

Universidad Nacional de Río Cuarto

FACULTAD DE CIENCIAS EXACTAS, FÍSICO QUÍMICAS Y NATURALES

ANALISTA EN COMPUTACIÓN

INTRODUCCION A LA ALGORITMICA Y PROGRAMACIÓN

# INTEGRADOR ALGORITMICA PROYECTO MALVINAS

Autores:

Debernardi, Álvaro; Astrada, Camila

## Algoritmo Proyecto Malvinas

### Léxico

NMax = 250

Tfecha = <dia ∈ (0,31), mes ∈ (0,12), año ∈ ℤ >

Tpersona = <

DNI ∈ Cadena,

nombre ∈ Cadena,

apellido ∈ Cadena,

nacimiento ∈ Tfecha,

edad ∈ ℤ,

ciudad ∈ Cadena,

ciudadAntes ∈ Cadena,

DP ∈ Cadena,

CE ∈ Cadena,

tel ∈ Cadena,

VF ∈ Logico,

fallecimiento ∈ Tfecha,

provincia ∈ Cadena,

fuerza ∈ Cadena,

destino ∈ Cadena,

funcionMalvinas ∈ Cadena,

grado ∈ Cadena,

secuelas ∈ Cadena>

Tcomunidad = <personas = arreglo[1...NMax] de Tpersonas, cant ∈ ℤ>

Tfallecido = < info ∈ Tpersona, next ∈ P de Tpersona >

Tlocalidad = < persona ∈ Tpersona, back ∈ P de Tpersona, next ∈ P de Tpersona >

**Funcion** Vacia(dato comunidad ∈ Tcomunidad) → Logico

#### Inicio

**si** (comunidad.cant = 0) **entonces**

← verdadero

**sino**

← falso

**Fsi**

**Ffuncion**

**Funcion** Llana(dato comunidad ∈ Tcomunidad) → Logico

#### Inicio

**si** (comunidad.cant = 150) **entonces**

← verdadero

**sino**

← falso

**Fsi**

**Ffuncion**

**Accion** Insertar(dato-resultado comunidad ∈ Tcomunidad)

#### Léxico Local

nueMie ∈ Tpersona

pos, aux ∈ ℤ

#### Inicio

**si** (Llana(comunidad)) **entonces**

salida: "No es posible agregar miembros. Lista llena."

←

**sino**

Entrada: nueMie

pos ← 1

**mientras** (nueMie.apellido > comunidad.personas[pos].apellido y pos ≤ comunidad.cant) **hacer**

pos ← pos + 1

**Fmientras**

**si** (nueMie.apellido = comunidad.personas[pos].apellido) **Entonces**

**Segun:**

nueMie.nombre < comunidad.personas[pos].nombre : NADA  
nueMie.nombre > comunidad.personas[pos].nombre : pos  $\leftarrow$  pos + 1  
nueMie.nombre = comunidad.personas[pos].nombre :

**Segun:**

nueMie.DNI < comunidad.personas[pos].DNI : NADA  
nueMie.DNI > comunidad.personas[pos].DNI : pos  $\leftarrow$  pos + 1  
nueMie.DNI = comunidad.personas[pos].DNI :  
    Salida: "DNI repetido, operacion fallida"  
     $\leftarrow$

**Para** (aux  $\leftarrow$  comunidad.cant, pos  $\leq$  aux, aux  $\leftarrow$  aux - 1) **hacer**  
    comunidad.personas[aux + 1]  $\leftarrow$  comunidad.personas[aux]

**Fpara**

comunidad.personas[pos]  $\leftarrow$  nueMie  
comunidad.cant  $\leftarrow$  comunidad.cant + 1

**Fsi**

**Faccion**

**Accion** Suprimir(dato-resultado comunidad  $\in$  Tcomunidad, dato dni  $\in \mathbb{Z}$ )

**Léxico Local**

pos  $\in \mathbb{Z}$   
encontrado  $\in$  Logico

**Inicio**

pos  $\leftarrow$  0  
encontrado  $\leftarrow$  falso  
**mientras** (pos < comunidad.cant y encontrado = falso) **hacer**  
    pos  $\leftarrow$  pos + 1  
    **si** (comunidad.personas[pos].DNI = dni) **Entonces**  
        encontrado  $\leftarrow$  verdadero

**Fsi**

**Fmientras**

**Para** (pos, pos < comunidad.cant, pos  $\leftarrow$  pos + 1)  
    comunidad.personas[pos]  $\leftarrow$  comunidad.personas[pos + 1]

**Fpara**

comunidad.cant  $\leftarrow$  comunidad.cant - 1

**Faccion**

**Accion** Mostrar(dato comunidad  $\in$  Tcomunidad)

**Léxico local**

i  $\in \mathbb{Z}$

**Inicio**

**para** (i  $\leftarrow$  1, i  $\leq$  comunidad.cant, i  $\leftarrow$  i + 1) **hacer**  
    salida: comunidad.personas[i]

**Fpara**

**Faccion**

**Accion** Modificar(dato-resultado comunidad  $\in$  Tcomunidad, dato dni  $\in \mathbb{Z}$ )

**Léxico Local**

pos  $\in \mathbb{Z}^+$   
msg  $\in$  Cadena  
opcion  $\in \mathbb{Z}$

**Inicio**

pos  $\leftarrow$  1  
msg  $\leftarrow$  "Que informacion desea cambiar:  
    1.Nombre  
    2.Apellido  
    3.Fecha de nacimiento  
    4.Edad  
    5.Ciudad de residencia  
    6.Direccion postal

7. Correo electronico
8. Telefono
9. Vive
10. Fecha de fallecimiento

**mientras** (pos <= comunidad.cant y dni <> comunidad.personas[pos].DNI) **hacer**  
 pos ← pos + 1

**Fmientras**

**Si** (pos <= comunidad.cant) **Entonces**

Salida: msg

Entrada: opcion

**Segun**

- opcion = 1 : Entrada: comunidad.personas[pos].nombre
- opcion = 2 : Entrada: comunidad.personas[pos].apellido
- opcion = 3 : Entrada: comunidad.personas[pos].nacimient
- opcion = 4 : Entrada: comunidad.personas[pos].edad
- opcion = 5 : Entrada: comunidad.personas[pos].ciudad
- opcion = 6 : Entrada: comunidad.personas[pos].DP
- opcion = 7 : Entrada: comunidad.personas[pos].CE
- opcion = 8 : Entrada: comunidad.personas[pos].tel
- opcion = 9 : Entrada: comunidad.personas[pos].VF
- opcion = 10 : Entrada: comunidad.personas[pos].fallecimiento

**Fsegun**

Salida: comunidad.personas[pos]

**Sino**

Salida: "No hay una persona con ese DNI"

←

**Fsi**

**Faccion**

**Accion** MostrarFallecidos(dato comunidad ∈ Tcomunidad)

**Léxico Local**

fallecidos ∈ P de Tfallecido

aux, aux2 ∈ P de Tfallecido

i ∈ ℤ+

**Inicio**

obtener(fallecidos)

(\*fallecidos).next ← nil

aux2 ← fallecidos

i ← 1

**mientras** (i <= comunidad.cant) **hacer**

**Si** (comunidad.personas[pos].VF = falso) **Entonces**

obtener(aux)

(\*aux).info ← comunidad.personas[i]

(\*aux2).next ← aux

(\*aux).next ← nil

aux2 ← nil

**Fsi**

i ← i + 1

**Fmientras**

aux ← (\*fallecidos).next

**mientras** (aux <> nil)

Salida: (\*aux).info

aux ← (\*aux).next

**Fmientras**

**Faccion**

**Accion** MostrarPorLocalidad(dato localidad ∈ Cadena, comunidad ∈ Tcomunidad)

**Léxico Local**

cabeza ∈ P de Tlocalidad

cola ∈ P de Tlocalidad

$i \in \mathbb{Z}$   
 nuevo, aux  $\in P$  de Tlocalidad  
 posicionado  $\in \text{Logico}$

### **Inicio**

obtener(cabeza)  
 (\*cabeza).back  $\leftarrow$  nil  
 obtener(cola)  
 (\*cola).next  $\leftarrow$  nil  
 (\*cabeza).next  $\leftarrow$  cola  
 (\*cola).back  $\leftarrow$  cabeza  
 posicionado  $\leftarrow$  falso  
 aux  $\leftarrow$  (\*cabeza).next

**Para** ( $i \leftarrow 1, i \leq \text{comunidad.cant}, i \leftarrow i + 1$ )

**Si** ( $\text{comunidad.personas}[i].\text{ciudad} = \text{localidad}$ ) **Entonces**

obtener(nuevo)  
 (\*nuevo).persona  $\leftarrow$  comunidad.personas[i]  
**mientras** ((\*aux).next  $\neq$  nil, posicionado = falso)

**Si** ((\*nuevo).persona.DNI  $<$  aux.persona.DNI) **Entonces**

(\*nuevo).next  $\leftarrow$  aux  
 (\*nuevo).back  $\leftarrow$  (\*aux).back  
 ((\*aux).back).next  $\leftarrow$  aux  
 (\*aux).back  $\leftarrow$  nuevo  
 posicionado  $\leftarrow$  verdadero

**Fsi**

aux  $\leftarrow$  (\*aux).next

**Fmientras**

**Si** (no posicionado) **Entonces**

(\*nuevo).next  $\leftarrow$  aux  
 (\*nuevo).back  $\leftarrow$  (\*aux).back  
 (\*aux).back  $\leftarrow$  nuevo  
 ((\*aux).back).next  $\leftarrow$  nuevo

**Fsi**

**Fsi**

**Fpara**

aux  $\leftarrow$  (\*cabeza).next

**mientras** ((\*aux).next  $\neq$  nil) **hacer**

Salida: (\*aux).info

**Fmientras**

**Faccion**

msg  $\in$  Cadena  
 opcion  $\in \mathbb{Z}^+$   
 salir  $\in \text{Logico}$   
 comunidad  $\in T\text{comunidad}$   
 dni  $\in \mathbb{Z}$   
 localidad  $\in$  Cadena

### **Inicio**

comunidad.cant  $\leftarrow$  0

msg  $\leftarrow$  "Ingrese la opcion que desea:

- 1.Insertar integrante
- 2.Eliminar integrante
- 3.Modificar datos de un integrante
- 4.Listar todos los integrantes
- 5.Listar integrantes fallecidos
- 6.Listar integrantes de una localidad
- 7.Salir

salir  $\leftarrow$  falso

**repetir**

Salida: msg

Entrada: opcion

**segun**

(opcion = 1) : Insertar(comunidad)

(opcion = 2) :

Salida: "Ingrese DNI"

Entrada: dni

Suprimir(comunidad, dni)

(opcion = 3) :

Salida: "Ingrese DNI"

Entrada: dni

Modificar(comunidad, dni)

(opcion = 4) : Mostrar(comunidad)

(opcion = 5) : MostrarFallecidos(comunidad)

(opcion = 6) :

Salida: "Ingrese localidad"

Entrada: localidad

MostrarPorLocalidad(localidad, comunidad)

(opcion = 7) : salir  $\leftarrow$  verdadero

**Fsegun**

**hasta que**(salir = verdadero)

**Fin**