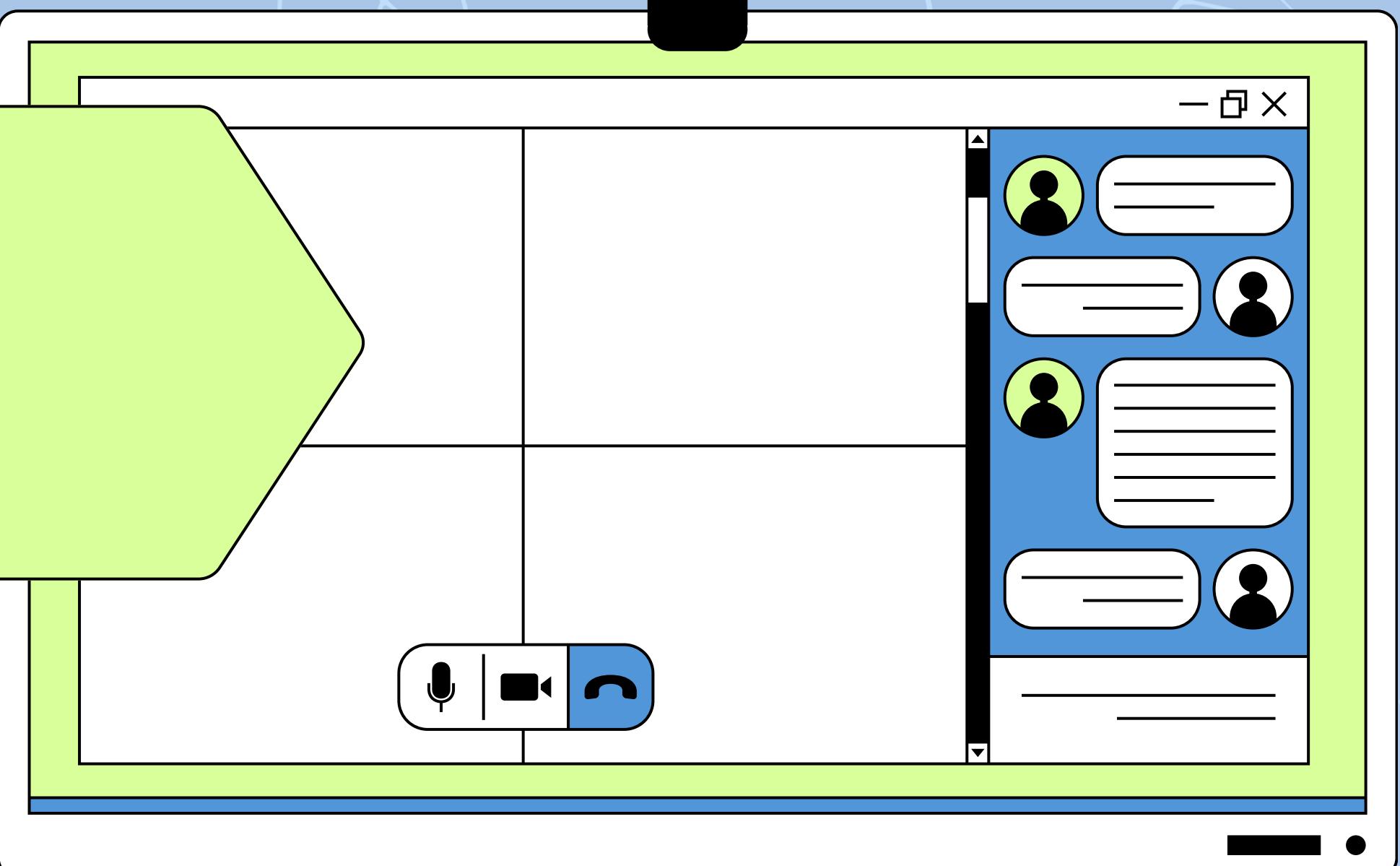


METODO DE INTERCAMBIO

- Hernandez Hernandez Brithany Rubi
- Castillo Cañas Maria Fernanda
- Ceballos Valenzuela Lidia Margarita





Los métodos de ordenamiento son algoritmos que realizan la operación de organizar datos en algún orden o secuencia específica, tal como creciente o decreciente, para datos numéricos, o alfabéticos, para datos de caracteres.

ORDENACION EXTERNA

ORDENACION INTERNA



La ordenación interna o de arreglos, recibe este nombre ya que los elementos o componentes del arreglo se encuentran en la memoria principal de la computadora.

Los métodos de ordenación interna a su vez se clasifican en:

- Métodos directos
- Métodos logarítmicos

METODOS LOGARITMICOS

Los métodos logarítmicos, son más complejos, difíciles de entender y son eficientes en grandes cantidades de datos.

METODOS DIRECTOS

Los métodos directos, son los más simples y fáciles de entender, son eficientes cuando se trata de una cantidad de datos pequeña.



El ordenamiento por intercambio es un algoritmo de ordenamiento de datos.

Se basa en comparar dos o más elementos de conjuntos o lista.

Este proceso se repite hasta que todos los elementos estén ordenados.

Entre estos algoritmos se encuentran el Burbuja (BubbleSort) y QuickSort

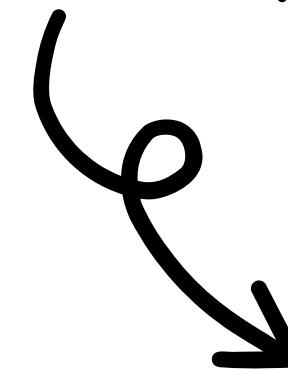


ALGORITMO

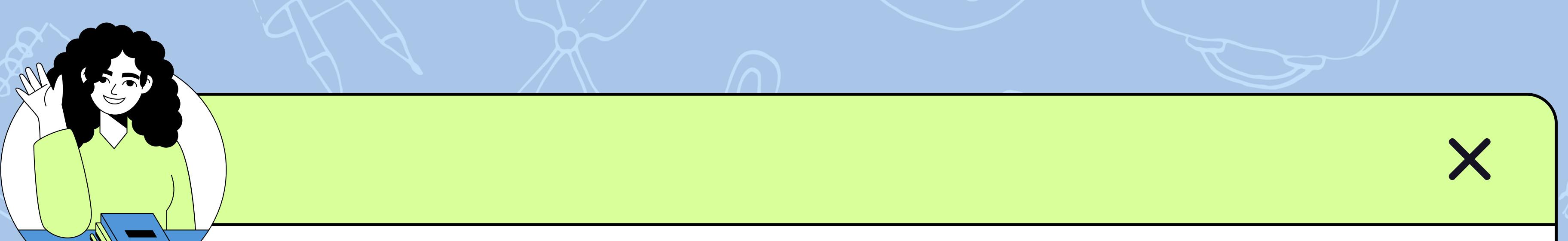


1

Se compara el primer elemento de la lista con el segundo

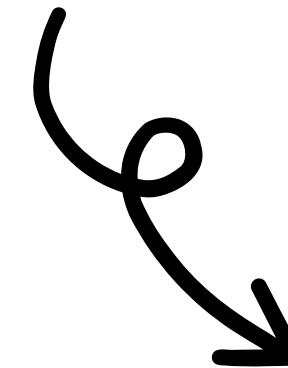


Si esta en orden(dependiendo del criterio de ordenación), se pasa al siguiente par de elementos; de lo contrario se intercambian



2

Se compara el segundo elemento con el tercero, y así sucesivamente hasta el final de la lista



el proceso se repite hasta que no se requieran mas cambios(dependiendo del criterio de ordenación).

```
#include<iostream>
using namespace std;
void intercambio(int[], int);
int main(){
    :
    void intercambio(int[], int);
    :
}
void intercambio(int[], int){
    :
    if(A[j] > A[j+1]) {
        swap ( A[j], A[j+1] );
    }
    :
}
```

```
if(A[j] > A[j+1]) {
    swap ( A[j], A[j+1] );
}
```

VENTAJAS

- EFICIENTE: Para pequeñas cantidades de datos. (especialmente si la lista está casi ordenada)
-
- Tiene una complejidad de tiempo lineal $O(n)$
- Fácil interpretación
- Puede aplicarse a arrays, listas enlazadas, vectores, etc.

DESVENTAJAS

- No adecuado para listas con un alto grado de desorden
- Consumo de memoria adicional: En algunos casos, el intercalamiento puede requerir el uso de memoria adicional para almacenar los datos intercalados temporalmente antes de copiarlos de vuelta al conjunto de salida.



**¡GRACIAS POR
SU ATENCIÓN!**



BIBLIOGRAFIA

- Hidalgo, J. (2001). C++ con clase. <https://conclase.net/c/orden>
- BlackeyeB. (2023, mayo 1). Algoritmos de ordenación explicados con ejemplos en JavaScript, Python, Java y C++. freecodecamp.org. <https://www.freecodecamp.org/espanol/news/algoritmos-de-ordenacion-explicados-con-ejemplos-en-javascript-python-java-y-c/>