



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en Software y tecnologías emergentes

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Antonio Ramos González

**Matrícula:** 372576

**Maestro:** Pedro Núñez Yépiz

**Actividad No. :** 13 (ANEXO)

**Tema - Unidad :** Archivos

**Ensenada Baja California a 19 de noviembre del 2023**

```

Antonio Ramos Gonzalez ID: 372576
// 14/11/2023 || 19/11/2023
// Se generara un archivo donde se almacenara datos generados por el usuario, y la lectura de archivos
// RGA_Act13_932
#include "Babilonia.h"
#define N 5000
> typedef struct _Reg...
} Treg;

int msg();
void menu();
int verMt(Treg reg[], int n, int mt);
Treg agregar(Treg reg[], int i);
void nombreAl(char nombre[], int sex);
void puestoAl(char puesto[]);
void apAl(char apellido[]);
int searchSec(Treg reg[], int n, int mt);
int searchBin(Treg reg[], int inf, int sup, int mt);
int orderBu(Treg reg[], int n);
void pintReg(Treg reg[], int n);
void pintOneReg(Treg reg[], int n);
void swap(Treg reg[], int i, int j);
void createTXT(Treg reg[], int n);
int partition(Treg reg[], int low, int high);
void quicksort(Treg reg[], int low, int high);
void regEliminados(Treg reg[], int n);
void modReg(Treg reg[], int n);
void MostrarArchivo();
void archivoBin(Treg reg[], int n);
int cargarArchBin(Treg reg[], int n);
void BinArchEl();

int main()
{
    srand(time(NULL));
    fflush(stdin);
    menu();
    return 0;
}

int msg()
{
    system("CLS");
    printf("Menu\n");
    printf("1.-Agregar\n");
    printf("2.-Editar registro\n");
    printf("3.-Eliminar registro\n");
    printf("4.-Buscar Registro\n");
    printf("5.-Ordenar\n");
    printf("6.-Imprimir\n");
    printf("7.-Archivo texto\n");
    printf("8.-Mostrar archivo de texto\n");
    printf("9.-Archivo Binario\n");
    printf("10.-Cargar Archivo binario\n");
    printf("11.-Registros eliminados\n");
    printf("0.-Salir\n");
    return valid("Elije una opcion: ", 0, 11);
}

```

```

void menu()
{
    int opc, kay;
    int i = 0, j, band, bus;
    int fileBand = 0;
    Treg reg[N];
    do
    {
        opc = msg();
        switch (opc)
        {
            case 1:
                if (i < N)
                {
                    for (j = 0; j < 100; j++)
                    {
                        if (i < N)
                        {
                            reg[i] = agregar(reg, i);
                            i++;
                            band = 1;
                        }
                    }
                }
                else
                {
                    printf("Base de datos llena\n");
                    system("PAUSE");
                }

                break;
            case 2:
                kay = valid("Ingrese matricula que desea modificar: ", 300000, 399999);
                if (band)
                {
                    bus = searchSec(reg, i, kay);
                    pintOneReg(reg, bus);
                }
                else
                {
                    bus = searchBin(reg, 0, i, kay);
                    pintOneReg(reg, bus);
                }
                modReg(reg, bus);

                break;
            case 3:
                kay = valid("Ingrese matricula que desea eliminar: ", 300000, 399999);
                if (band)
                {
                    bus = searchSec(reg, i, kay);
                    pintOneReg(reg, bus);
                }
                else
                {
                    bus = searchBin(reg, 0, i, kay);

```

```

        bus = searchBin(reg, 0, i, kay);
        pintOneReg(reg, bus);
    }
    if (valid("\nDESEA ELIMINAR REGISTRO\n1.-Si\n0.-NO\n", 0, 1))
    {
        reg[bus].status = 0;
    }

    break;
case 4:
    kay = valid("Ingrese matricula que desea Buscar: ", 300000, 399999);
    if (band)
    {
        bus = searchSec(reg, i, kay);
        pintOneReg(reg, bus);
    }
    else
    {
        bus = searchBin(reg, 0, i, kay);
        pintOneReg(reg, bus);
    }
    system("PAUSE");
    break;
case 5:
    if (band)
    {
        if (i < 500)
        {
            band = orderBu(reg, i);
        }
        else
        {
            band = 1;
            quicksort(reg, 0, i - 1);
        }
    }
    break;
case 6:
    pintReg(reg, i);
    break;
case 7:
    createTXT(reg, i);
    break;
case 8:
    MostrarArchivo();
    break;
case 9:
    archivoBin(reg, i);
    break;
case 10:
    if (fileBand == 0)
    {
        i = cargarArchBin(reg, i);
    }

    break;
case 11:
    BinArchEl();
    break;

```

```
    } while (opc != 0);
```

```
void nombreAl(char nombre[], int sex)
```

```
{
    int i, al;
    char nombH[50][20] = {
        "JUAN", "PEDRO", "LUIS", "CARLOS", "ANDRES", "MIGUEL", "DAVID", "RAFAEL", "FRANCISCO",
        "ALEJANDRO", "JORGE", "DANIEL", "GUILLERMO", "SERGIO", "PABLO", "JAVIER", "ALBERTO",
        "SERGIO", "OSCAR", "GUILLERMO", "MARIO", "RICARDO", "IGNACIO", "FEDERICO", "RODRIGO",
        "MANUEL", "ANTONIO", "RAUL", "ALFREDO", "FERNANDO", "ROBERTO", "EDUARDO", "HUGO", "ENRIQUE",
        "EMILIO", "ARMANDO", "JULIO", "ADRIAN", "GUSTAVO", "ARTURO", "ANGEL", "HORACIO",
        "RODOLFO", "LUCAS", "RAMON", "OMAR", "CESAR", "SANTIAGO", "GERARDO", "ERNESTO"};
    char nombM[50][20] = {
        "ANA", "SOFIA", "CARMEN", "ISABEL", "ANDREA", "PATRICIA", "ELENA", "MARTA", "BEATRIZ",
        "PAULA", "JULIA", "VALERIA", "RAQUEL", "NATALIA", "CLAUDIA", "VALENTINA", "LORENA",
        "ELENA", "MARIANA", "ADRIANA", "LAURA", "ANTONIA", "MARTINA", "ANGELA", "LUISA",
        "SARA", "VERONICA", "ALICIA", "ROSA", "LUCIA", "VICTORIA", "GLORIA", "INES",
        "ROCIO", "VANESA", "RAFAELA", "ANAI", "EVA", "SANDRA", "LOLA", "ANA", "ESTHER",
        "MIRIAM", "LIDIA", "YOLANDA", "INES", "MARINA", "CAROLINA", "PILAR", "IRENE"};
    for (i = 0; i < 10; i++)
    {
        al = rand() % 50;
    }
    if (sex == 1)
    {
        strcpy(nombre, nombH[al]);
    }
    else
    {
        strcpy(nombre, nombM[al]);
    }
}
```

```
void apAl(char apellido[])
```

```
{
    int i, al;
    char ap[87][20] =
    {
        "PEREZ", "LOPEZ", "GONZALEZ", "RODRIGUEZ", "MARTINEZ", "SANCHEZ", "FERNANDEZ", "GARCIA", "RODRIGUEZ", "DIAZ", "TORRES", "RAMIREZ", "JIMENEZ",
        "VARGAS", "MORALES", "RUIZ", "CASTRO", "HERRERA", "MEDINA", "NAVARRO", "DELGADO", "ROMERO", "MORENO", "GOMEZ", "VARGAS", "PEREZ", "FLORES", "GONZALEZ",
        "NUNEZ", "SILVA", "SOTO", "MENDOZA", "ROJAS", "MENDEZ", "SALAS", "AGUIRRE", "BENITEZ", "GUZMAN", "PAREDES", "RIOS", "VALENZUELA", "CORDERO", "ESCOBAR",
        "OCHOA", "MORALES", "PIZZARRO", "ARAYA", "CABRERA", "LOPEZ", "ORTEGA", "FERNANDEZ", "NAVARRO", "GUZMAN", "MENDEZ", "ROJAS", "SALAZAR", "VILLANUEVA", "PAREDES",
        "RIVAS", "SANDOVAL", "ESPINOZA", "MUNOZ", "ACOSTA", "DELGADO", "CASTRO", "NAVARRO", "GONZALEZ", "MENDOZA", "SILVA", "RIOS", "ROJAS", "AGUIRRE", "MORALES", "GOMEZ",
        "SOTO", "TORRES", "PEREZ", "MEDINA", "BENITEZ", "GARCIA", "VARGAS", "RODRIGUEZ", "RAMIREZ", "TORRES", "MORALES", "HERRERA", "JIMENEZ"};
    for (i = 0; i < 10; i++)
    {
        al = rand() % 87;
    }
    strcpy(apellido, ap[al]);
}
```

```
Treg agregar(Treg reg[], int i)
```

```
{
    int num, min = 18, max = 35;
    do
    {
        reg[i].kay = rand() % 100000 + 300000;
    } while (verMt(reg, i, reg[i].kay));
}
```

```

    } while (verMt(reg, i, reg[i].kay));

    num = rand() % 2 + 1;
    if (num == 1)
    {
        strcpy(reg[i].sex, "Masculino");
    }
    if (num == 2)
    {
        strcpy(reg[i].sex, "Femenino");
    }

    nombreAl(reg[i].nombre, num);

    apAl(reg[i].apP);
    apAl(reg[i].apM);

    reg[i].edad = rand() % (max - min + 1) + min;

    puestoAl(reg[i].puesto);

    return reg[i];
}

```

```

void puestoAl(char puesto[])
{
    int num;
    char pus[11][30]{
        "COMERCIAL",
        "TECNICO",
        "INGENIERO",
        "REPRESENTANTE",
        "INTENDENTE",
        "CONTADOR",
        "REPARTIDOR",
        "ADMINISTRADOR",
        "SECRETARIA",
        "AUXILIAR",
        "TELEFONISTA"};

    for (int i = 0; i < 11; i++)
    {
        num = rand() % 11;
    }
    strcpy(puesto, pus[num]);
}

```

```

int searchSec(Treg reg[], int n, int mt)
{
    int i;
    i = 0;
    for (i = 0; i <= n; i++)
    {
        if (reg[i].kay == mt)
        {
            return i;
        }
    }
    return -1;
}

```

```

int searchBin(Treg reg[], int inf, int sup, int mt)
{
    int med; // variable local
    while (inf <= sup)
    {
        med = (inf + sup) / 2;
        if (reg[med].kay == mt)
        {
            return med;
        }
        else
        {
            if (mt < reg[med].kay)
            {
                sup = med--;
            }
            else
            {
                inf = med++;
            }
        }
    }
    return -1;
}

```

```

int orderBu(Treg reg[], int n)
{
    int i, j;
    Treg temp;

    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (reg[j].kay < reg[i].kay)
            {
                temp = reg[i];
                reg[i] = reg[j];
                reg[j] = temp;
            }
        }
    }
    return 0;
}

```

```

void pintReg(Treg reg[], int n)
{
    int i = 0, j = 0, elec;

    do
    {
        i = 0;
        if (j < n)
        {
            system("CLS");
            printf("%-6s %-12s %-12s %-15s %-12s %-10s %-15s %s\n",
                "No.", "Matricula", "Nombre", "ApP", "ApM", "Edad", "Puesto", "Sexo");
            while (i < 40 && j < n)

```

```

while ((i < 40 && j < n))
{
    if (reg[j].status)
    {
        printf("%-6d.- %-10d %-10s %-15s %-13s %-8d %-15s %s\n",
            j + 1, reg[j].kay, reg[j].nombre, reg[j].apP,
            reg[j].apM, reg[i].edad, reg[j].puesto, reg[j].sex);
    }
    i++;
    j++;
}
elec = valid("1.-SI\n0.-NO\nSEGUIR IMPRIMIENDO: ", 0, 1);
}
else
{
    printf("LIMITE ALCANZADO\n");
    elec = 0;
    system("PAUSE");
}
} while (elec);
}

```

```

void pintOneReg(Treg reg[], int n)
{
    if (n != -1)
    {
        printf("MATRICULA: %d\nNOMBRE: %s \nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nEDAD: %d\nPUESTO: %s\nSEXO:%s",
            reg[n].kay, reg[n].nombre, reg[n].apP,
            reg[n].apM, reg[n].edad, reg[n].puesto, reg[n].sex);
    }
    else
    {
        printf("Registro no encontrado\n");
    }
}

```

```

int verMt(Treg reg[], int n, int mt)
{
    int i, j;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j <= i; j++)
        {
            if (reg[j].kay == mt)
            {
                return 1;
            }
        }
    }

    return 0;
}

```

```

void createTXT(Treg reg[], int n)
{
    FILE *fa;
    int i = 0;
    char namefile[30];

```



```

    }
    char namefile[30];

    validCad("Ingrese nombre del archivo: ", namefile);
    strcat(namefile, ".txt");
    fa = fopen(namefile, "w");
    fprintf(fa, "%-10s %-15s %-23s %-15s %-10s %-10s %-10s %s \n",
        "No.", "Matricula", "Nombre", "ApP", "ApM", "Edad", "Puesto", "Sexo");
    for (i = 0; i < n; i++)
    {
        if (reg[i].status != 0)
        {
            fprintf(fa, "%d.- %-10d %-10s %-15s %-10s %-8d %-10s %s\n",
                i + 1, reg[i].kay, reg[i].nombre, reg[i].apP,
                reg[i].apM, reg[i].edad, reg[i].puesto, reg[i].sex);
        }
    }
    fclose(fa);
}

void swap(Treg reg[], int i, int j)
{
    Treg temp = reg[i];
    reg[i] = reg[j];
    reg[j] = temp;
}

int partition(Treg reg[], int Low, int high)
{
    Treg pivot;
    pivot.kay = reg[high].kay;
    int i = Low - 1;

    for (int j = Low; j <= high - 1; j++)
    {
        if (reg[j].kay <= pivot.kay)
        {
            i++;
            swap(reg, i, j);
        }
    }
    swap(reg, i + 1, high);
    return i + 1;
}

void quicksort(Treg reg[], int Low, int high)
{
    if (Low < high)
    {
        int pi = partition(reg, Low, high);

        quicksort(reg, Low, pi - 1);
        quicksort(reg, pi + 1, high);
    }
}

void regEliminados(Treg reg[], int n)
{
    int i = 0, j = 0, elec;
}

```

```

do
{
    i = 0;
    if (j < n)
    {
        system("CLS");
        printf("%-6s %-12s %-12s %-15s %-12s %-10s %-15s %s\n",
            "No.", "Matricula", "Nombre", "ApP", "ApM", "Edad", "Puesto", "Sexo");
        while (i < 40 && j < n)
        {
            if (!reg[j].status)
            {
                printf("%-6d.- %-10d %-10s %-15s %-13s %-8d %-15s %s\n",
                    j + 1, reg[j].kay, reg[j].nombre, reg[j].apP,
                    reg[j].apM, reg[i].edad, reg[j].puesto, reg[j].sex);
            }
            i++;
            j++;
        }
        elec = valid("1.-SI\n0.-NO\nSEGUIR IMPRIMIENDO: ", 0, 1);
    }
    else
    {
        printf("LIMITE ALCANZADO\n");
        elec = 0;
    }
} while (elec);
system("PAUSE");
}

```

```

void modReg(Treg reg[], int n)
{
    int opc;
    printf("\nQUE CAMPO DESEA MODIFICAR?\n");
    printf("1.-NOMBRE\n");
    printf("2.-APELLIDO PATERNO\n");
    printf("3.-APELLIDO MATERNO\n");
    printf("4.-EDAD\n");
    printf("5.-PUESTO\n");
    printf("6.-SEXO\n");
    printf("0.-NINGUNO\n");
    opc = valid("INGRESE OPCION: ", 0, 6);

    switch (opc)
    {
        case 1:
            validCad("INGRESE NUEVO NOMBRE: ", reg[n].nombre);
            break;

        case 2:
            validCad("INGRESE NUEVO APELLIDO: ", reg[n].apP);
            break;

        case 3:
            validCad("INGRESE NUEVO APELLIDO: ", reg[n].apM);
            break;

        case 4:
            reg[n].edad = valid("INGRESE NUEVA EDAD", 18, 35);

```

```

        reg[n].edad = valid("INGRESE NUEVA EDAD", 18, 35);
        break;
    case 5:
        validCad("INGRESE PUESTO: ", reg[n].puesto);
        break;
    case 6:
        validCad("INGRESE SEXO: ", reg[n].sex);
        break;
}
}

```

```

void MostrarArchivo()
{
    FILE *fa;
    char namefile[30];
    char c;
    validCad("INGRESE NOMBRE DEL ARCHIVO: ", namefile);
    strcat(namefile, ".txt");
    fa = fopen(namefile, "r");
    if (fa == NULL)
    {
        printf("Error al abrir el archivo\n");
    }
    else
    {
        while ((c = fgetc(fa)) != EOF)
        {
            putchar(c);
        }
        fclose(fa);
    }
    system("PAUSE");
}

```

```

void archivoBin(Treg reg[], int n)
{
    FILE *fa;
    int i = 0;
    char namefile[30] = "Datos.dll";

    rename("Datos.dll", "Datos.tmp");

    fa = fopen(namefile, "wb");
    if (fa == NULL)
    {
        printf("Error al abrir el archivo\n");
    }
    else
    {
        for (i = 0; i < n; i++)
        {
            fwrite(&reg[i], sizeof(Treg), 1, fa);
        }
        fclose(fa);
        printf("Archivo creado\n");
        system("PAUSE");
    }
}

```

```

int cargarArchBin(Treg reg[], int n)
{
    FILE *fa;
    char namefile[30];
    Treg temp;
    int i = 0;
    validCad("Ingrese nombre del arvhivo: ", namefile);
    strcat(namefile, ".dll");
    fa = fopen(namefile, "rb");
    if (fa == NULL)
    {
        printf("Error al abrir el archivo\n");
        return -1;
    }
    else
    {
        while (fread(&temp, sizeof(Treg), 1, fa))
        {
            if (i < N)
            {
                reg[i] = temp;
                i++;
            }
        }
        printf("Archivo cargado\n");
        fclose(fa);
        return i;
    }
}

```

```

void BinArchEl()
{
    FILE *fa;
    char namefile[30];
    Treg Borrados[N], temp;
    int cont = 0;
    validCad("Ingrese nombre del arvhivo: ", namefile);
    strcat(namefile, ".dll");
    fa = fopen(namefile, "rb");
    if (!fa)
    {
        printf("Error al abrir el archivo\n");
        system("PAUSE");
    }
    else
    {
        while (fread(&temp, sizeof(Treg), 1, fa) == 1)
        {
            if (temp.status == 0)
            {
                Borrados[cont] = temp;
                cont++;
            }
        }
        regEliminados(Borrados, cont);
        fclose(fa);
    }
}

```

No.	Matricula	Nombre	ApP	ApM	Edad	Puesto	Sexo
1	.- 332527	VANESA	JIMENEZ	VALENZUELA	21	REPARTIDOR	Femenino
2	.- 331380	ALFREDO	GONZALEZ	MEDINA	33	TELEFONISTA	Masculino
3	.- 313155	MARTA	GOMEZ	GUZMAN	34	SECRETARIA	Femenino
4	.- 309095	RAFAEL	PEREDES	CORDERO	24	TECNICO	Masculino
5	.- 332462	PAULA	ROJAS	MORENO	33	TELEFONISTA	Femenino
6	.- 307064	CARLOS	GARCIA	RODRIGUEZ	31	AUXILIAR	Masculino
7	.- 319682	ERNESTO	CASTRO	MENDEZ	20	INGENIERO	Masculino
8	.- 314740	LOLA	MARTINEZ	VARGAS	26	ADMINISTRADOR	Femenino
9	.- 332009	EDUARDO	HERRERA	RIOS	27	CONTADOR	Masculino
10	.- 307389	SERGIO	RIOS	ROJAS	22	TECNICO	Masculino
11	.- 308188	MIRIAM	RUIZ	RAMIREZ	33	TELEFONISTA	Femenino
12	.- 321225	ANGEL	VARGAS	MARTINEZ	21	AUXILIAR	Masculino
13	.- 301356	MARTA	PEREZ	RAMIREZ	23	INTENDENTE	Femenino
14	.- 319648	ERNESTO	SALAZAR	RODRIGUEZ	22	REPRESENTANTE	Masculino
15	.- 306363	ANAIS	MENDEZ	FERNANDEZ	28	INGENIERO	Femenino
16	.- 309343	LAURA	BENITEZ	MEDINA	34	TECNICO	Femenino
17	.- 304194	ALICIA	JIMENEZ	ESCOBAR	18	COMERCIAL	Femenino
18	.- 313017	RODRIGO	GUZMAN	RODRIGUEZ	33	AUXILIAR	Masculino
19	.- 326859	PILAR	AGUIRRE	GONZALEZ	20	INGENIERO	Femenino
20	.- 329843	ARMANDO	FERNANDEZ	GONZALEZ	28	TECNICO	Masculino
21	.- 313397	OSCAR	BENITEZ	LOPEZ	24	COMERCIAL	Masculino
22	.- 316390	SANDRA	RODRIGUEZ	PEREZ	23	REPRESENTANTE	Femenino
23	.- 314037	EMILIO	MARTINEZ	ROMERO	19	REPRESENTANTE	Masculino
24	.- 319103	ANGEL	MUNOZ	GOMEZ	22	CONTADOR	Masculino
25	.- 321618	ADRIANA	ARAYA	ESCOBAR	22	REPRESENTANTE	Femenino
26	.- 332352	ARTURO	SANDOVAL	RAMIREZ	18	INTENDENTE	Masculino
27	.- 320102	MARINA	JIMENEZ	MORALES	24	TECNICO	Femenino
28	.- 308030	SANTIAGO	ARAYA	GUZMAN	32	AUXILIAR	Masculino
29	.- 320802	GLORIA	MUNOZ	RUIZ	35	INGENIERO	Femenino
30	.- 316706	PABLO	HERRERA	PEREZ	26	SECRETARIA	Masculino
31	.- 320777	ANDREA	MORALES	RODRIGUEZ	22	INGENIERO	Femenino
32	.- 303275	ANGEL	LOPEZ	MUNOZ	21	COMERCIAL	Masculino
33	.- 314481	RICARDO	HERRERA	TORRES	23	AUXILIAR	Masculino
34	.- 321832	EDUARDO	JIMENEZ	HERRERA	20	TELEFONISTA	Masculino
35	.- 325895	ALFREDO	PEREZ	RAMIREZ	19	COMERCIAL	Masculino
36	.- 332289	JAVIER	VARGAS	PIZARRO	28	INGENIERO	Masculino
37	.- 330058	SERGIO	TORRES	CORDERO	21	COMERCIAL	Masculino
38	.- 312365	RAFAELA	MEDINA	SALAS	32	CONTADOR	Femenino
39	.- 326158	FERNANDO	MENDOZA	MEDINA	23	TECNICO	Masculino
40	.- 324539	SANDRA	ACOSTA	TORRES	34	AUXILIAR	Femenino

MATRICULA: 332527

NOMBRE: VANESA

APELLIDO PATERNO: JIMENEZ

APELLIDO MATERNO: VALENZUELA

EDAD: 21

PUESTO: REPARTIDOR

SEXO:Femenino

QUE CAMPO DESEA MODIFICAR?

1.-NOMBRE

2.-APELLIDO PATERNO

3.-APELLIDO MATERNO

4.-EDAD

5.-PUESTO

6.-SEXO

0.-NINGUNO

INGRESE OPCION:



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

INGRESE NUEVO NOMBRE: ELENA

No.	Matricula	Nombre	ApP	ApM	Edad	Puesto	Sexo
1	.- 332527	ELENA	JIMENEZ	VALENZUELA	21	REPARTIDOR	Femenino
2	.- 331380	ALFREDO	GONZALEZ	MEDINA	33	TELEFONISTA	Masculino
3	.- 313155	MARTA	GOMEZ	GUZMAN	34	SECRETARIA	Femenino
4	.- 309095	RAFAEL	PAREDES	CORDERO	24	TECNICO	Masculino
5	.- 332462	PAULA	ROJAS	MORENO	33	TELEFONISTA	Femenino
6	.- 307064	CARLOS	GARCIA	RODRIGUEZ	31	AUXILIAR	Masculino
7	.- 319682	ERNESTO	CASTRO	MENDEZ	20	INGENIERO	Masculino
8	.- 314740	LOLA	MARTINEZ	VARGAS	26	ADMINISTRADOR	Femenino
9	.- 332009	EDUARDO	HERRERA	RIOS	27	CONTADOR	Masculino
10	.- 307389	SERGIO	RIOS	ROJAS	22	TECNICO	Masculino
11	.- 308188	MIRIAM	RUIZ	RAMIREZ	33	TELEFONISTA	Femenino
12	.- 321225	ANGEL	VARGAS	MARTINEZ	21	AUXILIAR	Masculino
13	.- 301356	MARTA	PEREZ	RAMIREZ	23	INTENDENTE	Femenino
14	.- 319648	ERNESTO	SALAZAR	RODRIGUEZ	22	REPRESENTANTE	Masculino
15	.- 306363	ANAIS	MENDEZ	FERNANDEZ	28	INGENIERO	Femenino
16	.- 309343	LAURA	BENITEZ	MEDINA	34	TECNICO	Femenino
17	.- 304194	ALICIA	JIMENEZ	ESCOBAR	18	COMERCIAL	Femenino
18	.- 313017	RODRIGO	GUZMAN	RODRIGUEZ	33	AUXILIAR	Masculino
19	.- 326859	PILAR	AGUIRRE	GONZALEZ	20	INGENIERO	Femenino
20	.- 329843	ARMANDO	FERNANDEZ	GONZALEZ	28	TECNICO	Masculino
21	.- 313397	OSCAR	BENITEZ	LOPEZ	24	COMERCIAL	Masculino
22	.- 316390	SANDRA	RODRIGUEZ	PEREZ	23	REPRESENTANTE	Femenino
23	.- 314037	EMILIO	MARTINEZ	ROMERO	19	REPRESENTANTE	Masculino
24	.- 319103	ANGEL	MUNOZ	GOMEZ	22	CONTADOR	Masculino
25	.- 321618	ADRIANA	ARAYA	ESCOBAR	22	REPRESENTANTE	Femenino
26	.- 332352	ARTURO	SANDOVAL	RAMIREZ	18	INTENDENTE	Masculino
27	.- 320102	MARINA	JIMENEZ	MORALES	24	TECNICO	Femenino
28	.- 308030	SANTIAGO	ARAYA	GUZMAN	32	AUXILIAR	Masculino
29	.- 320802	GLORIA	MUNOZ	RUIZ	35	INGENIERO	Femenino
30	.- 316706	PABLO	HERRERA	PEREZ	26	SECRETARIA	Masculino
31	.- 320777	ANDREA	MORALES	RODRIGUEZ	22	INGENIERO	Femenino
32	.- 303275	ANGEL	LOPEZ	MUNOZ	21	COMERCIAL	Masculino
33	.- 314481	RICARDO	HERRERA	TORRES	23	AUXILIAR	Masculino
34	.- 321832	EDUARDO	JIMENEZ	HERRERA	20	TELEFONISTA	Masculino
35	.- 325895	ALFREDO	PEREZ	RAMIREZ	19	COMERCIAL	Masculino
36	.- 332289	JAVIER	VARGAS	PIZARRO	28	INGENIERO	Masculino
37	.- 330058	SERGIO	TORRES	CORDERO	21	COMERCIAL	Masculino
38	.- 312365	RAFAELA	MEDINA	SALAS	32	CONTADOR	Femenino
39	.- 326158	FERNANDO	MENDOZA	MEDINA	23	TECNICO	Masculino
40	.- 324539	SANDRA	ACOSTA	TORRES	34	AUXILIAR	Femenino

1.-SI

0.-NO

SEGUIR IMPRIMIENDO:



MATRICULA: 332527

NOMBRE: ELENA

APELLIDO PATERNO: JIMENEZ

APELLIDO MATERNO: VALENZUELA

EDAD: 21

PUESTO: REPARTIDOR

SEXO: Femenino

DESEA ELIMINAR REGISTRO

1.-Si

0.-NO





# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

Actividad\_13

output

Datos.dll M

RGA\_Act1... M

Babilonia.h

RGA\_Act... 1, M

prub

603

604

605

606

validCad("Ingrese nombre del arvhivo

strcat(namefile, ".dll");

fa = fopen(namefile, "rb");

if (fopen(namefile, "rb") != NULL)

PROBLEMS 1

OUTPUT

DEBUG CONSOLE

TERMINAL

Archivo creado

Presione una tecla para continuar . . .



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

PROBLEMS **1** OUTPUT DEBUG CONSOLE TERMINAL PORTS

No.	Matricula	Nombre	ApP	ApM	Edad	Puesto	Sexo
1	.- 332527	ELENA	JIMENEZ	VALENZUELA	21	REPARTIDOR	Femenino

1.-SI  
0.-NO  
SEGUIR IMPRIMIENDO: █