Universidad Nacional de Río Cuarto

Facultad de ciencias exactas, físico-químicas y naturales.



Introducción a la Algorítmica y Programación

(Cod. 3300)

**Trabajo integrador final**

**PROFESOR RESPONSABLE**

FERREIRA SZPINIAK, Ariel

**PROFESORES COLABORADORES**

ROJO, Guillermo

**ELABORADO POR**

BORTOL, Juan

DNI:31932122

Comisión 3

VENIER, Andrés

DNI: 34884476

. Comisión 1

**Río Cuarto - Córdoba**

**2022**

**Anotaciones preliminares del proyecto**

Abrir el archivo:

* abrir con “a” pidiendo el nombre (cadena), si no existe se crea uno

Alta de un registro:

* considerar límite de días a 366
* considerar fechas repetidas (teniendo en cuenta el borrado lógico, sobreescribir si es verdadero)
* necesitamos una acción de búsqueda sobre el archivo

Suprimir

* exista el registro a suprimir
* que el registro a suprimir no tenga borrado lógico
* acordarse de restar 1 al realizar acceso directo

Modificar

* buscar por fecha y modificar los otros campos(podemos reciclar la acción de alta reescribiendo todo el registro o ver, si nos sobra tiempo, de modificar campos específicos)

Mostrar

* mostrar fechas de registros con borrado lógico = falso
* considerar el orden de los registros (consultar)

Buscar y mostrar los parámetros

* ¿Importa la posición del archivo o nos manejamos con la posición del arreglo?
* Ejecución de Función dentro de Acción (considerar parámetros necesarios para la función, como la fecha a buscar y el índice de recorrido del arreglo y lo que sea necesario para implementar la función, entre los parámetros de la acción)

Temperatura máxima

* Pasar los registros a una lista, buscar la temperatura máxima y usar ese valor para hacer la lista con los días que hayan registrado la misma temperatura
* Mostrar por salida la fecha de los registros iguales hasta que cambie la temperatura y no mostramos más.

Viento

* insertion invertido (cambiar la condición de comparación de mayor a menor)
* acordarse de mostrar la velocidad del viento junto con la fecha de los 10 primeros

Precipitaciones

* Consultar: ¿diez primeros días de precipitaciones diferentes?
* Bubble

Copia de seguridad

* Para el backup añadir fecha y hora del sistema al nombre del archivo de respaldo

**Consideraciones Especiales:**

Se debe considerar que el programa fue desarrollado sobre Linux y pensado para ese entorno.

**Algoritmo EstacionMeteorologica**

**Léxico**

nombreArchivo, nombreFecha ∈ Cadena

f, backup ∈ Archivo de Tregdiario

Tregdiario=<ddmmyyyy ∈ Z, tmax ∈ Z, tmin ∈ Z, HUM ∈ (0…100), PNM ∈ (900…3500), DV ∈

(0…360), FF ∈ Z, PP ∈ (0…1000), borrado ∈ Lógico>

registro ∈ Tregdiario

Tarreglo = arreglo de (1…366) de Tregdiario

arregloActivo ∈ Tarreglo

cantidad, opc, posicion, fechaBusqueda ∈ Z

TNodoRegistro = < info ∈ Tregdiario, next ∈ puntero a TNodoRegistro >

lista ∈ puntero a TNodoRegistro

**Acción** AltaRegistro (Dato: nombreArch ∈ Cadena, Dato-Resultado:g ∈ Archivo de Tregdiario)

**Léxico Local**

reg, aux ∈ Tregdiario *//variables auxiliares para lectura y escritura de registros*

msj ∈ Cadena

respuesta ∈ Caracter *//Condiciona la carga cíclica de registros*

**Inicio**

**Abrir**(“nombreArch.dat”, g, a) *// Para añadir o crear el archivo en su defecto*

**IrPos**(g, 0) *//Nos posicionamos al comienzo del archivo*

respuesta ← ‘s’

**mientras** ((respuesta = ‘s’) o (respuesta = ‘S’)) **hacer**

**Entrada:** reg

**mientras** noEOF(g) o (reg.ddmmyyyy = aux.ddmmyyyy) *//Búsqueda secuencial de fecha repetida con marca*

**Leer**(g,aux) *inicial*

**fmientras**

**si** (EOF(g)) **entonces** *//No hay registros cargados con esa fecha, se llega hasta la última posición*

**Escribir**(g, reg) //*Escritura del nuevo registro al final del archivo*

**sino** *//No se alcanzó el fin del archivo porque ya existe un registro con esa fecha*

msj ← “Ya existe un registro cargado para esa fecha”

**Salida:** msj

**fsi**

msj ← “¿Desea cargar otro registro? (s/n)”

**Salida:** msj

**Entrada:** respuesta *//Posibilita repetir el ciclo de carga*

**fmientras**

**Cerrar**(g)

**FAccion**

**Acción** Eliminar (Dato: nombreArch ∈ Cadena, Dato-Resultado:g ∈ Archivo de Tregdiario)

**Léxico Local**

fecha ∈ Z *//Variable parámetro para la búsqueda de registros*

aux, reg ∈ Tregdiario //Variables auxiliares para la lectura y escritura del archivo

msj ∈ Cadena

**Inicio**

**Abrir**(“nombreArchivo”, g, a) *//Apertura con la posibilidad de modificar un registro*

**Entrada:** fecha

**IrPos**(g,0) *//Nos posicionamos al comienzo del archivo*

**mientras** noEOF(g) o (fecha = aux.ddmmyyyy)*//Búsqueda secuencial de fecha repetida con marca inicial*

Leer(g, aux)

**fmientras**

**si** (EOF(g))  **entonces** *//La fecha buscada no se encuentra en el archivo*

msj ← “No existe un registro con esa fecha.”

**Salida:** msj

**sino** *//Existe un registro con esa fecha*

**irPos**(g, PosicionActual -1) *//Volver a la posición del registro con la fecha buscada*

reg ← aux *//Se almacena en la variable de escritura los datos del último registro leído*

reg.borrado ← Verdadero *//Modificación del campo borrado para su posterior escritura*

**Escribir**(g, reg)

**fsi**

**Cerrar**(g)

**FAccion**

**Acción** Modificar (Dato: nombreArch ∈ Cadena, Dato-Resultado: g ∈ ARCHIVO de Tregdiario)

**Léxico Local**

fecha ∈ Z *//Variable parámetro para la búsqueda de registros*

reg ∈ Tregdiario *//variable auxiliar para la modificación del registro*

msj ∈ Cadena

**Inicio**

**Abrir** (“nombreArch.dat”, g, a) *//Apertura con la posibilidad de modificar un registro*

**Entrada**: fecha

IrPos(g, 0) *//Nos posicionamos al comienzo del archivo*

**mientras** noEOF(g) o (aux.ddmmyyyy = fecha) **hacer** *//Búsqueda secuencial de fecha repetida con marca inicial*

Leer(g, aux)

**fmientras**

**si** (EOF(g)) **entonces** *//La fecha buscada no se encuentra en el archivo*

msj ← “no encontrado”

**sino** *//Existe un registro con esa fecha*

IrPos(g, PosicionActual - 1) *//Volver a la posición del registro con la fecha buscada*

**Entrada:** reg *//Carga de los nuevos valores para su posterior escritura*

**Escribir**(g,reg)

**fsi**

**Cerrar (**g**)**

**FAccion**

**Acción** MostrarRegistroClima **(**Dato: nombreArch ∈ Cadena, g ∈ ARCHIVO de Tregdiario)

**Léxico local**

aux ∈ Tregdiario *//variable auxiliar para la lectura de registros*

msj ∈ Cadena

**Inicio**

**Abrir** (“nombreArch”, g, l) *//Apertura del archivo para lectura*

**si** (EOF(g)) **entonces** *//Tratamiento de secuencia vacia*

**msj ← “**No hay registros cargados en el archivo”

**Salida**: msj

**sino**

**mientras** (noEOF(g)) **hacer** *//Recorrido secuencial con marca inicial*

**Leer** (g, aux)

**si** (aux.borrado <> verdadero) **entonces** *//El registro está activo y debe ser mostrado*

**Salida:** aux

**fmientras**

**fsi**

**Cerrar** (g)

**FAccion**

**Acción** PasaAlArreglo (Dato: nombreArch ∈ Cadena, g ∈ Archivo de Tregdiario, Resultado a ∈ Tarreglo, cant ∈ Z )

**Léxico local**

i ∈ Z *//Variable iteradora utilizada para recorrer el arreglo*

aux ∈ Tregdiario *//Variable auxiliar para la lectura de registros*

msj ∈ Cadena

**Inicio**

i **←** 1

cant ← 0

**Abrir** (“nombreArch.dat”, g, l) *//Apertura del archivo para lectura*

**si** (EOF(g) **entonces** *//Tratamiento de secuencia vacía*

msj ← “El archivo se encuentra vacío”

**Salida:** msj

**sino**

**mientras** (noEOF(g)) **hacer** *//Recorrido secuencial del archivo*

**Leer**(g, aux)

**si** (aux.borrado <> verdadero) **entonces** *//Existen registros activos para almacenar en el arreglo*

a[i] ← aux *//Almacenamiento del registro en el arreglo*

cant ← cant + 1 *//Incremento del contador*

i ← i + 1

**fsi**

**fmientras**

**fsi**

**Cerrar**(g)

**FAccion**

**Función** BusquedaRecursiva (Dato: a ∈ Tarreglo, cant ∈ Z, fecha ∈ Z) → Z

**Inicio**

**Según**:

(cant = 0): ← (-1) *//Condición de salida*

(cant >0 ^ a[cant] != fecha): ←(Busqueda(a,cant-1,fecha) *//Condición de llamada recursiva*

(cant >0 ^ a[cant] = fecha): ←cant; *//Condición de salida con la posición del registro buscado*

**fsegun**

**FFuncion**

**Función** MaxTemp (Dato: nombreArch ∈ Cadena, g ∈ Archivo de Tregdiario) → Z

**Léxico local**

aux ∈ Tregdiario *//Variable auxiliar para la lectura de registros*

maximo ∈ Z *//Variable auxiliar para almacenar el mayor valor de tmax*

**Inicio**

**Abrir** (“nombreArch.dat”, g, l) *//Apertura del archivo para lectura*

**si** (EOF (g)) **entonces** *//Tratamiento de secuencia vacía*

maximo **←** -100 *//Valor irreal que condiciona la carga de la lista ante archivo vacío de registros*

**sino**

**Leer** (aux, g)

maximo ← aux.tmax *//Tratamiento del primer elemento*

**mientras** noEOF(g) **hacer** *//Recorrido secuencial del archivo*

**Leer** (g, aux)

**si** (aux.tmax > maximo) **entonces** *//Existe un valor de tmax mayor que el almacenado*

maximo ← aux.tmax

**fsi**

**fmientras**

**fsi**

**Cerrar**(g)

**←** maximo *//Devuelve el mayor valor de tmax almacenado en el archivo*

**FFuncion**

**Accion** OrdenarMayorTemp (Dato: nombreArch ∈ Cadena, g ∈ Archivo de Tregdiario, l ∈ puntero a TNodoRegistro)

**Léxico local**

aux ∈ Tregdiario *//Variable auxiliar para la lectura de registros*

sec ∈ puntero a TNodoRegistro *//Variable auxiliar para la carga de registros en la lista*

maximo ∈ Z *//Variable auxiliar que almacena el mayor valor de tmax*

msj ∈ Cadena

**Inicio**

**Abrir** (“nombreArch.dat”, g, l)

**si** (EOF (g)) **entonces** *//Tratamiento de secuencia vacía*

msj ← “El archivo está vacío”

**Salida:** msj

**sino**

maximo ← MaxTemp(“nombreArch.dat”, g) *//Llamada de la función para encontrar el mayor tmax*

l ← nil *//Inicialización de la lista*

**mientras** (noEOF(g)) **hacer**

**Leer** (aux, g)

**si** (aux.tmax = maximo) **entonces** *//La tmax del registro coincide con la mayor encontrada*

**obtener**(sec) *//Creación del nodo para posterior carga del registro*

(^sec).info ← aux

(^sec).next ← l *//Inserción a la cabeza*

l ← sec *//posicionamiento del puntero a la cabeza*

**fsi**

**fmientras**

**Cerrar**(g)

**liberar**(sec) *//Supresión del puntero auxiliar*

msj ← “Los días con mayor temperatura registrados son:”

**Salida**: msj

**mientras** l <> nil **hacer** *//Recorrido secuencial de la lista*

**Salida:** (^l).info.ddmmyyyy *//Muestra por pantalla los días con mayor tmax*

l ← (^l).next

**fmientras**

**fsi**

**FAccion**

**Acción** OrdPrecipitMax (dato: r ∈ Tarreglo, t ∈ Z) //recibe el arreglo y una variable con la cantidad de elementos

**Léxico local**

i, j ∈ Z *//Variables iteradoras utilizadas para recorrer el arreglo*

aux ∈ Tregdiario *//Variable auxiliar para la lectura de registros*

**Inicio** *//Ordenamiento por bubble sort invertido (mueve los elementos menores al final del arreglo)*

i ← t *//Asigna la cantidad total de elementos del arreglo al primer iterador*

**mientras** i > 1 **hacer** *//Hay elementos en el arreglo*

j ← 1 *//Posiciona el segundo iterador al comienzo del arreglo*

**mientras** j < i **hacer**

**si** r.[ j ].PP < r [ j+1 ].PP **entonces** *//Intercambia la posición de los elementos del arreglo*

aux ← r [ j ]

r [ j ] ← r[ j+1 ]

r[ j+1 ] ← aux

**fs**i

j ← j + 1

**fmientras**

i ← i - 1

**fmientras**

i ← 1 *//Mueve el iterador de nuevo al primer elemento del arreglo*

**mientras** i <= 10 **hacer**

**Salida**: r[i].ddmmyyyy, r[i].PP*//Informa la fecha y precipitaciones de los 10 mayores registros*

i ← i+1

**fmientras**

**FAccion**

**Acción** OrdenarMayorViento (**dato**: r ∈ Tarreglo, t ∈ Z) //recibe el arreglo y una variable con la cantidad de elementos

**Léxico local**

aux ∈ Tregdiario *//Variable auxiliar para la lectura de registros*

i, j ∈ (1… t + 1) *//Variables iteradoras utilizadas para recorrer el arreglo*

**Inicio** *//Insertion sort invertido(mueve los elementos menores al final de arreglo )*

i ← 2

**mientras** i <= t **hacer**

aux ← r[i]

j ← i - 1

**mientras** j > 0 ^ r[j].FF < aux.FF **hacer** *//Intercambia la posición de los elementos del arreglo*

r[j+1] ← r[j]

j ← j - 1

**fmientras**

r[j+1] ← aux

i ← i + 1

**fmientras**

i ← 1 *//Mueve el iterador de nuevo al primer elemento del arreglo*

**mientras** i <= 10 **hacer**

**Salida**: r[i].ddmmyyyy, r[i].FF *//Informa la fecha y velocidad del viento de los 10 mayores registros*

i ← i + 1

**fmientras**

**FAccion**

**Acción** CopiaBackup (Dato: “nombreArch.dat” ∈ Cadena, g ∈ Archivo de Tregdiario, Resultado: “nombreFecha.dat” ∈ Cadena, h ∈ Archivo de Tregdiario)

**Léxico local**

aux, sec ∈ Tregdiario *//Variables auxiliares para la lectura y escritura de registros*

msj ∈ Cadena

**Inicio**

**Abrir**(“nombreArch.dat”, g, l)

**si** (EOF(g) **entonces** *//Tratamiento de secuencia vacía*

msj ← “El archivo se encuentra vacío”

**Salida:** msj

**sino**

**Abrir**(“nombreFecha.dat”, h, e)

**mientras** noEOF(g) **hacer** *//Recorrido secuencial con marca inicial*

**Leer**(g, aux)

**si** (aux.borrado <> verdadero) **entonces** *//El registro está activo*

sec ← aux

**Escribir**(h,sec) *//Escritura del registro en el archivo de respaldo*

**fsi**

**fmientras**

**fsi**

**Cerrar**(g)

**Cerrar**(h)

**FAcción**

*//------------------------------------------------PROGRAMA PRINCIPAL------------------------------------------------//*

**Inicio**

**repetir**

**Entrada:** opc

**segun**

opc = 1: AltaRegistro (“nombreArch.dat”, f) *//Carga un registro en el archivo*

opc = 2: Eliminar (“nombreArch.dat”, f) *//Elimina un registro del archivo*

opc = 3: ModificarRegistro (“nombreArch.dat”, f) *//Modifica los campos de un registro cargado*

opc = 4: Mostrar(“nombreArch.dat”, f) *//Muestra todos los registros activos del archivo*

opc = 5: PasaAlArreglo (“nombreArch.dat”, f, arregloActivo, cantidad) *//Busca por fecha y muestra posición*

**Entrada:** fechaBusqueda

posicion ← Busqueda (arregloActivo, cantidad, fechaBusqueda)

**Salida:** cant

opc = 6: OrdenarMayorTemp (“nombreArch.dat”, f, lista) *//Crea una lista con los registros con la mayor*

*temperatura almacenada y la muestra*

opc = 7: PasaAlArreglo (“nombreArch.dat”, f, arregloActivo, cantidad) *//Ordena el arreglo por precipitación e*

OrdPrecipitMax(arregloActivo, cantidad) *informa los 10 mayores*

opc = 8: PasaAlArreglo (“nombreArch.dat”, f, arregloActivo, cantidad) *//Ordena el arreglo por velocidad del*

OrdenarMayorViento (arregloActivo, cantidad) *viento e informa los 10 mayores*

opc = 9: Respaldo (“nombreArch.dat”, f, “nombreFecha.dat”, backup) *//Crea una copia de respaldo del*

**fsegun** *archivo*

**hasta que** opc = 0

**Fin**