

```
int i  
#include<iostream>
```

```
Return 0
```

```
While(i <= j){  
    }
```

```
if(x > 4){  
    }
```

Programación: Variables, Operadores y más

```
10100100100
```

```
00001100100
```

```
11101010010
```

```
0101
```

```
0  
1  
1  
1  
1  
0
```

Prof. Nicolás Hidalgo
nicolas.hidalgoc@mail.udp.cl



El inicio en C++

main()



Mansión C++

Un main()

muchas variables

muchas instrucciones

muchas funciones

Estructura base

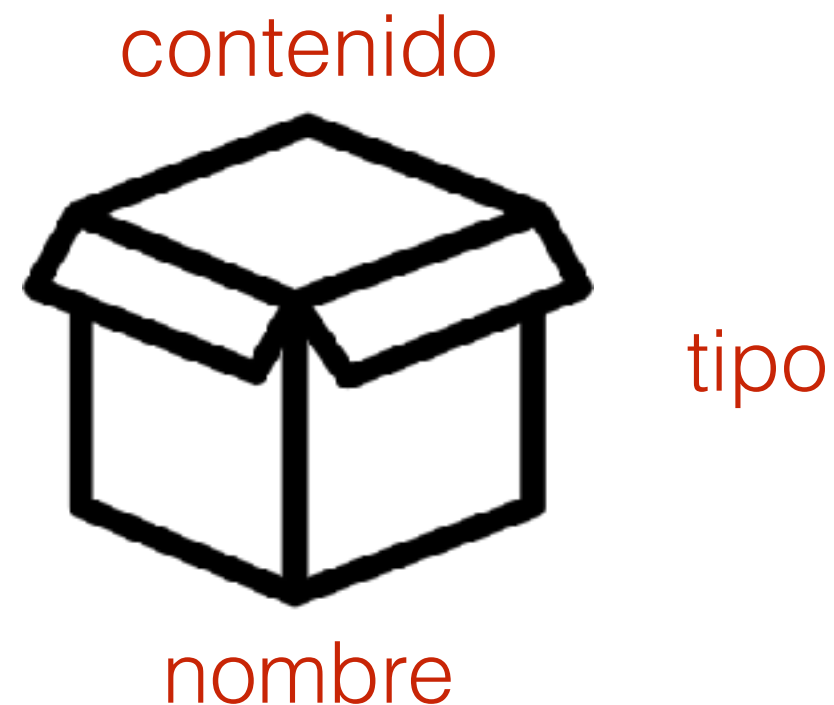
```
#include <iostream>

using namespace std;

int main(){
    // Esto es un comentario C++
    //
    // Aca defino el conjunto de instrucciones
    // para llevar a cabo una tarea en el computador
    █
    return 0;
}
~
~
```

Variables

- ¿Qué es una variable?
 - pensemos en una caja...
- Espacio en el computador
- Contenido puede variar en el tiempo

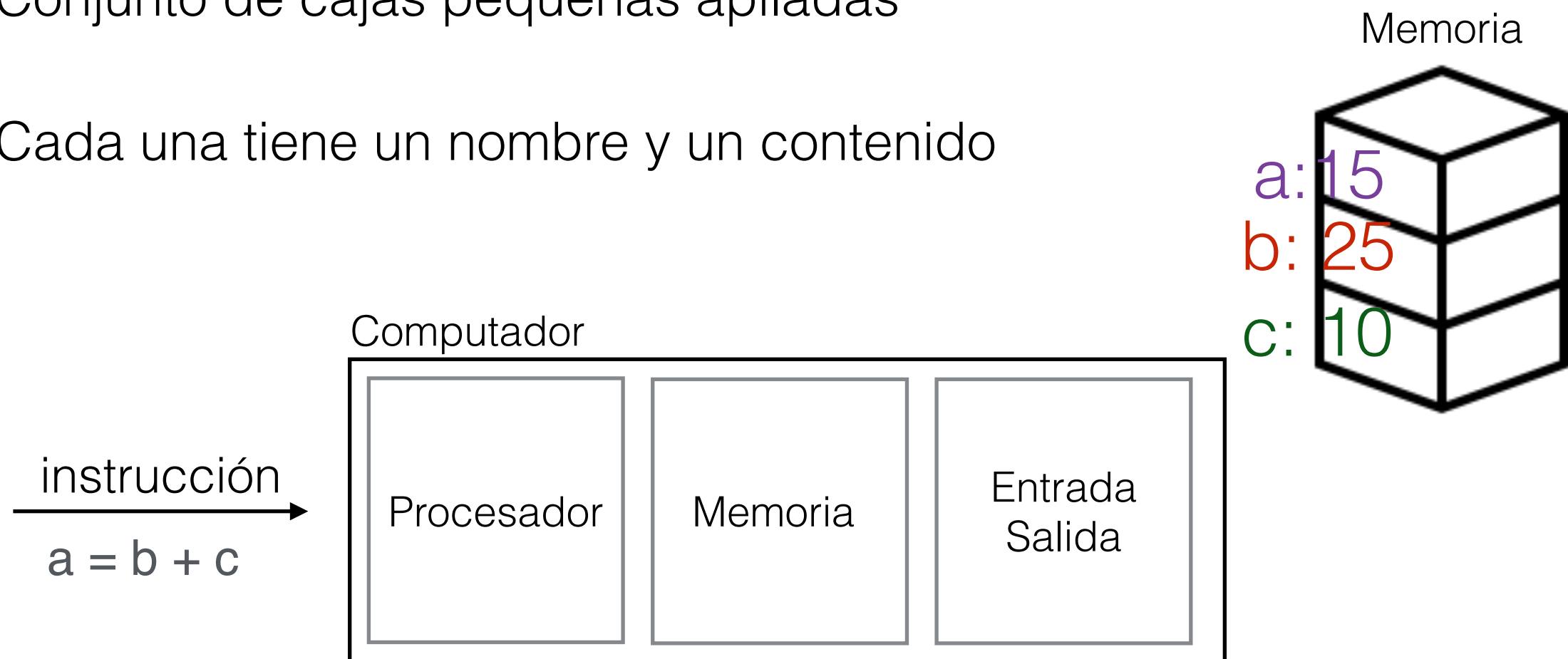


Variables

- Según donde se declaran pueden ser:
 - **Globales**: visibilidad en cualquier lugar del código
 - **Locales**: visibilidad solo en donde están definidas

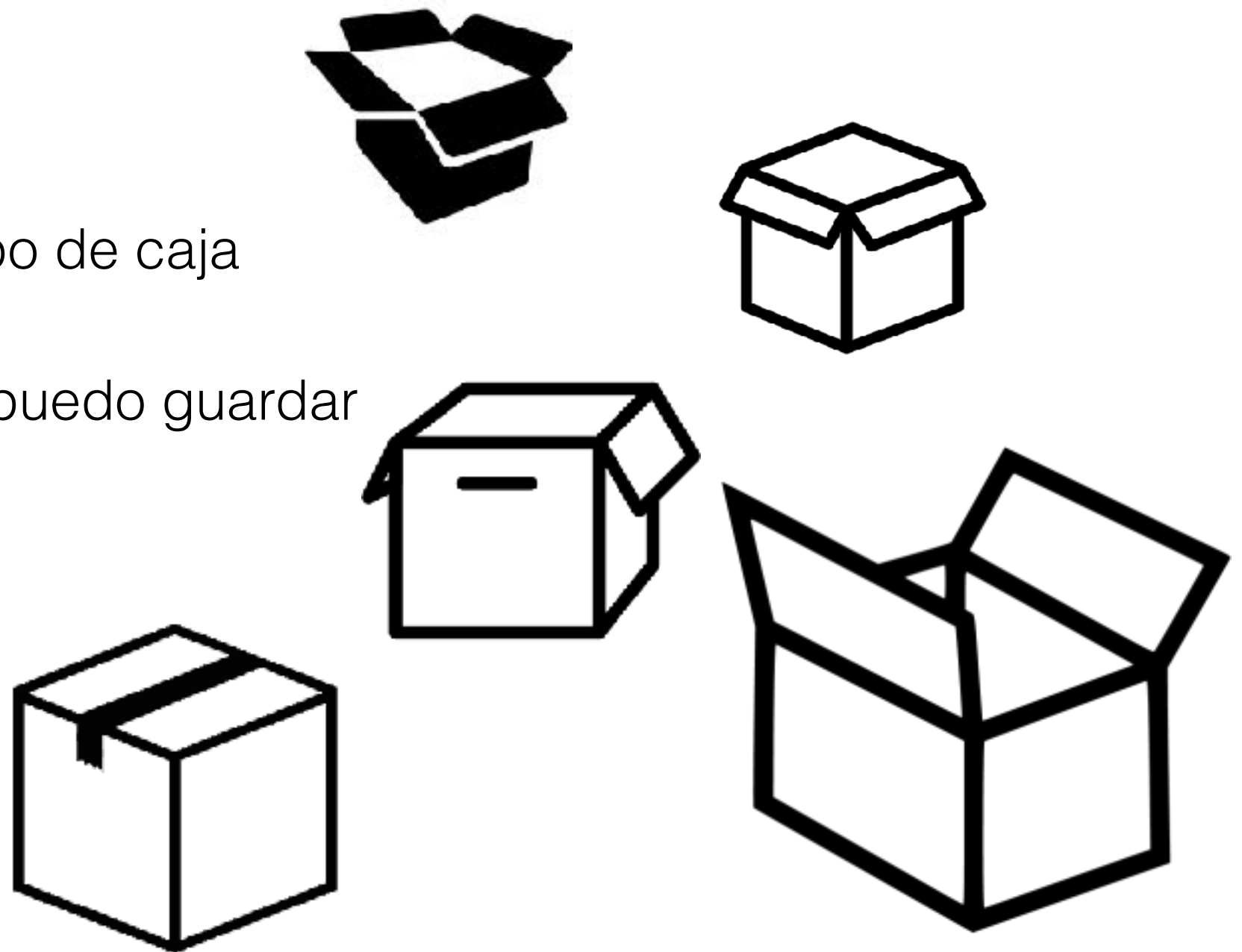
Memoria

- Conjunto de cajas pequeñas apiladas
- Cada una tiene un nombre y un contenido



Tipos de datos

- Ojo! no hay un solo tipo de caja
- forma determina que puedo guardar



Tipos de datos

- **int** (integer o entero): tipo de dato entero, se representan en 4 bytes
4 bytes -> 32 bits
- **char** (Character o caracter): tipo de dato para almacenar caracteres, se representan en 1 byte
1 byte -> 8 bits
- **float** (floating o .flotante): tipo de dato para almacenar números reales, se representan en 4 bytes
4 bytes -> 32 bits
- **double** (flotante doble) tipo de dato para almacenar reales, se representan en 8 bytes
8 bytes -> 64bits
- **boolean** (operador lógico): tipo de dato para representar valores verdaderos o falsos en 1 byte
1 byte -> 8 bits

Pregunta

- ¿Por qué se utiliza 1 byte para una variable boolean si con 1 bit basta?

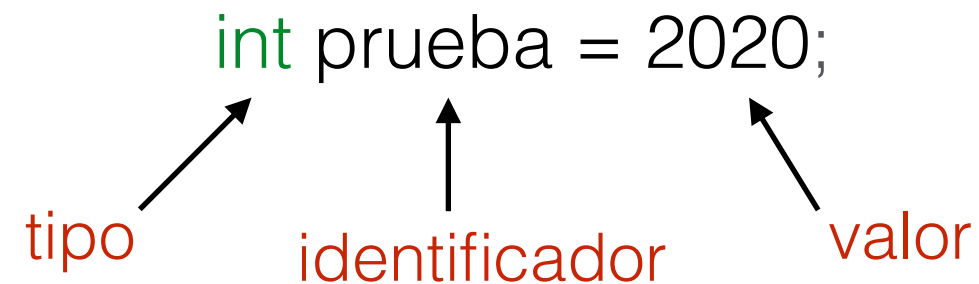
Tipos de datos

- **Modificadores**
 - short: entero corto (aplicable a int), acorta a 2 bytes (16 bits)
 - long: enteros o reales largos (aplicable a int y double)
 - unsigned: hace significativo al bit de signo (aplicable a int)
 - signed: enteros negativos y positivos (aplicable a int)

Variables

- ¿Cómo sería en C++?

`int prueba = 2020;`



tipo identificador valor

<tipo de dato> <nombre de la variable>; `int x;`

<tipo de dato> <lista de variables>; `float x, y, z;`

<tipo de dato> <lista de variables> = valor; `int x = 2017, y = 2020, z = 2021;`

Constantes

- **Constantes declaradas:** variable que no puede ser modificado su valor se designa con la palabra ***const***

const <tipo de dato> <nombre constante> = <valor> ;

- **Constantes definidas:** se declaran mediante la directiva ***#define***

#define <nombre constante> <valor>

Operadores

- Operador suma (+): $\text{int } i = x + y;$
- Operador resta (-): $\text{int } i = x - y;$
- Operador multiplicación (*): $\text{int } i = x * y;$
- Operador división (/): $\text{int } i = x / y;$
- Operador resto (%): $\text{int } i = x \% y;$

Operadores

- Asuma que la variable x posee el valor 4, e y posee el valor 2. ¿Qué valores tendría i para cada operación?

Operador suma (+): $\text{int } i = x + y;$

Operador resta (-): $\text{int } i = x - y;$


Operador multiplicación (*): $\text{int } i = x * y;$

Operador division (/): $\text{int } i = x / y;$

Operador resto (%): $\text{int } i = x \% y;$

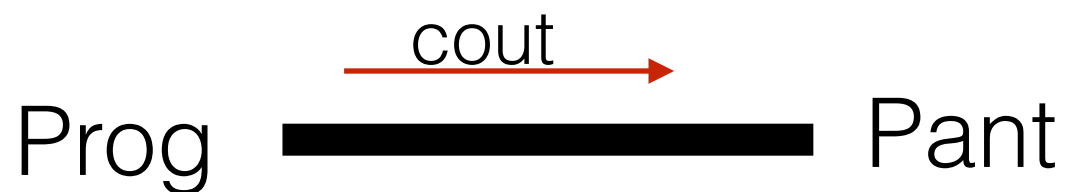
i	x	y
6	4	2
2		
8		
2		
0		

Entrada / Salida

- **¿Cómo ingresamos datos?** 
- la entrada es la parte del programa donde el usuario ingresa datos
- mecanismo más simple es a través del teclado
- En c++ podemos recibir datos y asociarlos a variables con el comando *cin*

cin >> variable;

Entrada / Salida

- **¿Cómo entregamos resultados?** 
- la salida es la parte del programa donde los resultados son entregados al usuario
- mecanismo más simple es a través de la pantalla
- En c++ podemos escribir información por pantalla por medio del comando *cout*

cout << texto;

cout << texto<<variable;

Leamos C++

```
#include<iostream>
using namespace std;

int main(){

cout << "Hola Mundo UDP!" << endl;

return 0;

}
```

Leamos C++

```
#include<iostream>
using namespace std;

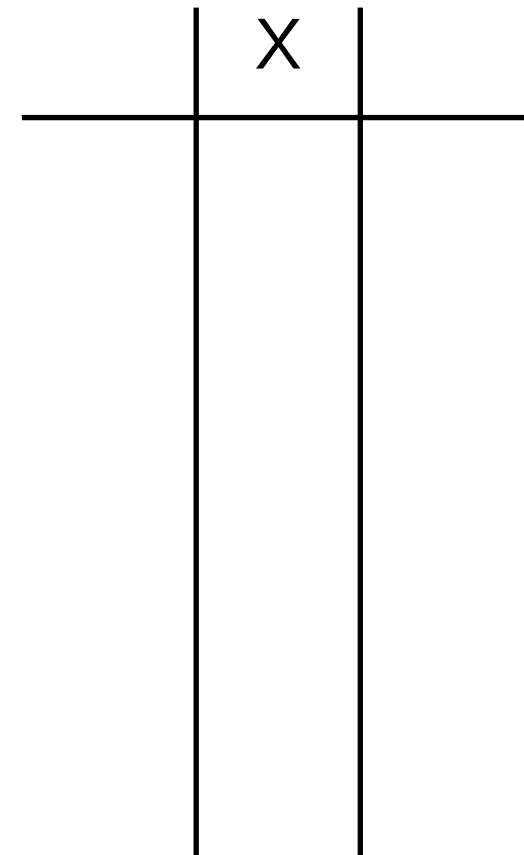
int main(){

int x;
cout << x << endl;

cout << "Escriba un numero: ";
cin >> x;

cout << "salida x : " << x << endl;

return 0;
}
```



Leamos C++

```
#include <iostream>
using namespace std;
```

```
int main (){

int x , y ;

cout << "Ingrese dos numeros" << endl;
cin >> x >> y ;

cout << "Si los sumamos da: " << x+y << endl;

return 0;

}
```

	x	y

Leamos C++

```
#include<iostream>
using namespace std;
```

```
int main(){
```

```
    int i =1;
    float j = 1.1;
```

```
    j = i;
```

```
    if (i == j){
```

```
        cout<< "Iguales!";
```

```
    }else
```

```
        cout << "Diferentes!";
```

```
    return 0;
}
```

	i	j
	1	1.1
		1.0

Leamos C++

```
#include<iostream>
using namespace std;
```

```
int main(){

    int i =1;
    float j = 1.1;

    if (i == j){

        cout<< "Iguales!";
    }else
        cout << "Diferentes!";

    return 0;
}
```

	i	j
	1	1.1

Escribamos en C++

- Realice 4 programas en c++, uno que permita sumar, otro restar, otro multiplicar, y otro dividir dos números definidos en el código.

Pregunta

- ¿Tuvo algún problema?
- ¿Qué problemas les surgieron?

Casting

- Convertir un valor de un tipo de datos a otro tipo de datos
- La conversión puede ser explícita o implícita
 - **Implícita:** no requieren de operador, son llevadas a cabo de manera automática cuando el valor es asignado un tipo de dato compatible
 - Ejemplo: `short a = 2000; int b; b=a;`
 - **Explícita:** uso de un operador para explicitar la conversión de datos
 - Ejemplo: `double x= 10.3; int y; y = (int) x;`
 - `Int x=2, y=5;`

Ejercicio

- Realice un programa en c++ que solicite al usuario ingresar 3 notas y calcule el promedio, donde nota 1 vale un 50%, nota 2 un 30%, y nota 3 un 20%.

Ejercicio

- Realice un programa en c++ que convierta de centímetros a pulgadas. Considere que una pulgada son 2.54 cm.
- E.g:
 - **Entrada:** Ingrese longitud: 45
 - **Salida:** 45cm = 17.7165 in

Ejercicio

- Realice un programa en c++ que permita obtener los ceros de una ecuación cuadrática del estilo $y = Ax^2 + Bx + C$. Considere sólo soluciones reales.