

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>ANALISIS Y DISEÑO DE ALGORITMOS</i>				
<b>TITULO DE LA PRÁCTICA:</b>	<i>Búsqueda en vector ordenado</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>II</i>	<b>AÑO LECTIVO:</b>	<i>2023-B</i>	<b>NRO. SEMESTRE:</b>	<i>B</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>dd/mm/aaaa</i>	<b>HORA DE PRESENTACIÓN</b>	<i>hh/mm/ss</i>		
<b>INTEGRANTE (s)</b> <i>DANIEL WILSTON CHURA MONROY</i>				<b>NOTA (0-20)</b>	<i>Nota colocada por el docente</i>
<b>DOCENTE(s):</b> <i>Nombres de docentes que elaboraron la guía de Laboratorio</i>					

RESULTADOS Y PRUEBAS
<b>I. EJERCICIOS RESUELTOS:</b>

## Búsqueda binaria

Este algoritmo de búsqueda es análogo al método que se utiliza para buscar un nombre en una guía telefónica antigua, esa que parece un libro. Por supuesto, la idea sólo funciona porque el vector está ordenado.

```
// Esta función toma un número entero x
// y un vector creciente v[0..n-1]
// y devuelve un índice j en 0..n
// tal que v[j-1] < x <= v[j].
int
busquedaBinaria (int x, int n, int v[]) {
    int i = -1, d = n; // atención!
    while (i < d-1) {
        int m = (i + d)/2;
        if (v[m] < x) i = m;
        else d = m;
    }
    return d;
}
```

¡Simple y limpio! Los nombres de las variables no fueron elegidos al azar: *i* recuerda “izquierda”, *m* recuerda “medio” y *d* recuerda a “derecha”. El resultado de dividir por 2 en la expresión  $(i+d)/2$  se trunca automáticamente, ya que las variables son de tipo `int`. Por ejemplo, si *i* es 6 y *d* es 9, la expresión  $(i+d)/2$  es 7.

0							<i>i</i>		<i>d</i>			<i>n</i> -1
111	222	333	444	555	555	666	777	888	888	888	999	999

```
#include <iostream>

using namespace std;

int busquedaBinaria(int x, int n, int v[]) {

    int i = -1, d = n;

    while (i < d - 1) {

        int m = (i + d) / 2;

        if (v[m] < x)
```

```
        i = m;

    else

        d = m;

    }

    return d;
}

int main() {

    int arreglo[] = {1, 3, 5, 7, 9, 11, 13, 15};

    int longitud = sizeof(arreglo) / sizeof(arreglo[0]);

    int numero = 8;

    int indice = busquedaBinaria(numero, longitud, arreglo);

    if (indice < longitud) {

        cout << "El elemento " << numero << " debería insertarse en la
posición " << indice << "." << endl;

    } else {

        cout << "El elemento " << numero << " es mayor que todos los
elementos en el arreglo." << endl;

    }

    return 0;
}
```

Link to this code: [\[copy\]](#)

```
options compilation execution
El elemento 9 debería insertarse en la posición 4.

Normal program termination. Exit status: 0
```

1. Para evitar valores de índice fuera del rango 0..n-1, podríamos cambiar la línea `i = -1; d = n` de la función `busquedaBinaria` a través de las siguientes tres líneas. Analice esta variante de código.

```
if (v[n-1] < x) return n;
if (x <= v[0]) return 0;
// ahora v[0] < x <= v[n-1]
i = 0; d = n-1;
```

```
int busquedaBinaria(int x, int n, int v[]) {
    // Si x es mayor que todos los elementos en el arreglo
    if (v[n-1] < x) return n;

    // Si x es menor o igual al primer elemento en el arreglo
    if (x <= v[0]) return 0;

    // Ahora sabemos que v[0] < x <= v[n-1]
    int i = 0, d = n - 1;

    while (i < d) {
        int m = (i + d) / 2;

        if (v[m] < x)
            i = m + 1;
        else
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 5</p>

```

        d = m;

    }

    return d;
}

```

2. Responda las siguientes preguntas sobre la función `busquedaBinaria`. ¿Qué sucede si `"while (i < d-1)"` se reemplaza por `"while (i < d)"`? o por `"while (i <= d-1)"`? ¿Qué pasa si intercambiamos `"(i + d) / 2"` por `"(i + d - 1) / 2"` o por `"(i + d + 1) / 2"` o por `"(d - i) / 2"`? ¿Qué sucede si `"if (v[m] < x)"` se reemplaza por `"if (v[m] <= x)"`? ¿Qué sucede si `"i = m"` se reemplaza por `"i = m+1"` o por `"i = m-1"`? ¿Qué sucede si `"d = m"` se reemplaza por `"d = m+1"` o `"d = m-1"`?

- Si reemplazamos `while (i < d-1)` por:

*while (i < d): La condición del bucle cambiaría y la iteración continuaría mientras i sea menor que d, lo que podría resultar en un bucle infinito si no se maneja correctamente. Esto dependería de cómo se actualicen las variables i y d en el bucle.*

*while (i <= d-1): Esto es equivalente a la condición original, ya que i siempre es menor que d-1 debido a cómo se actualizan en el bucle.*

- Si intercambiamos `(i + d) / 2` por:

- *(i + d - 1) / 2: Esto cambiaría el punto medio calculado durante la búsqueda binaria. Podría hacer que el algoritmo apunte a un índice diferente, lo que podría afectar la precisión de la búsqueda.*
- *(i + d + 1) / 2: Similar al caso anterior, esto cambiaría el punto medio calculado y podría afectar la precisión de la búsqueda.*
- *(d - i) / 2: Esto cambiaría el cálculo del punto medio, lo que podría llevar a errores en la búsqueda binaria.*

- Si reemplazamos `if (v[m] < x)` por:

- *if (v[m] <= x): Esto incluiría el caso en el que x es igual al elemento en v[m], lo que podría cambiar qué índice se devuelve al final del algoritmo.*

- Si reemplazamos `i = m` por:

- *i = m+1: Esto movería el índice i al siguiente elemento, lo que podría afectar la lógica de búsqueda y posiblemente llevar a un resultado incorrecto.*
- *i = m-1: Esto movería el índice i al elemento anterior, lo que también podría afectar la lógica de búsqueda.*

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 6</p>

- Si reemplazamos  $d = m$  por:

$d = m+1$ : Esto movería el índice  $d$  al siguiente elemento, lo que podría afectar la lógica de búsqueda y posiblemente llevar a un resultado incorrecto.

- $d = m-1$ : Esto movería el índice  $d$  al elemento anterior, lo que también podría afectar la lógica de búsqueda.

3. Ejecute la función **busquedaBinaria** con  $n = 16$ . ¿Cuáles son los valores posibles de  $m$  en la primera iteración? ¿Cuáles son los posibles valores de  $m$  en la segunda iteración? ¿En la tercera? ¿En la cuarta?

La función **busquedaBinaria** devolverá 7. Ahora analicemos los valores de  $m$  en cada iteración:

- **Primera iteración:**
  - $i$  se inicializa en 0.
  - $d$  se inicializa en 15.
  - $m$  se calcula como  $(0 + 15) / 2 = 7$ .
- **Segunda iteración:**
  - $i$  es 0 (no se ha actualizado).
  - $d$  es 7 (no se ha actualizado).
  - $m$  se calcula como  $(0 + 7) / 2 = 3$ .
- **Tercera iteración:**
  - $i$  es 0 (no se ha actualizado).
  - $d$  es 3 (no se ha actualizado).
  - $m$  se calcula como  $(0 + 3) / 2 = 1$ .
- **Cuarta iteración:**
  - $i$  es 0 (no se ha actualizado).
  - $d$  es 1 (no se ha actualizado).
  - $m$  se calcula como  $(0 + 1) / 2 = 0$ .

En cada iteración,  $m$  se calcula como el promedio de  $i$  y  $d$ , truncando hacia abajo (ya que  $m$  es un entero). Esto lleva a una reducción en el rango de búsqueda hasta que  $i$  y  $d$  son iguales o contiguos, lo que termina la búsqueda.

4. En la función **busquedaBinaria**, ¿es cierto que  $m$  está en  $0..n-1$  siempre que se ejecuta la sentencia `if (v[m] < x)?` (Tenga en cuenta que  $i$  y  $d$  pueden no estar en  $0..n-1$ .)

Sí, es cierto que cuando se ejecuta la sentencia `if (v[m] < x)` dentro de la función **busquedaBinaria**, el valor de  $m$  siempre estará en el rango de 0 a  $n-1$ , siempre que  $i$  y  $d$  estén en el rango de 0 a  $n-1$ .

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 7</p>

5. Compruebe la validez de la siguiente afirmación: cuando  $n+1$  es una potencia de 2, la expresión  $(i + d)$  es divisible por 2, sean cuales sean  $v$  y  $x$ .

*Cuando  $n+1$  es una potencia de 2, podemos expresar  $n+1$  como  $2^k$  para algún número entero  $k$ . Luego, en la fórmula  $(i + d) / 2$ ,  $i + d$  será una potencia de 2, y por lo tanto, siempre será divisible por 2.*

*Por lo tanto, independientemente de los valores específicos de  $v$  y  $x$ , cuando  $n+1$  es una potencia de 2, la expresión  $(i + d)$  siempre será divisible por 2.*

*Por lo que la afirmación es verdadera.*

6. **Variantes de algoritmos.** Escriba una versión de la función de `busquedaBinaria` que tenga la siguiente invariante: al comienzo de cada iteración,  $v[i-1] < x \leq v[d]$ . Repita con  $v[i-1] < x \leq v[d+1]$ . Repita con  $v[i] < x \leq v[d+1]$ .

- *invariante  $v[i-1] < x \leq v[d]$ :*

```
int busquedaBinaria(int x, int n, int v[]) {
    if (v[n-1] < x) return n;
    if (x <= v[0]) return 0;

    int i = 0, d = n - 1;

    while (i <= d) {
        int m = (i + d + 1) / 2;
        if (v[m] <= x)
            i = m + 1;
        else
            d = m - 1;
    }

    return i;
}
```

}

- *invariante  $v[i-1] < x \leq v[d+1]$ :*

```
int busquedaBinaria(int x, int n, int v[]) {  
  
    if (v[n-1] < x) return n;  
  
    if (x <= v[0]) return 0;  
  
    int i = 0, d = n - 1;  
  
    while (i <= d) {  
  
        int m = (i + d + 1) / 2;  
  
        if (v[m] <= x)  
            i = m + 1;  
  
        else  
            d = m - 1;  
  
    }  
  
    return i;  
}
```

- *invariante  $v[i] < x \leq v[d+1]$ :*

```
int busquedaBinaria(int x, int n, int v[]) {  
  
    if (v[n-1] < x) return n;  
  
    if (x <= v[0]) return 0;
```



```
int i = 0, d = n - 1;

while (i <= d) {

    int m = (i + d + 1) / 2;

    if (v[m] < x)

        i = m + 1;

    else

        d = m - 1;

}

return i;
}
```

## II. PRUEBAS

*¿Con que valores comprobaste que tu práctica estuvo correcta? ¿Qué resultado esperabas obtener para cada valor de entrada? ¿Qué valor o comportamiento obtuviste para cada valor de entrada?*

*Con números del 1 al 9. Con los que el programa soluciona de manera óptima.*

## III. CUESTIONARIO:

*Colocar la evidencia de las respuestas realizadas al cuestionario enunciado en la guía práctica de laboratorio.*

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 10

```

Link to this code: [link]
options compilation execution
El elemento 8 debería insertarse en la posición 7.

Normal program termination. Exit status: 0

```

## CONCLUSIONES

*Este es un laboratorio en el que analizamos las diferentes variantes del algoritmo de búsqueda binaria, los cambios que ocurren y cómo afectan al algoritmo*

## METODOLOGÍA DE TRABAJO

*Colocar la metodología de trabajo que ha utilizado el estudiante o el grupo para resolver la práctica, es decir el procedimiento/secuencia de pasos en forma general.*

## REFERENCIAS Y BIBLIOGRAFÍA

*Colocar las referencias utilizadas para el desarrollo de la práctica en formato IEEE*

- <https://github.com/Dan1elMon/ADA-LABS.git>
- [https://www.w3schools.com/cpp/cpp\\_functions\\_recursion.asp](https://www.w3schools.com/cpp/cpp_functions_recursion.asp)
-