

# **Ingeniero en Software y tecnologías emergentes**

**Materia:** Programación Estructurada / Clave 36276

**Alumno:** Solano Meza Angel Daniel

**Matrícula:** 372453

**Maestro:** Pedro Núñez Yépiz

**Actividad No.** : 13

**Tema-Unidad :** Archivos Indexados

**Ensenada Baja California a 28 de Noviembre del 2023**

```
File Edit Selection View Go Run ... Repositorio
C ADSM_ACT14.c 5 C junior.h 4
Actividad14 > C ADSM_ACT14.c > ...
1 // Solano Meza Angel Daniel Matr. 372453
2 // 28/11/2023
3 // Uso de archivos binarios
4 // ADSM_ACT14_932
5
6 #define N 5000
7 #define AUTO 100
8 #include "junior.h"
9
10 /*** PROTOTIPOS DE FUNCIONES *****/
11 int msges();
12 void menu();
13 void NombreArch(char archivo[]);
14 int ContarReg(char nombre[]);
15 int CrearIndices(TIndex vect[], int *i);
16 TReg AgregarAuto(void);
17 void RespaldoBIN(char nom[]);
18 TKey Busquedas(TIndex vect[], int n, int i, int *ordenado);
19 void AgregarRegistro(TIndex vect[], int *i, int ordenado);
20 void EliminarRegistro(TIndex vect[], int *i, int ordenado);
21 void BuscarRegistro(TIndex vect[], int *i, int ordenado);
22 void MostrarNormal(void);
23 void MostrarOrdenado(int i, int ordenado, TIndex vect[], int movimientos);
24 void GenerarOrdenado(char nom[], int ordenado, TIndex vect[], int movimientos, int i);
25 void GenerarNormal(char nom[]);
26 void Empaquetar();
27
28 > int main()...
34
35 int msges()
36 {
37     int op;
```

```
int msges()
{
    int op;
    system("CLS");
    printf(" M E N U \n");
    printf("1.- AGREGAR \n");
    printf("2.- ELIMINAR \n");
    printf("3.- BUSCAR \n");
    printf("4.- ORDENAR \n");
    printf("5.- IMPRIMIR REGISTROS ARCHIVO ORIGINAL \n");
    printf("6.- IMPRIMIR REGISTROS ARCHIVO ORDENADO \n");
    printf("7.- GENERAR ARCHIVO TXT ORDENADO\n");
    printf("8.- GENERAR ARCHIVO TXT NORMAL\n");
    printf("9.- EMPAQUETAR \n");
    printf("0.- SALIR \n");
    printf("ESCOGE UNA OPCION: ");
    scanf("%d", &op);
    return op;
}

void menu()
{
    srand(time(NULL));
    int op, n, i = 0, ordenado = 0, movimientos = 0;
    char nom[50];
    n = ContarReg("datos");
    n *= 1.25;
    TIndex IndexReg[n];
    CrearIndices(IndexReg, &i);
    RespaldoBIN("datos");
    do
```

```

op = msges();
switch (op)
{
case 1: // AGREGAR
    system("CLS");
    AgregarRegistro(IndexReg, &i, ordenado);
    movimientos++;
    ordenado = 0;
    system("PAUSE");
    break;
case 2: // EDITAR REGISTRO
    system("CLS");
    EliminarRegistro(IndexReg, &i, ordenado);
    system("PAUSE");
    break;
case 3: // BUSCAR REGISTRO
    system("CLS");
    BuscarRegistro(IndexReg, &i, ordenado);
    system("PAUSE");
    break;
case 4: // ORDENAR REGISTROS
    system("CLS");
    if (ordenado == 0)
    {
        printf("INDICES ORDENADOS\n");
        ordenado = 1;
        movimientos = 0;
    }
}

```

```

else
{
    printf("INDICES YA ORDENADOS\n");
}
system("PAUSE");
break;
case 5: // ORDENAR REGISTROS
    system("CLS");
    MostrarNormal();
    system("PAUSE");
    break;
case 6: // IMPRIMIR REGISTROS
    system("CLS");
    MostrarOrdenado(i, ordenado, IndexReg, movimientos);
    system("PAUSE");
    break;
case 7: // GENERAR TXT
    system("CLS");
    GenerarOrdenado(nom, ordenado, IndexReg, movimientos, i);
    system("PAUSE");
    break;
case 8: // MOSTRAR ARCHIVO TXT
    system("CLS");
    GenerarNormal(nom);
    system("PAUSE");
    break;
case 9:
    Empaquetar();
    break;

```

```
void NombreArch(char archivo[])
{
    system("CLS");
    printf("NOMBRE DEL ARCHIVO SIN EXTENSION: \n");
    fflush(stdin);
    gets(archivo);
}
```

```
TReg AgregarAuto(void)
{
    TReg trabajador;
    int sexo = 0, cuatro, tres, telefono;
    char puestos[][11] = {"GERENTE", "TECNICO", "EMPLEADO", "ASISTENTE", "TECNICO", "ANALISTA"};
    char NombresHombre[][15] = {"", "ALBERTO", "FERNANDO", "MARIO", "ALEJANDRO", "GABRIEL", "SERGIO", "CARLOS", "HECTOR", "LUIS", "ENRIQUE", "
char NombresMujer[][15] = {"", "LUCIA", "RAQUEL", "NATALIA", "BEATRIZ", "LOURDES", "CLARA", "LINDA", "GLORIA", "ADRIANA", "VALERIA", "ALIC
char Apellidos[][30] = {"HARO", "SOLANO", "SANDEZ", "LOPEZ", "MEZA", "PEREZ", "SANCHEZ", "FERNANDEZ", "TORRES", "RAMIREZ", "RUIZ", "GONZALE
char Estados[33][30] = {"AGUASCALIENTES", "BAJA CALIFORNIA", "BAJA CALIFORNIA SUR", "CAMPECHE", "CHIAPAS", "CHIHUAHUA", "COAHUILA", "COLIMA
char sexos[2][8] = {"HOMBRE", "MUJER"};
```

```

char sexos[2][8] = {"HOMBRE", "MUJER"};
sexo = NumAleatorio(0, 1);
if (sexo == 1)
{
    strcpy(trabajador.name, NombresHombre[NumAleatorio(1, 29)]);
    strcpy(trabajador.LastName1, Apellidos[NumAleatorio(0, 88)]);
    strcpy(trabajador.LastName2, Apellidos[NumAleatorio(0, 88)]);
    strcpy(trabajador.sex, sexos[0]);
}
else
{
    strcpy(trabajador.name, NombresMujer[NumAleatorio(1, 29)]);
    strcpy(trabajador.LastName1, Apellidos[NumAleatorio(0, 88)]);
    strcpy(trabajador.LastName2, Apellidos[NumAleatorio(0, 88)]);
    strcpy(trabajador.sex, sexos[1]);
}
strcpy(trabajador.JobPosition, puestos[NumAleatorio(0, 5)]);
strcpy(trabajador.State, Estados[NumAleatorio(0, 31)]);

cuatro = rand() % 100 * 1000;
tres = rand() % 1000;

telefono = 1000000 + cuatro + tres;
trabajador.cellPhone = telefono;

trabajador.age = NumAleatorio(18, 65);
trabajador.enrollment = NumAleatorio(300000, 399999);

```

```

    trabajador.status = 1;

    return trabajador;
}

void RespaldoBIN(char nom[])
{
    char archivo1[30], archivo2[30];
    FILE *fa;
    FILE *respaldo;
    TReg reg;
    strcpy(archivo1, nom);
    strcat(archivo1, ".dat");
    fa = fopen(archivo1, "rb");
    if (fa)
    {
        strcpy(archivo2, nom);
        strcat(archivo2, ".tmp");
        respaldo = fopen(archivo2, "wb");
        while (fread(&reg, sizeof(TReg), 1, fa))
        {
            fwrite(&reg, sizeof(TReg), 1, respaldo);
        }
        fclose(fa);
        fclose(respaldo);
    }
}

```

> void AgregarRegistro(TIndex vect[], int \*i, int ordenado) ...

```

void AgregarRegistro(TIndex vect[], int *i, int ordenado)
{
    TReg reg;
    FILE *fa;
    reg = AgregarAuto();
    while (Busquedas(vect, *i, reg.enrollment, ordenado) != -1)
    {
        reg.enrollment = NumAleatorio(300000, 399999);
    }
    fa = fopen("datos.dat", "a+b");
    fseek(fa, 0, SEEK_END);
    fwrite(&reg, sizeof(TReg), 1, fa);
    fclose(fa);

    vect[*i].index = *i;
    vect[*i].enrollment = reg.enrollment;
    (*i)++;
    printf("REGISTRO AGREGADO\n");
}

TKey Busquedas(TIndex vect[], int n, int i, int *ordenado)
{
    TKey j;

    if (ordenado)
    {
        j = BusquedaBinaria(vect, 0, i, n);
    }
    else

```

```

        j = BusquedaBinaria(vect, 0, i, n);
    }
    else
    {
        j = BusquedaTIndex(vect, n, i);
    }
    return j;
}

void EliminarRegistro(TIndex vect[], int *i, int ordenado)
{
    int eliminar, encontrado, op;
    TReg reg;
    FILE *fa;
    printf("NO. DE EMPLEADO A ELIMINAR: ");
    eliminar = Validar(300000, 399999);
    encontrado = Busquedas(vect, *i, eliminar, ordenado);
    if (encontrado != -1)
    {
        fa = fopen("datos.dat", "rb");
        fseek(fa, sizeof(TReg) * vect[encontrado].index, SEEK_SET);
        fread(&reg, sizeof(TReg), 1, fa);
        fclose(fa);
        if (reg.status == 0)
        {
            printf("MATRICULA YA ELIMINADA\n");
        }
    }
}

```

```

    else
    {
        printf("NOMBRE: %s\nAPELLIDOS: %s %s\nSEXO: %s \nPUESTO: %s \nESTADO: %s\nEDAD: %d\nTELEFONO: %d\n", reg.name, reg.LastName1, reg.
        printf("ELIMINAR\n1-SI\n2-NO\nSELECCIONE UNA OPCION: ");
        op = Validar(1, 2);
        system("CLS");
        if (op == 1)
        {
            fa = fopen("datos.dat", "r+b");
            if (fa)
            {
                fseek(fa, sizeof(TReg) * vect[encontrado].index, SEEK_SET);
                fread(&reg, sizeof(TReg), 1, fa);
                reg.status = 0;
                fseek(fa, sizeof(TReg) * vect[encontrado].index, SEEK_SET);
                fwrite(&reg, sizeof(TReg), 1, fa);
                printf("REGISTRO ELIMINADO\n");
                fclose(fa);
            }
            else
            {
                printf("ERROR, NO ABIERTO\n");
            }
        }
        else
        {
            printf("REGISTRO NO ELIMINADO\n");
        }
    }
}

```

```

    }
}
else
{
    printf("MATRICULA NO ENCONTRADA\n");
}

void BuscarRegistro(TIndex vect[], int *i, int ordenado)
{
    int buscar, encontrado;
    TReg reg;
    FILE *fa;
    printf("NO. DE EMPLEADO A BUSCAR: ");
    buscar = Validar(300000, 399999);
    encontrado = Busquedas(vect, *i, buscar, ordenado);
    if (encontrado != -1)
    {
        fa = fopen("datos.dat", "rb");
        fseek(fa, sizeof(TReg) * vect[encontrado].index, SEEK_SET);
        fread(&reg, sizeof(TReg), 1, fa);
        fclose(fa);
        if (reg.status == 0)
        {
            printf("MATRICULA ELIMINADA\n");
        }
        else
        {

```

```

        else
        {
            printf("INDICE: %d\nNOMBRE: %s\nAPELLIDOS: %s %s\nSEXO: %s \nPUESTO: %s \nESTADO: %s\nEDAD: %d\nTELEFONO: %d\n", encontrado, reg.name, reg.LastName1, reg.LastName2, reg.age, reg.sex, reg.job, reg.status, reg.age, reg.phone);
        }
    }
    else
    {
        printf("MATRICULA NO ENCONTRADA\n");
    }
}

int Ordenacion(TIndex vect[], int i, int movimientos)
{
    if (movimientos == 0)
    {
        Quicksort(vect, 0, i - 1);
    }
    else
    {
        if (movimientos < 3)
        {
            Burbuja(vect, i);
        }
        else
        {
            Quicksort(vect, 0, i - 1);
        }
    }
}

return 1;

```

```

void MostrarNormal(void)
{
    FILE *fa;
    TReg reg;
    int j = 0;
    fa = fopen("datos.dat", "rb");
    while (fread(&reg, sizeof(TReg), 1, fa))
    {
        printf("%-9d %-11d %-15s %-20s %-17s %-7d %-13s %-18s %-13s %d\n", j++, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.age, reg.sex, reg.job, reg.status, reg.phone);
    }
    fclose(fa);
}

void MostrarOrdenado(int i, int ordenado, TIndex vect[], int movimientos)
{
    FILE *fa;
    TReg reg;
    if (ordenado == 0)
    {
        Ordenacion(vect, i, movimientos);
    }
    fa = fopen("datos.dat", "rb");
    if (fa)
    {
        for (int j = 0; j < i; j++)
        {
            fseek(fa, vect[j].index * sizeof(TReg), SEEK_SET);
            fread(&reg, sizeof(TReg), 1, fa);
            printf("%-9d %-11d %-15s %-20s %-17s %-7d %-13s %-18s %-13s %d\n", vect[j].index, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.age, reg.sex, reg.job, reg.status, reg.phone);
        }
        fclose(fa);
    }
}

```

```

if (fa)
{
    for (int j = 0; j < i; j++)
    {
        fseek(fa, vect[j].index * sizeof(TReg), SEEK_SET);
        fread(&reg, sizeof(TReg), 1, fa);
        printf("%-9d %-11d %-15s %-20s %-17s %-7d %-13s %-18s %-13s %d\n", vect[j].index, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.age, reg.sex, reg.job, reg.status, reg.phone);
    }
    fclose(fa);
}

void GenerarOrdenado(char nom[], int ordenado, TIndex vect[], int movimientos, int i)
{
    FILE *fa, *bin;
    TReg reg;
    NombreArch(nom);
    strcat(nom, ".txt");
    if (ordenado == 0)
    {
        Ordenacion(vect, i, movimientos);
    }
    bin = fopen("datos.dat", "rb");
    fa = fopen(nom, "w");
    for (int j = 0; j < i; j++)
    {
        fseek(bin, vect[j].index * sizeof(TReg), SEEK_SET);
        fread(&reg, sizeof(TReg), 1, bin);
        if (reg.status == 1)
        {
            fprintf(fa, "%-9d %-11d %-15s %-20s %-17s %-7d %-13s %-18s %-13s %d\n", vect[j].index, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.age, reg.sex, reg.job, reg.status, reg.phone);
        }
    }
    fclose(fa);
}

```



```

        if (reg.status == 1)
        {
            fprintf(fa, "%10d || %10d || %9s || %16s || %16s || %12s || %16s || %16s || %4d || %10d\n", vect[j].index, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.status, reg.age, reg.phone, reg.email, reg.index);
        }
    }
    fclose(fa);
    fclose(bin);
}

void GenerarNormal(char nom[])
{
    FILE *fa, *bin;
    TReg reg;
    int j = 0;
    NombreArch(nom);
    strcat(nom, ".txt");
    bin = fopen("datos.dat", "rb");
    fa = fopen(nom, "w");

    while (fread(&reg, sizeof(TReg), 1, bin))
    {
        if (reg.status == 1)
        {
            fprintf(fa, "%-10d || %10d || %9s || %16s || %16s || %12s || %16s || %16s || %4d || %10d\n", j++, reg.enrollment, reg.name, reg.LastName1, reg.LastName2, reg.status, reg.age, reg.phone, reg.email, reg.index);
        }
    }

    fclose(fa);
}

```

```

    fclose(bin);
}

void Empaquetar()
{
    FILE *doc = fopen("datos.dat", "rb");
    FILE *fa = fopen("datos.bak", "wb");

    TReg reg;

    if (doc)
    {
        while (fread(&reg, sizeof(TReg), 1, doc))
        {
            if (reg.status == 1)
            {
                fwrite(&reg, sizeof(TReg), 1, fa);
            }
        }

        fclose(doc);
        fclose(fa);
    }
}

```

REGISTRO AGREGADO

Presione una tecla para continuar . . . █

NO. DE EMPLEADO A ELIMINAR: 330295

NOMBRE: JOSE

APELLIDOS: CONTRERAS CASTILLO

SEXO: HOMBRE

PUESTO: AnlSist

ESTADO: YN

EDAD: 46

TELEFONO: 1025268

ELIMINAR

1-SI

2-NO

SELECCIONE UNA OPCION: █

NO. DE EMPLEADO A BUSCAR: 330295

INDICE: 3350

NOMBRE: JOSE

APELLIDOS: CONTRERAS CASTILLO

SEXO: HOMBRE

PUESTO: AnlSist

ESTADO: YN

EDAD: 46

TELEFONO: 1025268

Presione una tecla para continuar . . . █

INDICES ORDENADOS

Presione una tecla para continuar . . . █

INDICES YA ORDENADOS

Presione una tecla para continuar . . . █

3340	305968	JAVIER	MUNGUIA	AGUILAR	21	HOMBRE	Arquitect	TS
3341	315561	SUSANA	MACIAS	OSORIO	50	MUJER	AsistAdm	BC
3342	324147	LAURA	ZAMORA	CABRERA	38	MUJER	DisGraf	GT
3343	304142	LAURA	CONTRERAS	MEDINA	19	MUJER	ProffPrim	SP
3344	323053	HECTOR	DIAZ	ALVARADO	45	HOMBRE	IngElec	CS
3345	309775	LAURA	GOMEZ	MEDINA	28	MUJER	EspRRHH	MS
3346	318319	HECTOR	MOLINA	AGUILAR	29	HOMBRE	AnMktDg	SP
3347	321778	ELENA	MORA	JIMENEZ	19	MUJER	ConsFin	DF
3348	317557	GABRIELA	ESTRELLA	CRUZ	46	MUJER	PsOrg	ZS
3349	321101	MARTA	DURAN	VARGAS	39	MUJER	IngElec	HG
3350	330295	JOSE	CONTRERAS	CASTILLO	46	HOMBRE	AnlSist	YN
3351	327050	PILAR	LARA	ARIAS	38	MUJER	Traduct	NE
3352	311893	ELENA	ORTIZ	MACIEL	49	MUJER	Traduct	SL
3353	321005	CARMEN	CONTRERAS	MENDOZA	25	MUJER	IngRed	BC
3354	331050	AURORA	HERRERA	VALENCIA	34	MUJER	EspLog	SR
3355	328851	ENRIQUE	MACIEL	PARDO	47	HOMBRE	RepVInt	HG
3356	311976	SUSANA	CAMACHO	ROJAS	25	MUJER	PsOrg	SR
3357	328201	ARMANDO	PARDO	FLORES	45	HOMBRE	Arquitect	MN
3358	308728	ALEJANDRA	ESTRELLA	VALENZUELA	23	MUJER	TrabSoc	OC
3359	310512	RAQUEL	DURAN	IGLESIAS	22	MUJER	AsServ	TS
3360	321701	CARLOS	LUGO	SALAZAR	53	HOMBRE	TECNICO	DURANGO
3361	324074	MARTA	TORRES	REYES	41	MUJER	ASISTENTE	SINALOA

Presione una tecla para continuar █

40	332514	ENRIQUE	CAMACHO	RAMIREZ	47	HOMBRE	IngRed	GR	1031960
2484	332530	MARTA	NUNEZ	CORDERO	38	MUJER	InvCien	DG	1005691
1927	332543	SALVADOR	MOLINA	VARGAS	48	HOMBRE	ChefEje	CM	1025107
3327	332561	JUAN	CHACON	RIOS	38	HOMBRE	EspRRHH	TS	1013784
1236	332572	CONSUELO	ESCOBAR	CABRERA	32	MUJER	DisUX	ZS	1018491
2386	332574	GLORIA	VASQUEZ	MUNGUIA	48	MUJER	DevSoft	OC	1000328
1107	332592	ROBERTO	TOVAR	REYES	26	HOMBRE	PsOrg	BS	1001674
1984	332617	MARIO	ORTIZ	MEDINA	21	HOMBRE	ChefEje	AG	1011038
1636	332620	MIGUEL	ALVAREZ	URIBE	30	HOMBRE	DisGraf	CC	1024358
1151	332626	SALVADOR	LARA	SOLIS	23	HOMBRE	EspSeg	CS	1002437
712	332679	MERCEDES	PEREDES	BELTRAN	46	MUJER	GteProj	JC	1011613
1132	332686	MERCEDES	SUAREZ	CAMACHO	22	MUJER	TecSup	TL	1010523
29	332687	JOSE	CASTANEDA	VASQUEZ	22	HOMBRE	InvCien	ZS	1008462
2465	332692	ARTURO	GONZALES	TORRES	31	HOMBRE	Traduct	OC	1007788
505	332698	PAULA	ESPINOSA	ZAMORA	30	MUJER	TecSup	NT	1016706
1962	332706	PILAR	ROJAS	ARIAS	38	MUJER	Contado	MN	1025346
1048	332720	CARMEN	ALVARADO	HIDALGO	44	MUJER	AsServ	GT	1014616
944	332723	PILAR	LEON	VASQUEZ	35	MUJER	Abogado	CC	1006386
431	332727	PEDRO	TOVAR	TORRES	48	HOMBRE	ChefEje	HG	1000943
1565	332732	SILVIA	CASILLAS	TORRES	31	MUJER	IngElec	SP	1032505
2121	332757	PEDRO	QUINTERO	VALDES	44	HOMBRE	EspLog	JC	1023786
1090	332761	MERCEDES	TORRES	FLORES	49	MUJER	EspSeg	SP	1004974

Presione una tecla para continuar

1	2101	300019	MIGUEL	HIDALGO	FLORES	HOMBRE	InvCien	21	1010529
2	3308	300044	PAULA	ROSALES	HIDALGO	MUJER	EspLog	22	1010870
3	3287	300046	ALEJANDRA	REYES	DURAN	MUJER	TecSup	23	1005412
4	1386	300077	CARLOS	MORA	VALENCIA	HOMBRE	Arquitect	43	1006753
5	31	300086	JOSE	GOMEZ	MACIAS	HOMBRE	IngElec	40	1017883
6	2950	300087	BEATRIZ	AGUILAR	VALENCIA	MUJER	Abogado	37	1027023
7	3008	300092	SOFIA	ROMAN	ALVAREZ	MUJER	EspSeg	41	1028675
8	1586	300112	MIGUEL	PACHECO	VEGA	HOMBRE	AnlVent	49	1015089
9	1640	300122	RAUL	SERRANO	BARRIOS	HOMBRE	GteProj	44	1029062
10	2953	300133	ROSA	LOPEZ	ESPINOZA	MUJER	ProfPrim	46	1011987
11	950	300142	ANA	CERVANTES	DELGADO	MUJER	ChefEje	49	1007730
12	2920	300148	MARIO	URIBE	VARGAS	HOMBRE	DevSoft	32	1007812
13	2984	300158	SOFIA	LEON	VALDES	MUJER	Contado	26	1030988
14	2194	300171	MARIO	SERRANO	RAMOS	HOMBRE	TrabSoc	28	1016729
15	782	300178	MARIO	PARDO	CORDERO	HOMBRE	AnMktDg	26	1010069
16	2339	300182	GLORIA	LEAL	PARDO	MUJER	PsOrg	28	1007186
17	2257	300187	ARMANDO	CONTRERAS	GONZALES	HOMBRE	AnlSist	50	1020450
18	2709	300197	CLARA	GUERRERO	ROJAS	MUJER	DisUX	29	1029019
19	80	300209	HECTOR	SILVA	DURAN	HOMBRE	EspLog	38	1024691
20	351	300224	ALEJANDRO	IGLESIAS	SUAREZ	HOMBRE	PsOrg	26	1025510
21	1960	300244	SALVADOR	RAMOS	CHACON	HOMBRE	Traduct	44	1007814
22	2360	300245	GABRIELA	ORTIZ	ROMERO	MUJER	EspRRHH	48	1015364
23	3015	300250	PILAR	HERRERA	ALVARADO	MUJER	TrabSoc	35	1010009
24	1865	300271	PATRICIA	ESPINOZA	REYES	MUJER	DisGraf	43	1006922
25	2940	300297	NATALIA	SANDOVAL	LEAL	MUJER	EspRRHH	44	1018431
26	2466	300310	ROBERTO	SALAZAR	URIBE	HOMBRE	EspRRHH	42	1029569
27	611	300311	MARIO	CERVANTES	BAUTISTA	HOMBRE	RepVInt	21	1010044
28	1192	300319	ROBERTO	OSORIO	ALVAREZ	HOMBRE	Arquitect	33	1001038
29	2650	300323	MARIA	LEAL	MENENDEZ	MUJER	ProfPrim	19	1030996
30	3129	300326	ARMANDO	TORRES	RIVERA	HOMBRE	IngRed	21	1030595
31	146	300347	JORGE	GOMEZ	SOTO	HOMBRE	AsistAdm	45	1013896
32	296	300349	EDUARDO	SOSA	ARIAS	HOMBRE	ConsFin	34	1012779

1	0		331399	ALEJANDRO	IGLESIAS	ARIAS	HOMBRE	RepVInt	BC
2	1		328162	ALBERTO	CASTILLO	LEON	HOMBRE	GteProj	OC
3	2		321166	MIGUEL	MERCADO	MALDONADO	HOMBRE	DisUX	CH
4	3		320571	CLARA	GARZA	CRUZ	MUJER	MedGen	CL
5	4		308286	CONSUELO	VASQUEZ	GUERRERO	MUJER	ProfPrim	ZS
6	5		314627	MARIA	MENDOZA	SANDOVAL	MUJER	InvCien	BC
7	6		306881	RAUL	TOVAR	PENA	HOMBRE	DevSoft	NT
8	7		322504	ALEJANDRO	ROJAS	ROJAS	HOMBRE	Traduct	DG
9	8		329950	JOSE	TORRES	ROMAN	HOMBRE	TrabSoc	YN
10	9		310655	PILAR	GARZA	FLORES	MUJER	EspRRHH	NL
11	10		310480	JORGE	DURAN	VILLANUEVA	HOMBRE	DevSoft	VZ
12	11		319891	MARTA	GUZMAN	RUBIO	MUJER	Contado	QT
13	12		331486	JAVIER	LARA	SUAREZ	HOMBRE	EspLog	MS
14	13		332448	SUSANA	CAMACHO	MACIEL	MUJER	DisGraf	CS
15	14		314814	SUSANA	CASTANEDA	SOLANO	MUJER	AnlSist	JC
16	15		322393	LAURA	PACHECO	CASILLAS	MUJER	DisUX	AG
17	16		301488	JOSE	ESPINOSA	CASTANEDA	HOMBRE	DisUX	CS
18	17		314993	MANUEL	AGUILAR	ESPINOSA	HOMBRE	Enferm	BS
19	18		327953	SALVADOR	LEON	ROLDAN	HOMBRE	AnlVent	JC
20	19		309371	ROSA	ALVAREZ	HERRERA	MUJER	AsServ	YN
21	20		308765	SALVADOR	AGUILAR	GONZALES	HOMBRE	AnlDatos	MC
22	21		328983	VICTORIA	TORRES	VEGA	MUJER	EspRRHH	MC
23	22		320227	ISRAEL	CAMACHO	ARIAS	HOMBRE	IngRed	CS
24	23		316658	SALVADOR	PARDO	ESTRELLA	HOMBRE	AnMktDg	NE
25	24		326642	EDUARDO	IGLESIAS	GOMEZ	HOMBRE	Traduct	MC
26	25		317533	MARIA	LOPEZ	MACIAS	MUJER	AsServ	HG
27	26		309080	SUSANA	IGLESIAS	MORA	MUJER	Traduct	NE
28	27		318084	BEATRIZ	GUERRERO	PENA	MUJER	IngRed	OC
29	28		314588	MERCEDES	CASILLAS	ESPINOZA	MUJER	AsistAdm	NL
30	29		332687	JOSE	CASTANEDA	VASQUEZ	HOMBRE	InvCien	ZS
31	30		306592	VICTORIA	SOLIS	CERVANTES	MUJER	RepVInt	SR
32	31		300086	JOSE	GOMEZ	MACIAS	HOMBRE	IngElec	YN

☰	datos.bak	M
☰	datos.dat	M
🕒	datos.tmp	M