

**ALGORITMOS Y PROGRAMACIÓN I ( 75 – 40 )**  
*Cátedra Azcurra*

---

*Curso: 002      Cuatrimestre : Primero      Año : 2019*  
*Turno: Miércoles de 16 a 19 hrs. y Miércoles de 19 a 22 hrs.*

---

**TRABAJO PRÁCTICO 1**

---

**Tutor : MARCOS OTERO   Grupo N° 2**

---

Apellido/s, Nombre/s	Número de Padrón
<b>BAILAQUE, FEDERICO</b>	<b>104.764</b>
<b>BRASBURG, AGUSTIN</b>	<b>104.733</b>
<b>GIADAS, JUAN IGNACIO</b>	<b>101.713</b>
<b>LEDERKREMER, MIGUEL</b>	<b>61.719</b>
<b>TREZZA, FEDERICO</b>	<b>104.632</b>

<b>Entregado el :</b>		<b>Re-Entregado el :</b>	
<b>Aprobado el :</b>		<b>Concepto :</b>	<b>Firma :</b>
<b>Observaciones:</b>			



# Indice

<b>Enunciado</b>	<b>4</b>
<b>Datos, estructuras y sus relaciones</b>	<b>4</b>
Módulos	4
Constantes globales	4
Estructuras	4
1. Fecha	4
2. Nacionalidades	5
3. Nombres y apellidos	5
4. Sexo	5
5. Registro de cada empleado	5
6. Tabla principal de empleados	5
<b>main.c</b>	<b>6</b>
Corrida	6
Listado	6
<b>funciones.h</b>	<b>7</b>
<b>TP_1_Modulo_A.c</b>	<b>9</b>
Corrida	9
Listado	9
<b>TP_1_Modulo_B.c</b>	<b>13</b>
Corrida	13
Listado	13
<b>TP_1_Modulo_C.c</b>	<b>16</b>
Corrida	16
Listado	16
<b>TP_1_Modulo_D.c</b>	<b>18</b>
Corrida	18
Listado	18
<b>TP_1_Modulo_E.c</b>	<b>20</b>
Corrida	20
Listado	20
<b>mostrarBase.c</b>	<b>21</b>
Corrida	21
Listado	21

## Enunciado

Una empresa que se dedica a la reubicación de empleados tiene una lista de 500 empleados (como máximo) con la siguiente información:

- Nombre y apellido. Con el formato "Apellidos, Nombres". - Fecha de nacimiento del empleado. - Sexo (M, F) - Nacionalidad

Las nacionalidades pueden ser: Argentina, Uruguay, Chilena, Peruana, Boliviana, Paraguay, Brasileña. Un empleado puede tener más de una nacionalidad.

Se pide armar las estructuras y el programa que permita:

A- Realizar la carga de datos. Validar todos los datos ingresados.

B- Mostrar un listado de personas nacidas antes de 2000 ordenado por Fecha de nacimiento. Mostrar todos los datos de cada empleado.

C- Ingresar un Apellido por teclado y listar todas las mujeres que se apellidan igual y además nacieron en verano. Mostrar todos los datos de cada empleado.

D- Emitir un informe de empleados uruguayos y argentinos que tengan más de una nacionalidad. Mostrar todos los datos de cada empleado.

E- Indicar el porcentaje de empleados argentinos

## Datos, estructuras y sus relaciones

### Módulos

Para trabajar eficientemente en equipo, distribuimos el TP **varios archivos .c**, uno por módulo y colocamos todas las definiciones de tipos, librerías a incluir y constantes globales en un archivo **funciones.h** a ser incluido en todos los módulos.

### Constantes globales

Utilizamos sólo tres:

<b>MAXIMOFISICO</b>	La máxima cantidad de registros de la tabla principal
<b>NACIONALIDADES</b>	La cantidad de nacionalidades permitidas
<b>LARGOCAMPOSTEXTO</b>	El largo máximo de los strings nombres y apellidos

### Estructuras

A continuación describimos las estructuras globales utilizadas por todos los módulos.

#### 1. Fecha

```
typedef struct
{
    int dia;
    int mes;
    int anio;
} tipoFecha;
```

Para las fechas utilizamos la struct **tipoFecha** compuesta por los **int: dia, mes y anio**. Tomamos esta decisión, porque, si bien guardar las fechas en un string "yyyymmdd" facilita el ordenamiento, tener los componentes separados en forma numérica hace menos complejas las búsquedas como "nacidos en verano" o "antes del 2000".

## 2. Nacionalidades

```
typedef int tVectorNacionalidades[NACIONALIDADES];
```

Luego de considerar varias alternativas, decidimos que la solución más simple para manejar las nacionalidades de cada empleado, es un vector de valores booleanos de tipo **tVectorNacionalidades** con tantas posiciones como empleados. Cada posición del vector representa una nacionalidad: 0 Argentina, 1 Uruguay, 2 Chilena, 3 Peruana, 4 Boliviana, 5 Paraguaya, 6 Brasileña. Cada elemento del vector tiene un 1 si el empleado tiene la nacionalidad correspondiente a esa posición o, de lo contrario 0. Esta estructura resultó muy práctica en los módulos de carga de datos y de búsqueda de argentinos y uruguayos con múltiples nacionalidades.

## 3. Nombres y apellidos

```
char apellidos[LARGOCAMPOSTEXTO];
char nombres[LARGOCAMPOSTEXTO];
```

Los nombres y los apellidos de cada registro están guardados en las cadenas **apellidos** y **nombres** de largo **LARGOCAMPOSTEXTO**.

## 4. Sexo

```
char sexo;
```

Para guardar el sexo alcanza con un campo de tipo char **sexo** de un sólo caracter.

## 5. Registro de cada empleado

```
typedef struct
{
    char apellidos[LARGOCAMPOSTEXTO];
    char nombres[LARGOCAMPOSTEXTO];
    tipoFecha fecha;
    char sexo;
    tVectorNacionalidades nacionalidades;
} typeEmpleado;
```

Los datos de cada empleado se guardan en un registro de tipo **typeEmpleado** es una struct compuesta por las 4 estructuras y el campo char descriptos en los puntos anteriores.

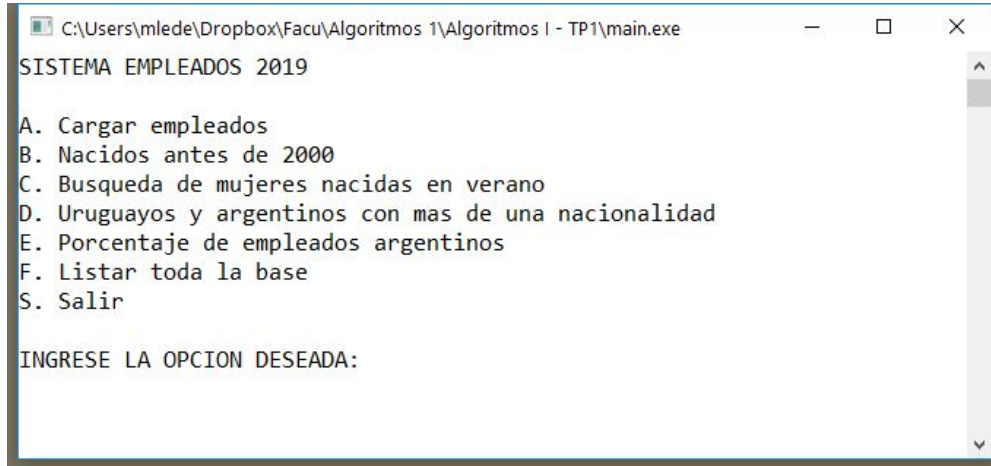
## 6. Tabla principal de empleados

```
typedef typeEmpleado tVectorEmpleados[MAXIMOFISICO];
```

Finalmente, la tabla principal con la información de todos los empleados de la empresa, es un vector de tipo **tVectorEmpleados** de tamaño **MAXIMOFÍSICO**. Cada elemento del vector es una estructura del tipo **typeEmpleado** descripto en el punto anterior.

# main.c

## Corrida



## Listado

```
#include "funciones.h"
```

```
int main()
{
    int maximoLogico, salir;
    char opcion;
    maximoLogico = 0;
    salir = 0;
    tVectorEmpleados empleados;
    cargarDatosDeEjemplo(empleados, &maximoLogico);
    while (!salir)
    {
        printf("SISTEMA EMPLEADOS 2019\n\n");
        printf("A. Cargar empleados\nB. Nacidos antes de 2000\nC. Busqueda de mujeres nacidas en verano\n");
        printf("D. Uruguayos y argentinos con mas de una nacionalidad\nE. Porcentaje de empleados argentinos\n");
        printf("F. Listar toda la base\nS. Salir\n\nINGRESE LA OPCION DESEADA: ");
        scanf(" %c", &opcion);
        fflush(stdin);
        switch (opcion)
        {
            case 'A':
            case 'a':
                cargarEmpleados(empleados, &maximoLogico);
                pausa();
                break;
            case 'B':
            case 'b':
                antesDel2000(empleados, maximoLogico);
                break;
            case 'C':
            case 'c':
                mujeres(empleados, maximoLogico);
                break;
```

```

    case 'D':
    case 'd':
        dobleNacionalidad(empleados, maximoLogico);
        break;
    case 'E':
    case 'e':
        argentinos(empleados, maximoLogico);
        break;
    case 'S':
    case 's':
        salir = 1;
        break;
    case 'F':
    case 'f':
        borrarPantalla();
        printf("BASE DE EMPLEADOS\n\n");
        mostrarBase(empleados, maximoLogico);
        break;
    default:
        printf("Por favor digite una opcion valida. Pulse Enter para continuar");
        getchar();
        borrarPantalla();
}
}
borrarPantalla();
printf("\nGRACIAS POR UTILIZAR SISTEMA EMPLEADOS 2019!\n\n");
return 0;
}

void borrarPantalla()
{
#ifdef _WIN32
    system("cls");
#else
    system("clear");
#endif
}

void pausa()
{
    printf("\nPulse Enter para volver al Menu Principal");
    fflush(stdin);
    getchar();
    borrarPantalla();
    return;
}

```

---

## funciones.h

```

#include <stdio.h>
#include <string.h>
// CONSTANTES Y TIPOS GLOBALES
#ifndef DEFINICIONES_CARGADAS
#define DEFINICIONES_CARGADAS
#define MAXIMOFISICO 500
#define NACIONALIDADES 7

```

```
#define LARGOCAMPOSTEXTO 20
// ESTRUCTURA FECHA Y VECTOR NACIONALIDADES A INCLUIR EN CADA REGISTRO
typedef struct
{
    int dia;
    int mes;
    int anio;
} tipoFecha;
typedef int tVectorNacionalidades[NACIONALIDADES];
/* TIPO DE DATOS DE CADA REGISTRO: typeEmpleado
   TIPO DE DATOS DEL VECTOR PRINCIPAL: tVectorEmpleados */
typedef struct
{
    char apellidos[LARGOCAMPOSTEXTO];
    char nombres[LARGOCAMPOSTEXTO];
    tipoFecha fecha;
    char sexo;
    tVectorNacionalidades nacionalidades;
} typeEmpleado;
typedef typeEmpleado tVectorEmpleados[MAXIMOFISICO];
#include "mostrarBase.c"
#include "cargarDatosDeEjemplo.c"
#include "TP_1_Modulo_A.c"
#include "TP_1_Modulo_B.c"
#include "TP_1_Modulo_C.c"
#include "TP_1_Modulo_D.c"
#include "TP_1_Modulo_E.c"
#endif
void borrarPantalla();
void pausa();
```

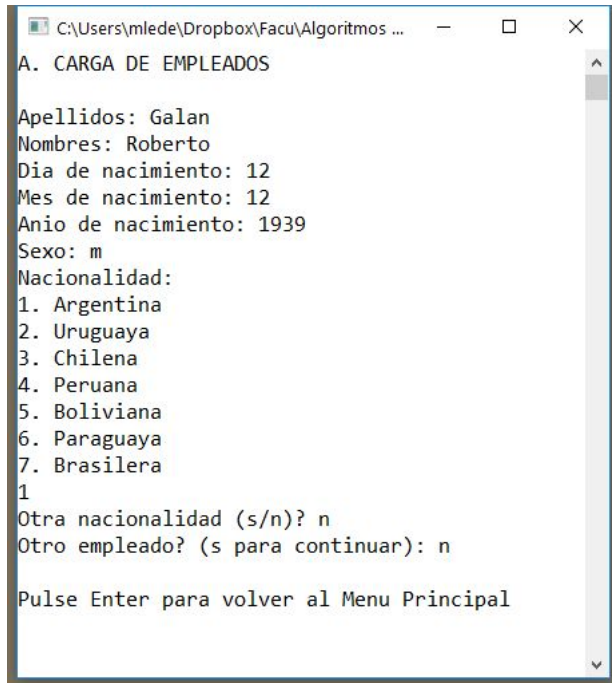
---



# TP\_1\_Modulo\_A.c

A. Realizar la carga de datos. Validar todos los datos ingresados.

## Corrida



## Listado

```
// A. CARGA Y VALIDACION DE DATOS
#include "funciones.h"
#define ANIOACTUAL 2019

int esCadenaValida(char texto[LARGOCAMPOSTEXTO]);
void cargarNacionalidad(tVectorNacionalidades nacEmpleados);
char validarSexo(char sexo);
void validarFecha(int *dia, int *mes, int *anio);
void validarnacionalidad(int *numNacionalidad);

void cargarEmpleados(tVectorEmpleados empleados, int *maximoLogico)
{
    int i;
    char seguimos;
    i = *maximoLogico;
    borrarPantalla();
    printf("A. CARGA DE EMPLEADOS\n\n");
    do
    {
        do
        {
            printf("Apellidos: ");
            fgets(empleados[i].apellidos, LARGOCAMPOSTEXTO, stdin);
            fflush(stdin);
        } while (!esCadenaValida(empleados[i].apellidos));
```

```

do
{
    printf("Nombres: ");
    fgets(empleados[i].nombres, LARGOCAMPOSTEXTO, stdin);
    fflush(stdin);
} while (!esCadenaValida(empleados[i].nombres));
printf("Dia de nacimiento: ");
scanf("%d", &empleados[i].fecha.dia);
fflush(stdin);
printf("Mes de nacimiento: ");
scanf("%d", &empleados[i].fecha.mes);
fflush(stdin);
printf("Año de nacimiento: ");
scanf("%d", &empleados[i].fecha.ano);
validarFecha(&empleados[i].fecha.dia, &empleados[i].fecha.mes, &empleados[i].fecha.ano);
fflush(stdin);
printf("Sexo: ");
scanf(" %c", &empleados[i].sexo);
fflush(stdin);
empleados[i].sexo = validarSexo(empleados[i].sexo);
cargarNacionalidad(empleados[i].nacionalidades);
i++;
printf("Otro empleado? (s para continuar): ");
scanf(" %c", &seguimos);
fflush(stdin);
} while ((seguimos == 's') && (i < 100));
*maximoLogico = i;
return;
}

int esCadenaValida(char texto[LARGOCAMPOSTEXTO])
{
    int esValida = 1;
    int i = 0;
    if (texto[0] <= 32)
        esValida = 0; // si comienza con espacio o la cadena es vacia no es valida
    while (esValida && (texto[i] != 10) && (texto[i] != 0))
    { // mientras las letras sean validas y no llegamos al final, continua recorriendo la cadena
        if ((texto[i] >= 97) && (texto[i] <= 122))
        { // si es una letra minuscula pasar a mayuscula
            texto[i] -= 32;
        }
        else
        {
            if (((texto[i] < 65) || (texto[i] > 90)) && (texto[i] != 32))
            { // si la letra no es una mayuscula o espacio no es valida
                esValida = 0;
                printf("Solo letras y espacios por favor\n");
            }
        }
        i++;
    }
    texto[i] = 0; // le quita el new line
    return esValida;
}

void cargarNacionalidad(tVectorNacionalidades nacEmpleados)
{
    int i;

```

```

int nacionalidad;
char seguimos;
i = 0;
do
{
    printf("Nacionalidad: \n");
    printf("1. Argentina \n2. Uruguay\n3. Chilena\n4. Peruana\n5. Boliviana \n6. Paraguaya\n7. Brasileira\n");
    scanf("%i", &nacionalidad);
    fflush(stdin);
    validarnacionalidad(&nacionalidad);
    nacEmpleados[nacionalidad - 1] = 1; // colocamos un 1 en la posicion correspondiente del vector nacionalidades
    i++;
    printf("Otra nacionalidad (s/n)? ");
    fflush(stdin);
    scanf("%c", &seguimos);
    while (seguimos != 's' && seguimos != 'n')
    {
        printf("El dato no es valido. Ingrese 's' o 'n': ");
        scanf("%c", &seguimos);
        fflush(stdin);
    }
} while ((seguimos == 's') && (i < 7));
return;
}

char validarSexo(char sexo)
{
    while ((sexo != 'm') && (sexo != 'f'))
    {
        if ((sexo != 'f' && sexo != 'm'))
        {
            printf("Sexo invalido. Ingrese 'm' o 'f' : ");
            scanf("%c", &sexo);
            fflush(stdin);
        }
    }
    sexo -= 32; // pasamos a mayiç%sculas
    return sexo;
}

void validarFecha(int *dia, int *mes, int *anio)
{
    int correcto;
    correcto = 0;
    while (correcto == 0)
    {
        if (*mes >= 1 && *mes <= 12)
        {
            switch (*mes)
            {
                case 1:
                case 3:
                case 5:
                case 7:
                case 8:
                case 10:
                case 12:
                    if (*dia >= 1 && *dia <= 31)
                    {
                        correcto = 1;
                    }
                }
            }
        }
    }
}

```

```

        else
        {
            correcto = 0;
        }
        break;
case 4:
case 6:
case 9:
case 11:
    if (*dia >= 1 && *dia <= 30)
    {
        correcto = 1;
    }
    else
    {
        correcto = 0;
    }
    break;
case 2:
    if (*anio % 4 == 0 && *anio % 100 != 0 || *anio % 400 == 0)
        if (*dia >= 1 && *dia <= 29)
            correcto = 1;
        else
            correcto = 0;
    else if (*dia >= 1 && *dia <= 28)
        correcto = 1;
    else
        correcto = 0;
}
}
else
    correcto = 0;
if (correcto == 0 || (*anio > ANIOACTUAL || *anio < 1900))
{
    printf("Fecha invalida. Ingrese otra (DD/MM/AAAA)\n");
    fflush(stdin);
    printf("Dia de nacimiento: ");
    scanf("%i", &(*dia));
    fflush(stdin);
    printf("Mes de nacimiento: ");
    scanf("%i", &(*mes));
    fflush(stdin);
    printf("Anio de nacimiento: ");
    scanf("%i", &(*anio));
    fflush(stdin);
    correcto = 0;
}
}
}
void validarnacionalidad(int *numNacionalidad)
{
    while (*numNacionalidad > 7 || *numNacionalidad < 1)
    {
        printf("El dato ingresado es invalido. Ingrese un numero entre 1 y 7: \n");
        scanf("%i", numNacionalidad);
        fflush(stdin);
    }
}
}

```

---

## TP\_1\_Modulo\_B.c

B. Mostrar un listado de personas nacidas antes de 2000 ordenado por Fecha de nacimiento. Mostrar todos los datos de cada empleado.

### Corrida

APELLIDOS	NOMBRES	FECHA NAC.	SEXO	NACIONALIDAD
LEGRAND	MIRTA	08-08-1919	F	Brasilenia
GALAN	ROBERTO	12-12-1929	M	Argentina Paraguaya
CASAN	MORIA	02-02-1932	F	Argentina
GALAN	ROBERTO	12-12-1939	M	Argentina
ALFANO	GRACIELA	13-05-1940	F	Argentina
BORGES	GRACIELA	13-05-1940	F	Uruguay Chilena Peruana
DE LA V	FLOR	13-06-1940	F	Chilena
BORGES	JORGE LUIS	12-12-1966	M	Argentina
SPINETTA	LUIS ALBERTO	04-11-1978	M	Argentina

Pulse Enter para volver al Menu Principal.

### Listado

// B- Mostrar un listado de personas nacidas antes de 2000 ordenado por Fecha de nacimiento. Mostrar todos los datos de cada empleado.

```
#include "funciones.h"
```

```
void ordenarPorFechaNacimiento(tVectorEmpleados vector, int maximoLogico, tVectorEmpleados vectorOrdenado);
int esPosterior(tipoFecha fecha, tipoFecha fecha2);
```

```
void antesDel2000(tVectorEmpleados empleados, int maximoLogico)
{
    tVectorEmpleados nacidosAntesDel2000, ordenadosAntesDel2000;
    int i, i2000, maximoLogico2000;
    i2000 = 0;
    borrarPantalla();
    // GENERAMOS NUEVO VECTOR FILTRADO CON SOLO LOS NACIDOS LUEGO DEL 2000
    for (i = 0; i < maximoLogico; i++)
    {
        if (empleados[i].fecha.anio < 2000)
        {
            nacidosAntesDel2000[i2000] = empleados[i];
            i2000++;
        }
    }
    maximoLogico2000 = i2000;
    // ORDENAMOS EL VECTOR FILTRADO Y LO MOSTRAMOS
    ordenarPorFechaNacimiento(nacidosAntesDel2000, maximoLogico2000, ordenadosAntesDel2000);
    printf("B. EMPLEADOS NACIDOS ANTES DEL 2000 ORDENADOS POR FECHA DE NACIMIENTO\n\n");
    mostrarBase(ordenadosAntesDel2000, maximoLogico2000);
}
```

```

    return;
}

// ORDENA VECTORES POR FECHA
void ordenarPorFechaNacimiento(tVectorEmpleados vector, int maximoLogico, tVectorEmpleados vectorOrdenado)
{
    int i, j, posicion;
    for (i = 0; i < maximoLogico; i++)
    {
        vectorOrdenado[i].fecha.dia = 0;
    }
    /* ORDENAMOS POR EL METODO INSERTION SORT: PARA CADA ELEMENTO, RECORREMOS EL VECTOR CONTANDO EN posicion
    TODOS LOS ELEMENTOS QUE SON MENORES AL ELEMENTO ACTUAL. LUEGO LO COPIAMOS EN EL NUEVO VECTOR EN LA UBICACION
    posicion */
    borrarPantalla();
    for (i = 0; i < maximoLogico; i++)
    {
        posicion = 0;
        // COMPARAMOS LA FECHA DEL ELEMENTO ACTUAL CONTRA LA FECHA DE TODOS LOS OTROS ELEMENTOS DEL VECTOR:
        for (j = 0; j < maximoLogico; j++)
        {
            if (i != j)
            { // LO VAMOS COMPARANDO CONTRA TODOS LOS OTROS ELEMENTOS Y ESTABLECEMOS LA POSICION EN Q DEBE QUEDAR
                if (esPosterior(vector[i].fecha, vector[j].fecha) == 1)
                {
                    posicion++;
                }
            }
        }
        // SI EN EL VECTOR ORDENADO YA HAY ELEMENTOS CON IGUAL FECHA, LO UBICAMOS EN LA SIGUIENTE POSICION
        while (esPosterior(vectorOrdenado[posicion].fecha, vector[i].fecha) == 2)
        {
            posicion++;
        }
        vectorOrdenado[posicion] = vector[i];
    }
}

/* COMPARA DE FECHAS. DEVUELVE:
0 SI FECHA2>FECHA
1 SI FECHA>FECHA2,
2 SI FECHA==FECHA2
*/
int esPosterior(tipoFecha fecha, tipoFecha fecha2)
{
    int esPost = 0;
    if (fecha.anio > fecha2.anio)
    {
        esPost = 1;
    }
    if (fecha.anio == fecha2.anio)
    {
        if (fecha.mes > fecha2.mes)
        {
            esPost = 1;
        }
        if (fecha.mes == fecha2.mes)
        {
            if (fecha.dia > fecha2.dia)

```

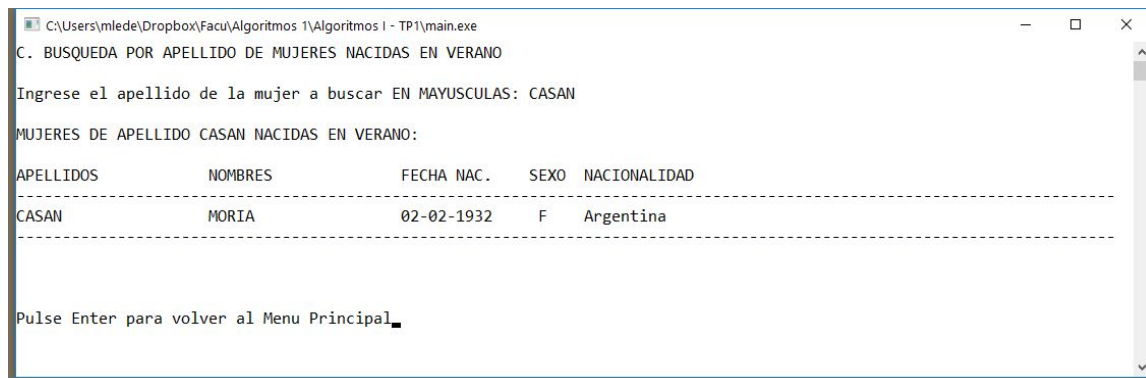
```
{
    esPost = 1;
}
if (fecha.dia == fecha2.dia)
{
    esPost = 2; // misma fecha de nacimiento!!!
}
}
}
return esPost;
}
```

---

## TP\_1\_Modulo\_C.c

C. Ingresar un Apellido por teclado y listar todas las mujeres que se apellidan igual y además nacieron en verano. Mostrar todos los datos de cada empleado.

### Corrida



### Listado

/\* C- Ingresar un Apellido por teclado y listar todas las mujeres que se apellidan igual y ademas nacieron en verano. Mostrar todos los datos de cada empleado.

\*/

```
#include "funciones.h"
```

```
void mujeres(tVectorEmpleados empleados, int maximoLogico) /* modificacion: saque el puntero del maximoLogico */
```

```
{
```

```
    int i;
    int posicion = 0;
    char apellido[LARGOCAMPOSTEXTO];
    tipoFecha empieza_verano;
    empieza_verano.dia = 21;
    empieza_verano.mes = 12;
    tipoFecha termina_verano;
    termina_verano.dia = 20;
    termina_verano.mes = 3;
    borrarPantalla();
    printf("C. BUSQUEDA POR APELLIDO DE MUJERES NACIDAS EN VERANO\n\n");
    printf("Ingrese el apellido de la mujer a buscar EN MAYUSCULAS: ");
    fgets(apellido, LARGOCAMPOSTEXTO, stdin);
    apellido[strlen(apellido) - 1] = '\0';
    printf("\nMUJERES DE APELLIDO %s NACIDAS EN VERANO:\n\n", apellido);
    mostrarEncabezado();
    for (i = 0; i < maximoLogico; i++)
    {
        if (empleados[i].sexo == 'F')
        { /* Asignamos el anio del empleado a el anio del verano para poder reusar nuestra funcion que compara fechas*/
```



```

    empieza_verano.anio = empleados[i].fecha.anio;
    termina_verano.anio = empleados[i].fecha.anio;
    if ((strcmp(empleados[i].apellidos, apellido) == 0) && ((esPosterior(empleados[i].fecha, empieza_verano) >=
1) || (esPosterior(termina_verano, empleados[i].fecha) >= 1)))
    /*          Pido que tengan mismo apellido          Que haya nacido a partir del 21 de Diciembre
Que haya nacido hasta el 20 de marzo          */
    {
        mostrarRegistro(empleados, i); /* Muestro el empleado que cumple las condiciones */
    }
}
mostrarPie();
pausa();
}

```

---

## TP\_1\_Modulo\_D.c

D. Emitir un informe de empleados uruguayos y argentinos que tengan más de una nacionalidad. Mostrar todos los datos de cada empleado.

### Corrida



D. ARGENTINOS Y URUGUAYOS CON MAS DE UNA NACIONALIDAD

APELLIDOS	NOMBRES	FECHA NAC.	SEXO	NACIONALIDAD
GARCIA	CHARLY	30-01-2010	M	Argentina Uruguay
ESPOSITO	LALI	05-05-2002	F	Argentina Uruguay Chilena Boliviana Paraguaya Brasilenia
BORGES	GRACIELA	13-05-1940	F	Uruguay Chilena Peruana
CASAN	GALA	21-03-2001	F	Argentina Brasilenia
GALAN	ROBERTO	12-12-1929	M	Argentina Paraguaya

Pulse Enter para volver al Menu Principal

### Listado

```
#include "funciones.h"

void dobleNacionalidad(tVectorEmpleados empleados, int maximoLogico)
{
    int i;
    int posicion;
    int otranacionalidad;
    borrarPantalla();
    printf("D. ARGENTINOS Y URUGUAYOS CON MAS DE UNA NACIONALIDAD\n\n");
    mostrarEncabezado();
    for (i = 0; i < maximoLogico; i++)
    {
        if (empleados[i].nacionalidades[0] == 1 || empleados[i].nacionalidades[1] == 1) //Me fijo que sea Argentino o Uruguayo
        {
            otranacionalidad = 0;
            posicion = 1; // Lo igualo a 1 ya que la posicion 0 es Argentina y lo primero que hago en el while es sumarle uno porque la posicion 1 es Uruguay
            if(empleados[i].nacionalidades[0] == 1 && empleados[i].nacionalidades[1] == 1)//Me fijo el caso de que sea Argentino y Uruguayo
            {
                mostrarRegistro(empleados, i);
            }
            else
            {
                while (otranacionalidad == 0 && posicion < 7)
                {
                    posicion++;
                    if (empleados[i].nacionalidades[posicion] == 1) //Si se da que es Argentino o Uruguayo y de otra
```

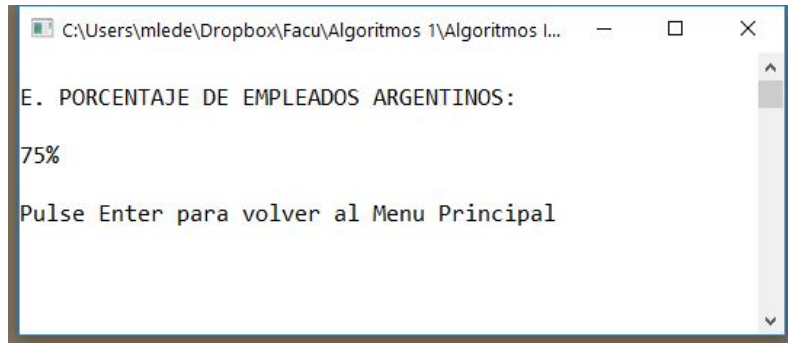
```
nacionalidad cambio otracionalidad por lo que deberia salir del while
    otracionalidad = 1;
}
if (otracialidad == 1 ) //Me fijo si cumple que sea Arg o Uru y de otra nacionalidad
{
    mostrarRegistro(empleados, i);
}
}
}
mostrarPie();
pausa();
return;
}
```

---

# TP\_1\_Modulo\_E.c

E. Indicar el porcentaje de empleados argentinos

## Corrida



## Listado

```
#include "funciones.h"
void argentinos(tVectorEmpleados empleados, int maximoLogico)
{
    int i, CantidadArgentinos = 0, CodigoArgentina = 0;
    borrarPantalla();
    for (i = 0; i < maximoLogico; i++)
    {
        if (empleados[i].nacionalidades[CodigoArgentina] == 1)
        {
            CantidadArgentinos++;
        }
    }
    printf("\nE. PORCENTAJE DE EMPLEADOS ARGENTINOS:\n\n%.0f%%\n", ((float)CantidadArgentinos / maximoLogico) *
100);
    pausa();
    return;
}
```

---

## mostrarBase.c

Muestra la base completa o un registro en particular. Se utiliza en varios módulos

### Corrida



APELLIDOS	NOMBRES	FECHA NAC.	SEXO	NACIONALIDAD
SPINETTA	LUIS ALBERTO	04-11-1978	M	Argentina
GARCIA	CHARLY	30-01-2010	M	Argentina Uruguay
CASAN	MORIA	02-02-1932	F	Argentina
LEGRAND	MIRTA	08-08-1919	F	Brasilenia
ESPOSITO	LALI	05-05-2002	F	Argentina Uruguay Chilena Boliviana Paraguaya Brasilenia
ALFANO	GRACIELA	13-05-1940	F	Argentina
BORGES	GRACIELA	13-05-1940	F	Uruguay Chilena Peruana
DE LA V	FLOR	13-06-1940	F	Chilena
BORGES	JORGE LUIS	12-12-1966	M	Argentina
CASAN	GALA	21-03-2001	F	Argentina Brasilenia

Pulse Enter para volver al Menu Principal

### Listado

```
// MUESTRA LA BASE COMPLETA O UN REGISTRO i. SE UTILIZA EN VARIOS MODULOS
#include "funciones.h"
typedef char typeNombreNacion[10];
typedef typeNombreNacion tVectorNombreNacion[NACIONALIDADES];
void mostrarRegistro(tVectorEmpleados empleados, int i);
void mostrarEncabezado();
void mostrarPie();

void mostrarBase(tVectorEmpleados empleados, int maximoLogico)
{
    int i;
    mostrarEncabezado();
    for (i = 0; i < maximoLogico; i++)
    {
        mostrarRegistro(empleados, i);
    }
    mostrarPie();
    pausa();
    return;
}

void mostrarEncabezado()
{
    printf("%-20s %-20s %-13s %-5s %s\n", "APELLIDOS", "NOMBRES", "FECHA NAC.", "SEXO", "NACIONALIDAD");
}
```

```

printf("-----
-----\n");
}

void mostrarPie()
{

printf("-----
-----\n\n\n");
}

void mostrarRegistro(tVectorEmpleados empleados, int i)
{
    int codigoNacion;
    tVectorNombreNacion nombreNacionalidad = {"Argentina", "Uruguay", "Chilena", "Peruana", "Boliviana",
"Paraguay", "Brasilenia"};
    printf("%-20s %-20s %02d-%02d-%d      %-5c", empleados[i].apellidos, empleados[i].nombres,
empleados[i].fecha.dia, empleados[i].fecha.mes, empleados[i].fecha.anio, empleados[i].sexo);
    // mostrar las nacionalidades
    for (codigoNacion = 0; codigoNacion < NACIONALIDADES; codigoNacion++)
    {
        if (empleados[i].nacionalidades[codigoNacion] == 1)
        {
            printf("%s ", nombreNacionalidad[codigoNacion]);
        }
    }
    printf("\n");
    return;
}

```

---