Introducción a la Algorítmica y Programación

## Trabajo Práctico Integrador

## Integrantes

* Augusto Lovagnini
* Giovanni Buchieri
* Genaro Pennone

**Actividad**

**Ej. 15)** Para administrar una lista de personas se requiere hacer diversas acciones que permitan ese trabajo. Para almacenar los nombres se utilizará un registro que contiene dos campos, uno con un arreglo donde se guardan los nombres y otro con la cantidad actual de nombres almacenados en el arreglo (Utilice un registro con un campo arreglo y otro campo con la cantidad de datos que se han cargado al arreglo). La cantidad máxima que se podrá almacenar será de 1000 nombres. Para poder administrar la lista se deben desarrollar las siguientes acciones y/o funciones:

a) Una función llamada Vacia que reciba como parámetro el registro (con el arreglo de nombres y la cantidad de nombres cargados), y devuelva verdadero si la lista está vacía y sino debe devolver falso.

b) Una función llamada Llena que reciba como parámetro el registro (con el arreglo de nombres y la cantidad de nombres cargados), y devuelva Verdadero si el arreglo está completamente lleno (es decir si ya tiene 1000 nombres cargados) y sino debe devolver Falso.

c) Una acción que permita insertar un nombre en el arreglo. Para ello se pasará como parámetros el registro (con el arreglo de nombres y la cantidad de nombres cargados), y el nuevo nombre a insertar. El nuevo nombre se inserta siempre al final de la lista. Después de ejecutada la acción, la cantidad debe quedar incrementada en una unidad.

d) Una acción que permita suprimir al primer nombre de la lista, no importa cuál es. Simplemente suprime el primero cada vez que la acción es ejecutada. El parámetro que debe recibir es el registro (con el arreglo de nombres y la cantidad de nombres cargados). Después de ejecutada la acción, la cantidad debe quedar decrementada en una unidad.

e) Una acción que permita mostrar todos los nombres de la lista. Los parámetros que debe recibir son: el arreglo y la cantidad de nombres que tiene cargados.

## Análisis

**Datos**: arreglo, nuevoNombre, selección, volver //corregir si es que NuevoNombre, selección y volver van como datos.

**Resultados**: arreglo, msg

**Relaciones y Subproblemas:**

Salir = falso

Usaremos la estructura iterativa “Mientras” con la condición de continuación “salir = falso”, esto quiere decir que hasta que la variable lógica “Salir” no sea verdadera se continuara ciclando.

Dentro de esta estructura:

Menu() //Esta acción lo que hace es mostrar un menú con todas las opciones posibles en el programa. (cargar los nombres, eliminar primer nombre, mostrar lista de nombres, etc)

Se pedirá la entrada de la variable “Selección” la cual tiene que tener un valor del 1 al 5, ya que son 5 opciones las cuales el usuario puede elegir dentro del programa, para garantizar que es un numero entre el 1 y el 5, utilizaremos de nuevo la estructura iterativa “Mientras”, esta hará que se siga pidiendo la entrada hasta que ingrese algún numero valido.

Usaremos la estructura “según” con las siguientes condiciones:

Cuando selección sea 1 (selección = 1): se ejecutará “CargarArreglo(arreglo)”

//Esta acción le permite al usuario ingresar la cantidad de nombres que él quiera, con un máximo de 1000

Cuando selección sea 1 (selección = 2): se ejecutará “CargarNuevoNombre(arreglo)”

//Esta acción permite cargar un nuevo nombre en el arreglo, primero se fija que el arreglo no esté lleno, mediante la función “Llena” la cual devuelve verdadero si la función ya contiene 1000 nombres cargados o falso.

Cuando selección sea 1 (selección = 3): se ejecutará “SuprimirNombre(arreglo)”

//Esta acción permite suprimir el primer nombre del arreglo. Si el arreglo este vacío no se podrá eliminar nada y mostrará un mensaje aclarando que no existe ningún nombre cargado, para saber si esta vacío, esta acción utiliza la función “Vacía” la cual devuelve verdadero si la función no tiene ningún nombre cargado.

Cuando selección sea 1 (selección = 4): se ejecutará “Mostrar(arreglo)”

//Esta acción muestra en pantalla todos los nombres previamente ingresados, en caso que este vacío (utilizando la acción “Vacía” previamente mencionada), se mostrara un mensaje diciendo que no se ha encontrado ningún nombre.

Cuando selección sea 1 (selección = 5): a la variable salir se le asignara verdadero (salir ← verdadero)

Luego de ejecutar el condicional, se le preguntara al usuario si quiere volver al menú del inicio o salir del programa. Cuando seleccione 1, se volverá a ejecutar el menú. Cuando seleccione 2 a la variable “salir” se le asignara verdadero (salir ← verdadero).

## Diseño

**Algoritmo**

**Léxico**

Max = 1000

TNombres= arreglo[1.. Max] de Cadena

TData = < nombres ∈ TNombres, cant ∈ (0..Max) >

arreglo ∈ TData

msg ∈ Cadena  
   
 salir ∈ Lógico

selección, volver ∈ Z

*Funcion* Vacia(dato nom ∈ TData) → Logico

*Inicio*

Si(nom.cant = 0 ) entonces

← verdadero

Sino

← falso

fsi

*ffuncion*

*Funcion* Llena(dato nom ∈ TData)→ Logico

*Inicio*

Si (nom.cant = 1000 ) entonces

← verdadero

Sino

← falso

fsi

*ffuncion*

*Acción* CargarArreglo (resultado nom ∈ Tdata)

*Inicio*

//Ingresa la cantidad de nombres

Entrada: nom.cant

Mientras (nom.cant < 0 o nom.cant > 1000) hacer

//La cantidad de nombres tiene que estar entre 0 y 1000, por favor ingrese nuevamente la cantidad:

Entrada: nom.cant

Fmientras

Para(i ← 1, i <= nom.cant, i ← i + 1) hacer

//Ingresa los nombres

Entrada: nom.nombres[i]

fpara

*facción*

*Acción* cargarNuevoNombre (dato-resultado nom ∈ TData)

*Léxico Local*

nuevoNombre ∈ Caracter

msg ∈ Cadena

*Inicio*

Si(Llena(nom)) entonces

msg ← “No es posible insertar el nuevo nombre ya que el arreglo está lleno”

salida: msg

sino

//Ingresa el Nuevo Nombre que quieres agregar

Entrada: nuevoNombre

nom.nombres[nom.cant+1] ← nuevoNombre

nom.cant ← nom.cant + 1

msg ← “Nuevo nombre cargado con éxito”

salida: msg

fsi

*facción*

*Acción* SuprimirNombre(dato-resultado nom ∈ TData)

Lexico Local

msg ∈ cadena

*Inicio*

Si (Vacia(arreglo)) entonces

Msg ← “No se pudo suprimir ningún nombre, porque el arreglo está vacío”

Salida: msg

Sino

nom.nombres[1] ← nom.nombres[nom.cant]

nom.cant ← nom.cant – 1

fsi

*facción*

*Accion* Mostrar (dato nom ∈ TData)

*Léxico Local*

msg ← Cadena

*Inicio*

Si(Vacio(nom)) entonces

msg ← “el arreglo esta vacío”

sino

Para(i ← 1, i <= nom.cant, i ← i + 1) hacer

Salida: nom.nombres[i]

fpara

*faccion* ***INICIO***  
salir ← falso

Mientras (salir = falso) hacer

msg ← "Que quieres hacer?"

Salida: msg

msg ← "[1]Cargar Nuevo Arreglo."

Salida: msg

msg ← "[2]Cargar Nuevo Nombre."

Salida: msg

msg ← "[3] Suprimir el primer Nombre."

Salida: msg

msg ← "[4] Mostrar lo que contiene el arreglo”

Salida: msg

msg ← “[5] Salir"

Salida: msg

msg ← “\*INGRESA AQUI EL NUMERO: "

Salida: msg

Entrada: selección

Mientras (selección < 1 o selección > 5) hacer

msg ← “No existe ninguna operación con ese número, por favor vuelve a ingresar uno nuevo”

Menu()

Entrada: seleccion

Fmientras

Según

(selección ==1): CargarArreglo(arreglo)

(selección = 2): CargarNuevoNombre(arreglo)

(selección = 3): SuprimirNombre(arreglo)

(selección = 4): Mostrar(arreglo)

(selección = 5): salir ← verdadero

Fsegun

Si (selección <> 5) entonces

msg ← “Si quiere volver al MENU Presione [Cualquier Numero] – Si quiere SALIR presione [1]”

salida: msg

Entrada: volver

Si(volver = 1) entonces

salir ← verdadero

fsi

fsi

fmientras

**FIN**