Universidad Nacional de Río Cuarto

Facultad de Cs. Exactas, Fco-Qcas y Naturales - Dpto de Computación

Asignatura: **INTRODUCCIÓN A LA ALGORÍTMICA Y PROGRAMACIÓN**

Código: 3300

Año: 2022

**Proyecto integrador de mitad de año**

***Alumnos: Celina Gillo Mayer, Valentìn Cardozo, Joaquín Pablo Tissera***

Sea presente el siguiente proyecto (dictado por los profesores de la materia):

| Para administrar la nómina de empleados de una sucursal de un banco se requiere hacer un algoritmo (y luego un programa) con un menú de opciones que permite realizar estas tareas:   1. Alta de empleado (se carga apellido). ⃫ 2. Baja de empleado (se elimina un empleado, se lo busca por apellido). ⃫ 3. Modificar datos de empleado (se lo busca por apellido y se ofrece modificarlo). ⃫ 4. Listar (muestra todos los apellidos almacenados). ⃫ 5. Buscar a un empleado (si el apellido existe en la lista lo muestra, sino informa apellido inexistente). ⃫ 6. Salir. ⃫   La cantidad máxima que se podrá almacenar será de 1000 empleados (apellidos).  Para poder administrar la lista se deben desarrollar las siguientes acciones y/o funciones:   1. Una función llamada Vacía que reciba como parámetro el registro (con el arreglo de apellidos y la cantidad de apellidos cargados), y devuelva verdadero si la lista está vacía y si no debe devolver Falso. 2. Una función llamada Llena que reciba como parámetro el registro (con el arreglo de apellidos y la cantidad de apellidos cargados), y devuelva Verdadero si el arreglo está completamente lleno (es decir si ya tiene 1000 apellidos cargados) y sino debe devolver Falso. 3. Una acción que permita insertar un apellido en el arreglo. Para ello se pasarán como parámetros el registro (con el arreglo de apellidos y la cantidad de apellidos cargados), y el nuevo apellido a insertar. El nuevo apellido se inserta siempre al final. Después de ejecutada la acción, la cantidad debe quedar incrementada en una unidad. 4. Una acción que permita suprimir un apellido de la lista. Debe ubicarlo por el apellido y eliminarlo (desplazando los otros registros). Después de ejecutada la acción, la cantidad debe quedar decrementada en una unidad. 5. Una acción que permita mostrar todos los apellidos. Los parámetros que debe recibir son: el arreglo y la cantidad de apellidos que tiene cargados. 6. Una función que permita encontrar un empleado por apellido y que devuelva la ubicación en el arreglo. ⃫ 7. Una función que dados dos apellidos, devuelve Verdadero si están repetidos. ⃫   **Nota: no debe haber apellidos repetidos.** |
| --- |

Bajo el compromiso de realizar las consignas:

| **Consignas a entregar**  La resolución del Trabajo Práctico Integrador debe incluir:  **1)** Un documento en formato PDF con:  *a)* Análisis de cada módulo (incisos a hasta g). Identificar los datos de prueba y relaciones.   | Aclaración: no hace falta el análisis del problema general. | | --- |   *b)* Diseño (algoritmo en notación algorítmica)  *c)* Pruebas (completar el resultado de cada uno de los 6 casos de prueba)  **2)** Un archivo de texto con el Programa fuente en C. |
| --- | --- |

Se incluye a continuación la resolución del Trabajo Práctico, dando las respuestas a las consignas:

1. **a) Análisis de cada módulo:**

**Inciso A:**

Datos:

Resultados:

Relación:

**Inciso B:**

Datos:

Resultados:

Relación:

**Inciso C:**

Datos:

Resultados:

Relación:

**Inciso D:**

Datos:

Resultados:

Relación:

**Inciso E:**

Datos:

Resultados:

Relación:

**Inciso F:**

Datos: apellido (Cadena), arr (TData) *// recibe un apellido y el registro del arreglo (con campos; el arreglo y la cantidad cargada)*

Resultados: i (Z) *// devuelve un entero, en particular, el iterador*

Relación: Se recorre todo el arreglo; va comparando “apellido” con los componentes del arreglo. Ni bien coincidan con algún elemento del arreglo, devuelve el índice en el cuál coincidió.

**Inciso G:**

Datos: apellido1 (Cadena), apellido2 (Cadena) *// recibe dos apellidos*

Resultados: (Lógico) *// retorna un valor lógico*

Relación: Se compara “apellido1” con “apellido2”, si son iguales devuelve Verdadero. Si no son iguales, devuelve Falso.

**b) Diseño del proyecto:**

**Algoritmo** NóminaEmpleadosBanco

**Léxico**

AMax = 1000

TArreglo = arreglo [1..AMax] de Cadena

TData = < arreglo є TArreglo, cant є (0..AMax) >

// variables globales

// implementación de módulos

Función UbicarApellido (dato apellido є Cadena, arr є TData) → Z

Léxico local

i є Z

Inicio

Para ( i ← 1 , i ≤ arr.cant , i ← i + 1 ) hacer

Si ( apellido = arr.a[i] ) entonces

← i

Fsi

Fpara

Ffunción

Función ApellidosRepetidos (dato apellido1, apellido2 є Cadena) → Lógico

Inicio

← ( apellido1 = apellido2 )

Ffunción

**Inicio**

**Fin**

**c) Pruebas:**

**Inciso A:**

**Inciso B:**

**Inciso C:**

**Inciso D:**

**Inciso E:**

**Inciso F:**

**Inciso G:**

**Algoritmo principal:**