

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes
Programación Estructurada

Actividad 14. Archivos Indexados

ALUMNO: Fernando Haro Calvo

MATRICULA: 372106

GRUPO: 932

PROFESOR: Pedro Núñez Yépiz

28 de noviembre del 2023

Funciones Utilizadas:

```
/** PROTOTIPOS DE FUNCIONES *****/
int msges(void);
void menu(void);

// Auxiliares
void leerNomArch(char nomArchivo[]);
Twkr genPersAlea(void);
void imprArchBin(Tindice indice[], int n, int ordenado);

// Registros
void imprReg(Twkr pers);
void agregarReg(Tindice indice[], int *n, int *ordenado);
void eliminarReg(Tindice indice[], int *n, int ordenado);
void buscarReg(Tindice indice[], int n, int ordenado);
void ordenarReg(Tindice indice[], int n, int *ordenado);

// Archivos
void escrArchTXT(char nomArchivo[], Tindice vect[], int n, int ordenado);

// Archivos binarios
void empaquetar(Tindice indice[], int *n);
bool cargarIndice(Tindice vect[], int *n);
int contarRegArch(char nomArchivo[]);
```

Menú:

```
int msges()
{
    int op;
    system("CLS");
    printf("\n  M  E  N  U  \n");
    printf("1.- AGREGAR REGISTRO \n");
    printf("2.- ELIMINAR REGISTRO \n");
    printf("3.- BUSCAR REGISTRO \n");
    printf("4.- ORDENAR INDICE \n");
    printf("5.- IMPRIMIR ARCHIVO \n");
    printf("6.- IMPRIMIR INDICE \n");
    printf("7.- GENERAR ARCHIVO TEXTO \n");
    printf("8.- EMPAQUETAR \n");
    printf("9.- SALIR  \n");
    printf("ESCOGE UNA OPCION: ");
    op = valiNum(1, 9);
    return op;
}
```

Ejercicio 1:

```
case 1:
    agregarReg(indice, &nPers, &ordenado);
    break;
```

```
// Agrega un registro al vector de indices y al final del archivo binario.
void agregarReg(Tindice indice[], int *n, int *ordenado)
{
    Twrkr reg;
    FILE *fa;

    reg = genPersAlea();

    while (busqOpt(indice, *n, reg.enrollment, *ordenado) != -1)
    {
        reg.enrollment = matriAlea();
    }

    fa = fopen("datos.dat", "ab");
    fwrite(&reg, sizeof(Twrkr), 1, fa);
    fclose(fa);

    indice[*n].key = reg.enrollment;
    indice[*n].indice = *n;
    (*n)++;

    printf("Registro agregado con exito\n");

    if (*ordenado == 1)
    {
        *ordenado = 2;
    }
}
```

Salida 1:

Registro agregado con exito

Presione una tecla para continuar . . .

Registros 3322 - 3361

No	MATRICULA	TELEFONO	ESTADO	PUESTO	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
3320.-	331660	1020699	HG	Contado	PILAR	MALDONADO	DELGADO	43	MUJER
3321.-	326580	1021718	NL	InvCien	GLORIA	DIAZ	VALENZUELA	20	MUJER
3322.-	324056	1022268	GR	IngRed	ARTURO	MERCADO	SOLANO	34	HOMBRE
3323.-	314132	1004155	TC	AnlDatos	MANUEL	ORTIZ	CASTILLO	46	HOMBRE
3324.-	320138	1029683	BC	DisUX	RAUL	CERVANTES	CORDERO	31	HOMBRE
3325.-	328746	1005355	HG	AnlSist	GLORIA	ARIAS	SUAREZ	26	MUJER
3326.-	304141	1001984	QT	PsOrg	LAURA	PENA	MENDEZ	20	MUJER
3327.-	332561	1013784	TS	EspRRHH	JUAN	CHACON	RIOS	38	HOMBRE
3328.-	313703	1013028	TS	PsOrg	PATRICIA	VARGAS	ESPINOZA	43	MUJER
3329.-	301003	1013119	SP	TecSup	MARIA	NUNEZ	CASILLAS	49	MUJER
3330.-	317589	1000441	NL	RepVInt	JORGE	ESPINOZA	HERNANDEZ	32	HOMBRE
3331.-	317394	1008270	SP	AnMktDg	ISABEL	SANDOVAL	SOSA	36	MUJER
3332.-	306770	1006328	MC	ProfPrim	NATALIA	MACIAS	CASTANEDA	40	MUJER
3333.-	319845	1024138	MN	TrabSoc	ALEJANDRO	MOLINA	VEGA	18	HOMBRE
3334.-	303094	1009847	TS	IngRed	SONIA	RIVERA	RIOS	49	MUJER
3335.-	329717	1031145	TC	ProfPrim	JOSE	MERCADO	ARIAS	22	HOMBRE
3336.-	305034	1000685	SR	InvCien	CARMEN	ALVARADO	MIRANDA	20	MUJER
3337.-	316519	1002677	JC	ChefEje	JORGE	ROMAN	VILLANUEVA	31	HOMBRE
3338.-	324467	1015604	CS	PsOrg	ANA	ORTIZ	GARZA	50	MUJER
3339.-	314382	1013988	MS	AsServ	SOFIA	BARRIOS	ROMERO	46	MUJER
3340.-	305968	1009816	TS	Arquitect	JAVIER	MUNGUIA	AGUILAR	21	HOMBRE
3341.-	315561	1013521	BC	AsistAdm	SUSANA	MACIAS	OSORIO	50	MUJER
3342.-	324147	1010432	GT	DisGraf	LAURA	ZAMORA	CABRERA	38	MUJER
3343.-	304142	1001402	SP	ProfPrim	LAURA	CONTRERAS	MEDINA	19	MUJER
3344.-	323053	1005095	CS	IngElec	HECTOR	DIAZ	ALVARADO	45	HOMBRE
3345.-	309775	1032366	MS	EspRRHH	LAURA	GOMEZ	MEDINA	28	MUJER
3346.-	318319	1014722	SP	AnMktDg	HECTOR	MOLINA	AGUILAR	29	HOMBRE
3347.-	321778	1026083	DF	ConsFin	ELENA	MORA	JIMENEZ	19	MUJER
3348.-	317557	1023441	ZS	PsOrg	GABRIELA	ESTRELLA	CRUZ	46	MUJER
3349.-	321101	1023203	HG	IngElec	MARTA	DURAN	VARGAS	39	MUJER
3350.-	330295	1025268	YN	AnlSist	JOSE	CONTRERAS	CASTILLO	46	HOMBRE
3351.-	327050	1009239	NE	Traduct	PILAR	LARA	ARIAS	38	MUJER
3352.-	311893	1032127	SL	Traduct	ELENA	ORTIZ	MACIEL	49	MUJER
3353.-	321005	1016430	BC	IngRed	CARMEN	CONTRERAS	MENDOZA	25	MUJER
3354.-	331050	1023983	SR	EspLog	AURORA	HERRERA	VALENCIA	34	MUJER
3355.-	328851	1017951	HG	RepVInt	ENRIQUE	MACIEL	PARDO	47	HOMBRE
3356.-	311976	1010268	SR	PsOrg	SUSANA	CAMACHO	ROJAS	25	MUJER
3357.-	328201	1025737	MN	Arquitect	ARMANDO	PARDO	FLORES	45	HOMBRE
3358.-	308728	1003841	OC	TrabSoc	ALEJANDRA	ESTRELLA	VALENZUELA	23	MUJER
3359.-	310512	1010993	TS	AsServ	RAQUEL	DURAN	IGLESIAS	22	MUJER
3360.-	331950	1024803	NE	DevSoft	LUIS	GUZMAN	HIDALGO	18	HOMBRE

Ejercicio 2:

```
// Eliminar registro
case 2:
    eliminarReg(indice, &nPers, ordenado);
    break;
```

```
void eliminarReg(Tindice indice[], int *n, int ordenado)
{
    Tkey i;
    int num, op;
    FILE *fa;
    TWkr reg;

    printf("Ingrese la matricula del estudiante que desea eliminar: ");
    num = valiNum(300000, 399999);
    system("CLS");

    i = busqOpt(indice, *n, num, ordenado);

    if (i != -1)
    {
        fa = fopen("datos.dat", "rb+");

        fseek(fa, indice[i].indice * sizeof(TWkr), SEEK_SET);
        fread(&reg, sizeof(TWkr), 1, fa);
```

```

    if (reg.status == 0)
    {
        printf("Matricula se encuentra eliminada\n");
    }
    else
    {
        imprReg(reg);
        printf("\n\nDesea eliminar el registro? (1 - Si, 2 - No) ");
        op = valiNum(1, 2);
        system("CLS");

        if (op == 1)
        {
            reg.status = 0;
            fseek(fa, indice[i].indice * sizeof(TWrkr), SEEK_SET);
            fwrite(&reg, sizeof(TWrkr), 1, fa);

            printf("Matricula eliminada con exito\n");
        }
        else
        {
            printf("Matricula no eliminada\n");
        }
    }

    fclose(fa);
}
else
{
    printf("Matricula no encontrada\n");
}
}

```

Salida 2:

```
Ingrese la matricula del estudiante que desea eliminar: 304915
```

```
STATUS: ACTIVO  
MATRICULA: 304915  
TELEFONO: 1004293  
ESTADO: NL  
PUESTO: InvCien  
NOMBRE: SALVADOR  
AP. PATERNO: PACHECO  
AP. MATERNO: CHACON  
EDAD: 21  
SEXO: HOMBRE
```

```
Desea eliminar el registro? (1 - Si, 2 - No) 1
```

```
Matricula eliminada con exito
```

```
Presione una tecla para continuar . . .
```

```
Ingrese la matricula del estudiante que desea eliminar: 304915
```

```
Matricula se encuentra eliminada
```

```
Presione una tecla para continuar . . .
```


Ejercicio 3:

```
// Buscar registros
case 3:
    buscarReg(indice, nPers, ordenado);
    break;
```

```
void buscarReg(Tindice indice[], int n, int ordenado)
{
    Tkey i;
    int num;

    FILE *fa;
    TWkr reg;

    printf("Ingrese la matricula del estudiante que desea buscar: ");
    num = valiNum(300000, 399999);
    system("CLS");

    i = busqOpt(indice, n, num, ordenado);

    if (i != -1)
    {
        printf("Matricula encontrada\n\n", num);
        fa = fopen("datos.dat", "rb");

        fseek(fa, indice[i].indice * sizeof(TWkr), SEEK_SET);
        fread(&reg, sizeof(TWkr), 1, fa);
        imprReg(reg);

        fclose(fa);
    }
    else
    {
        printf("La matricula %d no se encuentra en el vector\n", num);
    }
}
```

Salida 3:

```
Ingrese la matricula del estudiante que desea buscar: 314588
```

```
Matricula encontrada
```

```
STATUS: ACTIVO
```

```
MATRICULA: 314588
```

```
TELEFONO: 1023105
```

```
ESTADO: NL
```

```
PUESTO: AsistAdm
```

```
NOMBRE: MERCEDES
```

```
AP. PATERNO: CASILLAS
```

```
AP. MATERNO: ESPINOZA
```

```
EDAD: 31
```

```
SEXO: MUJER
```

```
Presione una tecla para continuar . . .
```

Ejercicio 4:

```
// Ordenar registros  
case 4:  
    ordenarReg(indice, nPers, &ordenado);  
    break;
```

```
void ordenarReg(Tindice indice[], int n, int *ordenado)  
{  
    if (*ordenado == 0 || *ordenado == 2)  
    {  
        *ordenado = ordOpt(indice, n, *ordenado);  
        printf("El vector ha sido ordenado\n");  
    }  
    else  
    {  
        printf("El vector ya estaba ordenado\n");  
    }  
}
```

```
// Ordena un vector usando el algoritmo optimo
int ordOpt(Tindice vect[], int n, int ordenado)
{
    int i;

    if (ordenado == 0)
    {
        if (n <= 1000)
        {
            i = ordQuick(vect, n);
            printf("Ordenado usando Quick Sort\n");
        }
        else
        {
            i = ordMerge(vect, n);
            printf("Ordenado usando Merge Sort\n");
        }
    }
    else
    {
        if (ordenado == 2)
        {
            i = ordBubbleMejorado(vect, n);
            printf("Ordenado usando Bubble Sort Mejorado\n");
        }
    }

    return 1;
}
```

Salida 4:

Caso menos de 1000 datos (no ordenado)

```
Ordenado usando Quick Sort  
El vector ha sido ordenado
```

```
Presione una tecla para continuar . . . █
```

Caso más de 1000 datos (no ordenado)

```
Ordenado usando Merge Sort  
El vector ha sido ordenado
```

```
Presione una tecla para continuar . . . █
```

Caso casi ordenado

```
Ordenado usando Bubble Sort Mejorado  
El vector ha sido ordenado
```

```
Presione una tecla para continuar . . . █
```

Ejercicio 5:

```
// Imprimir registros originales desde el archivo
case 5:
    imprArchBin(indice, nPers, 0);
    break;
```

```
void imprArchBin(Tindice indice[], int n, int ordenado)
{
    int i, activos, op;

    FILE *fa;
    Twrkr reg;
    fa = fopen("datos.dat", "rb");

    printf("Registros 1 - 40\n");
    printf("-----\n");
    printf(" No | MATRICULA | TELEFONO | ESTADO | PUESTO\n");
    printf("-----\n");
    for (i = 0, activos = 1; i < n; i++)
    {
        if (ordenado == 1)
        {
            fseek(fa, indice[i].indice * sizeof(Twrkr), SEEK_SET);
        }
        else
        {
            fseek(fa, i * sizeof(Twrkr), SEEK_SET);
        }

        fread(&reg, sizeof(Twrkr), 1, fa);

        if (reg.status == 1)
        {
            printf("%4d.- %6d %7d %-2s %-11s\n",
                    reg.no, reg.matricula, reg.telefono, reg.estado, reg.puesto);
            activos++;
        }

        if (activos % 41 == 0 && activos < n)
        {
            printf("\n\n");
            printf("Desea continuar? (0 - Si, 1 - No): ");
            op = valiNum(0, 1);
        }
    }
}
```

```
        if (op == 0)
        {
            system("CLS");
            printf("Registros %d - %d\n", activos + 1, (activos + 40) > n ? n : (activos + 40));
            printf("-----\n");
            printf("  No   | MATRICULA | TELEFONO   | ESTADO | PUESTO          | NOMBRE          | APELLIDOS\n");
            printf("-----\n");
        }
        else
        {
            fclose(fa);
            return;
        }
    }

    fclose(fa);
}
```

Salida 5:

Registros 1 - 40

No	MATRICULA	TELEFONO	ESTADO	PUESTO	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	331399	1010412	BC	RepVInt	ALEJANDRO	IGLESIAS	ARIAS	33	HOMBRE
1.-	328162	1004600	OC	GteProj	ALBERTO	CASTILLO	LEON	43	HOMBRE
2.-	321166	1032541	CH	DisUX	MIGUEL	MERCADO	MALDONADO	46	HOMBRE
3.-	320571	1025025	CL	MedGen	CLARA	GARZA	CRUZ	27	MUJER
4.-	308286	1008058	ZS	ProfPrim	CONSUELO	VASQUEZ	GUERRERO	44	MUJER
5.-	314627	1002320	BC	InvCien	MARIA	MENDOZA	SANDOVAL	50	MUJER
6.-	306801	1015627	NT	DevSoft	RAUL	TOVAR	PENA	31	HOMBRE
7.-	322504	1022724	DG	Traduct	ALEJANDRO	ROJAS	ROJAS	21	HOMBRE
8.-	329950	1032126	YN	TrabSoc	JOSE	TORRES	ROMAN	42	HOMBRE
9.-	310655	1023008	NL	EspRRHH	PILAR	GARZA	FLORES	23	MUJER
10.-	310480	1029226	VZ	DevSoft	JORGE	DURAN	VILLANUEVA	45	HOMBRE
11.-	319891	1000514	QT	Contado	MARTA	GUZMAN	RUBIO	25	MUJER
12.-	331486	1015956	MS	EspLog	JAVIER	LARA	SUAREZ	28	HOMBRE
13.-	332448	1006144	CS	DisGraf	SUSANA	CAMACHO	MACIEL	49	MUJER
14.-	314814	1019386	JC	AnlSist	SUSANA	CASTANEDA	SOLANO	34	MUJER
15.-	322393	1017857	AG	DisUX	LAURA	PACHECO	CASILLAS	39	MUJER
16.-	301488	1021234	CS	DisUX	JOSE	ESPINOSA	CASTANEDA	38	HOMBRE
17.-	314993	1028615	BS	Enferm	MANUEL	AGUILAR	ESPINOSA	30	HOMBRE
18.-	327953	1021347	JC	AnlVent	SALVADOR	LEON	ROLDAN	22	HOMBRE
19.-	309371	1028688	YN	AsServ	ROSA	ALVAREZ	HERRERA	20	MUJER
20.-	308765	1028378	MC	AnlDatos	SALVADOR	AGUILAR	GONZALES	30	HOMBRE
21.-	328983	1004399	MC	EspRRHH	VICTORIA	TORRES	VEGA	35	MUJER
22.-	320227	1001804	CS	IngRed	ISRAEL	CAMACHO	ARIAS	41	HOMBRE
23.-	316658	1028620	NE	AnMktDg	SALVADOR	PARDO	ESTRELLA	35	HOMBRE
24.-	326642	1001933	MC	Traduct	EDUARDO	IGLESIAS	GOMEZ	44	HOMBRE
25.-	317533	1003063	HG	AsServ	MARIA	LOPEZ	MACIAS	49	MUJER
26.-	309080	1001264	NE	Traduct	SUSANA	IGLESIAS	MORA	39	MUJER
27.-	318084	1008420	OC	IngRed	BEATRIZ	GUERRERO	PENA	36	MUJER
28.-	314588	1023105	NL	AsistAdm	MERCEDES	CASILLAS	ESPINOZA	31	MUJER
29.-	332687	1008462	ZS	InvCien	JOSE	CASTANEDA	VASQUEZ	22	HOMBRE
30.-	306592	1016366	SR	RepVInt	VICTORIA	SOLIS	CERVANTES	49	MUJER
31.-	300086	1017883	YN	IngElec	JOSE	GOMEZ	MACIAS	40	HOMBRE
32.-	312463	1003029	MC	Traduct	DANIEL	SERRANO	JIMENEZ	28	HOMBRE
33.-	301962	1019758	MN	ProfPrim	JORGE	CASILLAS	GARZA	44	HOMBRE
34.-	319038	1028704	CC	TrabSoc	SONIA	RAMOS	CERVANTES	38	MUJER
35.-	321539	1005261	SL	InvCien	MARIA	HERRERA	MACIEL	27	MUJER
36.-	319020	1014084	JC	Traduct	CLARA	SILVA	DURAN	32	MUJER
37.-	304915	1004293	NL	InvCien	SALVADOR	PACHECO	CHACON	21	HOMBRE
38.-	309384	1013229	ZS	AsServ	CONSUELO	VILLANUEVA	CORDERO	44	MUJER
39.-	301773	1031284	TL	InvCien	MARIA	HIDALGO	FERNANDEZ	35	MUJER

Desea continuar? (0 - Si, 1 - No): ☐

Ejercicio 6:

```
// Imprimir registros ordenados desde el indice
case 6:
    imprArchBin(indice, nPers, 1);
    break;
```

Salida 6:

Registros 1 - 40

No	MATRICULA	TELEFONO	ESTADO	PUESTO	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	300019	1016520	DF	InvCien	MIGUEL	HIDALGO	FLORES	21	HOMBRE
1.-	300044	1010870	CL	EspLog	PAULA	ROSALES	HIDALGO	22	MUJER
2.-	300046	1005412	SR	TecSup	ALEJANDRA	REYES	DURAN	23	MUJER
3.-	300077	1006753	DF	Arquitect	CARLOS	MORA	VALENCIA	43	HOMBRE
4.-	300086	1017883	YN	IngElec	JOSE	GOMEZ	MACIAS	40	HOMBRE
5.-	300087	1027023	GR	Abogado	BEATRIZ	AGUILAR	VALENCIA	37	MUJER
6.-	300092	1028675	DF	EspSeg	SOFIA	ROMAN	ALVAREZ	41	MUJER
7.-	300112	1015089	YN	AnlVent	MIGUEL	PACHECO	VEGA	49	HOMBRE
8.-	300122	1029062	TL	GteProj	RAUL	SERRANO	BARRIOS	44	HOMBRE
9.-	300133	1011987	TC	ProfPrim	ROSA	LOPEZ	ESPINOZA	46	MUJER
10.-	300142	1007730	MC	ChefEje	ANA	CERVANTES	DELGADO	49	MUJER
11.-	300148	1007812	MN	DevSoft	MARIO	URIBE	VARGAS	32	HOMBRE
12.-	300158	1030988	MC	Contado	SOFIA	LEON	VALDES	26	MUJER
13.-	300171	1016729	CS	TrabSoc	MARIO	SERRANO	RAMOS	28	HOMBRE
14.-	300178	1010069	QT	AnMktDg	MARIO	PARDO	CORDERO	26	HOMBRE
15.-	300182	1007186	NT	PsOrg	GLORIA	LEAL	PARDO	28	MUJER
16.-	300187	1020450	YN	AnlSist	ARMANDO	CONTRERAS	GONZALES	50	HOMBRE
17.-	300197	1029019	PL	DisUX	CLARA	GUERRERO	ROJAS	29	MUJER
18.-	300209	1024691	CL	EspLog	HECTOR	SILVA	DURAN	38	HOMBRE
19.-	300224	1025510	VZ	PsOrg	ALEJANDRO	IGLESIAS	SUAREZ	26	HOMBRE
20.-	300244	1007814	GR	Traduct	SALVADOR	RAMOS	CHACON	44	HOMBRE
21.-	300245	1015364	QR	EspRRHH	GABRIELA	ORTIZ	ROMERO	48	MUJER
22.-	300250	1010009	BC	TrabSoc	PILAR	HERRERA	ALVARADO	35	MUJER
23.-	300271	1006922	CC	DisGraf	PATRICIA	ESPINOZA	REYES	43	MUJER
24.-	300297	1018431	DG	EspRRHH	NATALIA	SANDOVAL	LEAL	44	MUJER
25.-	300310	1029569	MC	EspRRHH	ROBERTO	SALAZAR	URIBE	42	HOMBRE
26.-	300311	1010044	MS	RepVInt	MARIO	CERVANTES	BAUTISTA	21	HOMBRE
27.-	300319	1001038	CL	Arquitect	ROBERTO	OSORIO	ALVAREZ	33	HOMBRE
28.-	300323	1030996	JC	ProfPrim	MARIA	LEAL	MENENDEZ	19	MUJER
29.-	300326	1030595	NT	IngRed	ARMANDO	TORRES	RIVERA	21	HOMBRE
30.-	300347	1013896	HG	AsistAdm	JORGE	GOMEZ	SOTO	45	HOMBRE
31.-	300349	1012779	YN	ConsFin	EDUARDO	SOSA	ARIAS	34	HOMBRE
32.-	300377	1007384	GR	AnlVent	MANUEL	GONZALES	GUERRERO	20	HOMBRE
33.-	300381	1002919	PL	MedGen	ALEJANDRO	LARA	MENDOZA	37	HOMBRE
34.-	300389	1011320	BC	ProfPrim	GLORIA	ROMERO	REYES	29	MUJER
35.-	300401	1028305	NE	IngCivil	MIGUEL	HERRERA	MACIAS	39	HOMBRE
36.-	300410	1023147	DF	InvCien	ISRAEL	CASTILLO	MORA	50	HOMBRE
37.-	300442	1001938	QR	DisUX	NATALIA	CHACON	PARDO	39	MUJER
38.-	300445	1004349	MN	DevSoft	ISRAEL	ESTRELLA	VARELA	49	HOMBRE
39.-	300454	1021799	NT	IngElec	GABRIELA	LEAL	MERCADO	25	MUJER

Desea continuar? (0 - Si, 1 - No): ☐

Ejercicio 7:

```
// Escribir archivo de texto
case 7:
    leerNomArch(nomArchivo);

    system("CLS");
    printf("1 - Archivo\n2 - Indice\n\nElija una opcion: ");
    if (valiNum(1, 2) == 1)
    {
        escrArchTXT(nomArchivo, indice, nPers, 0);
    }
    else
    {
        escrArchTXT(nomArchivo, indice, nPers, 1);
    }

    break;
```

```
// Escribe un archivo de texto con los registros especificados. Escribir
void escrArchTXT(char nomArchivo[], Tindice indice[], int n, int ordenado)
{
    int i, cont = 0;
    TWrkr reg;

    FILE *fa;
    FILE *fb;

    char temp[30];
    strcpy(temp, nomArchivo);
    strcat(temp, ".txt");

    fa = fopen(temp, "w");
    fb = fopen("datos.dat", "rb");

    fprintf(fa, "-----\n");
    fprintf(fa, " No | MATRICULA | TELEFONO | ESTADO | PUESTO | \n");
    fprintf(fa, "-----\n");

    for (i = 0, cont = 1; i < n; i++)
    {
        if (ordenado == 1)
        {
            fseek(fb, indice[i].indice * sizeof(TWrkr), SEEK_SET);
        }
        else
        {
            fseek(fb, i * sizeof(TWrkr), SEEK_SET);
        }

        fread(&reg, sizeof(TWrkr), 1, fb);
```

```
        if (reg.status == 1)
        {
            fprintf(fa, "%4d.- %6d %7d %2s %11s %10s\n",
                cont++, reg.no, reg.matricula, reg.telefono, reg.estado, reg.puesto);
        }
    }

    fprintf(fa, "\n-----\n");
    fprintf(fa, "-- TODOS LOS DERECHOS RESERVADOS @Scalaptia ONLYCODES\n");
    fprintf(fa, "-----\n");

    fclose(fa);
    fclose(fb);
}
```

Salida 7:

```
Ingresa el nombre del archivo (sin extension): datos
```

```
1 - Archivo  
2 - Indice
```

```
Elija una opcion:
```

Caso archivo:

ACTIVIDAD14 > datos.txt

No	MATRICULA	TELEFONO	ESTADO	PUESTO	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	331399	1010412	BC	RepVInt	ALEJANDRO	IGLESIAS	ARIAS	33	HOMBRE
1.-	328162	1004600	OC	GteProj	ALBERTO	CASTILLO	LEON	43	HOMBRE
2.-	321166	1032541	CH	DisUX	MIGUEL	MERCADO	MALDONADO	46	HOMBRE
3.-	320571	1025025	CL	MedGen	CLARA	GARZA	CRUZ	27	MUJER
4.-	308286	1008058	ZS	ProfPrim	CONSUELO	VASQUEZ	GUERRERO	44	MUJER
5.-	314627	1002320	BC	InvCien	MARIA	MENDOZA	SANDOVAL	50	MUJER
6.-	306801	1015627	NT	DevSoft	RAUL	TOVAR	PENA	31	HOMBRE
7.-	322504	1022724	DG	Traduct	ALEJANDRO	ROJAS	ROJAS	21	HOMBRE
8.-	329950	1032126	YN	TrabSoc	JOSE	TORRES	ROMAN	42	HOMBRE
9.-	310655	1023008	NL	EspRRHH	PILAR	GARZA	FLORES	23	MUJER
10.-	310480	1029226	VZ	DevSoft	JORGE	DURAN	VILLANUEVA	45	HOMBRE
11.-	319891	1000514	QT	Contado	MARTA	GUZMAN	RUBIO	25	MUJER
12.-	331486	1015956	MS	EspLog	JAVIER	LARA	SUAREZ	28	HOMBRE
13.-	332448	1006144	CS	DisGraf	SUSANA	CAMACHO	MACIEL	49	MUJER
14.-	314814	1019386	JC	AnlSist	SUSANA	CASTANEDA	SOLANO	34	MUJER
15.-	322393	1017857	AG	DisUX	LAURA	PACHECO	CASILLAS	39	MUJER
16.-	301488	1021234	CS	DisUX	JOSE	ESPINOSA	CASTANEDA	38	HOMBRE
17.-	314993	1028615	BS	Enferm	MANUEL	AGUILAR	ESPINOSA	30	HOMBRE

Caso índice:

ACTIVIDAD14 > datos.txt

No	MATRICULA	TELEFONO	ESTADO	PUESTO	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	300019	1016520	DF	InvCien	MIGUEL	HIDALGO	FLORES	21	HOMBRE
1.-	300044	1010870	CL	EspLog	PAULA	ROSALES	HIDALGO	22	MUJER
2.-	300046	1005412	SR	TecSup	ALEJANDRA	REYES	DURAN	23	MUJER
3.-	300077	1006753	DF	Arquitect	CARLOS	MORA	VALENCIA	43	HOMBRE
4.-	300086	1017883	YN	IngElec	JOSE	GOMEZ	MACIAS	40	HOMBRE
5.-	300087	1027023	GR	Abogado	BEATRIZ	AGUILAR	VALENCIA	37	MUJER
6.-	300092	1028675	DF	EspSeg	SOFIA	ROMAN	ALVAREZ	41	MUJER
7.-	300112	1015089	YN	AnlVent	MIGUEL	PACHECO	VEGA	49	HOMBRE
8.-	300122	1029062	TL	GteProj	RAUL	SERRANO	BARRIOS	44	HOMBRE
9.-	300133	1011987	TC	ProfPrim	ROSA	LOPEZ	ESPINOZA	46	MUJER
10.-	300142	1007730	MC	ChefeJe	ANA	CERVANTES	DELGADO	49	MUJER
11.-	300148	1007812	MN	DevSoft	MARIO	URIBE	VARGAS	32	HOMBRE
12.-	300158	1030988	MC	Contado	SOFIA	LEON	VALDES	26	MUJER
13.-	300171	1016729	CS	TrabSoc	MARIO	SERRANO	RAMOS	28	HOMBRE
14.-	300178	1010069	QT	AnMktDg	MARIO	PARDO	CORDERO	26	HOMBRE
15.-	300182	1007186	NT	PsOrg	GLORIA	LEAL	PARDO	28	MUJER
16.-	300187	1020450	YN	AnlSist	ARMANDO	CONTRERAS	GONZALES	50	HOMBRE
17.-	300197	1029019	PL	DisUX	CLARA	GUERRERO	ROJAS	29	MUJER

Ejercicio 8:

```
// Empaquetar registros
case 8:
    empaquetar(indice, &nPers);
    break;
}
```

```
// Actualiza el archivo binario eliminando los registros no activos.
void empaquetar(Tindice indice[], int *n)
{
    TWkrkr reg;
    FILE *fa;
    FILE *fb;

    rename("datos.dat", "datos.bak");

    fa = fopen("datos.dat", "wb");
    fb = fopen("datos.bak", "rb");

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

    fclose(fa);
    fclose(fb);




    *n = 0;
    cargarIndice(indice, n);

    printf("Archivo empaquetado con exito\n");
}
```

Salida 8:

```
Archivo empaquetado con éxito
```

```
Presione una tecla para continuar . . . █
```

 **datos.bak** **datos.dat** **datos.txt**