

Notação Polonesa Reversa

Na Notação Polonesa Reversa o operador procede os operandos, ou seja, quando há uma operação, esse operador aparece logo depois dos números envolvidos na conta. Dessa forma, a ordem em que os operadores aparecem é a ordem em que a equação deve ser solucionada. Isso retira a necessidade de parênteses para se dar prioridade à certas operações em uma equação e por isso é bastante utilizada na computação Citation[?, p.219]. Here's Another Citation[?]

Arquitetura do Projeto

O projeto está dividido apenas em três arquivos. Pilha.h, com a declaração de todas as funções, não apenas as funções pilha. Pilha.c, onde as funções são desenvolvidas. Mainpilha.c onde as funções são chamadas. Esse projeto foi criado no sistema Ubuntu 16.04, escrito no VisualCode e Sublime. Compilado e executado no GDB com c99. As exatas linhas utilizadas para compilação foram:

```
c99 -Wall -g -c pilha.c
c99 -Wall -g -o ppe mainpilha.c pilha.o
O nome do programa executável, dessa forma é ppe.
```

Funcionamento do programa

Esse programa funciona apenas com expressões contendo números, não letras.

O programa tem em sua main um menu com funcionamento baseado em switch, dessa forma assim que o programa for aberto, a seguinte tela deve aparecer

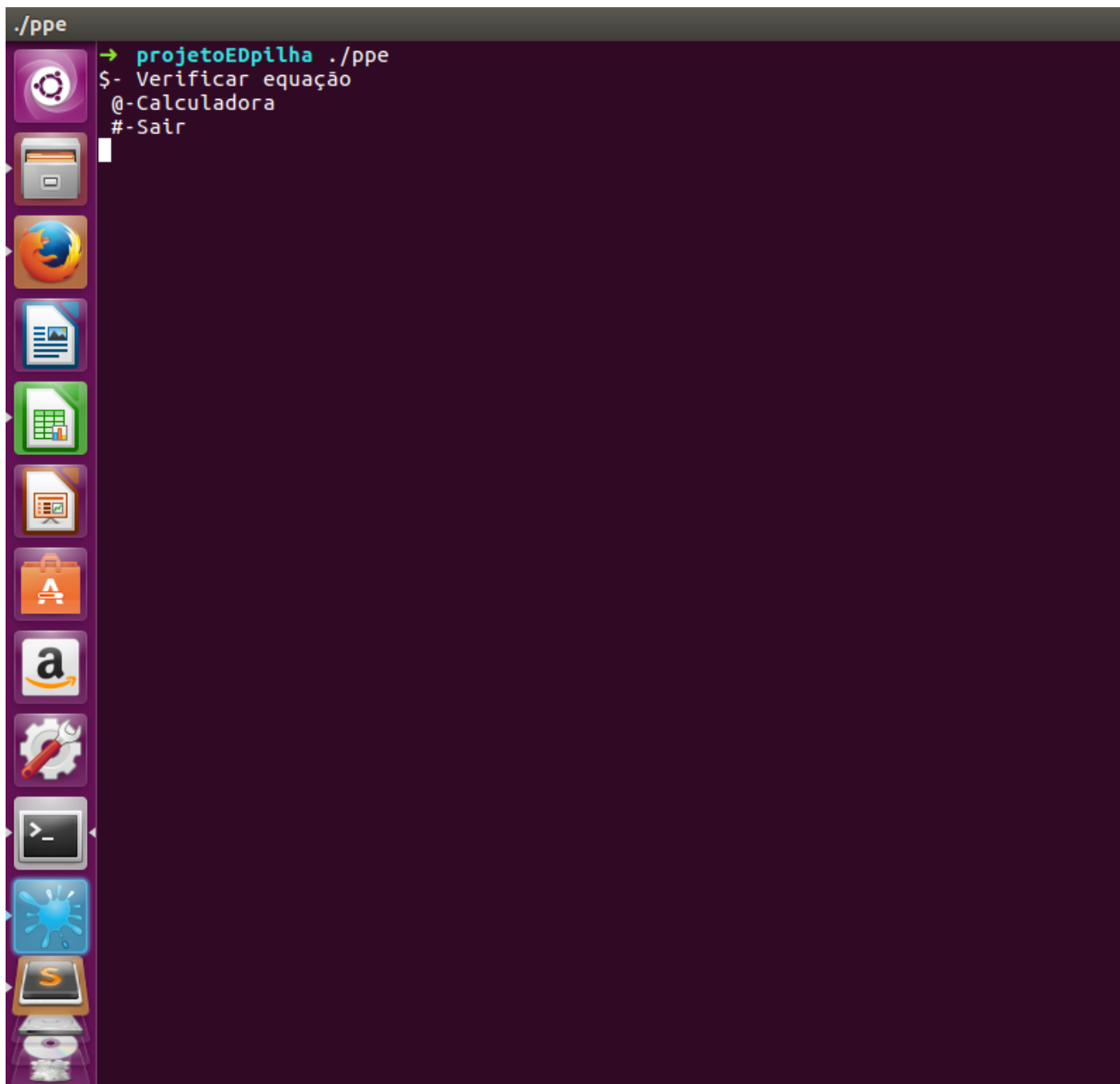


Figura 1: Menu Inicial

Ao escolher o caractere cifrão ativa-se o modo equação.

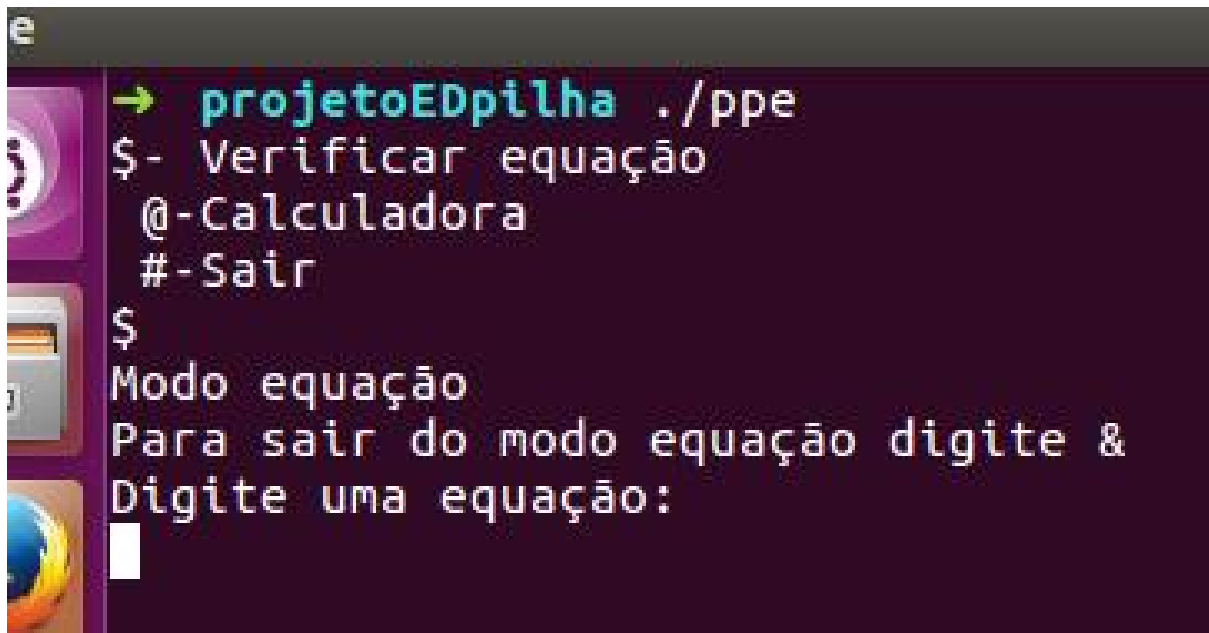


Figura 2: Menu Equação

Nessa hora, se digitar `&` você voltará ao menu inicial, caso contrario digite uma equação. O programa retornará se ela é válida, sua pos-fixa e seu resultado. Então retornará automaticamente ao menu inicial.

```
→ ~ cd Documentos
→ Documentos cd projetoEDpilha
→ projetoEDpilha ./ppe
$- Verificar equação
  @-Calculadora
  #-Sair
$
Modo equação
Para sair do modo equação digite &
Digite uma equação:
[20*(2+3)-1]
expressão verdadeira
  saída 20 2 3 + * 1 -
resultado 99.000000

$- Verificar equação
  @-Calculadora
  #-Sair
```

Figura 3: Calculo da Equação

Já ao digitar arroba, o usuário é mandado ao menu da Calculadora que inicia uma pilha vazia. Digite um número e ele te mostrará a pilha.

```
→ projetoEDpilha ./ppe
$- Verificar equação
@- Calculadora
#- Sair
@
Modo calculadora
Pilha vazia
█
```

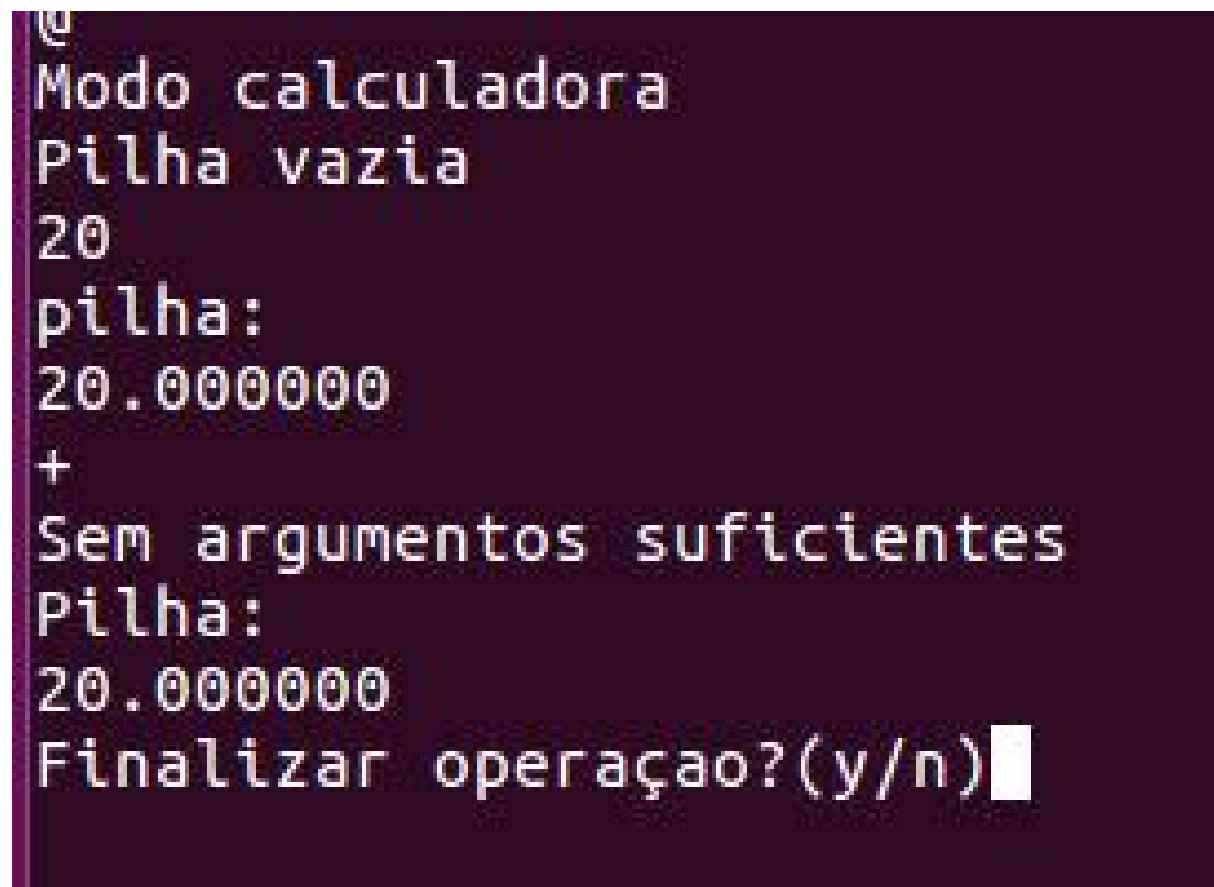
Figura 4: Calculadora com pilha vazia

```
@
Modo calculadora
Pilha vazia
20
pilha:
20.0000000
█
```

Figura 5: Pilha não vazia

Agora, digite um outro número e um operador. caso você digite apenas o operador, receberá

uma mensagem de erro.

A terminal window with a black background and green text. The text shows a calculator program in 'Modo calculadora' (Calculator mode). It displays 'Pilha vazia' (Empty stack) and the number '20'. Then it shows 'pilha:' (stack:) and '20.000000'. A '+' sign is entered. The program then displays an error message: 'Sem argumentos suficientes' (Not enough arguments). It then shows 'Pilha:' (stack:) and '20.000000'. Finally, it asks 'Finalizar operacao?(y/n)' (Finish operation?(y/n)) with a cursor at the end.

```
@
Modo calculadora
Pilha vazia
20
pilha:
20.000000
+
Sem argumentos suficientes
Pilha:
20.000000
Finalizar operacao?(y/n)
```

Figura 6: Argumento insuficiente

```
c
Modo calculadora
Pilha vazia
20
pilha:
20.0000000
2
pilha fim:
2.0000000
20.0000000
+
Pilha:
22.0000000
Finalizar operacao?(y/n) ☐
```

Figura 7: Calculo

Ao terminar, ele pergunta ao usuário se quer finalizar a operação. Caso queira, ele apresentará o resultado final e voltará ao menu inicial. Caso contrário, continuará na operação.

```
c
Modo calculadora
Pilha vazia
20
pilha:
20.000000
2
pilha fim:
2.000000
20.000000
+
Pilha:
22.000000
Finalizar operacao?(y/n)y
pilha fim:
22.000000
fim
$- Verificar equação
@-Calculadora
#-Sair
```

Figura 8: Fim da calculadora