

# CASIO NEWS

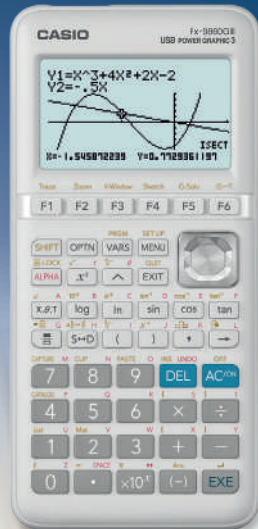
ANO 13 - ANUAL - JUNHO 2022 - NÚMERO 22

fx-991SPX



CLASSWIZ

fx-CG50



fx-CG9860GIII

C-Lab





## Editorial

A **Casio News** passou a ter uma edição anual. A decisão de passar a ser anual em vez de trimestral ou semestral está relacionada com as diversas formações online que o grupo dinamiza, disponibilizando em cada uma inúmeros materiais que podem ser distribuídos, copiados e replicados pelos diversos grupos disciplinares e alunos.

A **Casio News** neste ano letivo (2022/2023) quer mostrar o trabalho de excelência realizado pelo grupo de trabalho **CASIO+** fazendo uma mostra de diversos programas em *Python* e como a física pode ser lecionada recorrendo a vídeos que podem ser importados para dentro da calculadora gráfica.

Estivemos, estamos e estaremos sempre consigo!

Desejamos um excelente ano lectivo.

Ana Margarida Simões Dias  
School Coordinator

# CASIO NEWS

## Índice

### PROGRAMA 1:

Retas perpendiculares no plano . . . . . 3

### PROGRAMA 2:

Somas dos termos de uma progressão . . . 7

Determinação do valor da aceleração  
da queda de um objeto . . . . . 11

### Ficha técnica

Propriedade: Casio – Sucursal Portugal

Responsabilidade e Coordenação Geral: Casio - Sucursal Portugal • Ana Margarida S.M. Simões Dias O.S.

Toda a correspondência deve ser enviada para:

Morada: Parque das Nações • Rua do Pólo Sul, N.º 2 - 4.º Andar • 1990-273 Lisboa

Telefone: 21 893 91 70 • Fax: 21 893 91 79 • Email: margaridadias@casio.pt



**DIAS CASIO+ 2021/SP3**  
INICIAÇÃO À PROGRAMAÇÃO PYTHON NA CALCULADORA GRÁFICA  
ATIVIDADE "perpend.py"

## PROGRAMA 1

### Retas perpendiculares no plano

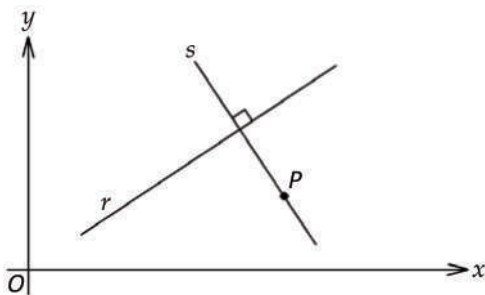
No plano, duas retas  $r$  e  $s$  de declives respetivamente iguais a  $m$  e  $m'$  são perpendiculares se e só se:

$$m \times m' = -1$$

#### Objetivo

Escreva um programa, em linguagem Python, que determina a equação reduzida da reta  $s$ , perpendicular à reta  $r$  definida por  $y = mx + b$  (com  $m \neq 0$ ) e que contém o ponto  $P$  de coordenadas  $(x_p, y_p)$ .

Nome do programa: **perpend.py**



Exemplo de execução do programa

Shell

Equacao da reta s perpendicular a r e que contem P:

r:  $y = mx + b$

$m = -2$

$b = 3$

P( $x_P, y_P$ )

$x_P = 2$

$y_P = 6$

s:  $y = 0.5x + 5.0$

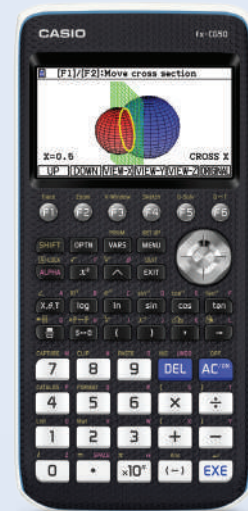
>>>

## Calculadora Gráfica CASIO fx-CG50

O ecrã de alta resolução com expressões em formato natural, permite a introdução de funções, raízes quadradas, expressões numéricas como vê representado no seu livro de texto.

#### Características técnicas:

- 3D
- Ecrã de alta definição com mais de 65.000 cores
- Função de colocar pontos numa imagem e vídeo
- 7 cores disponíveis em muitas aplicações
- Cor de ligação nos gráficos para fácil compreensão
- Simples ligação ao PC. Funciona como memória externa.
- Cabos incluídos
- Frações
- Passagem de decimal para fração e vice-versa
- 1ª e 2ª derivada numérica
- Sistema de equações (\* numérico \*) (max 6 incógnitas)
- Equações polinomiais (\* numérico \*) (max 6º grau)
- Cálculos estatísticos, Regressões e respetivos coeficientes
- Gráficos Estatísticos
- Tabela periódica (ADD-in)
- Folha de cálculo



PROGRAMA 1 – Proposta de resolução

Descrição	Programa em Python
algoritmo "perpend"	1 print("Equacao da reta s")
escreva "Equação da reta s	2 print("perpendicular a r")
perpendicular a r e que	3 print("e que contem P:")
contem P"	4
escreva "r: y=mx+b"	5 print("r: y=mx+b")
leia m <sub>1</sub> (declive de r)	6 m1 = float(input("m="))
leia b <sub>1</sub> (ordenada na origem de r)	7 b1 = float(input("b="))
escreva "P(x <sub>P</sub> ,y <sub>P</sub> )"	8 print("P(xP,yP)")
leia x (abscissa de P)	9 x = float(input("xP="))
leia y (ordenada de P)	10 y = float(input("yP="))
	11
m <sub>2</sub> = declive de s	12 m2 = -1 / m1
b <sub>2</sub> = ordenada na origem de s	13 b2 = y + x/m1
	14
escreva linha em branco	15 print()
escreva equação de s	16 print("s: y=",m2,"x+",b2)

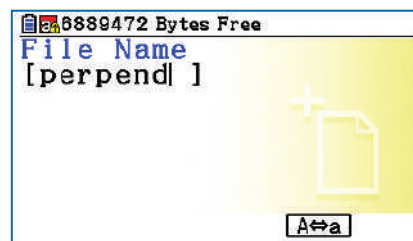
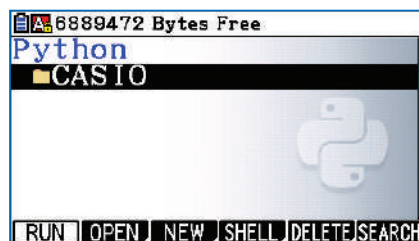
Explicação linha-a-linha:

- 01-03 Usando instruções *print*, informe o utilizador (no Shell), de que o programa determina a equação da reta *s*, perpendicular à reta *r* que contém o ponto *P*.
- 05 Escreva, no Shell, "*r: y = mx + b*". Considera-se *m* "≠" 0.
- 06-07 Peça ao utilizador os valores de *m* e de *b*.
- 08 Escreva, no Shell, "*P(xP,yP)*".
- 09-10 Peça ao utilizador as coordenadas do ponto *P*.
- 12 Calcule o declive da reta *s*.
- 13 Calcule a ordenada na origem da reta *s*.
- 15 Para separar a entrada da saída de dados, deixe uma linha em branco.
- 16 Apresente a equação da reta *s*.

CASIO fx-CG 50

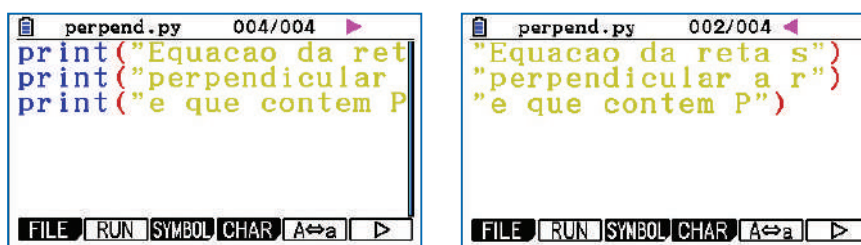
Vídeo – <https://youtu.be/VtsXBIVra4>

Entre no menu Python e pressione [F3] (New). Escreva "perpend" e carregue em [EXE].



Nas linhas 01, 02 e 03, escreva: `print("Equacao da reta s")`, `print("perpendicular a r")` e `print("e que contem P")`, de modo a adequar a mensagem à largura do Shell (21 caracteres).

**Nota:** Na função `print`, as expressões entre aspas (em amarelo) serão apresentadas *ipsis verbis* no Shell (sem aspas).



```
perpend.py 004/004
print("Equacao da reta s")
print("perpendicular
print("e que contem P

perpend.py 002/004
"Equacao da reta s")
"perpendicular a r")
"e que contem P")
```

Passa uma linha, para tornar o *script* mais claro (esta linha não irá aparecer no Shell).

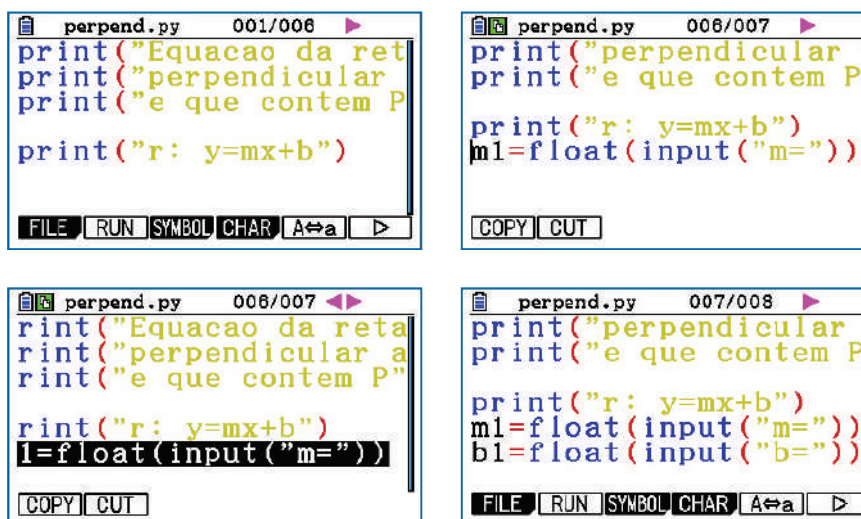
Na linha 05, escreva: `print("r: y=mx+b")`.

Na linha 06, escreva: `m1 = float(input("m="))`. Assim, o programa irá pedir ao utilizador um valor que será atribuído à variável "m1" (declive da reta r), do tipo float (número real).

A seguir, na linha 07, escreva: `b1 = float(input("b="))`, copiando a linha anterior (*copy/paste*) e trocando "m" por "b".

**Nota:** Para copiar uma expressão:

- 1) coloque o cursor no início da expressão, pressione **SHIFT** **8** (CLIP) e desloque o cursor até ao final da expressão (ficará selecionada);
- 2) carregue em **F1** (COPY);
- 3) coloque o cursor onde quer colar e pressione **SHIFT** **9** (PASTE).



```
perpend.py 001/006
print("Equacao da reta
print("perpendicular
print("e que contem P

print("r: y=mx+b")

perpend.py 006/007
print("perpendicular
print("e que contem P

print("r: y=mx+b")
m1=float(input("m="))

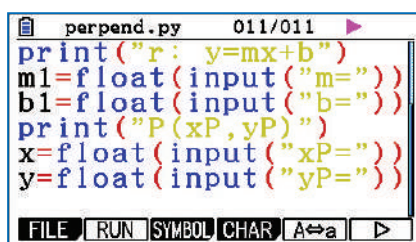
perpend.py 006/007
rint("Equacao da reta
rint("perpendicular a
rint("e que contem P"

rint("r: y=mx+b")
l=float(input("m="))

perpend.py 007/008
print("perpendicular
print("e que contem P

print("r: y=mx+b")
m1=float(input("m="))
b1=float(input("b="))
```

Nas linhas 08, 09 e 10, proceda do mesmo modo para pedir as coordenadas do ponto P.



```
perpend.py 011/011
print("r: y=mx+b")
m1=float(input("m="))
b1=float(input("b="))
print("P(xP, yP)")
x=float(input("xP="))
y=float(input("yP="))
```

Passa uma linha no *script*, para assinalar o fim da *entrada de dados*.

Seguidamente, nas linhas 11 e 12, defina as variáveis "m2" e "b2".

```
perpend.py 014/014
print("P(xP, yP)")
x=float(input("xP="))
y=float(input("yP="))

m2=-1/m1
b2=y+x/m1
```

Passe mais uma linha no *script*, para destacar a última parte correspondente à *saída de dados*.

Na linha 15, escreva *print()* para escrever uma linha em branco no Shell.

Por último, na linha 16, escreva: *print("s: y=",m2,"x+",b2)*.

**Nota:** A função *print* avalia cada uma das expressões separadas por vírgulas, após o que escreve os seus valores todos numa mesma linha do Shell, separados por um espaço em branco.

```
perpend.py 017/017
m2=-1/m1
b2=y+x/m1

print()
print("s: y=",m2,"x+"
```

```
perpend.py 016/017
/m1
x/m1

()
("s: y=",m2,"x+",b2)
```

Carregue em **F2** (RUN) **F1** (Sim) e experimente o programa.

**Nota:** No Shell, usa-se a tecla **-** (subtração) para escrever o sinal *menos* (-2).

```
|CASIO COMPUTER CO.,
>>>from perpend impor
Equacao da reta s
perpendicular a r
e que contem P:
r: y=mx+b
m=
```

```
perpendicular a r
e que contem P:
r: y=mx+b
m=-2
b=3
P(xP, yP)
xP=
```

```
e que contem P:
r: y=mx+b
m=-2
b=3
P(xP, yP)
xP=2
yP=6
```

```
b=3
P(xP, yP)
xP=2
yP=6

s: y= 0.5 x+ 5.0
>>>
```

Para usar novamente o programa, pressione **EXIT** e **F2** (RUN).

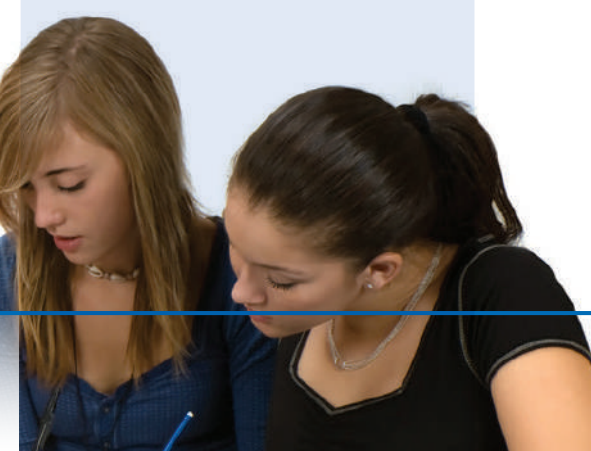
```
* SHELL Initialized *
>>>from perpend impor
Equacao da reta s
perpendicular a r
e que contem P:
r: y=mx+b
m=4
```

```
b=-1
P(xP, yP)
xP=3
yP=-7

s: y= -0.25 x+ -6.25
>>>
```

### Atividade realizada na Sessão Prática 3

“Iniciação à programação Python na calculadora gráfica” 2021/2022 por Manuel Marques e e António Cardoso



## DIAS CASIO+ 2021/SP3

INICIAÇÃO À PROGRAMAÇÃO PYTHON NA CALCULADORA GRÁFICA  
ATIVIDADE "somaprog.py"

### PROGRAMA 2

#### Soma dos termos de uma progressão

Uma sucessão  $(u_n)$  diz-se progressão *aritmética* quando o primeiro termo é igual a  $u_1$  e cada um dos restantes termos se obtém do anterior adicionando uma razão  $r$ .

A soma dos  $N$  primeiros termos de uma progressão aritmética é dada por:

$$S = \frac{u_1 + u_N}{2} \times N.$$

Uma sucessão  $(u_n)$  diz-se progressão *geométrica* quando o primeiro termo é igual a  $u_1$  e cada um dos restantes termos se obtém do anterior multiplicando por uma razão  $r$ .

A soma dos  $N$  primeiros termos de uma progressão geométrica é dada por:

$$S = u_1 \times \frac{1 - r^N}{1 - r}.$$

#### Objetivo

Escreva um programa, em linguagem Python, que calcula a soma dos  $N$  primeiros termos de uma progressão aritmética ou geométrica, conhecendo o primeiro termo e a razão.

Nome do programa: **somaprog.py**

#### Exemplo de execução do programa

Shell

```
Soma dos N primeiros termos de uma:
```

```
- progressão aritmética: prima [1]
```

```
- progressão geométrica: prima [2]
```

```
? 3
```

```
Valor inválido!
```

```
>>>
```

```
Soma dos N primeiros termos de uma:
```

```
- progressão aritmética: prima [1]
```

```
- progressão geométrica: prima [2]
```

```
? 1
```

```
u1 = 3
```

```
r = 5
```

```
N = 30
```

```
Soma: S = 2265.0
```

```
>>>
```

```
Soma dos N primeiros termos de uma:
```

```
- progressão aritmética: prima [1]
```

```
- progressão geométrica: prima [2]
```

```
? 2
```

```
u1 = 4
```

```
r = 1.5
```

```
N = 7
```

```
Soma: S = 128.6875
```

```
>>>
```

## Calculadora Científica CASIO - ClassWiz FX-82SPX e FX-350SPX

O ecrã de alta resolução com expressões em "formato natural" permite mostrar frações, raízes e outras operações como vê num livro de texto. O ecrã de alta resolução permite ter um menu por ícones e mensagens sem abreviatura, podendo visualizar 4 a 6 linhas.

#### Características técnicas:

- Menu por ícones
- Idiomas: castelano, catalán e português
- mensagens sem abreviaturas
- 4 a 6 linhas no visor
- Fatorização em números primos. MDC e MMC.
- Indicação de dízima infinita periódica

#### NOVIDADE:

- Tecla de simplificação de frações
- Memória PreAns
- Tecla de cálculo do resto da divisão
- Tabela numérica para 1 ou 2 funções



**RECOMENDADA  
PARA O  
2.º e 3.º CICLO**

Descrição	Programa em Python
<p>algoritmo "somaprog"</p> <p>leia p</p> <p>[1] = progressão aritmética</p> <p>[2] = progressão geométrica</p> <p>se p ≠ 1 e p ≠ 2</p> <p>então "Valor inválido!"</p> <p>senão</p> <p>leia u<sub>1</sub> (1.º termo)</p> <p>leia r (razão)</p> <p>leia N (n.º de termos)</p> <p>se p = 1 então</p> <p>u<sub>N</sub> = u<sub>1</sub> + (N-1)r</p> $soma = \frac{u_1 + u_N}{2} \times N$ <p>senão</p> $soma = u_1 \times \frac{1 - r^N}{1 - r}$ <p>escreva soma</p>	<pre> 1 print("Soma dos N primeiros \n termos de uma:") 2 print("- prog. aritmetica: \n prima [1]") 3 print("- prog .geométrica: \n prima [2]") 4 p = int(input("? ")) 5 6 if p!= 1 and p!= 2: 7     print("Valor inválido!") 8 else: 9     u1 = float(input("u1 = ")) 10    r = float(input("r = ")) 11    N = int(input("N = ")) 12    if p==1: 13        uN = u1+(N-1)*r 14        S = (u1+uN)*N/2 15    else: 16        S = u1*(1-r**N)/(1-r) 17    print("Soma: S =",S) </pre>

Explicação linha-a-linha:

- 01-03 Informe o utilizador de que deve digitar “1” ou “2” para calcular a soma dos *N* primeiros termos de uma progressão aritmética ou geométrica, respetivamente.
- 04 Atribua à variável *p* a escolha do utilizador.
- 06-07 Crie uma estrutura condicional *if – else* para separar o caso em que o valor digitado é inválido (diferente de “1” e “2”) do caso em que é válido.  
 Por exemplo, se o utilizador digitar “3” irá receber a mensagem “Valor inválido!” e o programa terminará.
- 08-11 Se o valor digitado for válido, peça ao utilizador o primeiro termo (*u*<sub>1</sub>) e a razão (*r*) da progressão, assim como o número de termos (*N*).
- 12-16 Acrescente uma segunda estrutura condicional *if – else* (dentro da primeira) para separar o caso em que o utilizador escolheu uma progressão aritmética (*p*==1) do caso em que optou por uma progressão geométrica (*p*==2).  
 Se o valor digitado for “1”, calcule o valor do último termo (*u*<sub>*n*</sub>) e a soma dos *N* primeiros termos da progressão aritmética.  
 Se o valor digitado for “2”, calcule a soma dos *N* primeiros termos da progressão geométrica, com a respetiva fórmula.
- 17 Por último, escreva a soma, usando a instrução *print*.

Entre no menu Python e pressione **F3** (New). Escreva “somaprog” e carregue em **EXE**.

Nas linhas 01 a 03, use instruções *print*, com o comando *\n* para “quebrar” a linha.



Na linha 04, escreva: `p=int(input("? "))`. Assim, o programa irá pedir ao utilizador um valor que será atribuído à variável “p”, do tipo *int* (número inteiro).

```
6873088 Bytes Free
File Name
[somaprogram]
```

```
somaprogram.py 002/002
print("Soma dos N pri
```

```
somaprogram.py 001/002
meiros \ntermos de um'
```

```
somaprogram.py 001/002
s \ntermos de uma:')
```

```
somaprogram.py 004/004
print("Soma dos N pri
print("-prog. aritmet
print("-prog. geometr
p=int(input("? "))
```

```
somaprogram.py 003/004
meiros \ntermos de u
tica: \n prima [1]')
rica: \n prima [2]')
```

Na linha 06, para criar uma estrutura *if – else*, pressione **F6**(**>**)**F1**(COMMAND)**F2**(if-else).

Com o cursor imediatamente antes de “:”, escreva: `p!=1 and p!=2`.

**Nota:** O símbolo “!” está disponível no separador CHAR (**F4**).

A seguir, preencha a linha 07, com indentação (dois espaços à esquerda).

```
somaprogram.py 006/010
print("-prog. geometr
p=int(input("? "))

if :
else:

if ifelseifelif for for-range while
```

```
Character Select
"#$%&'()*+,-./0123
456789:;<=>?@ABCDEF
GHIJKLMNOPQRSTUVWXYZ
Z[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

```
somaprogram.py 008/010
print("-prog. geometr
p=int(input("? "))

if p!=1 and p!=2:
    print("Valor invali
else:
```

```
somaprogram.py 007/010
("-prog. geometrica:
(input("? "))

=1 and p!=2:
nt("Valor invalido")
```

Coloque o cursor no final da linha 08 (após “else:”) e pressione **EXE**.

Nas linhas 09 a 11, com indentação, defina as variáveis “u1” (número real), “r” (número real) e “N” (número inteiro), cujos valores serão pedidos ao utilizador através de instruções *input*.

```
somaprog.py 011/012
if p!=1 and p!=2:
    print("Valor invalido")
else:
    u1=float(input("u1 = "))
    r=float(input("r = "))
    N=int(input("N = "))
```

```
somaprog.py 009/012
1 and p!=2:
    t("Valor invalido")

loat(input("u1 = "))
oat(input("r = "))
t(input("N = "))
```

Na linha 12 (com indentação), pressione novamente **F6** (▷) **F1** (COMMAND) **F2** (if-else).

Complete a estrutura condicional, tendo em atenção que as linhas 13, 14 e 16 têm um aumento do nível de indentação (quatro espaços).

Por último, na linha 17, com o mesmo nível de indentação da linha 15 (dois espaços), apresente a soma com uma instrução *print*.

```
somaprog.py 015/017
u1=float(input("u1 = "))
r=float(input("r = "))
N=int(input("N = "))
if :
else:
```

```
somaprog.py 014/018
u1=float(input("u1 = "))
r=float(input("r = "))
N=int(input("N = "))
if p==1:
    uN=u1+(N-1)*r
    S=(u1+uN)*N/2
else:
```

```
somaprog.py 017/018
if p==1:
    uN=u1+(N-1)*r
    S=(u1+uN)*N/2
else:
    S=u1*(1-r**N)/(1-r)
    print("Soma: S =", S)
```

```
somaprog.py 016/018
f p==1:
    uN=u1+(N-1)*r
    S=(u1+uN)*N/2
lse:
    S=u1*(1-r**N)/(1-r)
rint("Soma: S =", S)
```

Carregue em **F2** (RUN) **F1** (Sim) e experimente o programa.

```
Soma dos N primeiros
termos de uma:
-prog. aritmetica:
  prima [1]
-prog. geometrica:
  prima [2]
? 3
```

```
-prog. aritmetica:
  prima [1]
-prog. geometrica:
  prima [2]
? 3
Valor invalido
>>>
```

```
Soma dos N primeiros
termos de uma:
-prog. aritmetica:
  prima [1]
-prog. geometrica:
  prima [2]
? 1
```

```
prima [2]
? 1
u1 = 3
r = 5
N = 30
Soma: S = 2265.0
>>>
```

```
Soma dos N primeiros
termos de uma:
-prog. aritmetica:
  prima [1]
-prog. geometrica:
  prima [2]
? 2
```

```
prima [2]
? 2
u1 = 4
r = 1.5
N = 7
Soma: S = 128.6875
>>>
```

**Atividade realizada na Sessão Prática 3**

“Iniciação à programação Python na calculadora gráfica” 2021/2022 por Manuel Marques e e António Cardoso



### DIAS CASIO+ 2020/2021

ESTUDO DA FÍSICA E QUÍMICA COM A CALCULADORA  
ATIVIDADE "Determinação do valor da aceleração  
da queda de um objeto"

## Determinação do valor da aceleração da queda de um objeto

A "Queda Livre" é o movimento de um corpo sujeito apenas à força gravítica (movimento de um grave). O movimento retilíneo de queda livre, junto à superfície da Terra, ocorre quando a resistência do ar é desprezável.

### OBJETIVO:

Determinar a aceleração de um corpo, no seu movimento de queda vertical, a partir da lei do movimento retilíneo com aceleração constante.

### Tarefa 1: Realização do vídeo

Nesta tarefa deve realizar um vídeo que mostre a queda vertical de um corpo.

### Material:

- Dispositivo que filme;
- Régua ou fita métrica;
- Corpo (por exemplo, uma bola).

Durante a realização do vídeo, da queda vertical do corpo, deve manter-se o dispositivo de filmar fixo. Sugere-se o uso de um tripé.

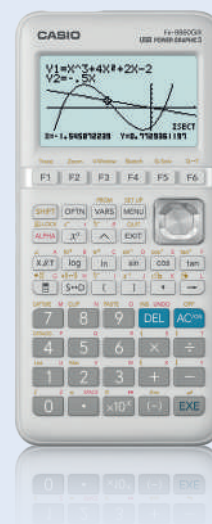
### Tarefa 2: Conversão do vídeo

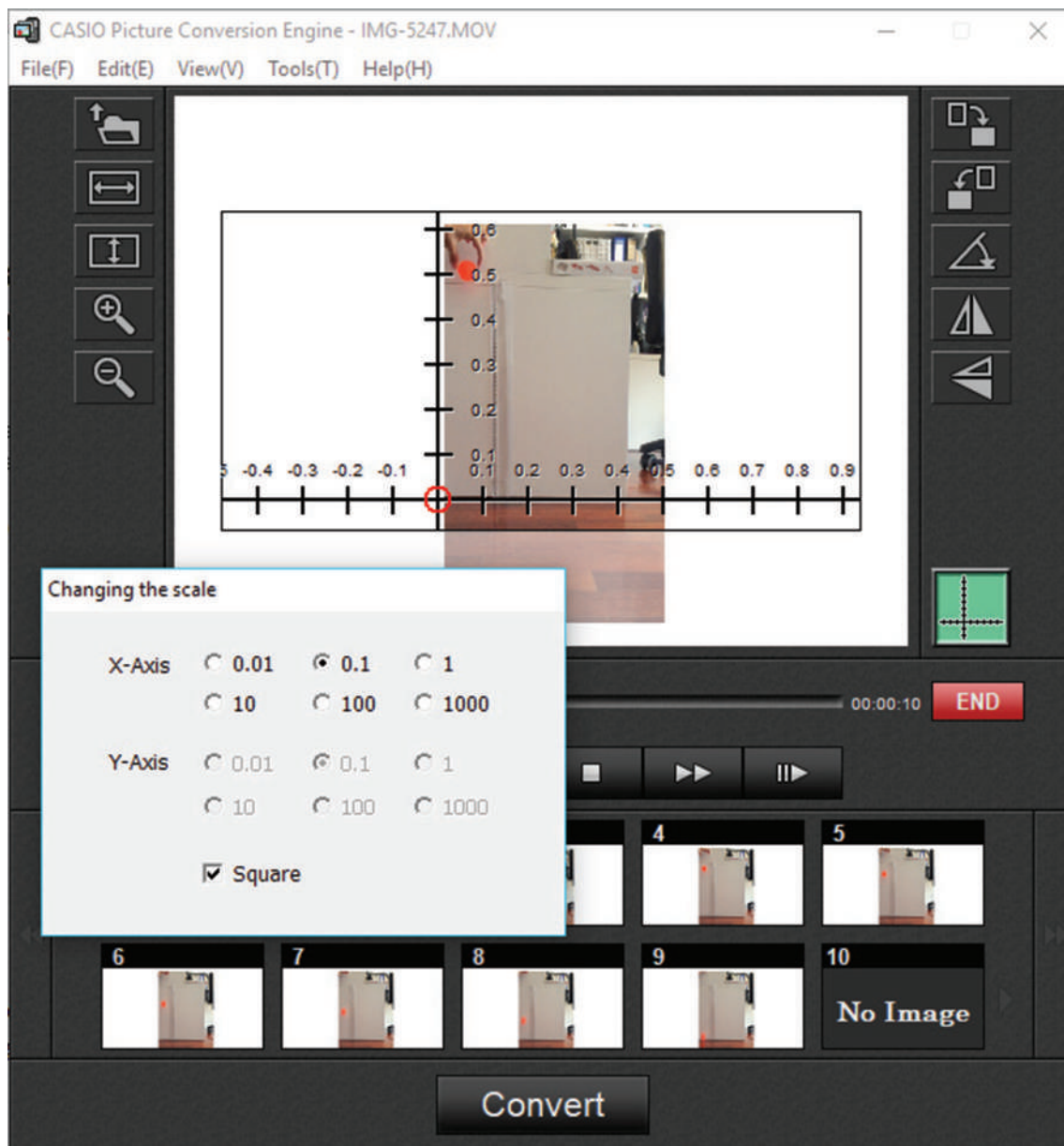
A partir do software *Casio Picture Conversion*, converta o vídeo para um ficheiro compatível com a calculadora (formato ".g3b").

## Calculadora Gráfica Casio FX-9860GIII

### Características técnicas:

- Escrita matemática. Leitura de resultados no modo matemático
- Conversões de unidades
- Tabela de valores
- Equação polinomiais (até 6.º grau)
- Cálculos com complexos de números
- cálculos matriciais
- cálculos vetoriais
- Sucessões
- Desenho de funções de coordenadas cartesianas, paramétricas, polares, Inequações
- Análise de gráfico de funções
- Regressão linear (12 modelos de regressão)
- Estatísticas descritivas
- Gráficos estatísticos: histogramas, diagramas de dispersão, diagrama de extremos e quartis, Gráfico circular e de barras
- Gerador de números aleatórios
- Cálculo integral numérico
- Cálculo diferencial numérico
- Programação Python
- Matemática Financeira
- Possível atualização do sistema operacional
- Folha de cálculo
- Tabela periódica (pré-instalado)
- eActivity
- Função de código QR
- Possível conexão com C-Lab - ECON 4
- Cabo entre calculadoras incluído
- Modo de Exame



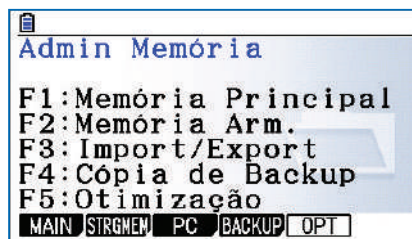
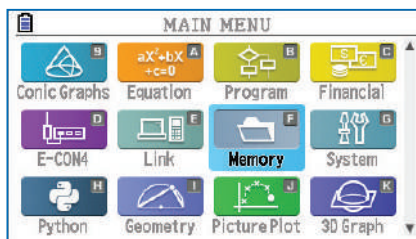


Na conversão do vídeo deve-se ter em consideração o seguinte:

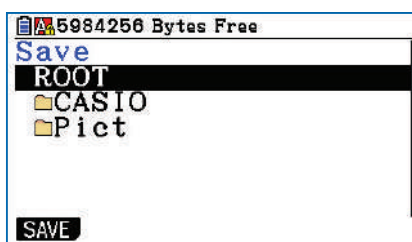
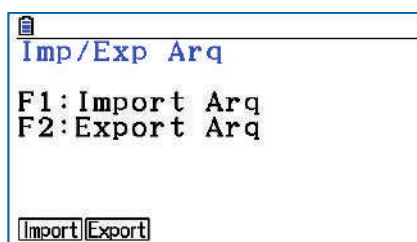
1. Selecionar, no vídeo, o início e o fim do movimento. No exemplo da figura anterior, o programa dividiu a parte do vídeo selecionado (movimento da bola de duração 0,3 s) em 9 *frames*. Para vídeos mais longos selecionar até 12 *frames*, no máximo.
2. A escala dos eixos deve ser escolhida em função da altura a que foi largada a bola. A origem dos eixos deve ser colocada no início ou no fim do movimento.
3. Depois de convertido, o vídeo deve ser enviado para a calculadora.

**Tarefa 3: Envio do vídeo para a calculadora**

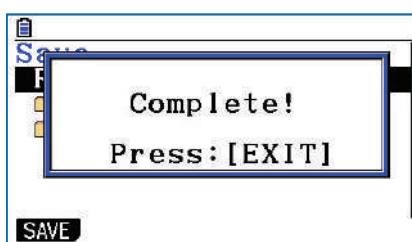
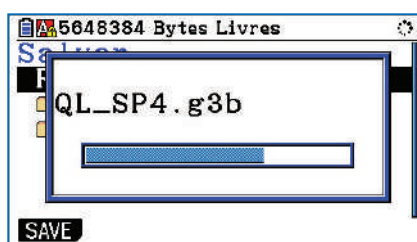
Entrar no menu Memory / Memória e selecionar **F3**: Import / Export



Selecionar **[F1]**: Import Arq. Selecione no computador o ficheiro que deseja importar e salve-o na raiz (ROOT), pressionando **[F1]** (SAVE).

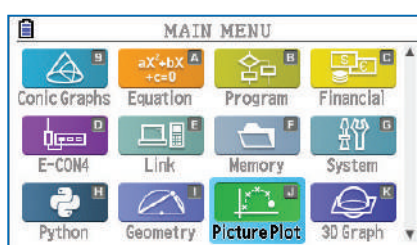


O ficheiro começa a ser guardado na calculadora. Quando terminar, pressione **[EXIT]**.

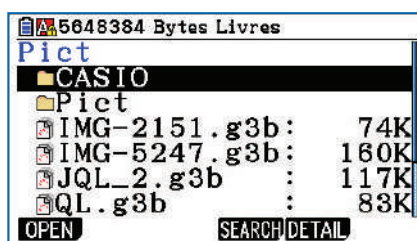


#### Tarefa 4: Visualização do vídeo e colocação de pontos

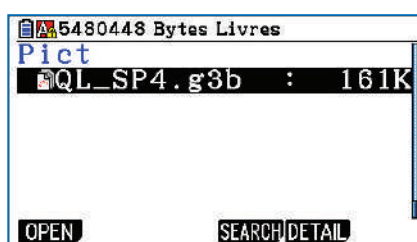
Abra o menu “Picture Plot” ou “Plot Imagem”.



Se surgir uma imagem de um trabalho anterior, deve pressionar **[OPTN]** **[F1]** (FILE), **[F1]** (OPEN).

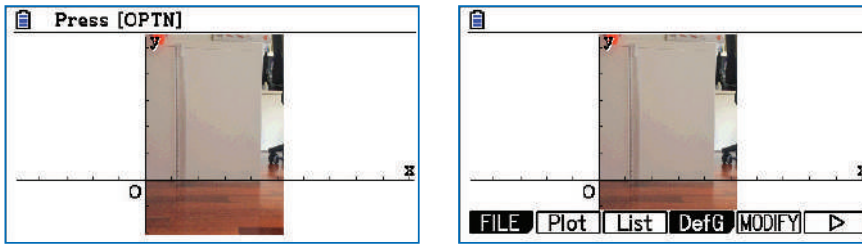


Escolha o ficheiro que colocamos na calculadora.

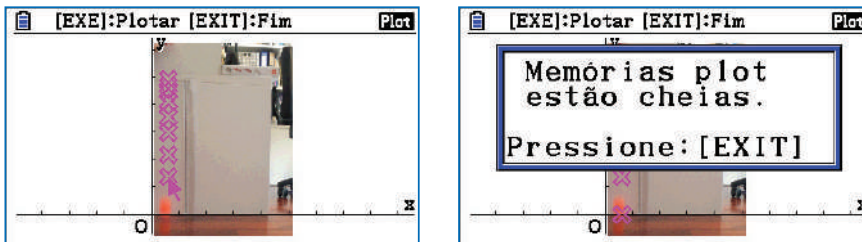


Colocação de pontos:

Pressione **[OPTN]**, seguido de **[F2]** (Plot).

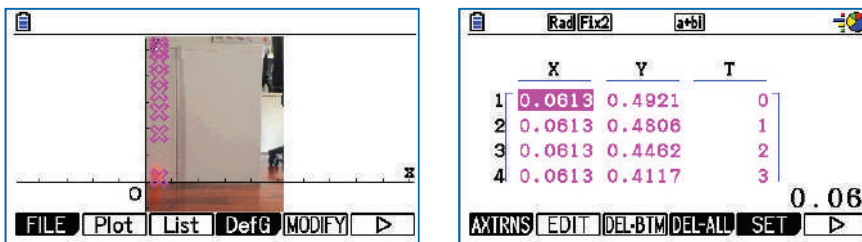


Coloca-se o cursor no limite inferior da bola e pressiona-se **[EXE]**. Passa para o *frame* seguinte. Tem de se andar com a seta cursor e marcar os pontos, pressionando **[EXE]**. Os primeiros pontos estão muito juntos, logo a deslocamento é mais pequeno e pode ser pouco visível. Quando terminar a marcação dos 9 pontos (9 *frames* da conversão do vídeo), surge uma mensagem. Pressione **[EXIT]**, e novamente **[EXIT]** para sair da marcação de pontos.



**Tarefa 5: Determinar a regressão que representa a posição da queda da bola em função do tempo**

Pressionar **[OPTN]** e escolher **[F3]** (List). Visualizamos 3 colunas onde X representa a abcissa da bola, Y a ordenada da bola durante a queda (diferentes alturas) e T representa o número do *frame*, com o início da contagem no zero.



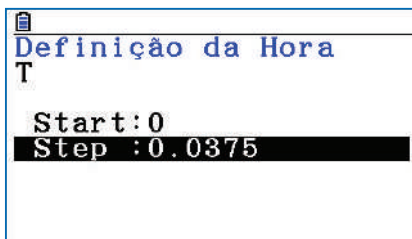
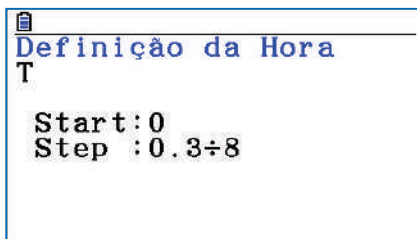
O T pode ser facilmente transformado na variável tempo com origem no início da queda da bola.

Com a ajuda de um *software* que consiga medir tempos pequenos (até centésima ou milésima de segundo), verificamos que a queda inicia-se no instante 3,26 s e a bola toca no chão no instante 3,56 s, logo temos um tempo de queda de 0,3 s.

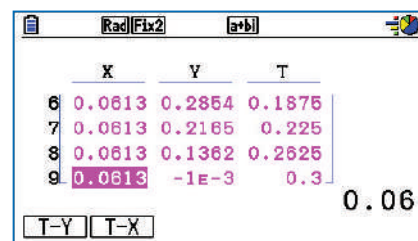
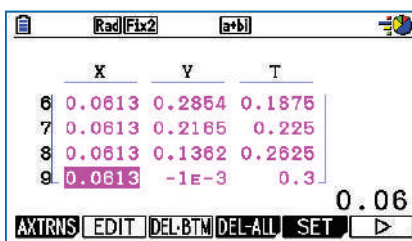
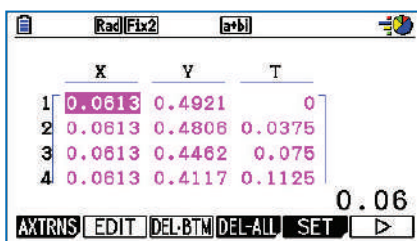


Tempo: 0:03.56-0:03.26=0.3 s

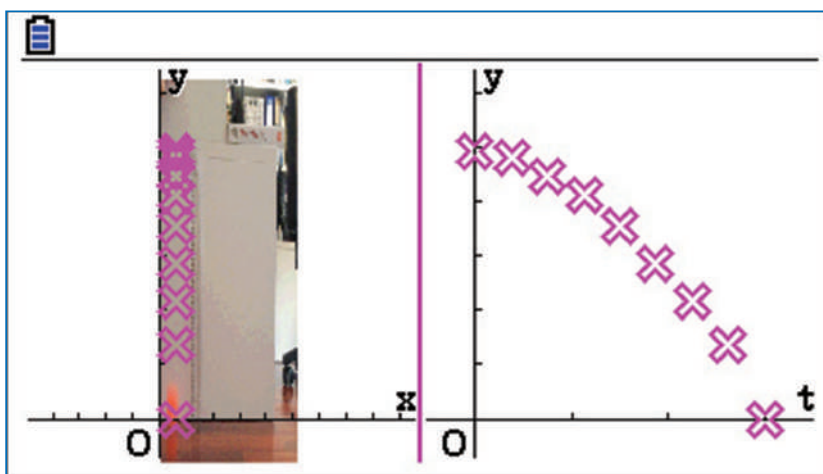
Com base nesta informação, em **F6** (SET), vamos definir o tempo. Considerando que temos 9 *frames*, e 8 tempos entre frames, o início é definido como “0”, o step: 0,3/8. Regressamos ao ecrã anterior, **EXIT**.



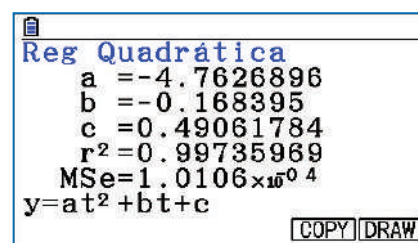
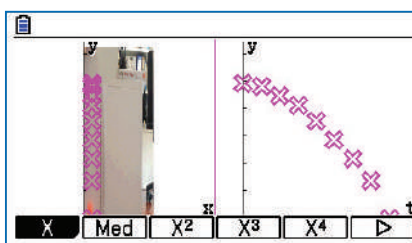
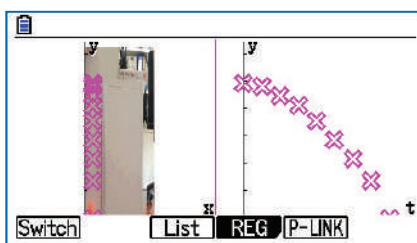
Como queremos a altura de queda em função do tempo, vamos solicitar que o gráfico tenha em consideração estas duas colunas. Fazemos **F1** (AXTRNS) e **F1** (T-Y)



Vemos do lado esquerdo a marcação dos pontos no ficheiro e do lado direito a transformação.



Para calcular a regressão, pressionamos **OPTN**, seguido de **F4** (REG). Escolhemos a regressão quadrática **F3** (X<sup>2</sup>)



Assim, a expressão obtida é **y = - 4,762 t<sup>2</sup> - 0,168 t + 0,491**

Uma vez que valor do coeficiente de correlação r<sup>2</sup> é aproximadamente 0,9958 verifica-se um bom ajuste da função quadrática a este movimento.

**Tarefa 6: Questões**

1. Comparando a expressão obtida com a expressão teórica  $y(t)$ , para um movimento retilíneo uniformemente acelerado, indique o significado físico dos coeficientes de cada termo e comente os respetivos valores obtidos.
2. Qual é a altura que a bola atinge ao no instante  $t=0,15$  segundos?
3. Em que instante a bola passa nas posições 0,20, 0,30, 0,40 m?
4. Para os instantes obtidos na alínea anterior obtenha os módulos da componente escalar da velocidade.

**Tarefa 7: Respostas às questões**

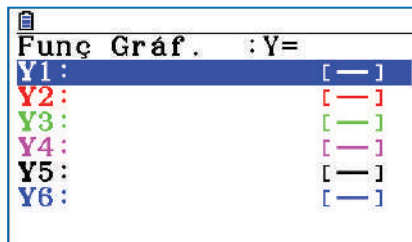
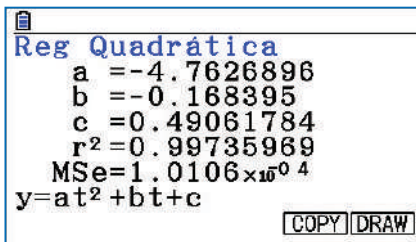
1. A expressão obtida foi, é  $y(t) = - 4,762 t^2 - 0,168 t + 0,491$ , em que:

$y_0 = 0,491m$  corresponde à altura no instante inicial (de onde a bola foi lançada);

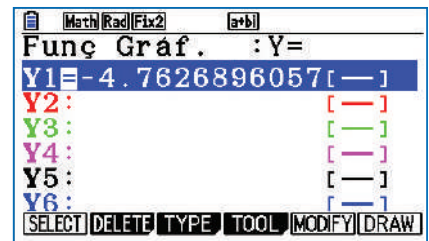
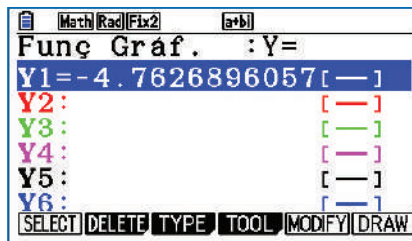
$v_0 = -0,168 \text{ ms}^{-1}$  corresponde ao valor da velocidade no instante inicial. Este valor deveria ser 0, pois a bola foi largada. Esta situação não se verificou provavelmente, por falta de rigor na execução da experiência e na obtenção dos dados. (Nota: Resolução do ecrã da máquina, incerteza em detetar o fundo da bola).

$\frac{1}{2}a = -4,762 \text{ m s}^{-2}$  donde  $a = -9,524 \text{ m s}^{-2}$  corresponde ao valor experimental da aceleração da gravidade. O sinal negativo deste valor significa que a força resultante (força gravítica) tem sentido contrário ao eixo dos  $yy$ .

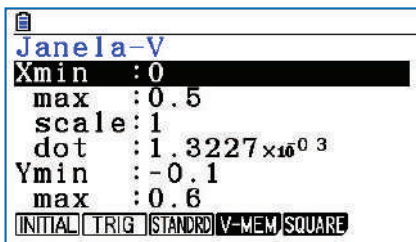
2. Depois de obter a regressão, vamos copiar a expressão para o menu gráfico, fazendo **[F5]** (COPY). Abre o editor de funções. Selecionamos um “Y” e pressionamos **[EXE]**. Ao fazer **[EXE]**, regressa ao ecrã anterior.



Entrar no menu gráfico (Menu 5). A função está disponível em Y1 (local onde a guardamos), mas temos de a ativar, fazendo **[F1]** (SELECT)



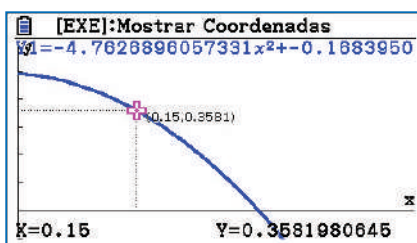
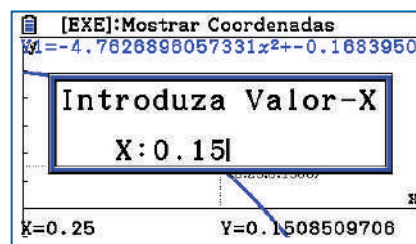
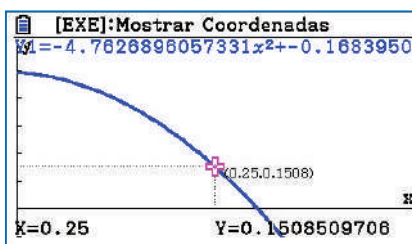
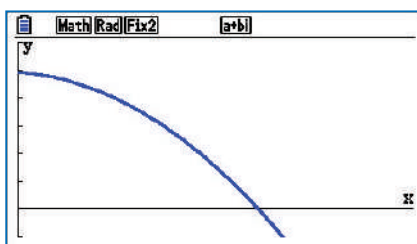
Há que alterar a janela de visualização **[SHIFT]** **[MENU]**



sugestão de janela

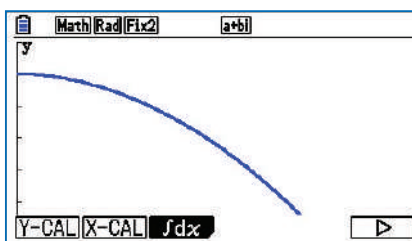
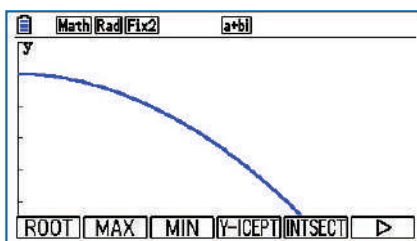


Desenhamos a função, fazendo **[F6]** (DRAW). Ativar o trace (**[F1]**) e introduzir 1,5. Pressionar **[EXE]** e obtemos o resultado.

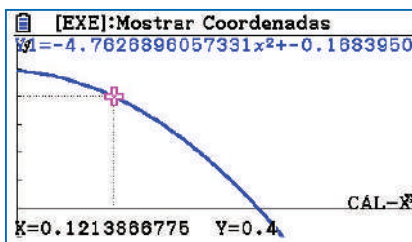
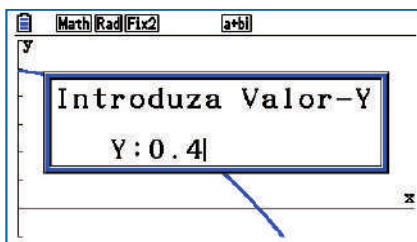
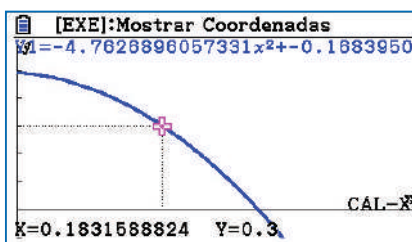
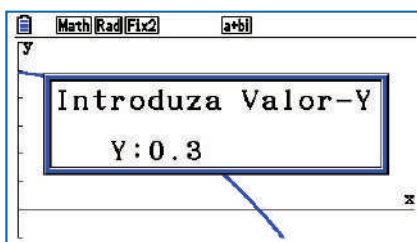
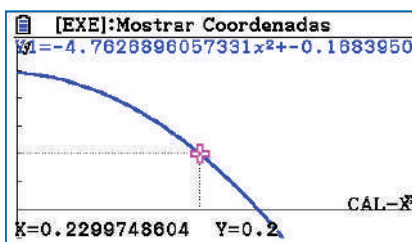
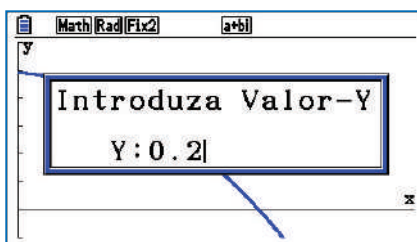


A bola encontrava-se a aproximadamente 36 cm do solo.

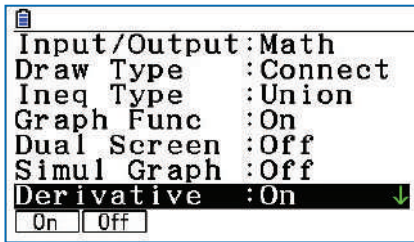
- Escolher **SHIFT [F5]** (G-SOLV), roda a barra de ferramentas (**[F6]**) e escolher **[F2]** (X-CAL). Introduzir a altura e **[EXE]**. O processo deve ser repetido para as três alturas.



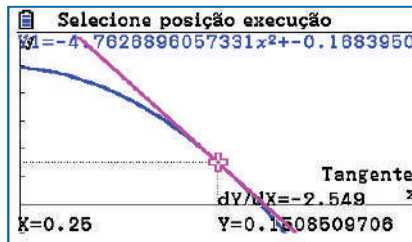
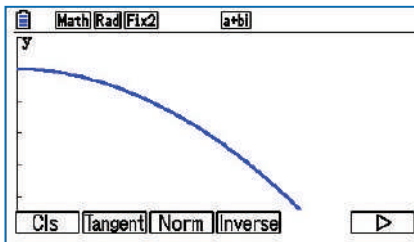
Introduzir a altura e **[EXE]**. O processo deve ser repetido para as três alturas.



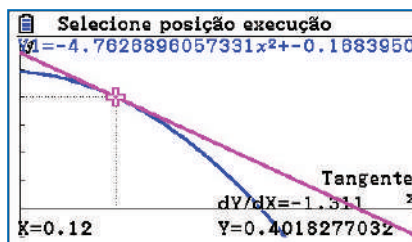
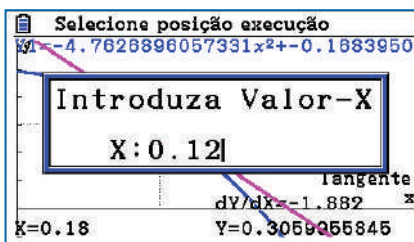
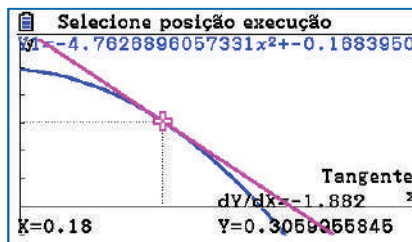
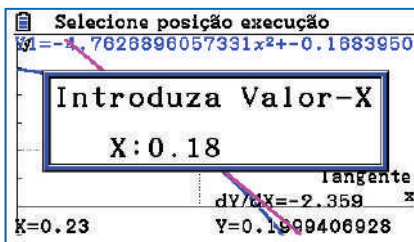
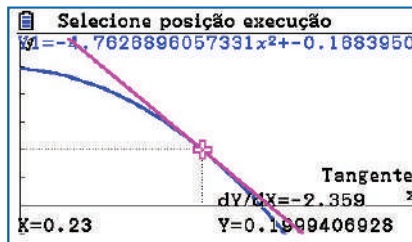
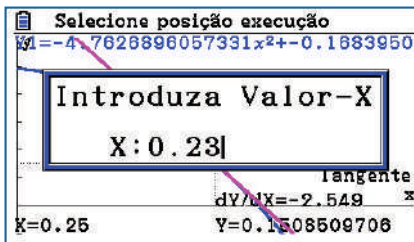
4. A componente escalar da velocidade obtém-se a partir do declive das retas tangentes nos instantes considerados. Se a derivada não estiver ativa, deve fazê-lo em **SHIFT** **MENU**



Com o gráfico desenhado, pressionar **SHIFT** **F4** (Sketch) e selecionar **F2** (Tangent).



Introduzir os instantes. O processo deve ser repetido para os três instantes.



**Atividade realizada na Sessão Prática 4**

“Estudo da Física e Química com a calculadora.”2020/2021 por Ana Margarida Dias

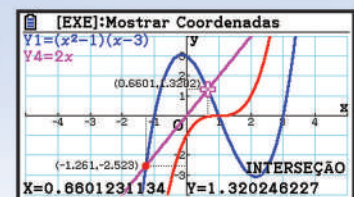
# ESTE ANO APOSTA FORTE NO TEU FUTURO, FAZ A MELHOR OPÇÃO

**A MELHOR OPÇÃO PARA O SECUNDÁRIO**



## VANTAGEM

- Visor a cores de alta resolução
- Modo exame
- Atualizáveis pela Internet
- Muito fáceis de operar
- Cabos incluídos
- Modelos anteriores com o mesmo modo de funcionamento
- Grande fiabilidade e autonomia
- Autorizadas pelo Ministério



## Casio Gráfica fx-CG50

- Menu por ícones
- Folha de cálculo
- Geometria
- Tabela periódica
- Ligação a sensores
- Estatística
- Gráficos em 3D (só na fx-CG50)
- Cálculo de regressões
- Análise gráfica
- Memória 16 Mb
- Imagens e vídeos
- Cálculo vetorial, diferencial e integral
- Reconhecimento automático de sensores (só na fx-CG50)

O grupo de trabalho “CASIO +” desenvolveu a sua participação na APM tendo como principal objetivo criar e desenvolver novas atividades e assim promover o ensino e aprendizagem da Matemática, recorrendo às calculadoras CASIO.

Em 2012 o grupo contava com 8 elementos. Em 2013, aumentou e em janeiro de 2016, o grupo conta com 12 elementos, 11 docentes de matemática do Ensino Básico e Secundário e 1 do Ensino Superior. O grupo está espalhado por Portugal Continental e região autónoma da Madeira, para fazer face às diversas solicitações de formação nas escolas.

Em 2023 o grupo vai crescer, podendo ingressar mais 4 elementos.

Ao longo do ano letivo, o grupo de trabalho realiza inúmeras formações *online*, ACD e anualmente os dias “Casio+” que a cada ano que passa o número de participantes tem vindo a aumentar.

Os elementos que compõem o grupo são:



Ana Margarida Dias  
Lisboa  
Casio School Coordinator



Ana Paula Jardim  
Madeira



António Cardoso  
Reguengos de Monsaraz



Dolcínia Almeida  
Anadia



Elisabete Martins  
Póvoa do Varzim



Elsa Barbosa  
Évora



Isabel Leite  
Vila Verde  
Coordenadora do Grupo



Jaime Carvalho e Silva  
Coimbra



Joaquim Rosa  
Lisboa



Manuela Labrusco  
Reguengos de Monsaraz



Manuel Marques  
Leiria