uso simultâneo de recursos didáticos e metodologias da Matemática mediante as aplicações computacionais.

"Os desafios para formar hoje o engenheiro do amanhã"

Essa linguagem permite observar o algoritmo em sua execução com as operações aritméticas promovendo, entre os alunos, análises e reflexões sobre elas.

Para Fernandes (2019), inúmeros problemas de aprendizado se iniciam na barreira do idioma, visto que apenas 5% da população brasileira apresenta capacidade de comunicação em inglês.

O endereço para acessar a LEP é: <https://egua.tech>. A interface inicial está representada na Figura 1.

Figura 1 – Apresentação da linguagem de programação egua.



Programação simples e moderna

```
classe Programacao {
    simples() {
        escreva("egua");
    }
}

var egua = funcao() {
    retorna "Olá Mundo!";
};
```

DOCS

FERRAMENTAS

GITHUB

Fonte: Autores

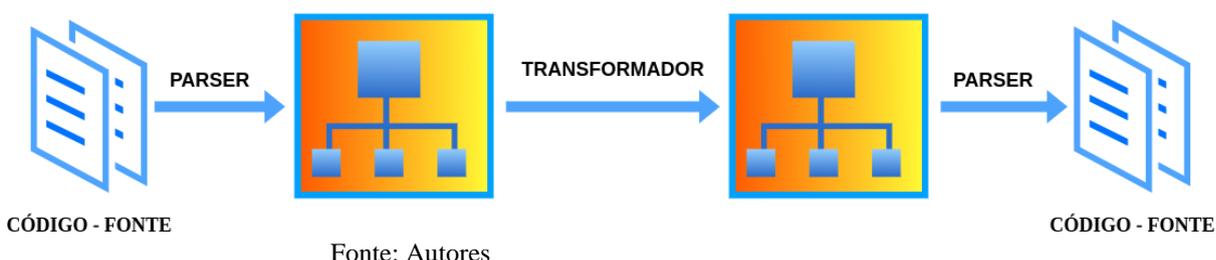
A Linguagem Egua propõe complementar e facilitar o aprendizado de linguagens de programação em propósito geral, seja de linguagens tipadas dinamicamente ou estaticamente. Para isso há uma subdivisão da linguagem em módulos, ainda mantendo a semântica semelhante, apenas com alterações na sintaxe da linguagem, tendendo a facilitar a transição entre os módulos da própria linguagem.

Da mesma maneira como a estrutura da linguagem se torna diferente a fim de atingir um maior número de alunos, há também uma distinção abstrata que rege os dois módulos, sendo um de processamento local, denominada eguap, e outro de processamento remoto com auxílio de servidores em nuvem, denominada eguac. Nesse trabalho serão tratados métodos específicos do módulo eguac.

Havia uma forte necessidade de atender ao aprendizado de estruturas mais robustas na programação, como tipagem estática, alocação estática, estrutura de dados e sintaxe mais otimizada, visando não apenas elegibilidade, mas processamento também. O módulo eguac tem influências fortemente ligadas a linguagem C, seja em escrita ou origem, visto que esta é uma linguagem transpilada em C. Um transpilador, compilador código fonte para código fonte ou

*source to source compiler*, é uma estrutura que viabiliza a transformação do código de uma linguagem para o código de outra linguagem (TEFERA, 2019), sem perder processamento, como em um interpretador. O transpilador recebe o código e utiliza o processo de *parser* de tokenização, conforme a Figura 2, processo responsável por transformar todas as palavras, reservadas ou não, do código fonte em uma lista e atrelar um identificador de função a cada uma delas, gerando a árvore de expressões para obter quais palavras serão transformadas na transposição para outra linguagem, após isso há o processo de transformar a árvore de expressão em Egua em uma árvore equivalente em C, para que haja o processo inverso de transformar a árvore em C para o código fonte em C, para que então haja o processo de compilação em C.

Figura 2 - Processo de transpilação.



Dada a transpilação da linguagem eguac para o C, há a necessidade de compilar o código em C, processo que executa o código escrito em C para código binário de máquina, este processo é executado por meio de um servidor auxiliar dedicado, onde a interface de programação do eguac envia o código transpilado por meio de requisição HTTPS (*Hyper Text Transfer Protocol Secure*), garantindo que a requisição não será interceptada ou interrompida, para que haja a execução do compilador.

A interface e o transpilador do eguac são escritos na linguagem JavaScript e utiliza o framework Angular para efetuar a comunicação com os servidores, assim como se utiliza da biblioteca Jison e Ace para compor a gramática e editor de texto, respectivamente. O servidor do eguac usa os serviços da *Heroku* para processamento, onde possui uma máquina de compilação dedicada para atender as demandas exigidas pelos utilizadores do eguac.

Assim como o módulo eguap, o módulo eguac, também foi submetido a estresses de utilização de rede, onde foram estudados e verificados os consumos em cada compilação do código e o consumo ao entrar na página. O consumo de rede inicial ao entrar na página do eguac possui média de 1.5MB (*MegaBytes*) e em cada requisição para compilar o código apresenta um consumo médio de 230B (*Bytes*), sendo assim há uma forte viabilidade na utilização da ferramenta mesmo com conexões lentas de rede, facilitando o acesso à plataforma para prática de programação em diversos aspectos, como acessibilidades, portabilidade e elegibilidade, conforme apresenta a Figura 3.

Figura 3 – Exemplo do código em eguac.

```
1 // eguac IDE [egua.tech]
2
3 main
4     flutuante X = 1.5;
5
6     se(X > 2)
7         escreva("X é maior que 2");
8     caso(X < 2)
9         escreva("X é menor que 2");
10    contrario
11        escreva("X é igual a 2");
12    fim
13
14    retorna 0;
15 fim
```

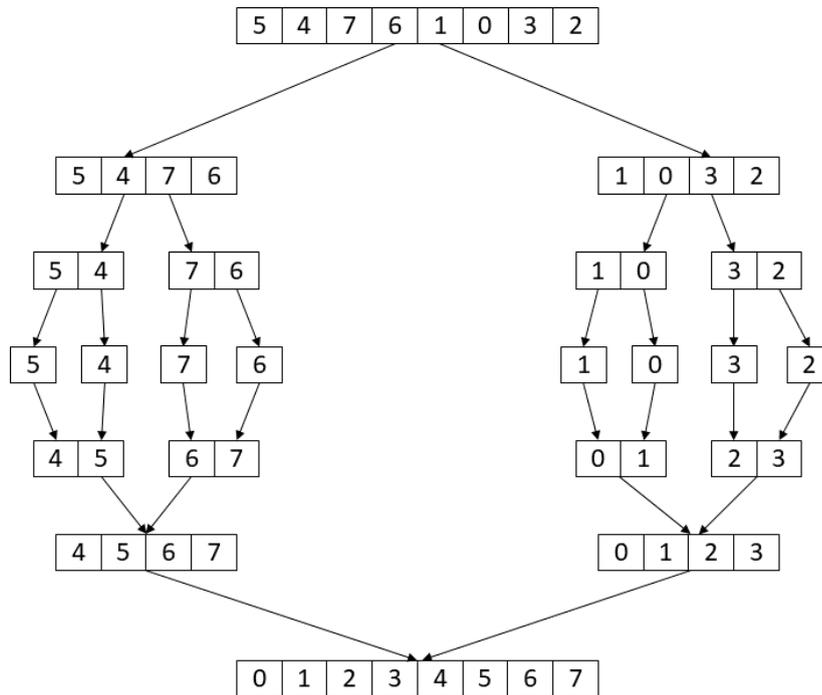
Fonte: Autores

Na figura 3 é apresentado um exemplo do código que utiliza como principal elemento uma estrutura condicional, nele são propostas condições com base no valor da variável X, resultando em uma saída diferente dependo de qual condição for satisfeita. Nesta aplicação são implantadas 3 condicionais, a primeira para  $X > 2$  (X maior que 2), a segunda para  $X < 2$  (X menor que 2), e a terceira para a ocorrência de nenhuma das anteriores serem atendidas. Esse tipo de estrutura foi essencial para a elaboração código que possibilitou a implementação do *Merge Sort* em eguac.

### 3 ALGORITMO MERGE SORT

O *Merge Sort* (ordenação por mistura), é um algoritmo de ordenação criado para sequenciar em determinada ordem, dados dispostos de forma desorganizada. Para realizar tal processo, o algoritmo passa por três pontos principais: dividir, conquistar e combinar. A Figura 4 retrata os passos elencados ao algoritmo.

Figura 4 – Exemplo do código em eguac.



Fonte: Autores

Na primeira etapa, divisão, o algoritmo divide o vetor original em subsequências de forma recursiva até que existam  $n$  vetores formados por apenas um elemento.

No segundo passo, a conquista, os subvetores são comparados aos pares e reorganizados de forma crescente, executando esse processo recursivamente até que restem 2 vetores, nesse momento algoritmo prossegue para o último passo.

No processo final do algoritmo, os 2 vetores restantes são comparados de forma paralela em ordem sequencial de seus elementos, que então, são postos de maneira crescente em um único vetor.

#### 4 SIMULAÇÃO DO MERGE SORT NA BIOLOGIA

Em uma situação problema, dispõe-se de um vetor formado por 4 elementos  $V[0 \dots n - 1]$ , esses elementos estão dispostos de forma aleatória e deseja-se ordená-los em ordem crescente  $V[0] \leq \dots \leq V[n - 1]$ , como representado na tabela abaixo.

Tabela 1 - Vetor 1

8	2	9	5
---	---	---	---

Fonte: Autores

Para realizar essa aplicação do *Merge Sort*, o algoritmo foi implementado na linguagem Egua, em especial no módulo eguac, que dispõe de ferramentas mais eficientes para esse tipo de abordagem. Aplicando o processo de *Merge Sort*, deve-se iniciar o programa dividindo o vetor em sequências menores e comparar os elementos aos pares.

Nesta aplicação em eguac, a divisão e a comparação foram feitas por meio de um

condicional “se” (*if*) dentro de um laço de repetição “para” (*for*). A condicional é responsável por comparar os pares de elementos e inverter sua ordem caso necessário, e o laço de repetição é a ferramenta que permite a recursividade da comparação para os outros elementos ao longo da cadeia, baseado na contagem por conta do incremento no comando “para”.

Em virtude, do vetor original apresentar apenas 4 elementos, basta que a função de comparação aos pares seja feita apenas 2 vezes, para que então o algoritmo prossiga em suas etapas. Após os pares serem organizados, os 2 vetores formados por 2 elementos cada são chamados de Vetor 2 ( $N/2$ ) e Vetor 3 ( $N/2$ ), ilustrados pelas tabelas a seguir.

Tabela 2 - Vetor 2.

2	8
---	---

Fonte: Autores

Tabela 3 - Vetor 3.

5	9
---	---

Fonte: Autores

A próxima etapa é unir os 2 vetores  $N/2$  por comparação paralela em um vetor resultante  $N$ , este último será chamado de Vetor 4 (tabela 4) e terá o mesmo comprimento do vetor original. Para realizar essa união, o *eguc* novamente utiliza estruturas condicionais para comparar os elementos de cada um dos vetores de forma sequencial e então inseri-los no Vetor 4.

Tabela 4 - Vetor 4.

2	5	8	9
---	---	---	---

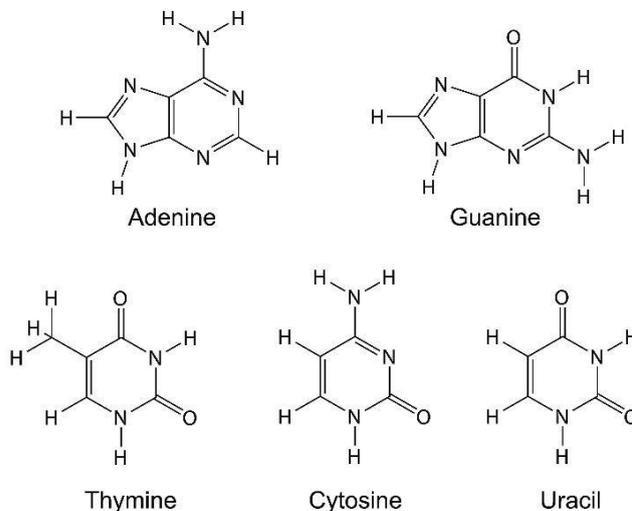
Fonte: Autores

## 5 RESULTADOS E DISCUSSÕES

De maneira análoga ao exemplo numérico, é possível aplicar o algoritmo *Merge Sort* no estudo dos nucleotídeos. O nucleotídeo é um conjunto formado pela associação de 3 moléculas – uma base nitrogenada, um grupamento fosfato e um glicídio do grupo das pentoses, conforme a Figura 5.

Na formação da estrutura do DNA (DeoxyriboNucleic Acid), temos as bases nitrogenadas Adenina (A), Guanina (G), Timina (T) e Citosina (C), e na composição do RNA (RiboNucleic Acid) temos a Adenina, Guanina, Uracila (U) e Citosina. Utilizou-se, para isso uma comparação entre duas sequências de proteínas com objetivo de detectar correspondências e incompatibilidades.

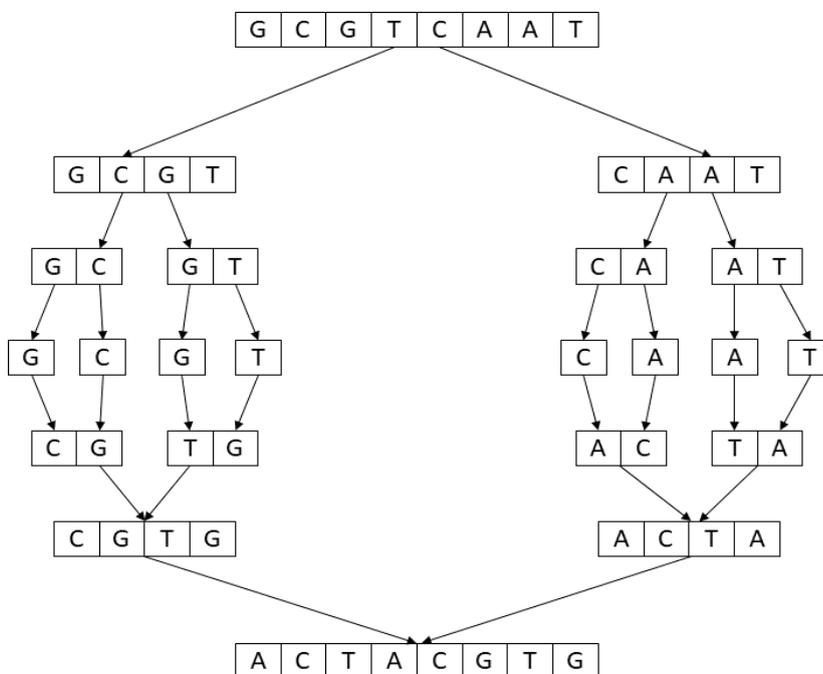
Figura 5 – Bases Nitrogenadas.



Fonte: greelane - [shorturl.at/bcpCG](http://shorturl.at/bcpCG)

No algoritmo *Merge Sort*, os padrões de compatibilidade são, normalmente, determinados para definir as características acumuladas durante o processo de recursividade. A Figura 6 apresenta um exemplo da aplicação desse algoritmo no estudo do DNA, realizado mediante a simulação computacional na Linguagem Egua de Programação.

Figura 6 – Aplicação do Algoritmo *Merge Sort*.



Fonte: Autores

Baseado nos resultados obtidos, estima-se aprimorar o processo de ensino-aprendizagem dos alunos dos cursos integrados do IFPA Campus Ananindeua em 30%, pois houve, de fato,

uma integralização das disciplinas Matemática, Lógica de Programação, Linguagem de Programação e Biologia. Com essa metodologia, os alunos do curso de Informática ajudaram a desenvolver o algoritmo, visto que eles receberam a colaboração dos alunos dos cursos de Meio Ambiente. Essa integralização de disciplinas e entre os cursos foi extremamente importante no processamento das informações que incrementaram essa pesquisa.

## 6 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento do algoritmo *Merge Sort* (MS), na Linguagem Egua de Programação (LEP), foi possível adaptar a sua execução para aplicações em várias disciplinas, como por exemplo a Matemática e a Biologia. A construção dessa ferramenta computacional despertou nos alunos um maior interesse pela integralização dos conteúdos estudados. Durante o planejamento, foi definido que, a intenção do MS e da LEP é de auxiliar de modo eficiente o processo de ensino e aprendizagem, no qual, pelo menos um dos professores envolvidos, detenha o conhecimento sobre os recursos computacionais dessa linguagem e os demais professores contribuíram com as suas expertises. Dessa forma, essa pesquisa foi eficiente em desenvolver um algoritmo com múltiplas aplicações utilizando uma linguagem de programação fácil e eficaz.

## REFERÊNCIAS

BONA, A. S. D. **Portfólio de Matemática**: um instrumento de análise do processo de aprendizagem. 2010. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul. 2011.

BORDA, Marcelo de Carvalho; PENTEADO, Miriam Godoy. **Informática e Educação Matemática. Coleção Tendências em Educação Matemática**. 4ª Edição. Belo Horizonte: Autêntica. 2011.

Brasil. Ministério da Educação. **ProInfo: Programa Nacional de Tecnologia Educacional**. <http://portal.mec.gov.br/proinfo>.

CARRAHER, D. W. Educação tradicional e educação moderna. In: CARRAHER, Terezinha Nunes (Org.). **Aprender pensando**. 15ª. ed. Petrópolis: Vozes, 2001. p. 11-30.

COSTA, Denis C. L.; COSTA, Heictor A. de O; NEVES, Lucas P. **Métodos Matemáticos Aplicados nas Engenharias via Sistemas Computacionais**. Pará: Coleção I – SINEPEM 2019. 2019.

FERNANDES, Vitória. **Apesar de estar no currículo, apenas 1% dos brasileiros realmente fala inglês fluente**. Disponível em: <https://www.moneytimes.com.br/apesar-de-estar-no-curriculo-apenas-1-dos-brasileiros-realmente-fala-ingles-fluente/>.

KAYAMA, M.; SATOH, M.; KOBAYASHI, K.; KUNIMUNE, H.; HASHIMOTO, M.; OTANI, M. Algorithmic thinking learning support system based on student-problem score table analysis. **International Journal of Computer and Communication Engineering**, v.3, n.2, p. 134–140, 2014.

MOYSÉS, L. **Aplicações de Vygotsky à Educação Matemática**. 5ª edição. São Paulo: Papyrus, 2003.

NEVES, Lucas P.; COSTA, Heictor A. de O.; COSTA, Denis C. L.; MENESES, Lair A. de. **Egua: a Linguagem de Programação Desenvolvida para o Ensino da Matemática**. Série Educar: Matemática, Tecnologia, Engenharia. V. 35. Cap. 12. Editora Poisson - Belo Horizonte – MG. Poisson. ISBN: 978-65-86127-44-7. DOI: 10.36229/978-65-86127-44-7. 2020.

PIAGET, J. **Seis estudos de psicologia**. 24ª ed. Rio de Janeiro: Forense Universitária, 1999. (Original de 1964).

TEFERA, Ermias. **Design of amharic programming language with a prototype of tokenizer and parser for selected constructs**. School of Applied Sciences of Near East University. 2020.

ZANCHETT, Guilherme A.; VAHLICK, Adilson; RAABE, André. Jogos de Programar: Abordagem para os Primeiros Contatos dos Estudantes com a Programação. **International Journal on Computational Thinking**, v. 1, n. 1, p. 39-51. 2017.

## MERGE SORT ALGORITHM DEVELOPED IN EGUA PROGRAMMING LANGUAGE USED IN SEQUENCING NITROGENATED BASES

**Abstract:** *The primary purpose of this work is to present an algorithm capable of performing an ordering, with the purpose of sequencing in a given structure. Merge Sort was developed in the Egua Programming Language. The project was created by researchers from the Federal Institute of Education, Science and Technology of Pará, with the objective of integrating the contents of Integrated Education. The Merge Sort was successfully applied in the sequencing of nitrogenous bases. Thus, students of Informatics and Environment, involved Mathematics, Programming Logic, Programming Language and Biology in the construction of a tool that optimized the teaching-learning process.*

**Keywords:** *RNA sequencing. Programming language. Interdisciplinarity. Teaching-Learning. Computational Algorithm.*