EIE
Escuela de
Ingeniería Eléctrica

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

UNIVERSIDAD DE COSTA RICA

`IE-0217` Estructuras abstractas de datos y algorítmos para ingeniería

Lab 4: Imprementacion de arreglos, clases, plantillas, algoritmos de búsqueda y ordenamiento genericos en C++

Timna Belinda Brown Ramírez

B61254

`timna.brown@ucr.ac.cr`

`belindabrownr@gmail.com`

I-2019

Tabla de contenidos

1. Consideraciones **1**

2. Abordaje y conclusiones **2**

3. Apéndice **2**

3.1. Código fuente 2

---

1. Consideraciones

- Laboratorio individual[1]

- Genere un reporte en LaTeX que incluya su código, su abordaje para la solución y sus conclusiones.[2]
- Suba su código y su documentación (doxygen, README, INSTALL) al git respectivo de su grupo y el directorio del laboratorio. Use su número de carné a diferenciar los trabajos de su grupo.
- Cada estudiante debe subir el reporte a *Schoology*.
- Recuerde que por cada día tardía de entrega se le rebajaran puntos de acuerdo con la formula: 3^d donde d > 1 es la cantidad de día tardíos.

2. Abordaje y conclusiones

Para la resolución del laboratorio presentado, se realizaron una serie de clases y funciones que cumplen con los objetivos del laboratorio 4. Las cuales incluyen los métodos de ordenamiento y búsqueda genéricos y hacer listas enlazadas.

Como conclusióon se puso en práctica el uso del lenguaje C++, además, del uso de la lógica para cumplir el objetivo planteado.

Los documentos con el cóodigoque permite la resolución de este laboratorio se encuentran en el git del grupo 7 en la subdivisión "Lab4-B61254".[4]

3. Apéndice

3.1. Código fuente

[3]

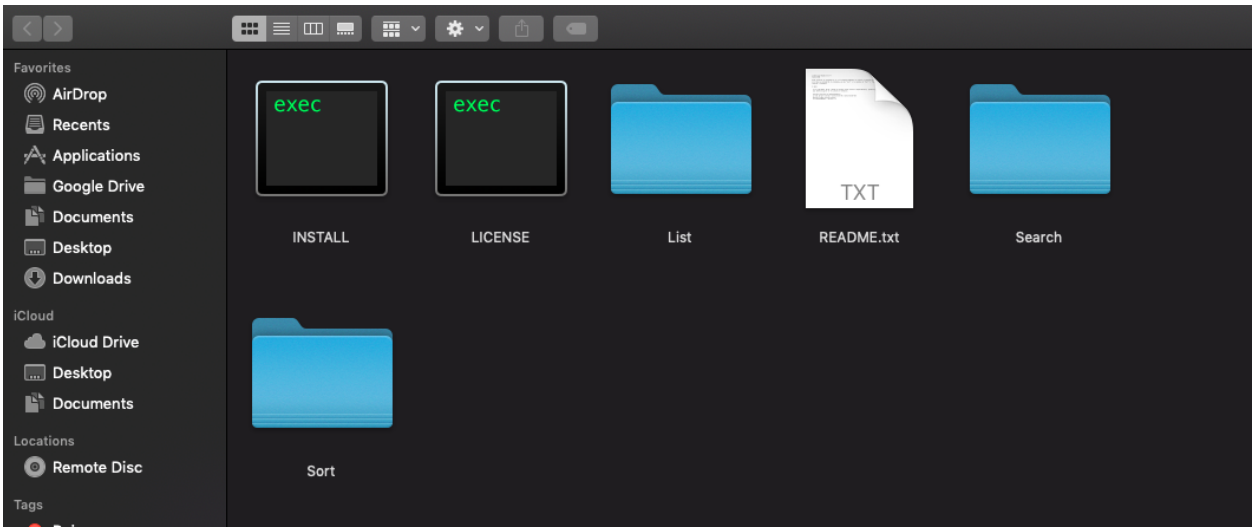

Figura 1: Dentro de Lab4

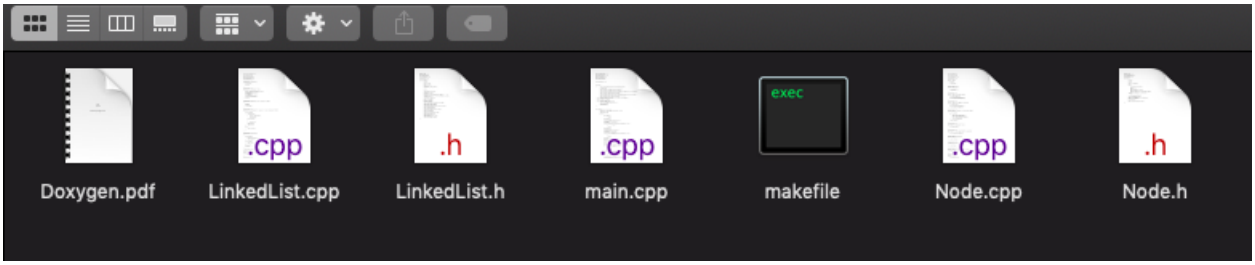

Figura 2: Dentro de List

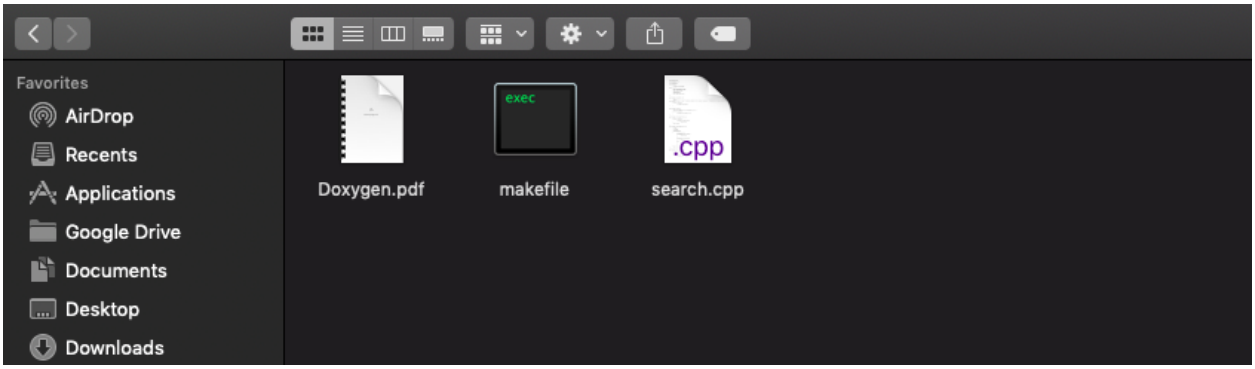

Figura 3: Dentro de Search

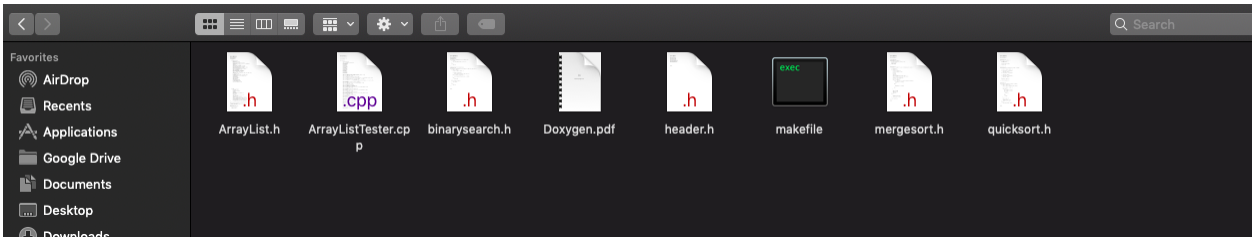

Figura 4: Dentro de Sort

Todos tienen en común lo siguiente:

```
Listas y arreglos en C++
Junio 2019

Para ejecutar el programa es ir a la carpeta mediante su consola o terminal al folder
en el que se encuentra el programa, ya sea "List" o la carpeta de "Sort" o "Search. Debe ingresar y digitar en la consola o
terminal:

$ make

 Si el copilador falla, puede ser porque tiene ciertos requerimeintos, posibles, por lo
 que debera digitar en la consola o terminal:

 Instale librerias en Debian/Ubuntu:
 $ sudo apt-get install libncurses5-dev libncursesw5-dev
 Necesita algun actualizacion:
 $ softwareupdate --install -a
```

Figura 5: Readme

```
Imprementacion de arreglos, clases, plantillas, algoritmos de busqueda
y ordenamiento genericos en C++

Belinda Brown - timna.brown@ucr.ac.cr
License Apache 2.0


Se distribuye un Makefile con 3 reglas:

    * build: compila los fuentes.
    * clean: borra los binarios.
    * run: ejecuta un corrida de ejemplo.
```

Figura 6: Install

```
all: build run clean

build:
	g++ -g --std=c++11 -Wall *.cpp -o a.exe


run:
	./a.exe

clean:
	rm a.exe
	rm -rf a.exe.dSYM
```

Figura 7: Makefile

search.cpp

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

const int MAX=20;
class Busqueda
{
  int numero_buscado[MAX];
  int n;
public:
  bool linearSearch(int key);
  bool binarySearch(int key);
  void accept(int num);
  void display();
  int fibo(int j);
  bool fibSearch(int key);

};

void Busqueda::accept(int num)
{
  n=num;
  cout<<"\nDigite "<<num<<" elementos, dandole enter al digitar cada opcion  \nllenandolo con numeros aleatorios que desee \n";
  for(int i=0;i<n;i++)
  {
    cin>>numero_buscado[i];
  }
}

void Busqueda::display()
{
  cout<<"\nLos numeros que se encuentran son:\n ";
  for(int i=0;i<n;i++)
  {
    cout<<numero_buscado[i]<<"\t";
  }
```

~/Desktop/Lab4/Search/search.cpp 1:1 LF UTF-8 C++ Not on branch GitHub Git (60737)

Figura 8: Código de search.cpp

search.cpp

```
36  }
37
38  bool Busqueda::linearSearch(int key)
39  {
40    for(int i=0;i<n;i++)
41    {
42      if(numero_buscado[i]==key)
43        return true;
44    }
45    return false;
46  }
47
48  bool Busqueda::binarySearch(int key)
49  {
50    int mid;
51    int low=0;
52    int high=n-1;
53    while(low<=high)
54    {
55      mid=(low+high)/2;
56      if(key==numero_buscado[mid])
57      {
58        return true;
59      }
60      if(key<numero_buscado[mid])
61      {
62        high=mid-1;
63      }
64      else
65      {
66        low=mid+1;
67      }
68    }
69    return false;
70  }
```

~/Desktop/Lab4/Search/search.cpp 1:1

Figura 9: Código de search.cpp

search.cpp

```
71
72  int Busqueda::fibo(int j)
73  {
74    if(j==0)
75      return 0;
76    if(j==1)
77      return 1;
78    else
79      return ((fibo(j-1))+(fibo(j-2)));
80  }
81
82  bool Busqueda::fibSearch(int key)
83  {
84    int f1,f2,j,mid;
85    j=1;
86    while(fibo(j)<=n)
87      j++;
88
89
90    f1=fibo(j-2);
91    f2=fibo(j-3);
92    mid=n-f1+1;
93
94    while(key!=numero_buscado[mid])
95    {
96      if(key>numero_buscado[mid])
97      {
98        if(f1==1)
99          break;
100       mid=mid+f2;
101       f1=f1-f2;
102       f2=f2-f1;
103     }
104     else
105     {
```

~/Desktop/Lab4/Search/search.cpp 1:1

Figura 10: Código de search.cpp

search.cpp

```
106         if(f2==0)
107           break;
108         mid=mid-f2;
109         int temp=f1-f2;
110         f1=f2;
111         f2=temp;
112       }
113     }
114     if(numero_buscado[mid]==key)
115       return true;
116     else
117       return false;
118   }
119   int main()
120   {
121     Busqueda s1;
122     int choice;
123     int key,n;
124     do
125     {
126       cout<<"\n1. Linear Search";
127       cout<<"\n2. Binary Search";
128       cout<<"\n3. Fibonacci Search";
129       cout<<"\n0. Salir";
130       cout<<"\nChoice: ";
131       cin>>choice;
132       switch(choice)
133       {
134       case 1:
135         cout<<"\nDigite el numero de elementos que desea ";
136         cin>>n;
137         s1.accept(n);
138         cout<<"\nIngrese el numero que desea buscar ";
139         cin>>key;
140         if(s1.linearSearch(key))
```

~/Desktop/Lab4/Search/search.cpp 1:1

Figura 11: Código de search.cpp

search.cpp

```
140         if(s1.linearSearch(key))
141           cout<<"\nNumero buscado... "<<key<<" ... Sí se encuentra";
142         else
143           cout<<"\nNumero buscado... "<<key<<" ... No se encuentra";
144         break;
145
146       case 2:
147         cout<<"\nDigite el numero de elementos que desea ";
148         cin>>n;
149         s1.accept(n);
150         cout<<"\nIngrese el numero que desea buscar ";
151         cin>>key;
152         if(s1.binarySearch(key))
153           cout<<"\nNumero buscado... "<<key<<" ... Sí se encuentra";
154         else
155           cout<<"\nNumero buscado... "<<key<<" ... No se encuentra";
156         break;
157       case 3:
158         cout<<"\nDigite el número de elementos que desea ";
159         cin>>n;
160         s1.accept(n);
161         cout<<"\nIngrese el numero que desea buscar ";
162         cin>>key;
163         if(s1.fibSearch(key))
164           cout<<"\nNumero buscado... "<<key<<" ... Sí se encuentra";
165         else
166           cout<<"\nNumero buscado... "<<key<<" ... No se encuentra";
167         break;
168       }
169
170     }while(choice!=0);
171
172     return 0;
173   }
174
```

~/Desktop/Lab4/Search/search.cpp* 171:2

Figura 12: Código de search.cpp

```
Belindas-MacBook-Air:Search belindabrown$ make
g++ -g --std=c++11 -Wall *.cpp -o a.exe
./a.exe

1. Linear Search
2. Binary Search
3. Fibonacci Search
0. Salir
Choice: 1

Digite el numero de elementos que desea 2

Digite 2 elementos, dandole enter al digitar cada opcion
llenandolo con numeros aleatorios que desee
32
43

Ingrese el numero que desea buscar 32

Numero buscado... 32 ... Sí se encuentra
```

Figura 13: Resultados de search

```
1. Linear Search
2. Binary Search
3. Fibonacci Search
0. Salir
Choice: 2

Digite el numero de elementos que desea 3

Digite 3 elementos, dandole enter al digitar cada opcion
llenandolo con numeros aleatorios que desee
23
21
54

Ingrese el numero que desea buscar 32

Numero buscado... 32 ... No se encuentra
1. Linear Search
2. Binary Search
3. Fibonacci Search
0. Salir
Choice: 3

Digite el número de elementos que desea 2

Digite 2 elementos, dandole enter al digitar cada opcion
llenandolo con numeros aleatorios que desee
32
43

Ingrese el numero que desea buscar 32

Numero buscado... 32 ... No se encuentra
1. Linear Search
2. Binary Search
```

Figura 14: Resultados de search

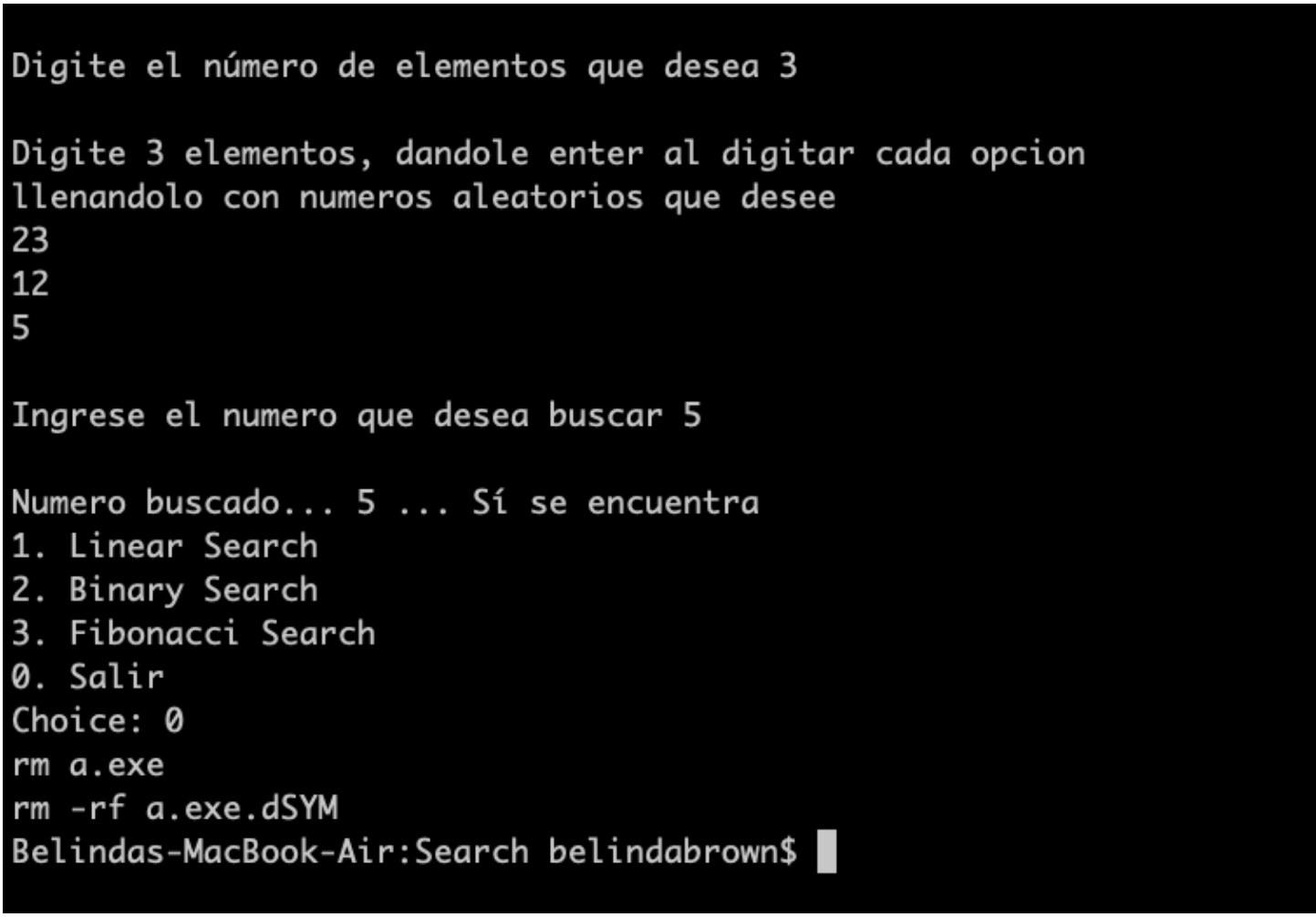

Figura 15: Resultados de search

```
Belindas-MacBook-Air:~ belindabrown$ cd Desktop/Lab4/Sort
Belindas-MacBook-Air:Sort belindabrown$ make
g++ -g --std=c++11 -Wall *.cpp -o a.exe
./a.exe
Agregando 16 valores

Los valores que hemos agregado son:
El tamano de la lista es de: 16
(438787, 17, 147, 3, -37, 584, -2147, 0, -98, 247, -37, 0, 27, 187, 824, -1

Obteniendo el elemento en la posicion 0
El elemento que se encuentra en esta posicion es: 438787

Obteniendo elemento en 15
El elemento en 15 es: -17

Eliminando el índice 22 (fuera de límites)
El tamaño de la lista es: 16
(438787, 17, 147, 3, -37, 584, -2147, 0, -98, 247, -37, 0, 27, 187, 824, -1

Eliminando el índice 0
Fue eliminado 17
Ahora, el tamano de esta lista es de: 15
(17, 147, 3, -37, 584, -2147, 0, -98, 247, -37, 0, 27, 187, 824, -17)

Eliminando el valor en el índice 5
Ahora, el tamano de esta lista es de:14
(17, 147, 3, -37, 584, 0, -98, 247, -37, 0, 27, 187, 824, -17)

Se eliminara el elemento: -2147
Ahora, el tamano de esta lista es de: 14
(17, 147, 3, -37, 584, 0, -98, 247, -37, 0, 27, 187, 824, -17)

Se eliminara el elemento 17
Ahora, el tamano de esta lista es de: 14
```

Figura 16: Resultados de sort

```
Se eliminara el elemento 17
Ahora, el tamano de esta lista es de: 14
(17, 147, 3, -37, 584, 0, -98, 247, -37, 0, 27, 187, 824, -17)

Probando Selection Sort
(-98, -37, -37, -17, 0, 0, 3, 17, 27, 147, 187, 247, 584, 824)

Haciendo aleatoria el ArrayList
Ahora la lista contiene:
(584, -37, 0, 3, -17, 0, 824, 27, -98, 247, 147, -37, 17, 187)

Probando Quick Sort
La lista ordenada es:
(-98, -37, -37, -17, 0, 0, 3, 17, 27, 147, 187, 247, 584, 824)

Volviendo aleatoria la lista
Ahora la lista contiene:
(17, 27, 584, 3, 0, -98, -17, -37, -37, 187, 0, 247, 824, 147)

Probando Merge Sort
(-98, -37, -37, -17, 0, 0, 3, 17, 27, 147, 187, 247, 584, 824)

Binary Search Test
Buscando el 584
El índice de 584 es: 12


Buscando el -888
El indice de -888 es: -1
Se espera un -1, lo que significa que no esta en la lista

Agregando el numero 49 al ArrayList
Ahora contiene
(-98, -37, -37, -17, 0, 0, 3, 17, 27, 49, 147, 187, 247, 584, 824)
```

Figura 17: Resultados de sort

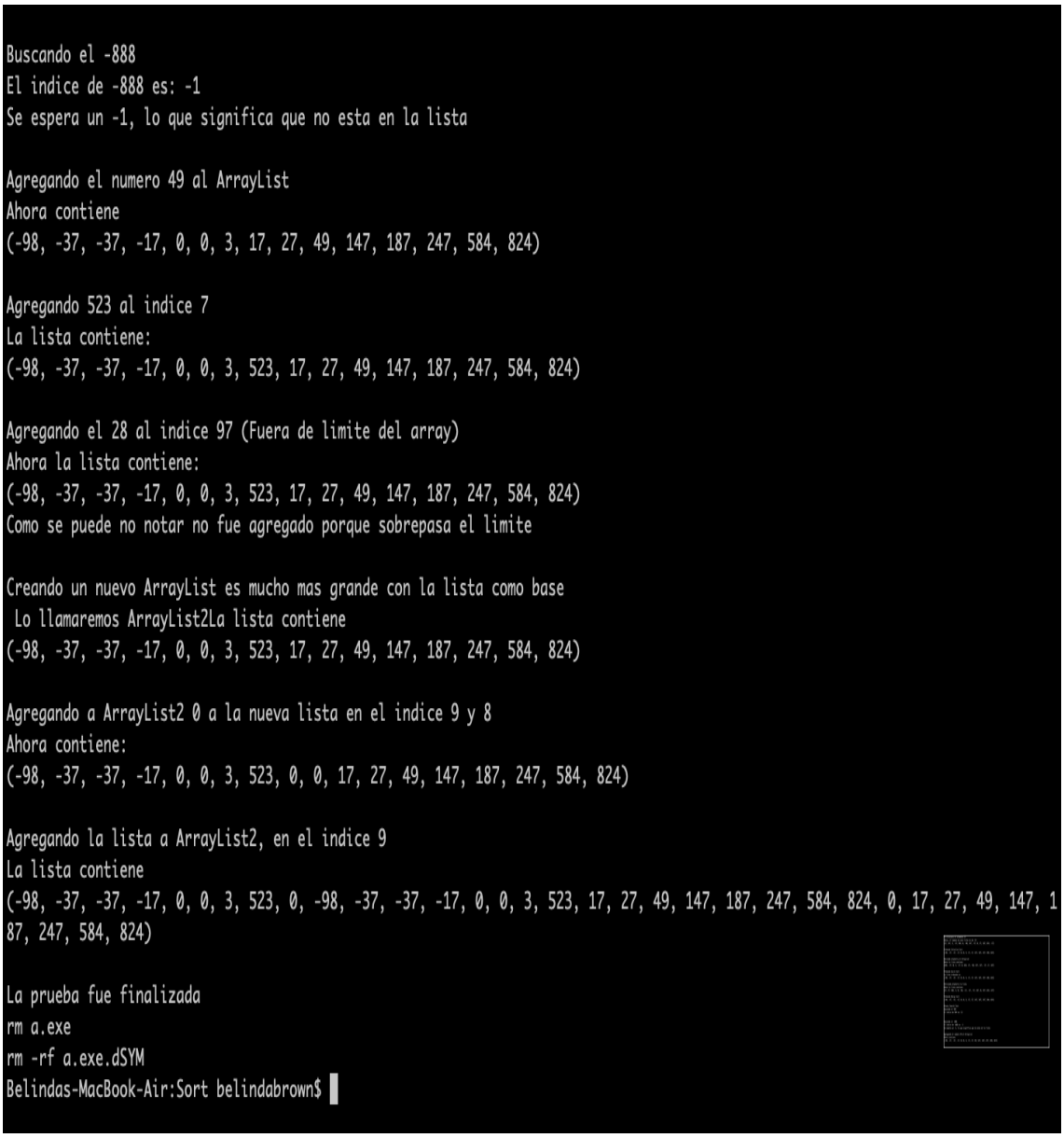

Figura 18: Resultados de sort

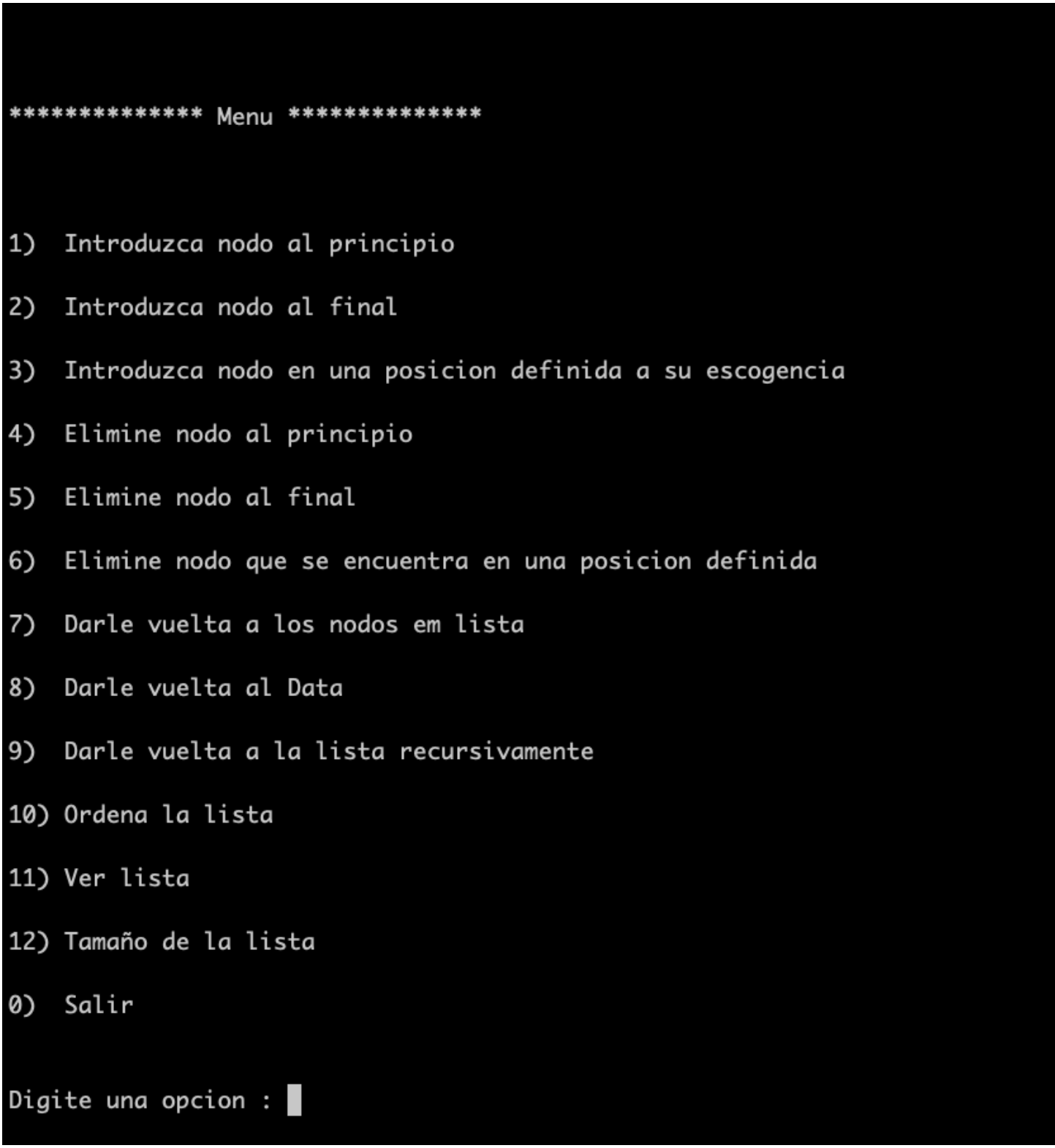

Figura 19: Resultados de List

```
12) Tamaño de la lista

0)  Salir

Digite una opcion : 1

Digite una opcion: 1

Proceso realizado exitosamente...


************** Menu **************


1)  Introduzca nodo al principio

2)  Introduzca nodo al final

3)  Introduzca nodo en una posicion definida a su escogencia

4)  Elimine nodo al principio

5)  Elimine nodo al final

6)  Elimine nodo que se encuentra en una posicion definida

7)  Darle vuelta a los nodos em lista

8)  Darle vuelta al Data
```

Figura 20: Resultados de List

```
5)  Elimine nodo al final

6)  Elimine nodo que se encuentra en una posicion definida

7)  Darle vuelta a los nodos em lista

8)  Darle vuelta al Data

9)  Darle vuelta a la lista recursivamente

10) Ordena la lista

11) Ver lista

12) Tamaño de la lista

0)  Salir

Digite una opcion : 2

Digite un valor: 21

Proceso realizado exitosamente...


********** Menu **********


1)  Introduzca nodo al principio
```

Figura 21: Resultados de List

```
9)  Darle vuelta a la lista recursivamente

10) Ordena la lista

11) Ver lista

12) Tamaño de la lista

0)  Salir


Digite una opcion : 11


     Lista

**** Nodo 1 ****
   0x7f8ae4400630
      1
   0x7f8ae4500000
**************

**** Nodo 2 ****
   0x7f8ae4500000
      21
      0
**************



************** Menu **************



1)  Introduzca nodo al principio
```

Figura 22: Resultados de List

```
4)  Elimine nodo al principio

5)  Elimine nodo al final

6)  Elimine nodo que se encuentra en una posicion definida

7)  Darle vuelta a los nodos em lista

8)  Darle vuelta al Data

9)  Darle vuelta a la lista recursivamente

10) Ordena la lista

11) Ver lista

12) Tamaño de la lista

0)  Salir

Digite una opcion : 12

Tamaño de la lista : 2


********** Menu **********


1)  Introduzca nodo al principio
```

Figura 23: Resultados de List

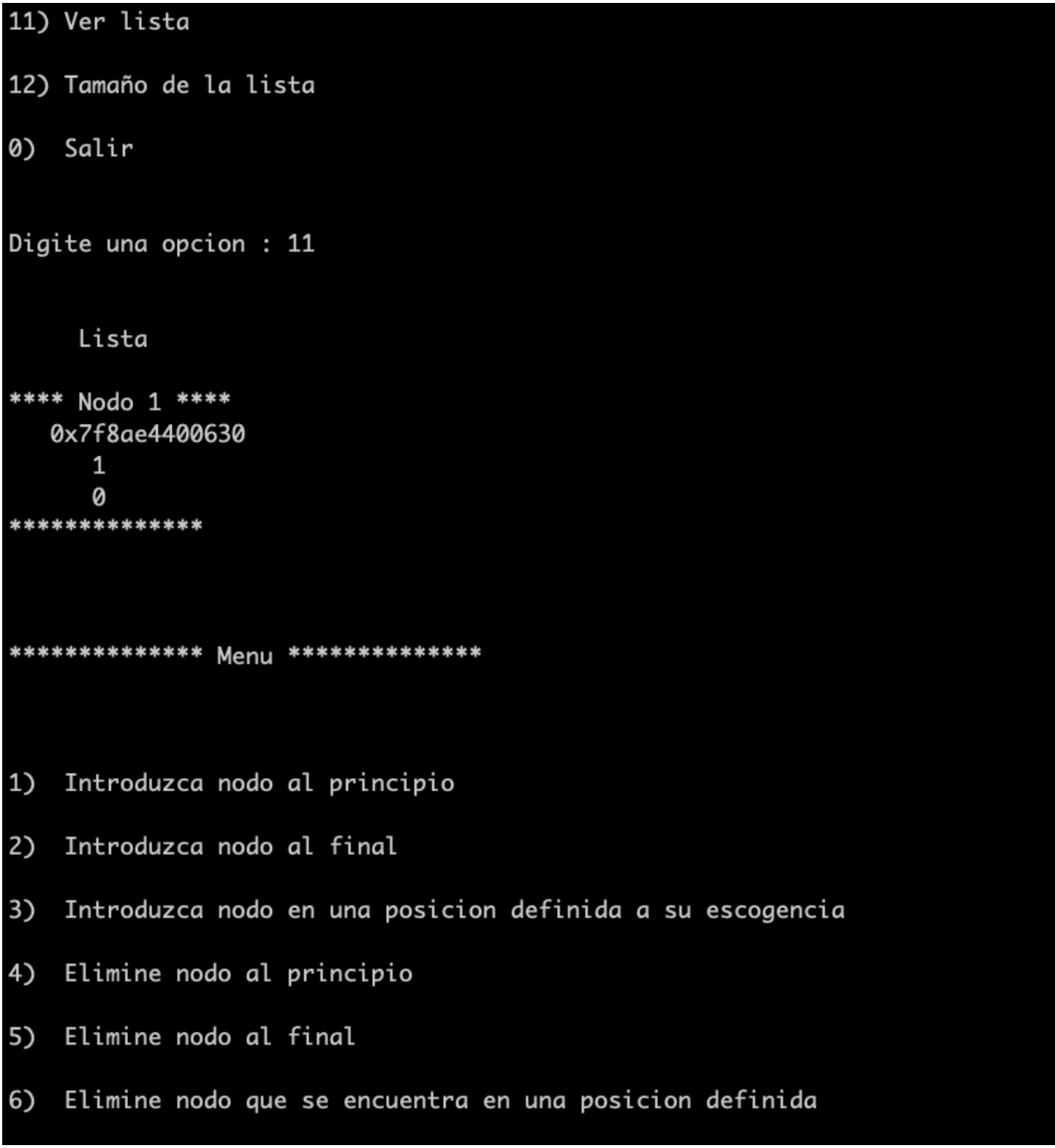

Figura 24: Resultados de List

```
10) Ordena la lista

11) Ver lista

12) Tamaño de la lista

0)  Salir

Digite una opcion : 5

Proceso realizado exitosamente...


************** Menu **************


1)  Introduzca nodo al principio

2)  Introduzca nodo al final

3)  Introduzca nodo en una posicion definida a su escogencia

4)  Elimine nodo al principio

5)  Elimine nodo al final

6)  Elimine nodo que se encuentra en una posicion definida

7)  Darle vuelta a los nodos em lista
```

Figura 25: Resultados de List

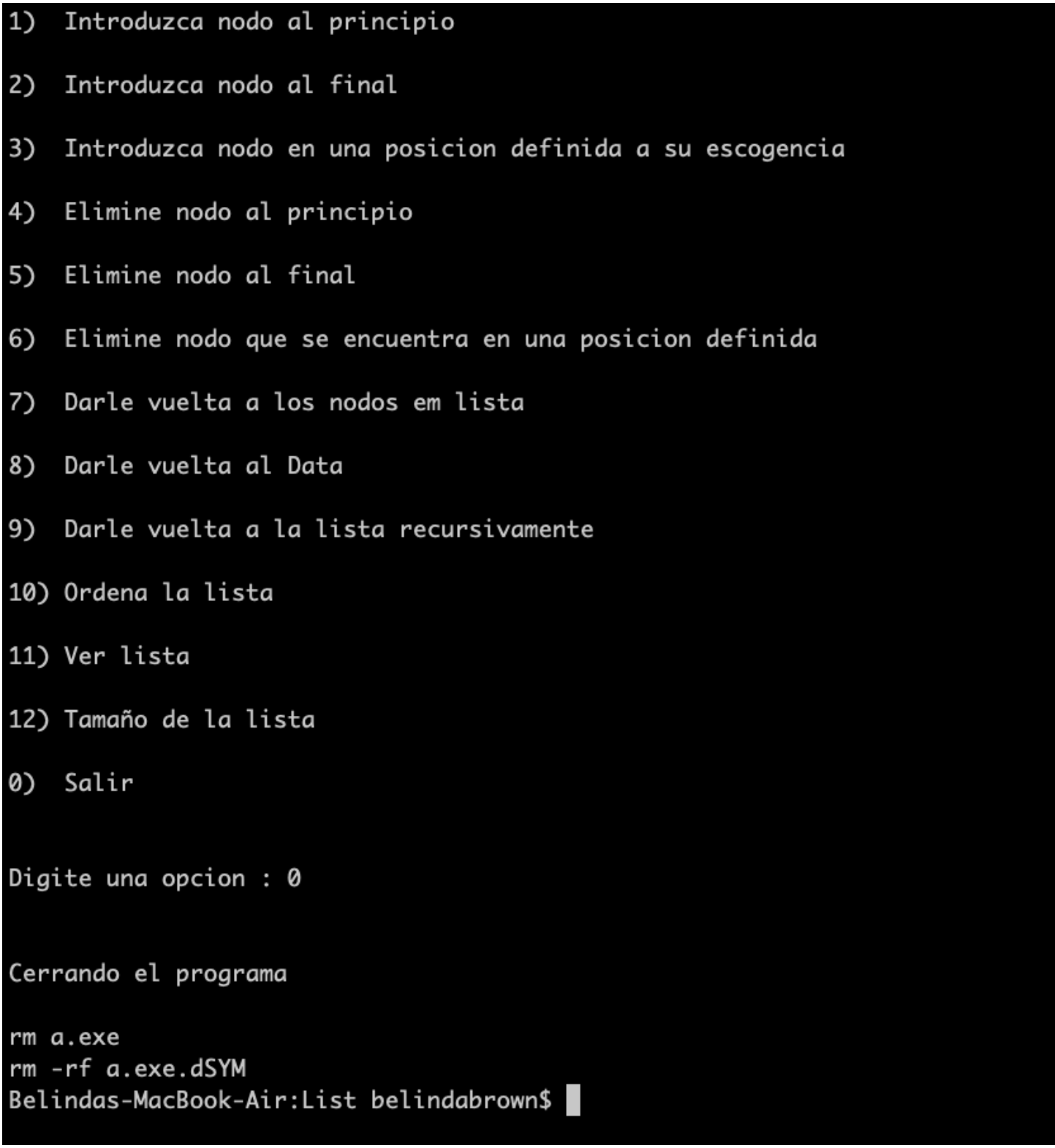

Figura 26: Resultados de List

Referencias

[1] Kroah-Hartman G Corbet. J, Rubini. A. *Linux Coding.* O'Reilly books, 1998.

[2] Computer Science Labs. *Tecnology- commands.* O'Reilly books, 2018.

[3] Mark Summerfield. *Programming in Python 3: A Complete Introduction to the Python Language.* Anaya Multimedia, 2009.

[4] A. M. Turing. *On computable numbers with an application to the Entscheidungs problem.* Proceedings of the london mathematical society, 1997.