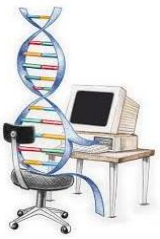




Computação  
Evolucionária

Implementação de um componente na  
linguagem Object Pascal (RAD Studio  
Delphi) para o desenvolvimento de  
algoritmos genéticos para fins didáticos.

Asiel Aldana Ortiz



## Motivação:



- Ausência de bibliotecas de funções para Algoritmos Genéticos na linguagem Object Pascal no mercado.
- Aumento da popularidade do IDE Delphi no desenvolvimento do desenvolvimento de aplicativos **MULTIPLATAFORMA** no Brasil.
- A crescente popularidade da aplicação de algoritmos genéticos em tarefas práticas.



## Objetivo:

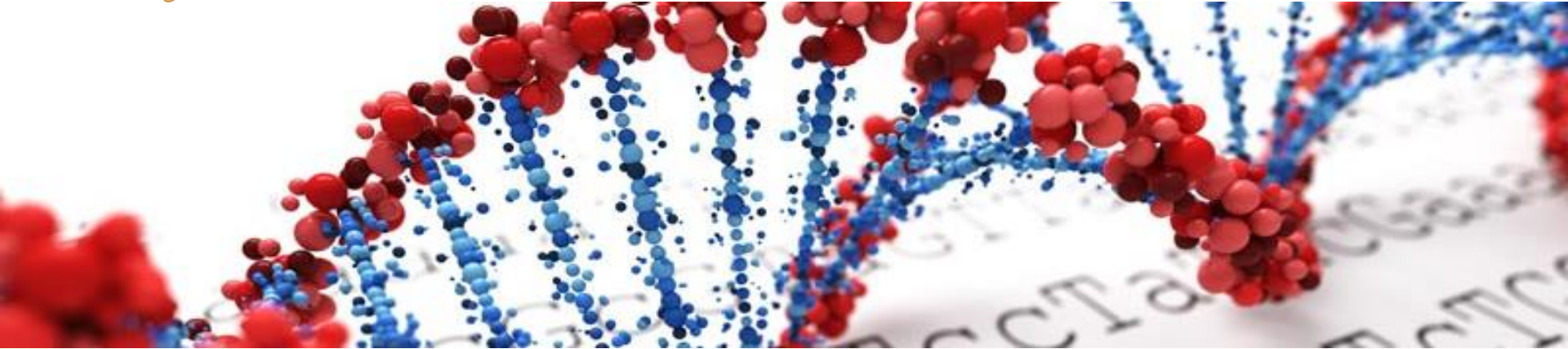


- Desenvolvimento de um componente do **IDE DELPHI** para a implementação de **ALGORITMOS GENÉTICOS** sob o padrão **MULTIPLATAFORMA**.





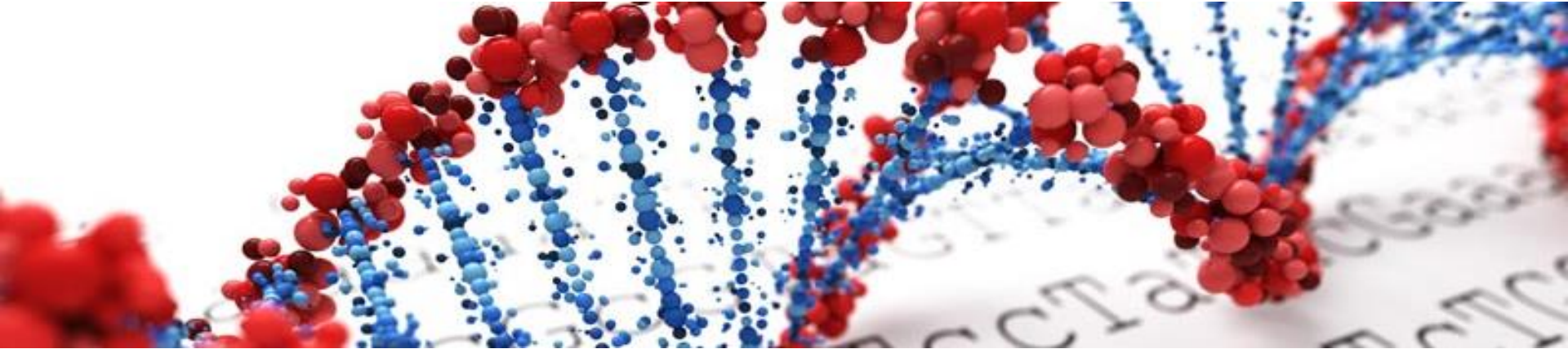
## Metodologia:



- Estudo de diferentes técnicas de programação para a implementação de Algoritmos Genéticos.
- Implementação de algoritmos de seleção, operadores de cruzamentos e mutações, configuração do tamanho da população e do cromossomo, da taxa de crossover e de mutação, do número de indivíduos mantidos pelo elitismo. etc.
- Encapsulamento de código em um componente não-Visual para **RAD STUDIO DELPHI**, mais testes em **MULTIPLATAFORMA**.



# GENETIC ALGORITHMS



constituem uma técnica de busca e otimização

inspirada no princípio Darwiniano de seleção NATURAL

utilizam elementos como a sobrevivência dos mais aptos

Este trabalho tem como objetivo, propor um  
componente para Algoritmos Evolutivos  
denominado

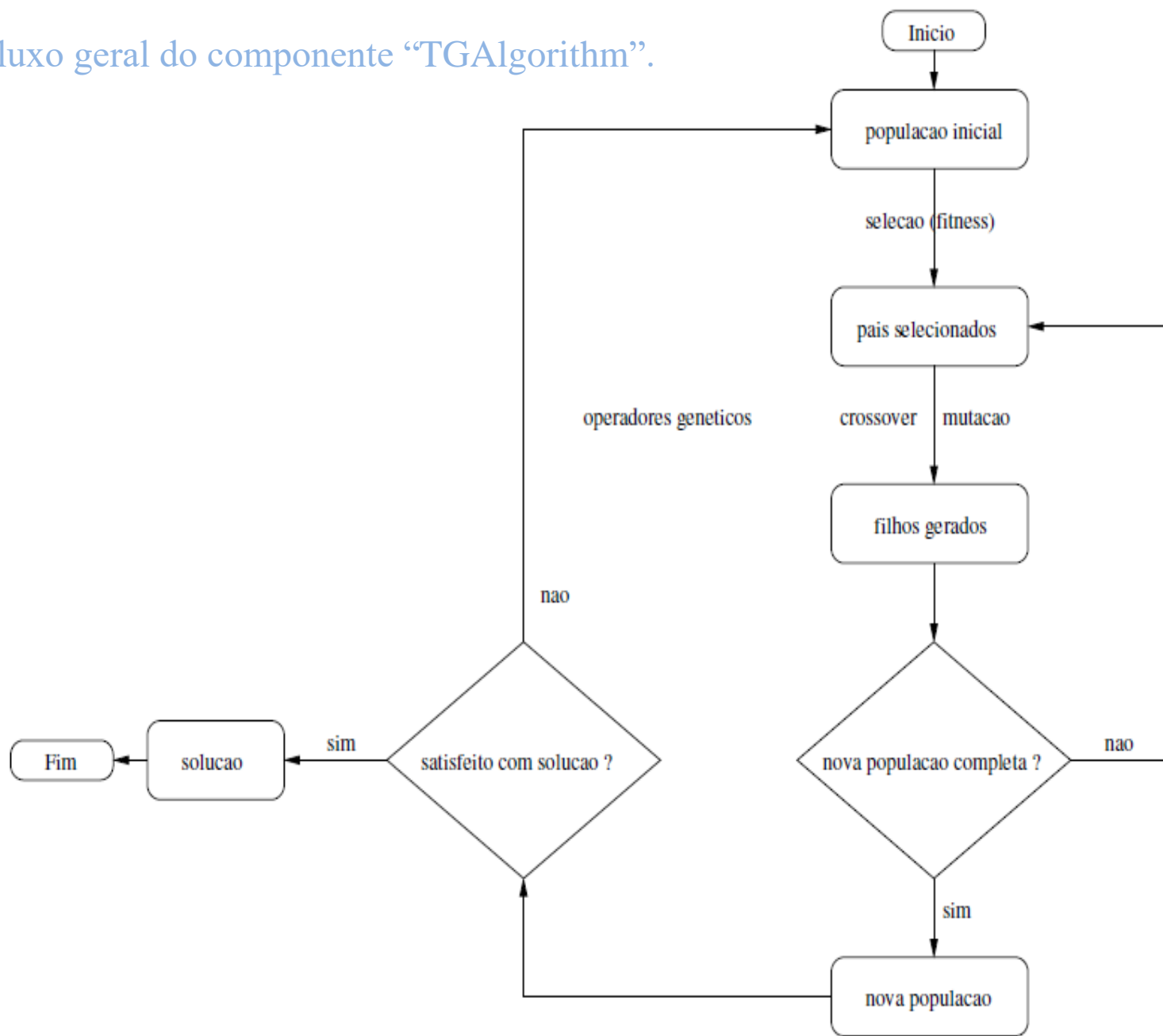
# TGAlgorithm

RAD Studio  
Delphi  
Rio

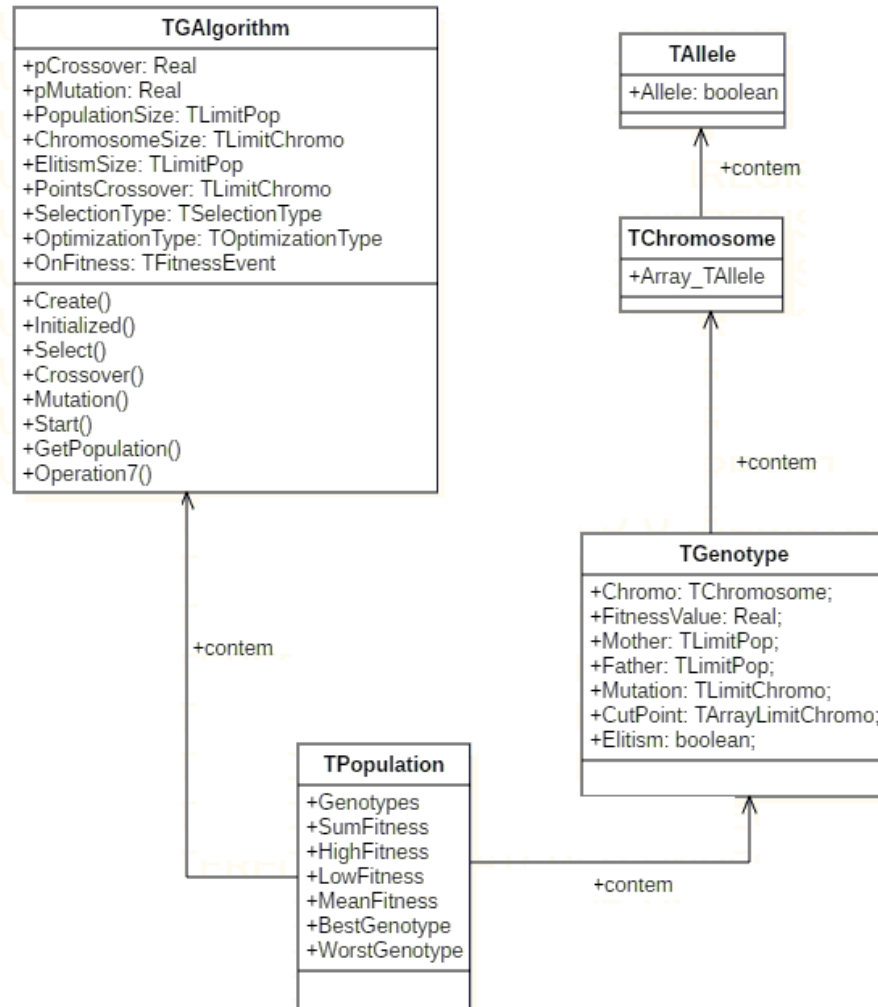
## PEARTICULARITIES DO COMPONENTE:

- Indivíduos binários
- TSelectionType = (stRoulette, stRank);
- TOptimizationType = (otMaximum, otMinimum);
- MaxPopSize = 1000;
- MaxChromoSize = 100
- OnFitness : TFitnessEvent
- crossover (Uniforme,n)

## Fluxo geral do componente “TGAlgorithm”.







Estrutura geral do componente “TGAAlgorithm”

## ATRIBUTOS

- pCrossover: Probabilidade de Crossover.
- pMutation: Probabilidade de mutação.
- PopulationSize: Tamanho da população
- ChromosomeSize: Tamanho do Cromosomo
- ElitismSize: Número de indivíduos mantidos por Elitism.
- PointsCrossover: Pontos de cruzamento.
- SelectionType: Tipo de seleção.
- OptimizationType: Tipo de otimização
- OnFitness: Função de validação

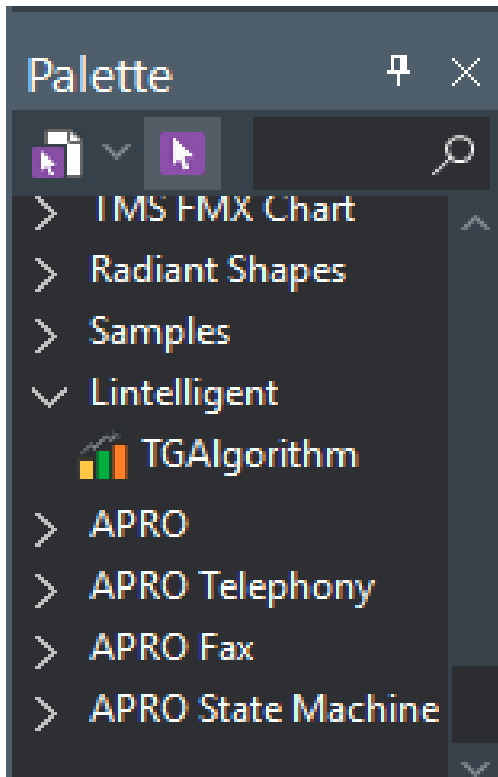
## FUNÇÕES

- Create()
- Initialized()
- BinToInt()
- BinToReal()
- ChromoToString()
- Select()
- Crossover()
- Mutation()
- Start()

# TGAlgorithm

# Resultados

**TGAlgorithm** em Paleta de componentes Delphi.



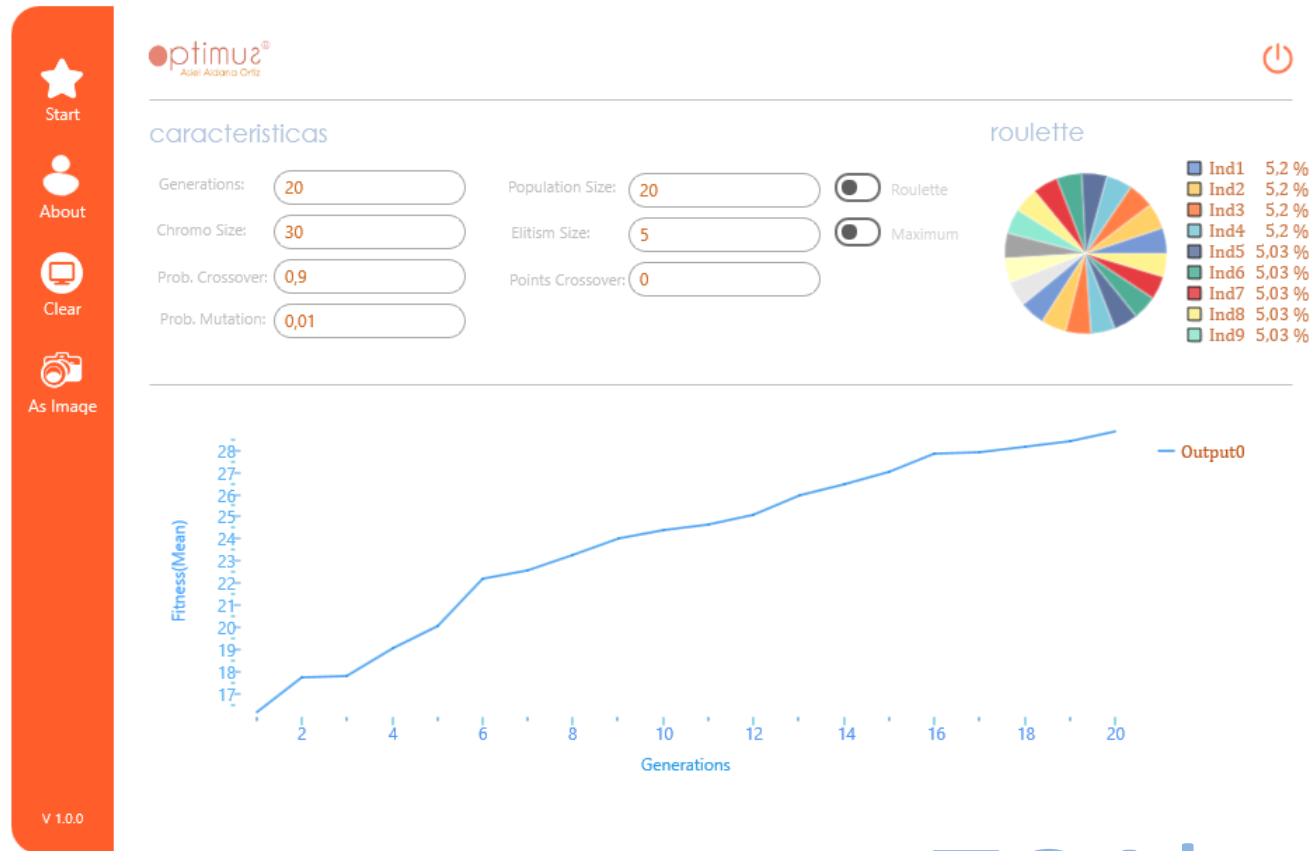
```
with(GA) do
  if(Initialized)then
    begin
      for j := 1 to NGenerations do
        begin
          Start;
          TSeriesA.AddXY(j,GetPopulation.MeanFitness);
        end;

      for i := 1 to GA.PopulationSize do
        begin
          Str := 'Ind' + IntToStr(i) ;
          Series1.Add(GetPopulation.Genotypes[i].FitnessValue,Str);
        end
      end
```

# TGAlgorithm

# Resultados

TGAlgorithm em Paleta de componentes Delphi.



# TGAlgorithm

```
procedure TfrMain.GAFitness(NewChromosome: TChromosome; var FitnessValue: Real);  
var  
    i:integer;  
    Sum:real;  
begin  
    Sum:=0;  
    for i:=1 to GA.ChromosomeSize do  
        if NewChromosome[i] then Sum:=Sum+1;  
    FitnessValue:=Sum;  
end;
```

Para realizar o teste, os seguintes parâmetros foram introduzidos:

pCrossover= 0,9  
pMutation=0,1  
PopulationSize=20  
ChromosomeSize=30  
ElitismSize=5  
PointsCrossover=0  
SelectionType="Roulette"  
OptimizationType="Maximun"

# Prova

# Max



Start



About



Clear



As Image



## características

Generations:

Population Size:

☒ Roulette

Chromo Size:

Elitism Size:

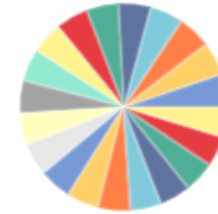
☒ Maximum

Prob. Crossover:

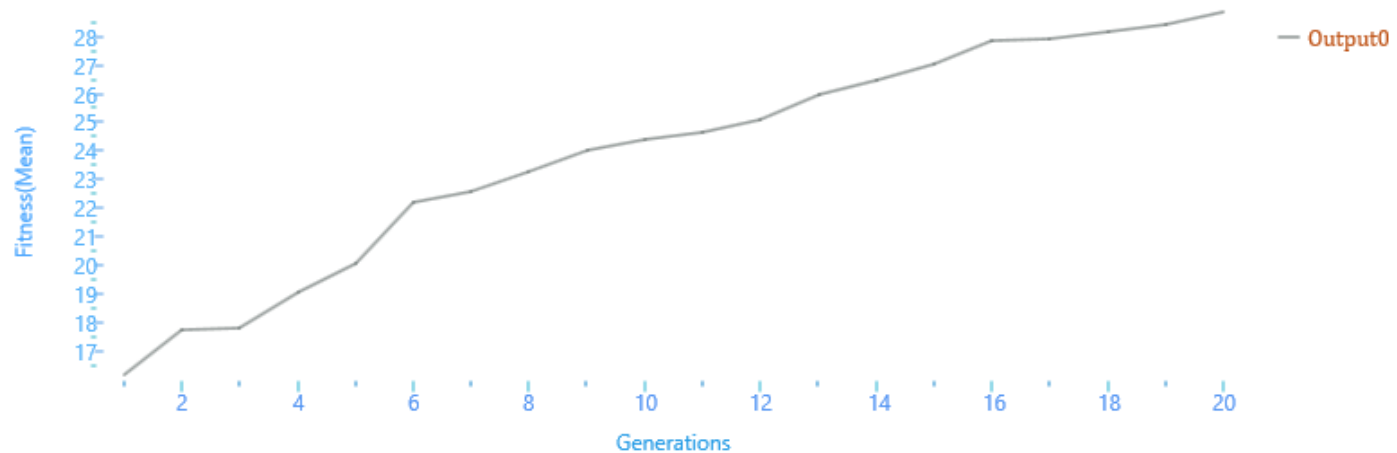
Points Crossover:

Prob. Mutation:

## roulette



Ind1	5,2 %
Ind2	5,2 %
Ind3	5,2 %
Ind4	5,2 %
Ind5	5,03 %
Ind6	5,03 %
Ind7	5,03 %
Ind8	5,03 %
Ind9	5,03 %



V 1.0.0



# Min



Start



About



Clear



As Image

V 1.0.0



## características

Generations:

Population Size:

☐ Roulette

Chromo Size:

Elitism Size:

☒ Minimum

Prob. Crossover:

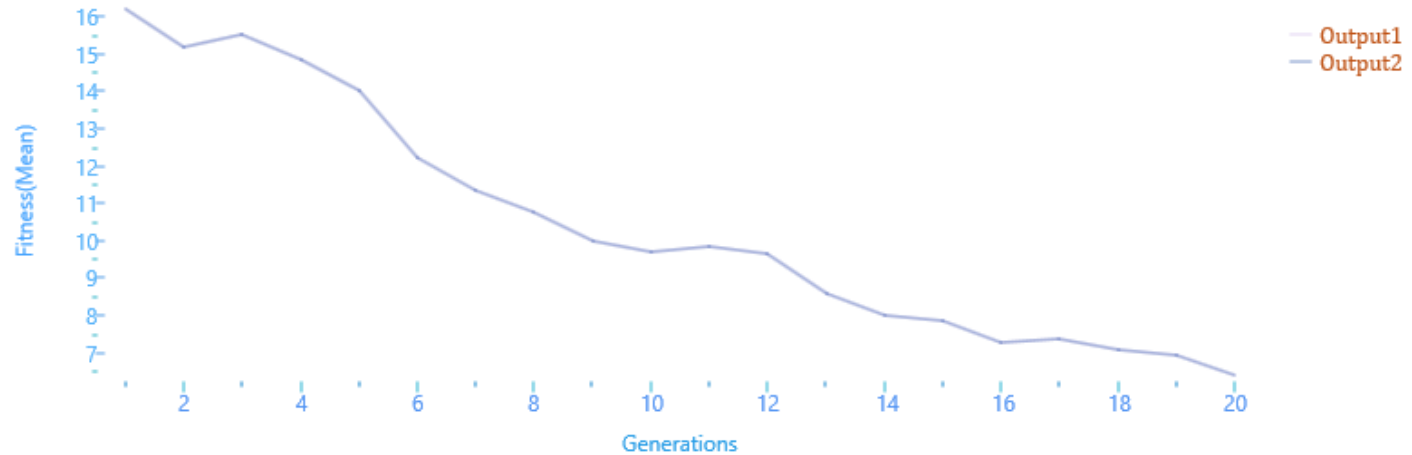
Points Crossover:

Prob. Mutation:

## roulette



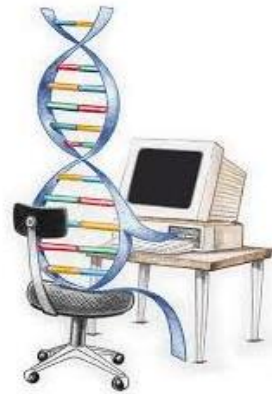
Ind1	2,34 %
Ind2	3,13 %
Ind3	3,91 %
Ind4	3,91 %
Ind5	3,91 %
Ind6	3,91 %
Ind7	3,91 %
Ind8	3,91 %
Ind9	3,91 %



## CONCLUSÕES

O Componente TGAAlgorithm mostrou-se eficaz no que diz respeito à extensibilidade e reusabilidade do código no desenvolvimento de aplicações. Com isto promoveu a simplicidade e a facilidade de manutenção no código das aplicações desenvolvidas.

A eficácia do algoritmo implementado foi demonstrada, e recomenda-se incorporar novos tipos de indivíduos, bem como outros algoritmos para mutação e cruzamento.



Obrigado!!!