

**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA**

**UNITEC**

**ASIGNATURA: ESTRUCTURAS DE DATOS I**

**SECCIÓN­: 489**

**PROYECTO I**

**TDAs FUNDAMENTALES**

**PRESENTADO POR:**

**ALUMNO: JUDÁ AARON PONCE VILLALTA**

**CTA: 11841248**

**CATEDRÁTICO:** ING**. REYNOD BOCANEGRA**

**CAMPUS TEGUCIGALPA 27 DE AGOSTO DE 2020**

**Introducción**

El siguiente manual mostrará cada uno de los metodos utilizados en la elaboración del proyecto. Dicho proyecto está hecho en c++ con IDE netbeans, que muestra la implementación de los tipos de datos abstractos con arreglos y nodos doblemente enlazados.

Se mostrará cada detalle usando capturas de pantalla y pseudocodigos de cada metodo para tener un mayor beneficio en cuanto al entendimiento de lo que se elaboró a lo largo del proyecto.

Menú principal

Captura de pantalla de un celular

Descripción generada automáticamente

Si el usuario de desea trabajar con listas se le mostrará el siguiente menú:

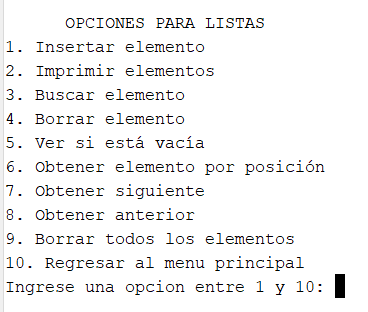
Menú de para listas

Imagen que contiene tabla, sostener, pantalla, computer

Descripción generada automáticamente

Ya sea que el usuario escoja trabajar con ArrayList o Linkedlist, se muestra el siguiente submenú:

Opciones para listas



Operaciones del Listas

1. Deberá solicitar al usuario la posición a insertarlo y el elemento a insertar y Luego deberá indicarle al usuario si la operación se realizó con éxito, y preguntar al usuario si desea introducir otro elemento, de ser así continuar introduciendo elementos nuevos, hasta que el usuario decida salir, en cuyo caso deberá regresar al Menú Operaciones de Listas.
   * 1. **Ejemplo:**

Imagen que contiene botella, tabla, pantalla, teléfono

Descripción generada automáticamente

Imagen que contiene tabla, teléfono, pájaro, ave

Descripción generada automáticamente

1. Desplegará de manera estética los elementos que estén en la lista.
   * 1. **Ejemplo:**

**Imagen que contiene naranja, cuarto

Descripción generada automáticamente**

1. Le solicitará al usuario que introduzca un número de cuenta, y deberá de buscar ese dato en los elementos de la lista, desplegando el primer Alumno encontrado en pantalla, junto con su posición. De no encontrar el dato en ningún elemento de la lista, entonces deberá desplegar un mensaje indicándolo.
   * 1. **Ejemplo 1:**

**Imagen que contiene objeto, reloj

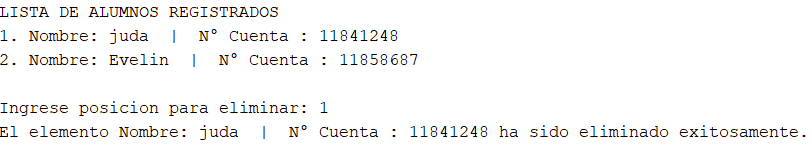
Descripción generada automáticamente**

* + 1. **Ejemplo 2:**

**Imagen que contiene persona, naranja, hombre, sostener

Descripción generada automáticamente**

1. Le solicitará al usuario una posición en la lista, y eliminará el elemento que esté en esa posición, deberá avisarle al usuario que la eliminación fue exitosa o no. Adicionalmente deberá desplegar el elemento borrado.
   * 1. **Ejemplo 1:**

****

* + 1. **Ejemplo 2:**

**Imagen que contiene tabla

Descripción generada automáticamente**

1. Mostrará un mensaje indicando que la lista está vacía o no está vacía. Si no está vacía mostrará cuántos elementos hay en ella.
   * 1. **Ejemplo 1:**

**Un conjunto de letras negras en un fondo blanco

Descripción generada automáticamente**

* + 1. **Ejemplo 2:**

****

1. Solicitará al usuario un entero, que representa una posición en la lista, desplegará el elemento que se encuentre en esa posición (si la posición es válida). Si no, mostrará un mensaje indicando que la posición no es válida.
   * 1. **Ejemplo 1:**

**Imagen que contiene pájaro

Descripción generada automáticamente**

* + 1. **Ejemplo 2:**

****

1. Pedirá un entero que representa una posición en la Lista y mostrará el elemento que está en la posición siguiente. Si no hay elemento en la posición siguiente mostrará un mensaje indicando que el espacio siguiente está vacío.
   * 1. **Ejemplo:**

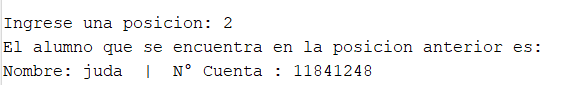
**Imagen que contiene reloj

Descripción generada automáticamente**

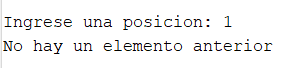
* + 1. **Ejemplo 2:**

****

1. Pedirá un entero que representa una posición en la Lista y mostrará el elemento que está en la posición anterior. Si no hay elemento en la posición anterior mostrará un mensaje indicando que el espacio anterior está vacío.
   * 1. **Ejemplo 1:**

****

* + 1. **Ejemplo 2:**

****

1. Borrará todos los elementos de la lista, liberando la memoria pertinente.
   * 1. **Ejemplo:**

**Imagen que contiene pájaro

Descripción generada automáticamente**

* Si el usuario de desea trabajar con pilas se le mostrará el siguiente menú:

Menú para pilas

Imagen que contiene pantalla, computer, tabla, teléfono

Descripción generada automáticamente

Ya sea que el usuario escoja trabajar con ArrayStack o LinkedStack, se muestra el siguiente submenú:

Opciones para pilas

Imagen que contiene ave, pájaro

Descripción generada automáticamente

Operaciones para pilas

1. Solicitará un Símbolo al usuario, para que se “empuje” en la pila.
   * 1. **Ejemplo:**

****

**Imagen que contiene pájaro

Descripción generada automáticamente**

****

1. Sacará un elemento de la Pila y mostrará el resultado (Si es exitoso, mostrará el elemento, si no lo es, indicará que no hay elementos en la pila).
   * 1. **Ejemplo 1:**

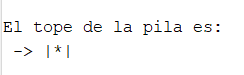
**Imagen que contiene tabla

Descripción generada automáticamente**

* + 1. **Ejemplo 2:**

****

1. Desplegará el elemento en el tope de la pila.
   * 1. **Ejemplo 1:**

****

* + 1. **Ejemplo 2:**

****

1. Desplegará un mensaje: “Está Vacía”, si está vacía, o, “No Está Vacía”, en el caso de que no esté vacía.
   * 1. **Ejemplo 1:**

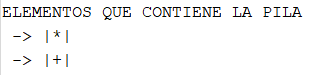
****

* + 1. **Ejemplo 2:**

**Imagen que contiene tabla

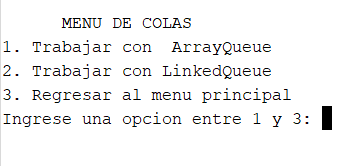
Descripción generada automáticamente**

1. Imprimirá los elementos de la pilaen el orden en el que deberían salir de ella.
   * 1. **Ejemplo:**

****

* Si el usuario de desea trabajar con colas se le mostrará el siguiente menú:

Menú para colas



Ya sea que el usuario escoja trabajar con ArrayQueue o LinkedQueue, se muestra el siguiente submenú:

Opciones para colas

Imagen que contiene ave, pájaro

Descripción generada automáticamente

Operaciones para colas

1. Solicitará un alumno al usuario, para meterlo en la cola.
   * 1. **Ejemplo:**

**Imagen que contiene naranja, sostener, foto, mano

Descripción generada automáticamente**

**Imagen que contiene naranja, cuarto, sostener, pantalla

Descripción generada automáticamente**

1. Sacará un elemento de la Cola y mostrará el resultado (Si es exitoso, mostrará el elemento, si no lo es, indicará que no hay elementos en la cola).
   * 1. **Ejemplo 1:**

****

* + 1. **Ejemplo 2:**

****

1. Desplegará el elemento de la frente de la cola.
   * 1. **Ejemplo:**

****

1. Desplegará un mensaje: “Está Vacía”, si está vacía, o, “No Está Vacía”, en el caso de que no esté vacía.
   * 1. **Ejemplo 1:**

****

* + 1. **Ejemplo 2:**

****

1. Imprimirá los elementos de la colaen el orden en el que deberían salir de ella.
   * 1. **Ejemplo:**

**Imagen que contiene naranja, computadora, cuarto

Descripción generada automáticamente**