



Instituto Infnet

**TESTE DE PERFORMANCE 1
FUNDAMENTOS DO DESENVOLVIMENTO OBJECTIVE C PARA APPLE IOS**

MAGNO VALDETARO DE OLIVEIRA

**RIO DE JANEIRO
FEVEREIRO - 2016**

MAGNO VALDETARO DE OLIVEIRA

TESTE DE PERFORMANCE 1
FUNDAMENTOS DO DESENVOLVIMENTO OBJECTIVE C PARA APPLE IOS

Trabalho apresentado ao Professor
Nabil Safatli
da disciplina **Fundamentos do Desenvolvimento Objective C para Apple iOS**
da turma **GADSLV_B1**, turno **NOITE 1**
do curso de **Computação Básica**.

Instituto Infnet
Rio de Janeiro - 4 de fevereiro de 2016

SUMÁRIO

[1 - INTRODUÇÃO](#)

[2 - DESENVOLVIMENTO DO TRABALHO](#)

[QUESTÃO 1](#)

[QUESTÃO 2](#)

[QUESTÃO 3](#)

1 - INTRODUÇÃO

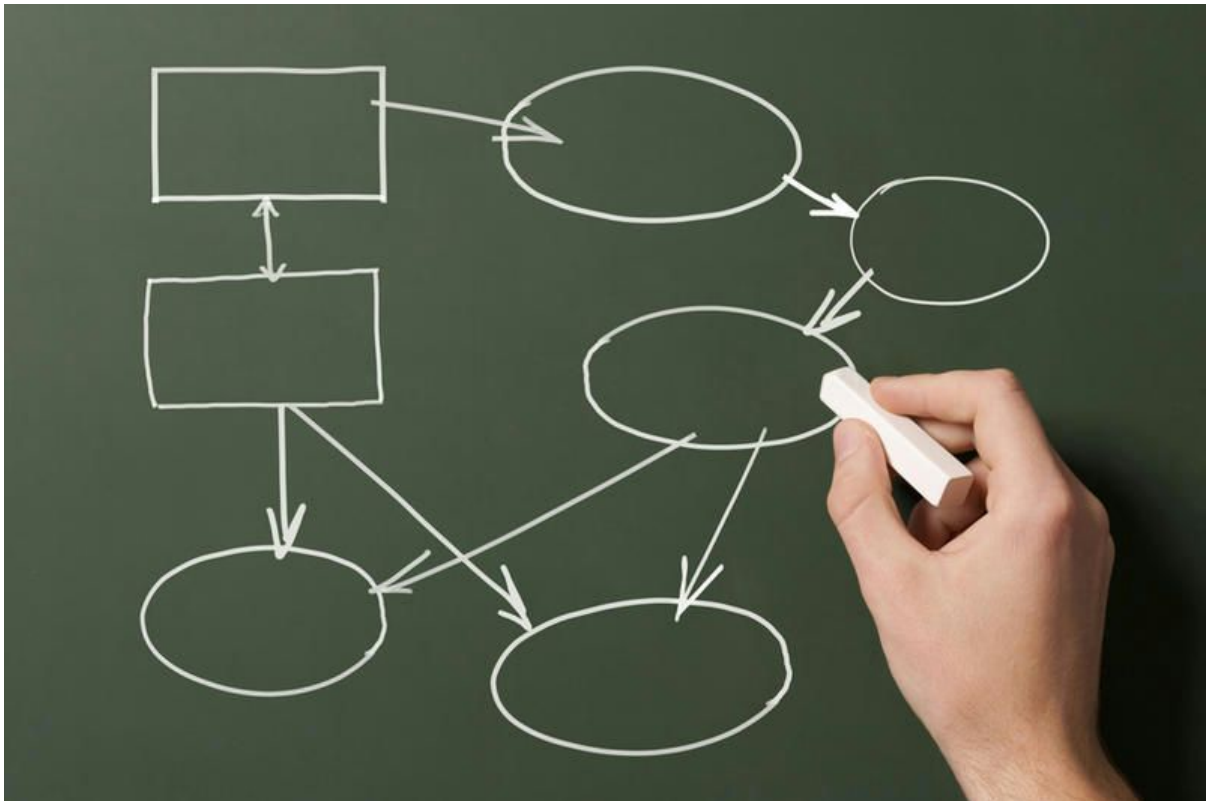
Neste teste de performance respondo as questões relacionadas ao conteúdo aplicado na etapas 1 e 2 da disciplina Fundamentos do Desenvolvimento Objective C para Apple iOS e desenvolvo algoritmos utilizando o VisuAlg.

2 - DESENVOLVIMENTO DO TRABALHO

QUESTÃO 1

a. O que é um algoritmo?

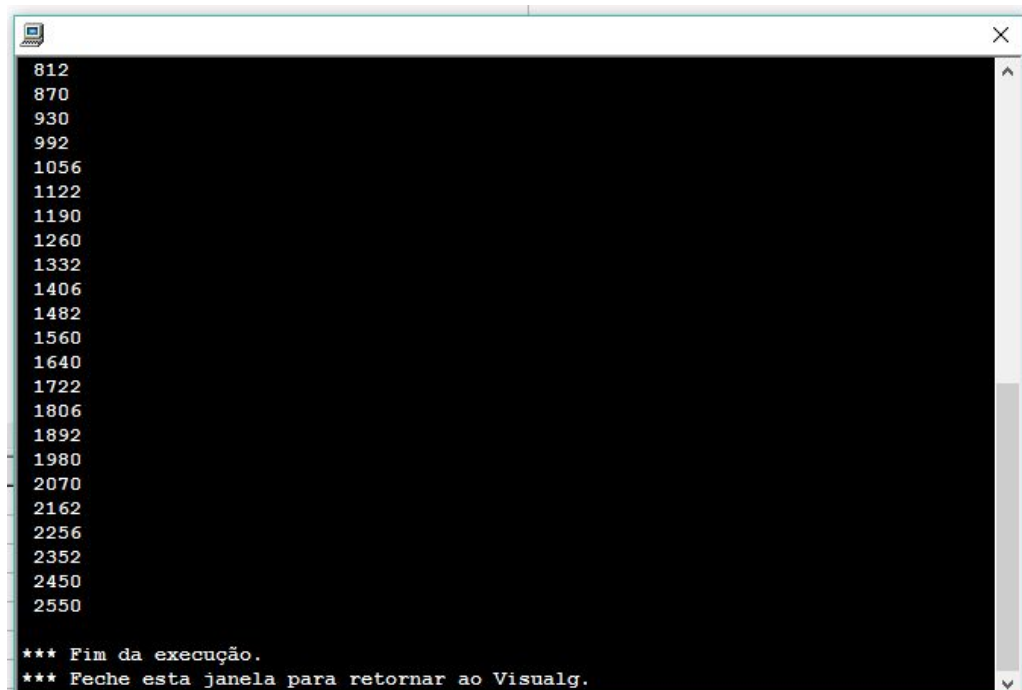
É uma sequência de instruções finita e ordenadas, visando atingir um objetivo.



QUESTÃO 2

a. Faça um programa que calcule e escreva a soma dos primeiros 50 números pares.

```
algoritmo "primeiros_50_pares"
// Seção de Declarações
var
    contador, num, soma : inteiro
inicio
// Seção de Comandos
    num <- 2
    soma <- num
    enquanto contador < 50 faça
        escreval(soma)
        contador <- contador + 1
        num <- num + 2
        soma <- soma + num
    fimenquanto
fimalgoritmo
```



```
812
870
930
992
1056
1122
1190
1260
1332
1406
1482
1560
1640
1722
1806
1892
1980
2070
2162
2256
2352
2450
2550

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

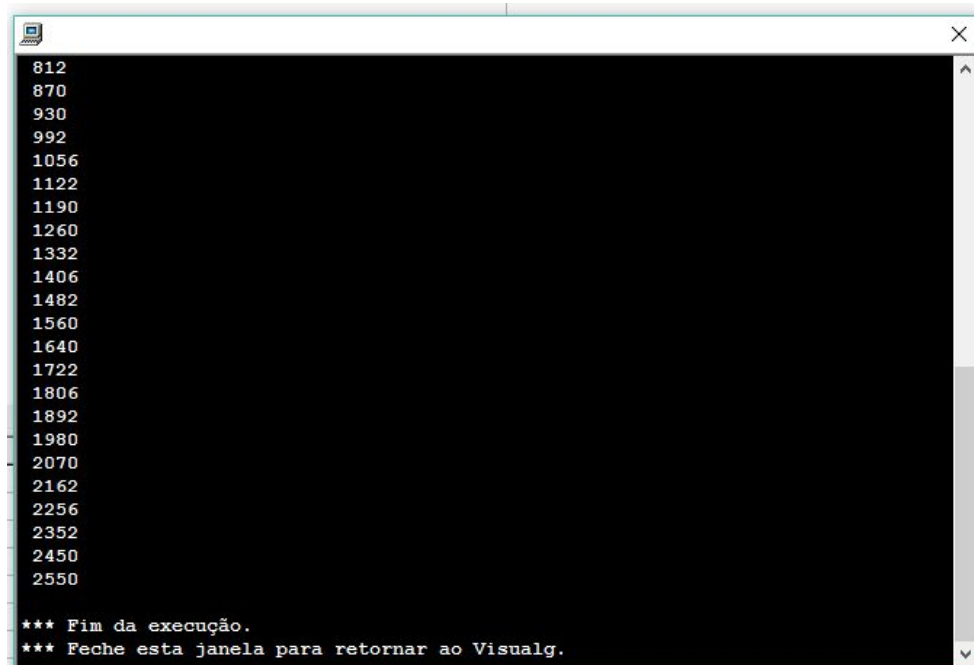
Resultado da Questão 2.a

b. De quantas maneiras você poderia escrever este algoritmo?

Utilizando estruturas de repetição (loop) este algoritmo poderia ser escrito de 3 maneiras.

Mais uma maneira agora utilizando PARA ... EXECUTE / FOR ... DO:

```
algoritmo "primeiros_50_pares"
// Seção de Declarações
var
  i, num, soma : inteiro
inicio
// Seção de Comandos
  num <- 2
  soma <- 2
  para i de 1 ate 50 faca
    escreval(soma)
    num <- num + 2
    soma <- soma + num
  fimpara
fimalgoritmo
```



```
812
870
930
992
1056
1122
1190
1260
1332
1406
1482
1560
1640
1722
1806
1892
1980
2070
2162
2256
2352
2450
2550

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

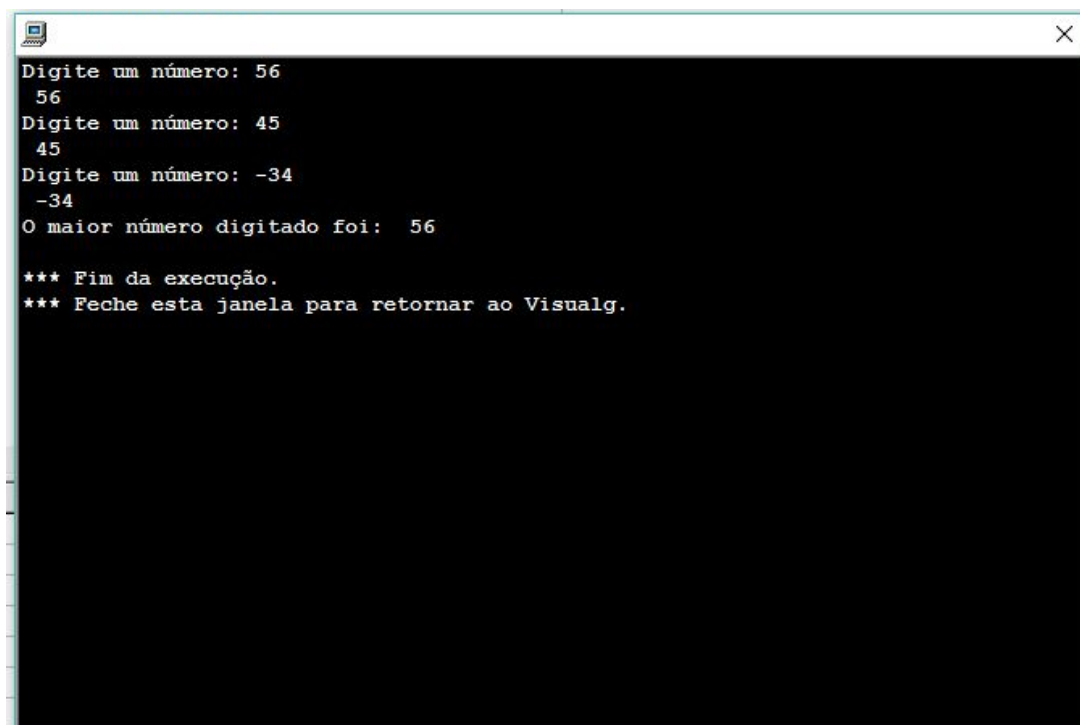
Resultado da Questão 2.b

QUESTÃO 3

a. Faça um programa que receba 3 números e escreva o número maior na tela.

```
algoritmo "qual_o_maior"
// Seção de Declarações
var
  i,n,maior : inteiro

inicio
// Seção de Comandos
  maior <- 0
  enquanto i < 3 faca
    escreva("Digite um número: ")
    leia(n)
    escreval(n)
    i <- i+1
    se n > maior entao
      maior <- n
    fimse
  fimenquanto
  escreval("O maior número digitado foi: ", maior)
finalgoritmo
```



```
Digite um número: 56
56
Digite um número: 45
45
Digite um número: -34
-34
O maior número digitado foi: 56

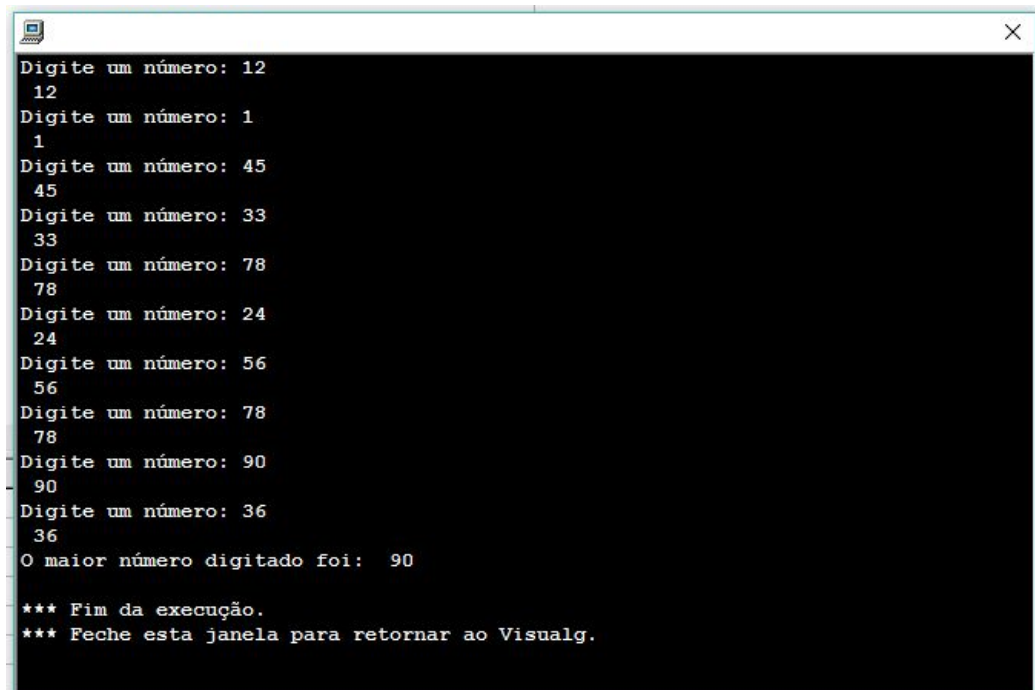
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Resultado da questão 3.a

b. Escreva um algoritmo que receba 10 números e escreva o maior utilizando apenas uma estrutura de repetição e uma condicional

```
algoritmo "qual_o_maior"
// Seção de Declarações
var
i,n,maior : inteiro

inicio
// Seção de Comandos
maior <- 0
enquanto i < 10 faça
    escreva("Digite um número: ")
    leia(n)
    escreval(n)
    i <- i+1
    se n > maior entao
        maior <- n
    fimse
fimenquanto
escreval("O maior número digitado foi: ", maior)
fimalgoritmo
```



```
Digite um número: 12
12
Digite um número: 1
1
Digite um número: 45
45
Digite um número: 33
33
Digite um número: 78
78
Digite um número: 24
24
Digite um número: 56
56
Digite um número: 78
78
Digite um número: 90
90
Digite um número: 36
36
O maior número digitado foi: 90

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Resultado da questão 3.b