



# Control de Flujo

Universidad Católica Boliviana

MSc, José Jesús Cabrera Pantoja

# Hoy veremos

- Condiciones
- Operaciones Lógicas



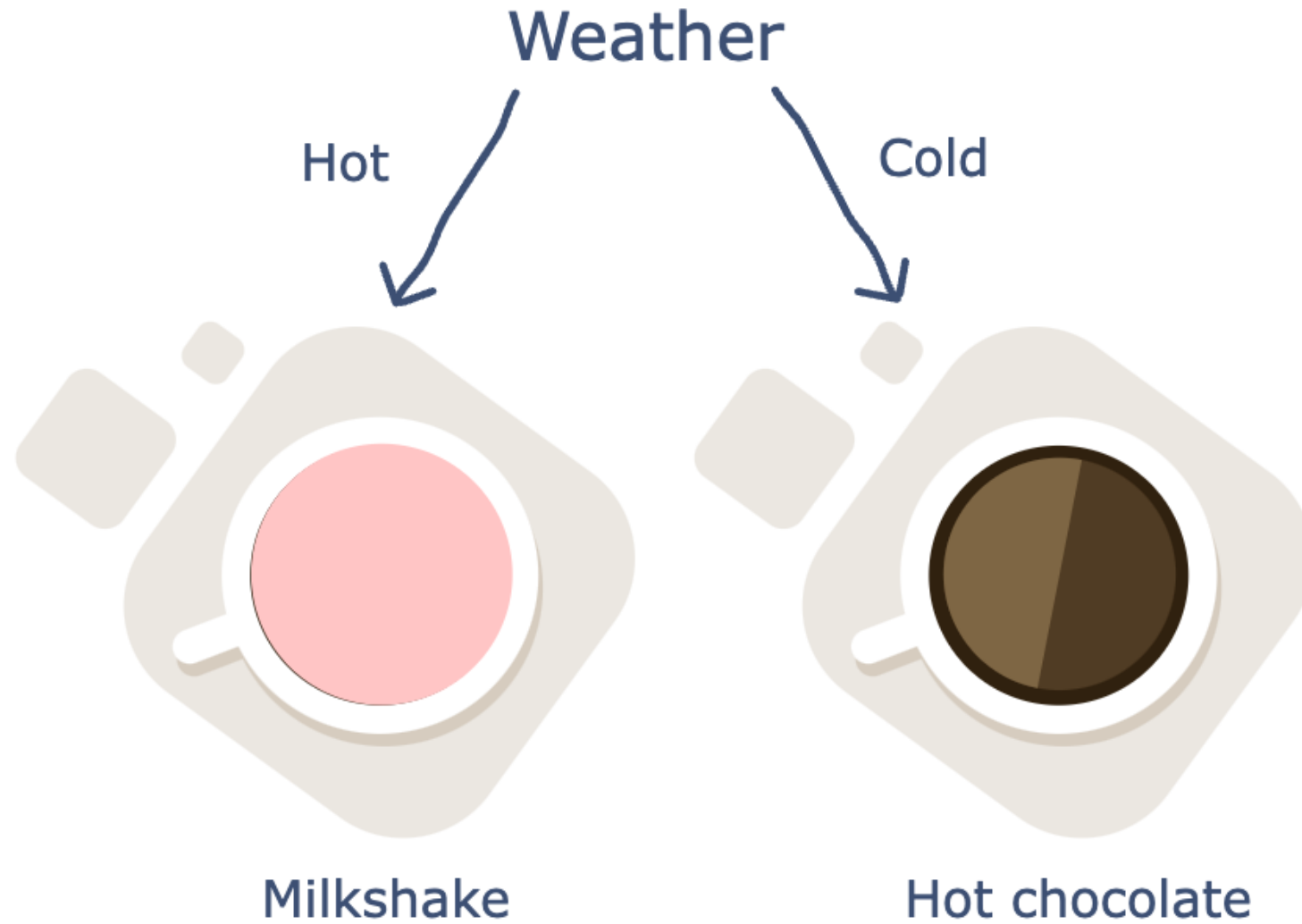


# Condiciones

# Condiciones

- A menudo, tomamos medidas en función de las circunstancias que nos rodean.
- Esto se llama **toma de decisiones**.
- Por ejemplo, si hace calor, podemos tomar una bebida fría como un batido, y si hace frío, podemos optar por beber **chocolate** caliente.

# Condiciones



# Condiciones

- De manera similar, queremos que nuestros programas de computadora decidan qué acción tomar dependiendo del resultado de una(s) condición(es).
- Aquí es donde entran en juego las declaraciones condicionales como **IF**, **IF-ELSE** y **IF-ELSE-IF**.

# Por que las condiciones?

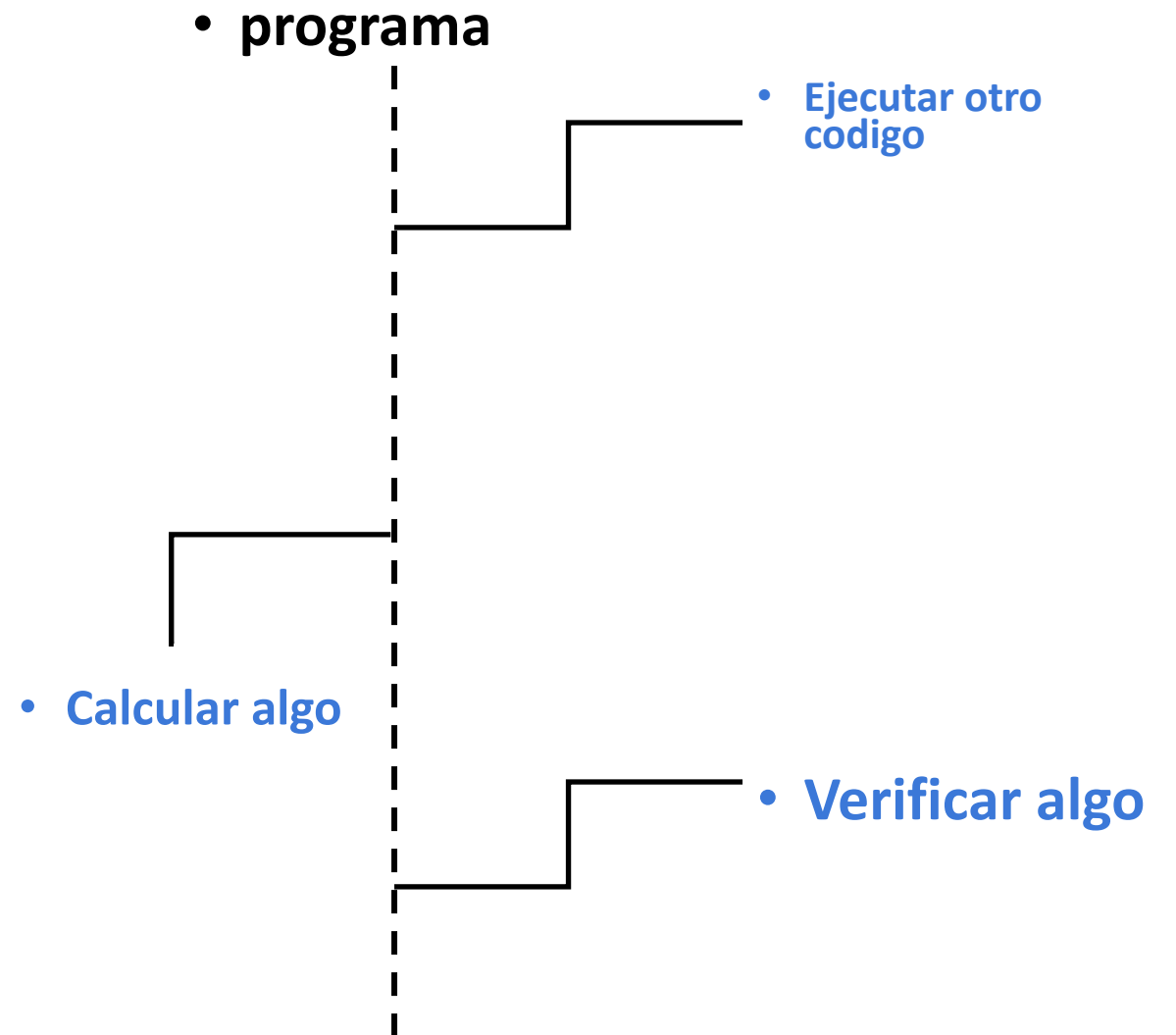
- Comienza por el principio, dijo el rey, muy gravemente, y continúa hasta que llegues al final: Entonces detente. **“Lewis Carroll, El maravilloso mundo de Alicia”**

- programa



# Por que las condiciones?

- Aquí es donde entran en juego las declaraciones condicionales como **IF**, **IF-ELSE** y **IF-ELSE-IF**.





# Condiciones

- Supongamos que queremos que nuestro programa muestre "**beber un batido**" solo si hace sol.
- La declaración **IF** sería útil. Si una condición es verdadera, solo entonces se ejecuta el conjunto de instrucciones en el bloque **IF**.
- **IF** es una **palabra clave**, lo que significa que es una palabra reservada con un significado especial.

# Condiciones: IF



```
1 IF CONDICION  
2     instrucciones
```

# Condiciones: IF


Una expresión que puede ser verdadera o false

Palabra  
Clave



Este bloque de código se  
ejecutara únicamente si la  
expresión es verdadera

# Condiciones: Pseudo codigo



```
1 INPUT weather
2 IF weather == sunny
3     OUTPUT drink a milkshake
```

- **IMPORTANTE:** El operador == verifica la igualdad mientras que el operador = asigna un valor a una variable.

# Condiciones

- Supongamos que queremos que nuestro programa muestre "beber un batido" si hace sol y "beber chocolate caliente" en caso contrario.



# Condiciones: IF-ELSE

Una expresión que puede ser verdadera o false

Palabra  
Clave

Palabra  
Clave

```
1 IF CONDICION
2   instrucciones
3
4 ELSE
5   instrucciones
```

Este bloque de código se ejecutará únicamente si la expresión es **verdadera**

Este bloque de código se ejecutará únicamente si la expresión es **falsa**

# Condiciones: Pseudo Codigo



```
1 INPUT weather
2 IF weather == sunny
3     OUTPUT drink a milkshake
4 ELSE
5     OUTPUT drink hot chocolate
```

- **IMPORTANTE:** El operador == verifica la igualdad mientras que el operador = asigna un valor a una variable.

# Condiciones: IF-ELSE-IF

- Supongamos que queremos verificar múltiples condiciones climáticas como soleado, moderado y frío en nuestro programa.

# Condiciones: IF-ELSE-IF

Una expresión que puede ser verdadera o false

Este bloque de código se ejecutará únicamente si la expresión 1 es **verdadera**. Si es **falsa** seguirá a verificar la siguiente condición

Palabra  
Clave

Palabras  
Clave

Palabras  
Clave

```
1 IF CONDITION 1
2   instrucciones
3
4 ELSE IF CONDITION 2
5   instrucciones
6
7 ELSE
8   instrucciones
```

Otra expresión que puede ser verdadera o false

Este bloque de código se ejecutará únicamente si la expresión 2 es **verdadera**

Este bloque de código se ejecutará únicamente si las expresiones son **falsa**

# Condiciones: Pseudo Codigo



```
1 INPUT weather
2 IF weather == sunny
3     OUTPUT drink a milkshake
4
5 ELSE-IF weather == moderate
6     OUTPUT drink iced tea
7
8 ELSE
9     OUTPUT drink hot chocolate
```

- **IMPORTANTE:** El operador == verifica la igualdad mientras que el operador = asigna un valor a una variable.



# Condiciones en C++



```
1 IF CONDICION
2     instrucciones
```

Pseudo codigo



```
1 if (condicion) {
2     codigo si la condicion es verdadera
3 }
```

C++

# Condiciones en C++



```
1  int main() {  
2      int a, b;  
3  
4      cin >> a >> b;  
5  
6      if (a == b) {  
7          cout << "equal"s << endl;  
8      }  
9  }
```

# Condiciones en C++



```
1 IF CONDICION
2     instrucciones
3
4 ELSE
5     instrucciones
```


Pseudo codigo



```
1 if (condicion) {
2     codigo, si es verdadero
3 } else {
4     codigo, si es falso
5 }
```

C++

# Condiciones en C++



```
1 int main() {  
2     int a, b;  
3     cin >> a >> b;  
4     if (a == b) {  
5         cout << "equal"s << endl;  
6     } else {  
7         cout << "not equal"s << endl;  
8     }  
9 }
```

# Condiciones en C++



```
1 IF CONDITION 1
2     instrucciones
3
4 ELSE IF CONDITION 2
5     instrucciones
6
7 ELSE
8     instrucciones
```

Pseudo codigo



```
1 if ( condicion 1 ) {
2     // ...
3 } else if ( condicion 2 ) {
4     // ...
5 } else {
6     // ...
7 }
```

C++



# Condiciones en C++



```
1 INPUT weather
2 IF weather == sunny
3     OUTPUT drink a milkshake
4
5 ELSE-IF weather == moderate
6     OUTPUT drink iced tea
7
8 ELSE
9     OUTPUT drink hot chocolate
```


Pseudo codigo



```
1 int main() {
2     string weather;
3     cin >> weather;
4
5     if ( weather == "sunny" ) {
6         cout << "drink a milkshake" << endl;
7     } else if ( weather == "moderate" ) {
8         cout << "drink iced tea" << endl;
9     } else {
10        cout << "drink hot chocolate" << endl;
11    }
12 }
```

C++

# Condiciones en C++



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     // Crear una variable con el valor de 21
7     int money = 21;
8
9     // Condicion
10    if (money >= 20) {
11        // instrccion si la condicion es verdadera
12        cout << "You can buy a watch";
13    }
14
15    // fuera de la condicion
16    cout << "This code will be executed" << endl;
17    return 0;
18 }
```

# Condiciones en C++

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

# Condiciones en C++

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     // Crear una variable de valor 6
7     int money = 6;
8
9     if (money >= 20) {
10         cout << "You can gift a watch" << endl;
11     }
12     else if (money >= 10) {
13         cout << "You can gift a comic book " << endl;
14     }
15     else if (money >= 5) {
16         cout << "You can gift a chocolate " << endl;
17     }
18     else {
19         cout << "You can gift a pen " << endl;
20     }
21
22     return 0;
23 }
```



# Condiciones logicas

# Conjunción: And lógico

- Una conjunción es una operación lógica "y" . Está formateado con dos símbolos `&&`. La expresión **a && b** devuelve verdadero si tanto `a` como `b` son verdaderos. De lo contrario, devolverá falso.



# Conjunción: And lógico



```
1 int n, lower_bound, upper_bound;
2
3 ...
4
5 if (lower_bound < n && n < upper_bound) {
6     // Este codigo solo se ejecutara cuando n
7     // se encuentre entre lower_bound Y upper_bound
8     cout << n << " is between "s << lower_bound << "
    and "s << upper_bound;
9 }
```

# Disyuncion: OR lógico

- La disyunción es una operación lógica "o". Está formateado con dos barras verticales `||`. Expresión `a || b` devuelve verdadero si al menos uno de `a` o `b` es verdadero. De lo contrario, devolverá falso.



# Disyuncion: Negacion

- La negación es una operación lógica "no". Está marcado con un signo de exclamación. La expresión  $\neg a$  devuelve falso si  $a$  es verdadero. De lo contrario, devolverá verdadero.

# Condiciones complejas

- Verificar cuando la condición es verdadera
  - Casa de tres pisos y color roja
  - Casa de tres pisos y color verde
  - Casa de dos pisos y color roja
  - Casa de dos pisos y color verde

```
1 string house_color; // color de casa
2 int num_floors;     // numero de pisos en la casa
3
4 ...
5
6 if ((house_color == "green"s && num_floors == 2) || num_floors == 3) {
7     cout << "Enter the "s << house_color << " house with "s
8         << num_floors << " floors"s << endl;
9 }
```

# Condiciones complejas



```
1 (false || true && true || false)
```

# Practica

- Tienes dos números  $n1$  y  $n2$ , y tienes que encontrar el número mayor.
  - Analice le problema (Entradas, salidas, pruebas)
  - Realice el pseudo código
  - Realice el código en C++

# Practica

- Tienes un número  $n$ , y tienes que determinar si  $n$  es par o impar
  - Analice le problema (Entradas, salidas, pruebas)
  - Realice el pseudo código
  - Realice el código en C++
- Debe utilizar el operador %.