

Trabalho 1 - PPLF

GUILHERME ZAMBERLAM POMINI

RA: 99345

Introdução

Introdução - Objetivos

- ▶ Criar um programa utilizando a linguagem Racket para fazer cálculos e simulações envolvendo ações da bolsa de valores.
- ▶ Funcionalidades:
 - ▶ Correlação
 - ▶ Media Móvel Simples e Exponencial
 - ▶ MACD e RSI
 - ▶ Ordenação das listas de ações por data
 - ▶ Calcular próxima/anterior data válida
 - ▶ Gerar gráficos
 - ▶ Simulação de compra e venda, dia a dia das ações

Implementação do Código

Leitura e construção

```
; Arquivo csv -> Lista de listas
; Função que recebe um arquivo .csv e retorna uma lista de listas separando por quebra de linha, sendo cada linha uma lista.
(define (ler_arquivo nome_arquivo)
  (call-with-input-file nome_arquivo
    csv->list))

; Le o .csv para arquivo
(define arquivo (ler_arquivo "dados.csv"))

; Define a estrutura dos dados das ações
(struct dados_acoes (nome date close) #:transparent)

; String -> String
; Inverte o ano com o dia de uma string de data do formato xx/xx/xxxx
(define (inverte string)
  (define x (reverse(string-split string "/" #:repeat? #t)))
  (string-append (first x) "/" (second x) "/" (third x)))

; Lista de strings -> dados_acoes
; Recebe uma lista de strings e devolve um dado_acoes com os valores das strings
(define (constroi dados)
  (dados_acoes (first dados) (inverte (second dados)) (string->number (sixth dados))))
```

Leitura e construção

```
; String, Lista de Listas de String -> Lista de Listas de dados_acoes
; Filtra a lista de todas as empresas pela string passada, retornando uma lista de listas com somente as empresas com o nome passado
(define (filtra_empresas nome lst)
  (cond [(empty? lst) empty]
        [(equal? nome (first (first lst))) (cons (constroi (first lst)) (filtra_empresas nome (rest lst)))]
        [else (filtra_empresas nome (rest lst))]))

;Estrutura para armazenar os dados do Google
(define google (filtra_empresas "Google" arquivo))

;Estrutura para armazenar os dados da Petrobras
(define petrobras (filtra_empresas "Petrobras" arquivo))

;Estrutura para armazenar os dados da Microsoft
(define microsoft (filtra_empresas "Microsoft" arquivo))
```

Ordenação por Data

```
; Listas desordenadas de cada uma das ações para testes de ordenação
(define google_desordenado (shuffle google))
(define petrobras_desordenado (shuffle petrobras))
(define microsoft_desordenado (shuffle microsoft))

; dados_acoes -> dados_acoes
; Recebe um dados_acoes e devolve o mesmo com os campos da data com o ano e o dia trocados
(define (inverte_data_acao acao)
  (dados_acoes (dados_acoes-nome acao) (inverte(dados_acoes-date acao)) (dados_acoes-close acao)))

; Lista de dados_acoes -> lista de dados_acoes
; Recebe uma lista de dados_acoes e devolve a mesma lista mas com todas as datas invertidas seguindo a função inverte_data
(define (inverte_acoes lst)
  (cond [(empty? lst) empty]
        [else (cons (inverte_data_acao (first lst)) (inverte_acoes (rest lst))))]))

; Lista de dados_acoes -> Lista de dados_acoes
; Ordena uma lista de dados_acoes por data, primeiramente declara uma nova lista que é igual a passada como parâmetro mas com a data
; de todos os elementos da forma ANO/MES/DIA. Em seguida é ordenado a string data seguindo em ordem crescente de caracteres.
; Por último é feita novamente a inversão da data da lista ordenada para voltar para o formato DIA/MES/ANO
(define (ordena_data lst)
  (define x (inverte_acoes lst))
  (inverte_acoes (sort x string<? #:key dados_acoes-date)))
```

Data Próxima e Anterior - Auxiliar

```
; Lista de dados_acoes -> Lista de Strings
; Gera uma lista com todas as datas válidas
(define (gera_datas acao)
  (cond [(empty? acao) empty]
        [else (cons (dados_acoes-date (first acao)) (gera_datas (rest acao)))]))

; Estrutura de lista que armazena todas as datas válidas
(define datas (gera_datas google))

; lista de strings -> lista de strings
; Recebe a lista de datas e retorna a mesma mas com no formato ANO/MES/DIA
(define (inverte_datas lista_datas)
  (cond [(empty? lista_datas) empty]
        [else (cons (inverte (first lista_datas)) (inverte_datas (rest lista_datas)))]))
```


Data Próxima e Anterior - Principal

```
; String, dados_acao -> String
; Devolve a proxima data válida
(define (proxima_data lista_datas data opcao)
  (cond [(empty? lista_datas) empty]
        [(opcao data (first lista_datas)) (first lista_datas)]
        [(string=? data (first lista_datas))
         (cond [(empty? (rest lista_datas)) empty]
               [else (first (rest lista_datas))])]
        [else (proxima_data (rest lista_datas) data opcao)]))

; String -> String
; Chama a proxima_data para calcular a data posterior da passada e a devolve
(define (posterior_data_valida data)
  (define nova_data (proxima_data (inverte_dadas datas) (inverte data) string<?))
  (cond [(empty? nova_data) empty]
        [else (inverte nova_data)]))

; String -> String
; Chama a proxima_data para calcular a data anterior da passada e a devolve
(define (anterior_data_valida data)
  (define nova_data (proxima_data (reverse (inverte_dadas datas)) (inverte data) string>?))
  (cond [(empty? nova_data) empty]
        [else (inverte nova_data)]))
```

Correlação - Auxiliares

```
; Lista de dados_acoes -> Numero
; Devolve a soma da coluna de fechamento de uma lista de ações
(define (somacoluna lst)
  (cond [(empty? lst) 0]
        [else (+ (dados_acoes-close (first lst)) (somacoluna (rest lst))))]))

; Lista de dados_acoes -> Numero
; Devolve a soma das colunas de fechamento de 2 ações multiplicados
; (multiplica o valor da lista1 pela lista2 para todos os elementos enquanto os soma)
(define (somamultiplicado lst1 lst2)
  (cond [(empty? lst1) 0]
        [else (+ (* (dados_acoes-close (first lst1)) (dados_acoes-close (first lst2))) (somamultiplicado (rest lst1) (rest lst2))))]))
```

Correlação - Principal

```
; Lista de dados_acoes, Lista de dados_acoes -> Numero  
; Calcula o indice de correlação entre duas empresas  
(define (correlacao acaol acao2)  
  (define x (soma coluna acaol))  
  (define y (soma coluna acao2))  
  (define xy (soma multiplicado acaol acao2))  
  (define xquadrado (soma multiplicado acaol acaol))  
  (define yquadrado (soma multiplicado acao2 acao2))  
  (define m (length acaol))  
  (/ (- xy (/ (* x y) m)) (sqrt (* (- yquadrado (/ (* y y) m)) (- xquadrado (/ (* x x) m))))))
```

Media Móvel Simples

```
; Lista de dados_acoes, Numero -> Numero
; Devolve a soma dos Numero primeiro termos da lista
(define (soma_gtd acao qtd)
  (cond [(< (length acao) qtd) -1]
        [(= 0 qtd) 0]
        [else (+ (dados_acoes-close (first acao)) (soma_gtd (rest acao) (sub1 qtd)))]))

; Lista de dados_acoes, Numero -> Lista de Numeros
; Calcula Media Movel ao longo dos dias para um determinado numero de dias, devolve uma lista de medias moveis ao longo do tempo
(define (media_movel acao dias)
  (cond [(> dias (length acao)) empty]
        [(empty? acao) empty]
        [else (cons ( / (soma_gtd acao dias) dias) (media_movel (rest acao) dias))]))
```

Media Móvel Exponencial

```
; Lista de dados_acoes, Numero, Numero -> Lista de Numeros
; Calcula Media Movel Exponencial
(define (media_movel_exponencial lst n media_anterior)
  (define k (/ 2 (+ n 1)))
  (cond [(empty? lst) empty]
        [else
         (define media_exponencial (+ (* k (- (dados_acoes-close (first lst)) media_anterior)) media_anterior))
         (cons media_exponencial (media_movel_exponencial (rest lst) n media_exponencial)))]))

; Lista, numero -> lista
; Avança na lista numero posições
(define (avanca_lista acao qtd)
  (cond [(= qtd 0) acao]
        [else (avanca_lista (rest acao) (sub1 qtd))]))

; Chama o calculo da media exponencial
(define (media_exponencial acao n)
  (cond [(> n (length acao)) "Valor Inválido, superior ao tamanho do período"]
        [else
         (define primeira_media (first (media_movel acao n)))
         (cons primeira_media (media_movel_exponencial (avanca_lista acao n) n primeira_media)))]))
```

MACD

```
; Lista, Lista -> Lista
; Diferença entre duas listas
(define (listal_menos_lista2 lst1 lst2)
  (cond [(empty? lst1) empty]
        [(empty? lst2) empty]
        [else (cons (- (first lst1) (first lst2)) (listal_menos_lista2 (rest lst1) (rest lst2)))]))

; Lista -> Lista
; MACD = Media exponencial de 12 dias - Media exponencial de 26 dias
(define (macd acao)
  (listal_menos_lista2 (media_exponencial acao 12) (media_exponencial acao 26)))
```

RSI - Auxiliares

```
; Soma Média de acordo com o tipo, > para Ganho Medio, < para Perda Media
(define (soma_media tipo lista quantidade)
  (cond [(= 0 quantidade) 0]
        [(tipo (dados_acoes-close (second lista)) (dados_acoes-close (first lista)))
         (define valor (abs (- (dados_acoes-close (second lista)) (dados_acoes-close (first lista)))))
         (+ valor (soma_media tipo (rest lista) (sub1 quantidade)))]
        [else (soma_media tipo (rest lista) (sub1 quantidade))]))

; Primeira Perda Média
(define (primeira_perda_media lista quantidade)
  (/ (soma_media < lista (sub1 quantidade)) quantidade))

; Primeiro Ganho Médio
(define (primeiro_ganho_medio lista quantidade)
  (/ (soma_media > lista (sub1 quantidade)) quantidade))
```

RSI - Principal

```
; Força Relativa
(define (forca_relativa acao quantidade)
  (/ (primeiro_ganho_medio acao quantidade) (primeira_perda_media acao quantidade)))

; RSI
(define (calcula_rsi acao original quantidade)
  (cond [(empty? acao) empty]
        [else
         (define valor (- 100 (/ 100(+ 1 (forca_relativa original quantidade)))))
         (cons valor (calcula_rsi (rest acao) (rest original) quantidade))]))

; Chama o cálculo do RSI
(define (rsi acao quantidade)
  (calcula_rsi (avanca_lista acao quantidade) acao quantidade))
```


Testes Unitários

```
; Testes unitários de ordenação
(define ordenacao-tests
  (test-suite "Testes Ordenacao"
    (check-equal? (ordena_data google_desordenado) google)
    (check-equal? (ordena_data petrobras_desordenado) petrobras)
    (check-equal? (ordena_data microsoft_desordenado) microsoft)))

; Testes unitários de correlação
(define correlacao-tests
  (test-suite "Testes Correlacao"
    (check-equal? (correlacao google microsoft) 0.1603697511597682)
    (check-equal? (correlacao google petrobras) -0.07983751056172986)
    (check-equal? (correlacao microsoft petrobras) 0.6372067611546479)))

; Testes unitários de proxima e anterior data
(define data-tests
  (test-suite "Testes Data"
    (check-equal? (posterior_data_valida "02/01/2018") "03/01/2018")
    (check-equal? (posterior_data_valida "30/04/2018") "01/05/2018")
    (check-equal? (anterior_data_valida "03/01/2018") "02/01/2018")
    (check-equal? (anterior_data_valida "01/05/2018") "30/04/2018")))

; Testes unitários de Media Movel Simples
(define media_movel-tests
  (test-suite "Teste Media Movel Simples"
    (check-equal? (first (media_movel google 10)) 1100.145996)
    (check-equal? (first (media_movel microsoft 10)) 87.794999600000001)
    (check-equal? (first (media_movel petrobras 10)) 11.171000000000001)))
```

Testes Unitários

```
; Testes unitários de Media Movel Exponencial
(define media_movel_exponencial-tests
  (test-suite "Teste Media Movel Exponencial"
    (check-equal? (first (media_exponencial google 14)) 1111.1821462857145)
    (check-equal? (first (media_exponencial microsoft 14)) 88.55714242857142)
    (check-equal? (first (media_exponencial petrobras 14)) 11.454999999999998)))

; Testes unitários de RSI
(define rsi-tests
  (test-suite "Teste RSI"
    (check-equal? (first (rsi google 14)) 93.30480987894174)
    (check-equal? (first (rsi microsoft 14)) 80.23506410256401)
    (check-equal? (first (rsi petrobras 14)) 91.00529100529101)))

; Testes unitários de MACD
(define macd-tests
  (test-suite "Teste MACD"
    (check-equal? (first (macd google)) -18.221595237179372)
    (check-equal? (first (macd microsoft)) -2.069808198717922)
    (check-equal? (first (macd petrobras)) -0.9373076923076944)))

; Função para executar os testes
(define (executa-testes . testes)
  (run-tests (test-suite "Todos os testes" testes))
  (void))

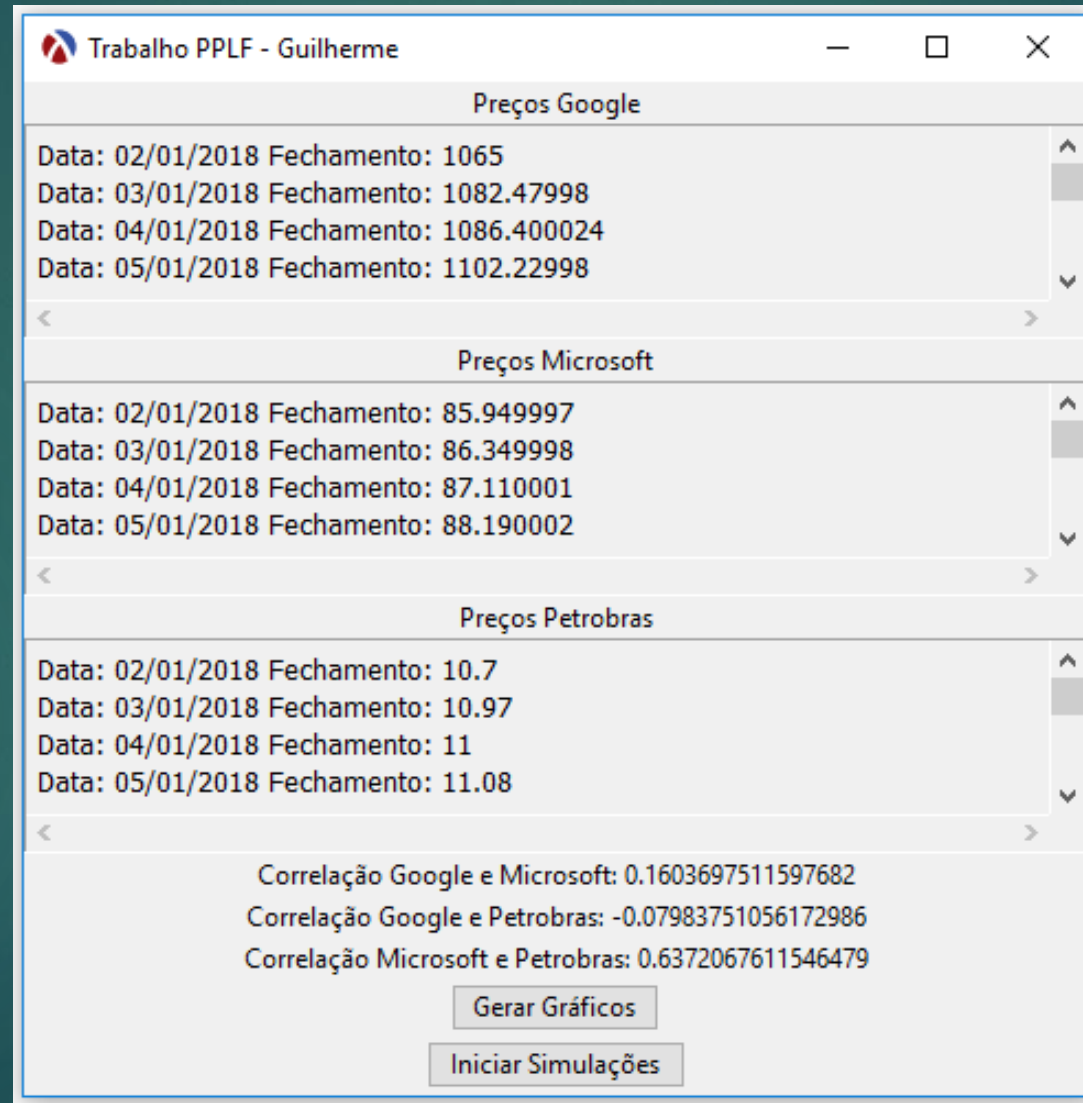
; Função para chamar todos os testes
(define (executa_todos_testes)
  (executa-testes correlacao-tests data-tests ordenacao-tests media_movel-tests media_movel_exponencial-tests macd-tests rsi-tests))
```

Exemplos

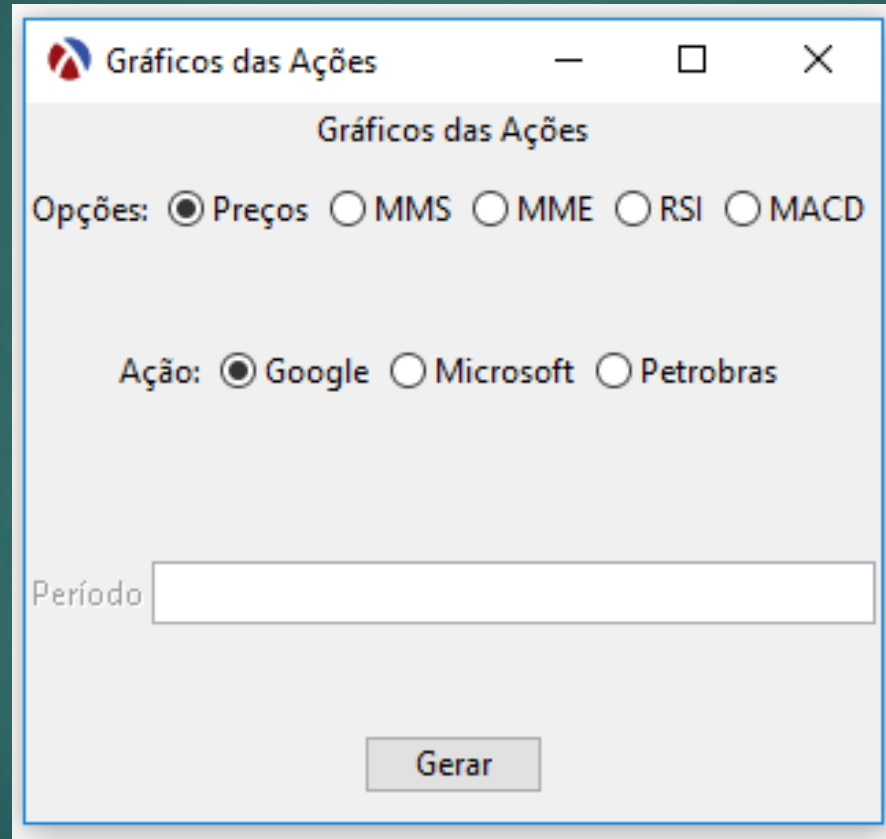
```
> (correlacao google microsoft)
0.1603697511597682
> (posterior_data_valida "02/01/2018")
"03/01/2018"
> (anterior_data_valida "05/01/2018")
"04/01/2018"
> (equal? google google_desordenado)
#f
> (equal? google (ordena_data google_desordenado))
#t
> (first (rsi microsoft 14))
80.23506410256401
> (first (macd google))
-18.221595237179372
> (first (media_movel google 10))
1100.145996
> (first (media_movel petrobras 10))
11.171000000000001
```

Interface

Tela Principal



Tela de Gerar Gráficos



The image shows a web form titled "Gráficos das Ações" (Stock Charts). The form has a title bar with a logo and window controls. Inside, the title "Gráficos das Ações" is repeated. Below the title, there are two rows of radio button options. The first row is labeled "Opções:" and includes "Preços" (selected), "MMS", "MME", "RSI", and "MACD". The second row is labeled "Ação:" and includes "Google" (selected), "Microsoft", and "Petrobras". Below these options is a text input field labeled "Período". At the bottom center of the form is a button labeled "Gerar".

Gráficos das Ações

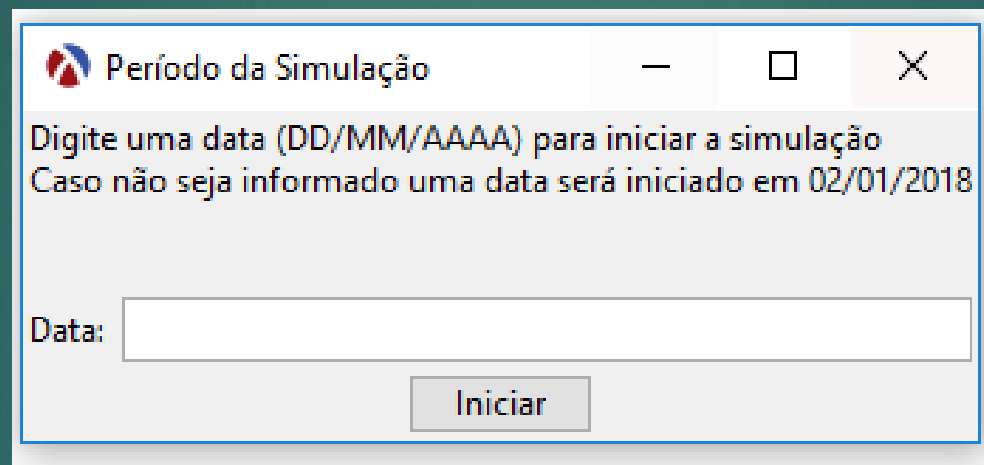
Opções: ☒ Preços ☐ MMS ☐ MME ☐ RSI ☐ MACD

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Período

Gerar

Tela Data Inicial Simulação



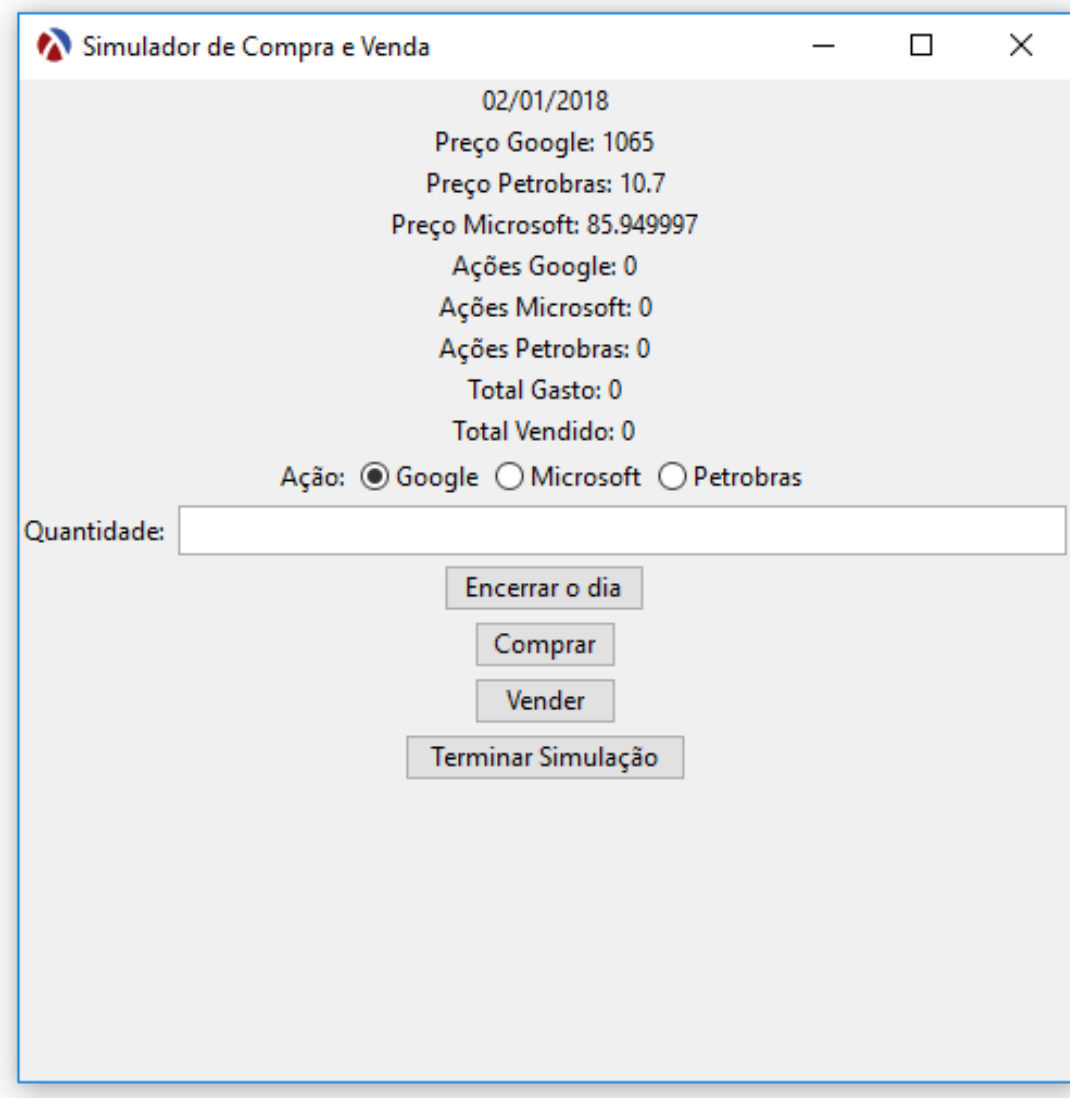
Período da Simulação

Digite uma data (DD/MM/AAAA) para iniciar a simulação
Caso não seja informado uma data será iniciado em 02/01/2018

Data:

Iniciar

Tela Simulador Compra e Venda



The screenshot shows a window titled "Simulador de Compra e Venda" with a standard Windows title bar. The main content area displays the following information:

- Date: 02/01/2018
- Preço Google: 1065
- Preço Petrobras: 10.7
- Preço Microsoft: 85.949997
- Ações Google: 0
- Ações Microsoft: 0
- Ações Petrobras: 0
- Total Gasto: 0
- Total Vendido: 0

Below this information, there is a line for selecting an action:

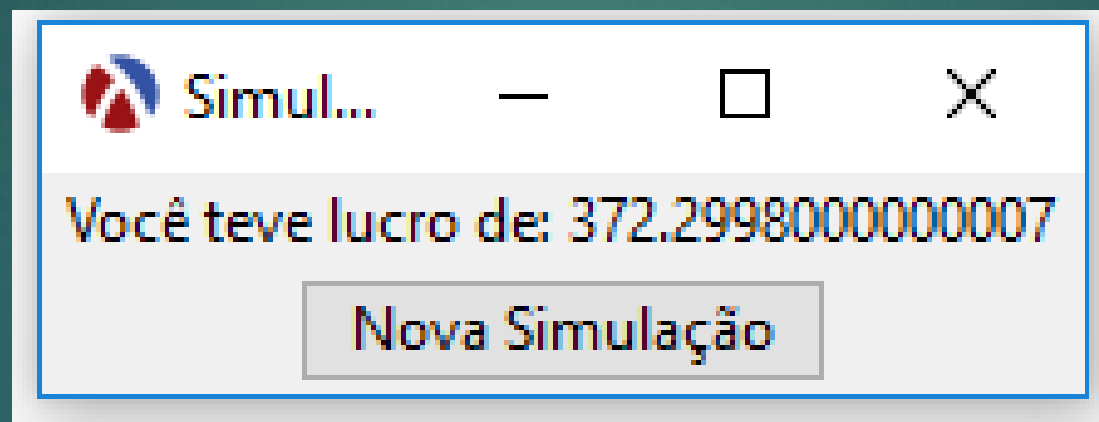
Ação: ☒ Google ☐ Microsoft ☐ Petrobras


Below the action selection, there is a text label "Quantidade:" followed by an empty input field.

At the bottom of the window, there are four buttons arranged vertically:

- Encerrar o dia
- Comprar
- Vender
- Terminar Simulação

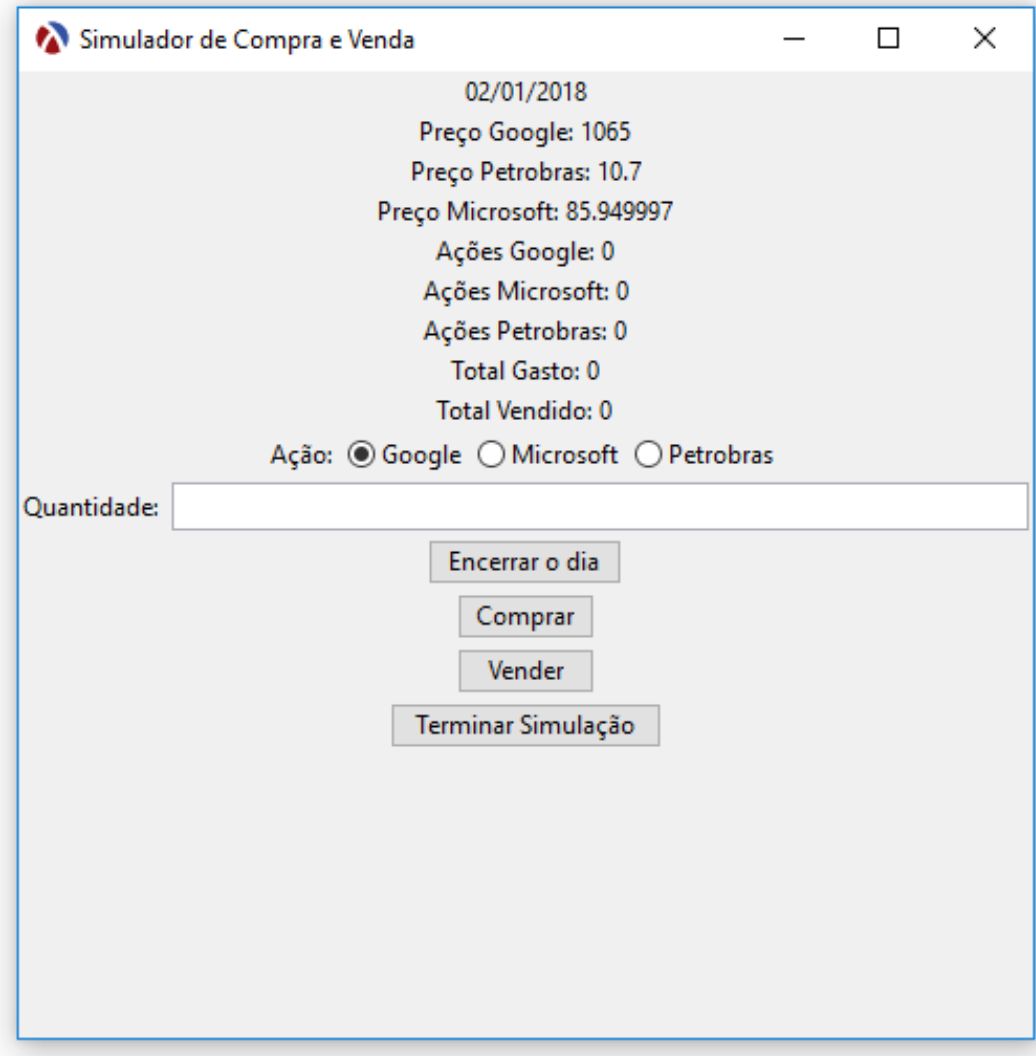
Tela Final Simulação





Simulação Compra e Venda

Simulação



02/01/2018

Preço Google: 1065
Preço Petrobras: 10.7
Preço Microsoft: 85.949997

Ações Google: 0
Ações Microsoft: 0
Ações Petrobras: 0

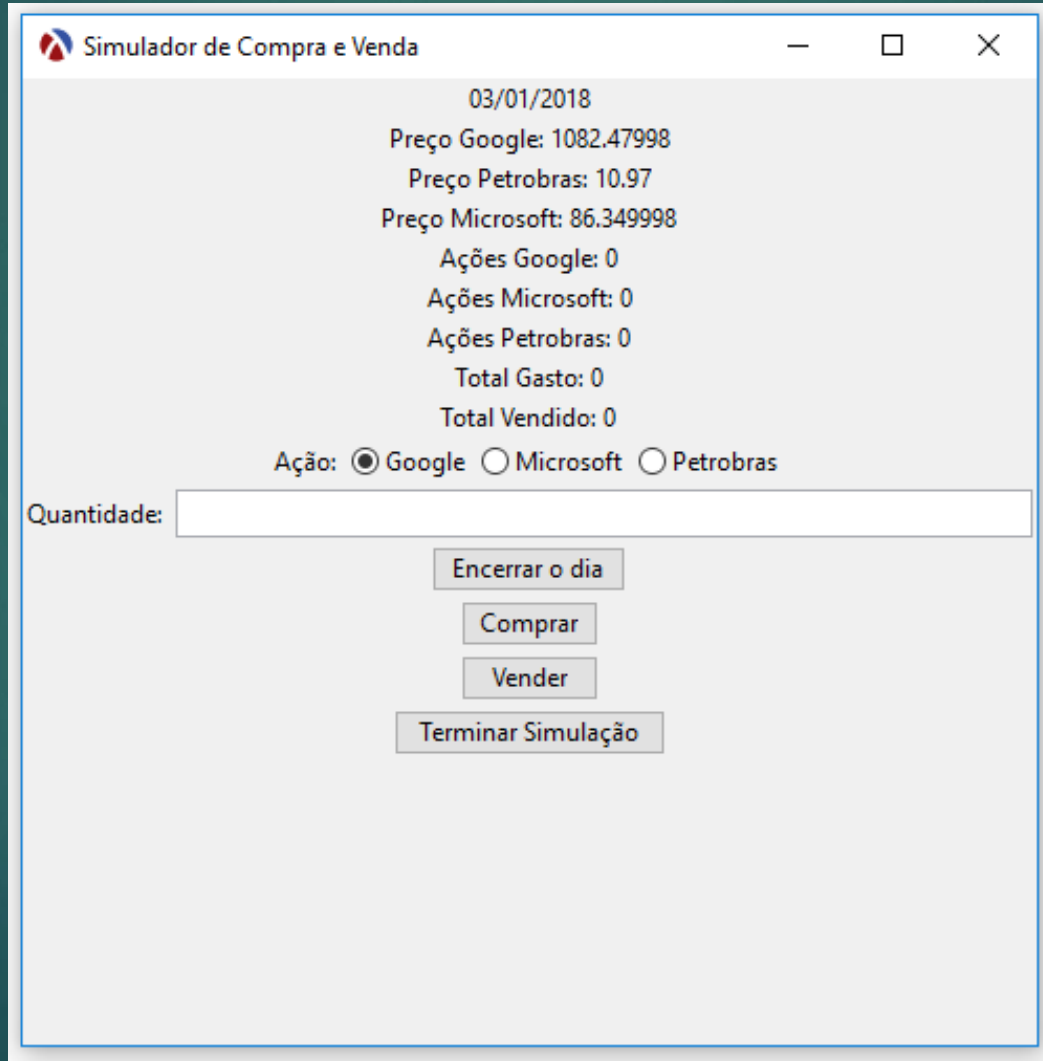
Total Gasto: 0
Total Vendido: 0

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Quantidade:

Encerrar o dia
Comprar
Vender
Terminar Simulação

Simulação



03/01/2018

Preço Google: 1082.47998

Preço Petrobras: 10.97

Preço Microsoft: 86.349998

Ações Google: 0

Ações Microsoft: 0

Ações Petrobras: 0

Total Gasto: 0

Total Vendido: 0

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Quantidade:

Encerrar o dia

Comprar

Vender

Terminar Simulação

Simulação

Simulador de Compra e Venda

03/01/2018

Preço Google: 1082.47998

Preço Petrobras: 10.97

Preço Microsoft: 86.349998

Ações Google: 10

Ações Microsoft: 0

Ações Petrobras: 0

Total Gasto: 10824.7998

Total Vendido: 0

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Quantidade: 10

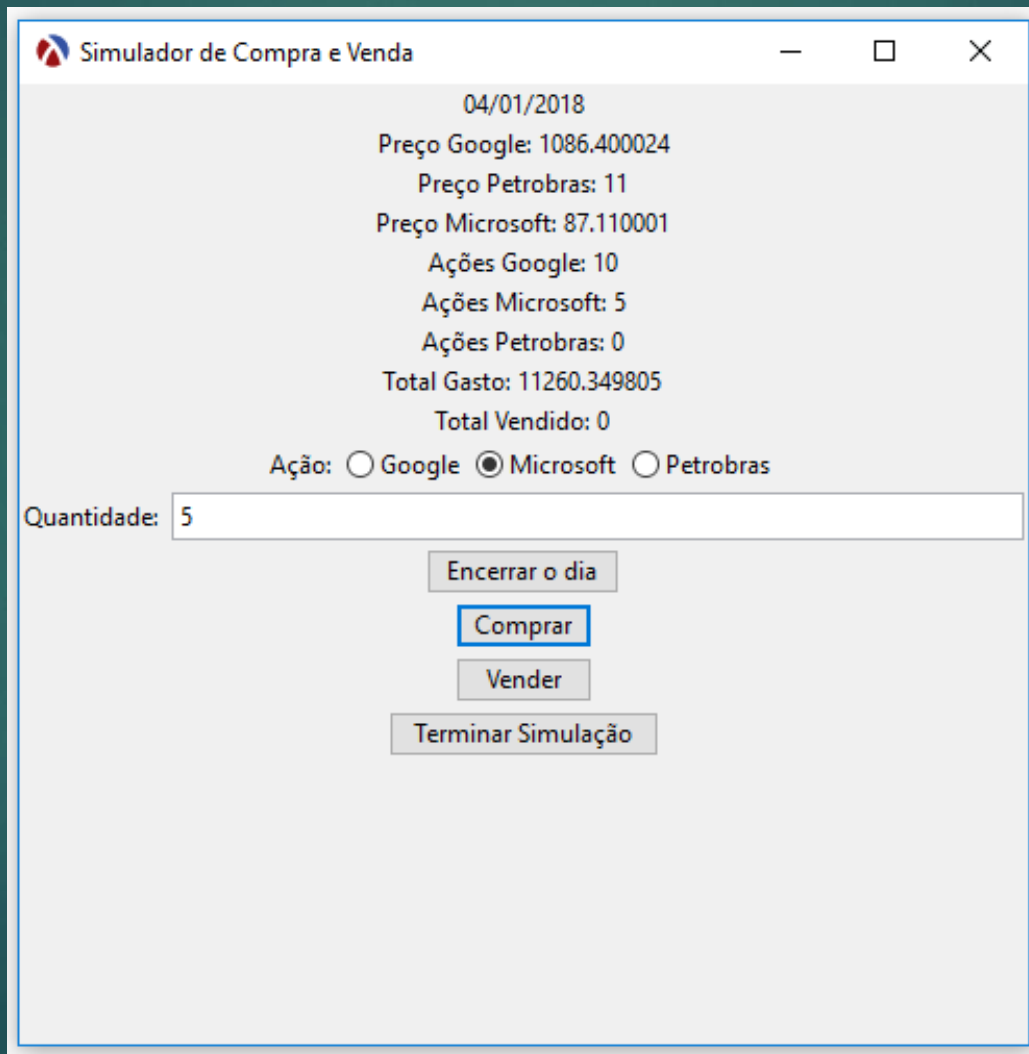
Encerrar o dia

Comprar

Vender

Terminar Simulação

Simulação



04/01/2018

Preço Google: 1086.400024

Preço Petrobras: 11

Preço Microsoft: 87.110001

Ações Google: 10

Ações Microsoft: 5

Ações Petrobras: 0

Total Gasto: 11260.349805

Total Vendido: 0

Ação: ☐ Google ☒ Microsoft ☐ Petrobras

Quantidade:

Encerrar o dia

Comprar

Vender

Terminar Simulação

Simulação

Simulador de Compra e Venda

04/01/2018

Preço Google: 1086.400024

Preço Petrobras: 11

Preço Microsoft: 87.110001

Ações Google: 5

Ações Microsoft: 5

Ações Petrobras: 0

Total Gasto: 11260.349805

Total Vendido: 5432.000120000001

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Quantidade: 5

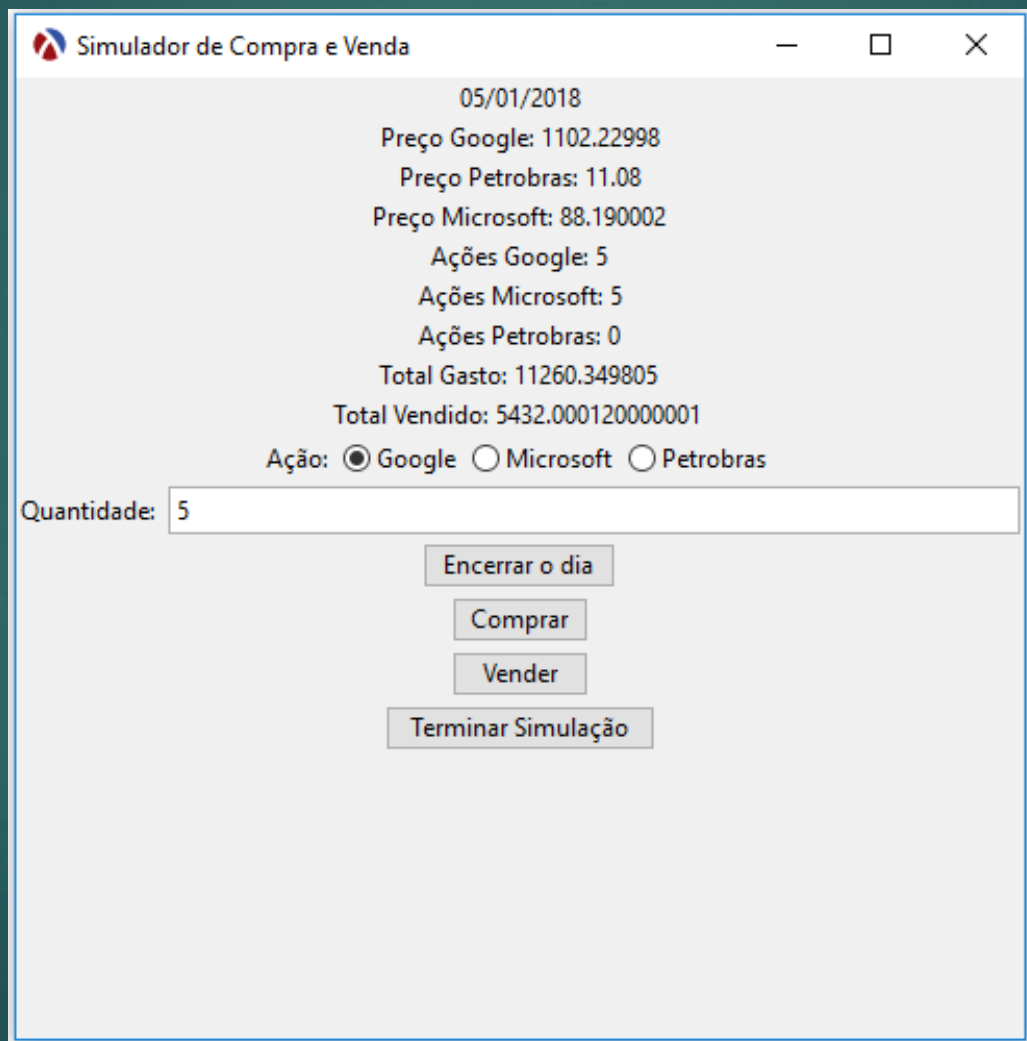
Encerrar o dia

Comprar

Vender

Terminar Simulação

Simulação



05/01/2018

Preço Google: 1102.22998

Preço Petrobras: 11.08

Preço Microsoft: 88.190002

Ações Google: 5

Ações Microsoft: 5

Ações Petrobras: 0

Total Gasto: 11260.349805

Total Vendido: 5432.000120000001

Ação: ☒ Google ☐ Microsoft ☐ Petrobras

Quantidade: 5

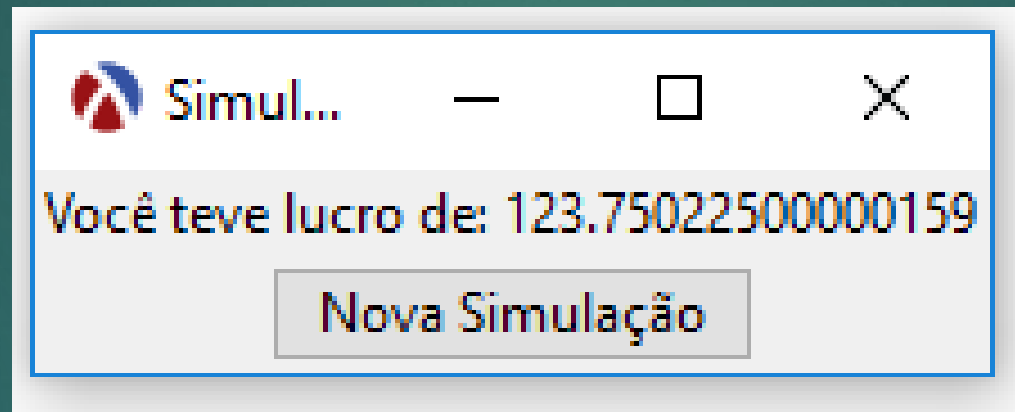
Encerrar o dia

Comprar

Vender

Terminar Simulação

Simulação Final



Referências

- ▶ The Racket Guide. Disponível em: <https://docs.racket-lang.org/guide/>. Acesso: 1 set. 2018
- ▶ Médias Moveis Simples e Exponencial. Disponível em: <https://www.tororadar.com.br/investimento/analise-tecnica/medias-moveis>. Acesso: 3 set. 2018
- ▶ Uma Cartilha Sobre o MACD. Disponível em: <https://www.tororadar.com.br/investimento/analise-tecnica/macd>. Acesso: 3 set. 2018
- ▶ IFR – Saiba o que é o índice de Força Relativa. Disponível em: <https://www.tororadar.com.br/investimento/analise-tecnica/indice-de-forca-relativa-ifr>. Acesso: 3 set. 2018