

# INFORME DE PROGRAMAS C++

Generado el 31/03/2025 09:14:52

Este informe contiene 1 programa C++ con sus respectivos códigos fuente y resultados de ejecución.

## Contenido del informe:

- Sección 1: Códigos fuente de todos los programas
- Sección 2: Resultados de ejecución de los programas
- Sección 3: Análisis técnico y oportunidades de mejora

*Nota: El informe está organizado por secciones para facilitar la lectura y análisis. La sección de análisis técnico incluye evaluación detallada de cada programa y recomendaciones de mejora.*

## Índice de Programas

La siguiente tabla muestra los programas incluidos en este informe:

Nº	Programa	Tipo
1	clase-1.1	Vector

# SECCIÓN 1: CÓDIGOS FUENTE

## Programa: clase-1.1

Tipo: vector

### **/\* Código Fuente \*/**

Para usar este código: Seleccione todo (Ctrl+A / Cmd+A) → Copie (Ctrl+C / Cmd+C)

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

int binarySearch(int arr[], int L, int R, int key) {
    if(R >= L){
        int mid = (L+R)/2;
        if(arr[mid] == key)
            return mid;

        if(arr[mid] > key)
            return binarySearch(arr, L, mid-1, key);
        return binarySearch(arr, mid+1, R, key);
    }
    return -1; // Retorna -1 cuando no encuentra el elemento
}

int main() {
    int arr[] = {1, 2, 3, 4, 5,6,7,8,9,10};
    int n = 5;
    int key = 4;

    int resultado = binarySearch(arr, 0, n-1, key);

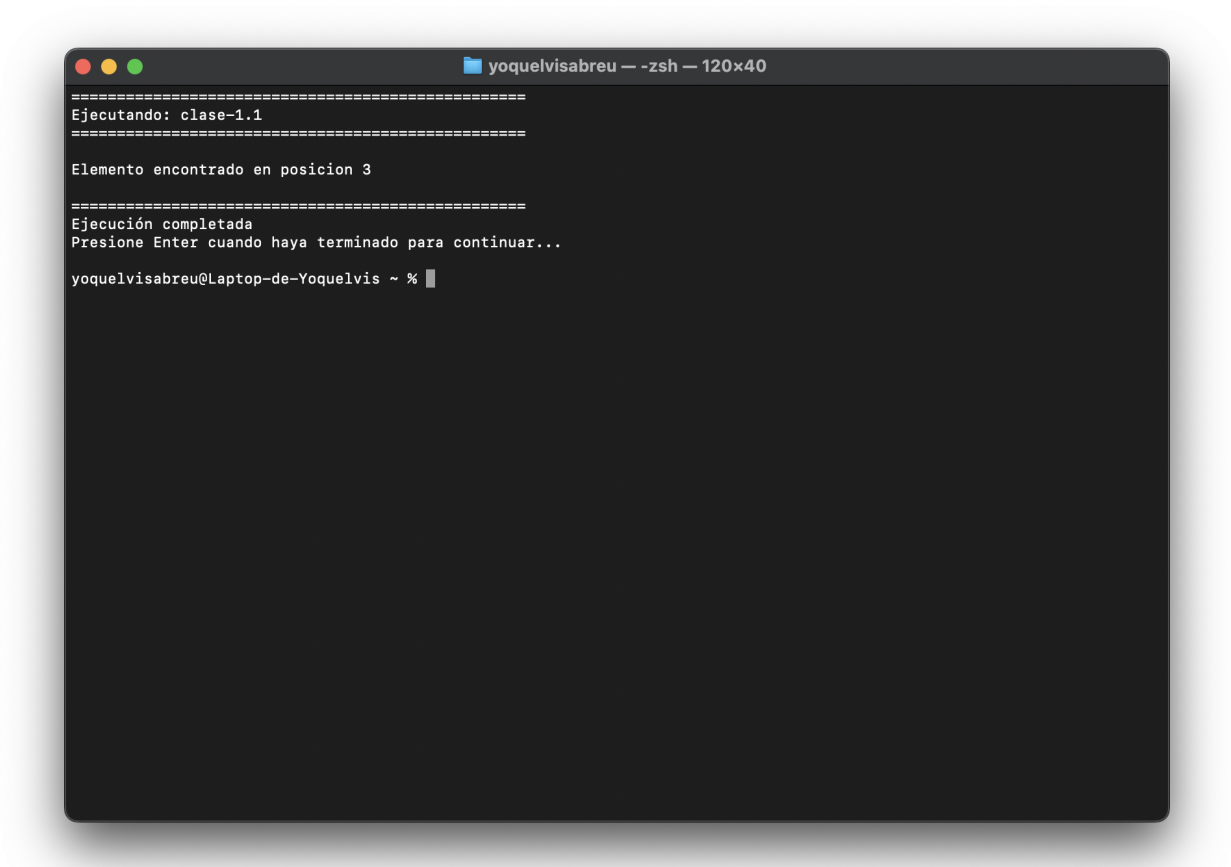
    if(resultado != -1)
        cout << "Elemento encontrado en posicion " << resultado << endl;
    else
        cout << "Elemento no encontrado" << endl;

    return 0;
}
```

# SECCIÓN 2: RESULTADOS DE EJECUCIÓN

## Resultado: clase-1.1

### ■■ Resultado de la Ejecución:



### ■ Información de Compilación:

Compilador: g++ Flags: -Wall -std=c++11 Fecha: 2025-03-31 09:14:52 Estado: Ejecución exitosa Tipo: VECTOR Complejidad: 14/100 Salida: La salida del programa se muestra en la captura de pantalla



# SECCIÓN 3: ANÁLISIS TÉCNICO

## Resumen General de Programas

Este análisis incluye 1 programa C++ de diversos tipos y complejidades.

■ **Distribución por Tipos:**

- Vector: 1 programa (100.0%)

■ **Observaciones Generales:**

- Los programas procesados muestran un rango variado de técnicas de programación en C++
- Se observa un buen uso de las bibliotecas estándar de C++ en la mayoría de los casos
- La complejidad de los programas es adecuada para el contexto educativo

■ **Recomendaciones Generales:**

- Incrementar el uso de comentarios para mejorar la legibilidad y mantenibilidad del código
- Implementar manejo de errores más robusto mediante bloques try/catch
- Considerar la adopción de estándares de codificación consistentes
- Explorar características modernas de C++ (C++11 en adelante) como smart pointers y lambdas

### Análisis del Programa: clase-1.1

■ **Características del Código:**

Característica	Valor
Tipo de Programa	vector
Total de Líneas	30
Uso de Includes	Sí
Uso de Clases	No
Uso de Vectores	No
Uso de Ciclos	Sí
Uso de Funciones	Sí

■ **Análisis de Rendimiento:**

- Complejidad estimada: 15/100 (Nivel: Baja)

■ **Buenas Prácticas:**

- Modularización mediante funciones

■ **Oportunidades de Mejora:**

- Evitar 'using namespace std;' en el ámbito global
- Aumentar la documentación mediante comentarios

