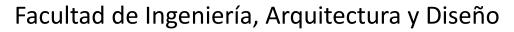


Universidad Autónoma de Baja California





ARCHIVOS INDEXADOS

PROGRAMACIÓN ESTRUCTURADA

ANEXO 14

Ingeniero en Software y Tecnologías Emergentes

Brayan Ivan Perez Ventura 372781

MENU:

```
void menu()
         int op;
          int max_registers, position;
          position = LoadBinaryFile();
         getIndexFile(position);
         getTempFiles(position);
         max_registers = position * 1.25;
95
         int ord = 0;
             system("CLS");
             op = msge_menu();
             system("CLS");
              switch (op)
                  if (position + 1 < max_registers)</pre>
                      addRegister(position, max_registers);
                      ord = 0;
                      position++;
                 }
                      printf("Vector lleno\n");
                 break;
              case 2:
                 deleteRegister(position, max_registers, ord);
                  findRegister(position, ord);
                 ord = OrderIndex(position, ord);
                 break:
                 ord = displayReg(position, ord);
                 break;
                 genTxT(position, ord);
                 break;
                 packageFile(position);
                 break;
                 break;
              if (op != 0)
                  system("PAUSE");
          } while (op != 0);
```

```
73  int msge_menu()
74  {
75     printf("----- M E N U -----\n");
76     printf("1.- Agregar\n");
77     printf("2.- Eliminar\n");
78     printf("3.- Buscar\n");
79     printf("4.- Ordenar\n");
80     printf("5.- Mostrar\n");
81     printf("6.- Generar archivo de texto\n");
82     printf("7.- Empaquetar\n");
83     printf("0.- Salir\n");
84     return valid("Selecciona una opcion: ", 0, 7);
85  }
```

```
PROBLEMS TERMINAL OUTPUT PORTS

----- M E N U -----

1.- Agregar

2.- Eliminar

3.- Buscar

4.- Ordenar

5.- Mostrar

6.- Generar archivo de texto

7.- Empaquetar

0.- Salir

Selecciona una opcion:
```

AÑADIR REGISTRO:

```
void addRegister(int position, int max_registers)
193 🖁
          TIndexStrct employees[max_registers];
          TWrkr tempEmployee;
195
          TIndexStrct temp;
          tempEmployee = generateRegister();
          fillIndexRegister(employees);
200
              tempEmployee.enrollment = numRandom(300000, 399999);
          } while (existElem(employees, position, tempEmployee.enrollment) != -1);
          addRegisterData(tempEmployee);
          temp.enrollment = tempEmployee.enrollment;
          temp.index = position;
          addRegisterIndex(temp);
          displayRegEmp(tempEmployee);
```

```
PROBLEMS TERMINAL OUTPUT PORTS GITLENS

Aniadido correctamente
Presione una tecla para continuar . . .
```

ENCONTRAR REGISTRO:

```
void findRegister(int position, int ord)
   int index;
   TIndexStrct indexFile[position];
   TWrkr regTemp;
   index = valid("Ingrese la matricula del estudiante a eliminar: ", 300000, 399999);
   fillIndexRegister(indexFile);
   if (ord)
       index = binarySearch(indexFile, 0, position, index);
       index = existElem(indexFile, position, index);
   if (index != -1)
       fa = fopen("datos.dat", "rb");
       fseek(fa, index * sizeof(TWrkr), SEEK_SET);
       fread(&regTemp, sizeof(TWrkr), 1, fa);
       fclose(fa);
       if (regTemp.status)
           displayRegEmp(regTemp);
           printf("Alumno eliminado con anterioridad\n");
```

```
PROBLEMS
          TERMINAL
Ingrese la matricula del estudiante a eliminar: 314318
Matricula:
            314318
Nombre:
            PTI AR
Ap. Paterno: ROMERO
Ap. Materno: SOTO
Sexo:
            MUJER
Edad:
            26
Posicion:
            Enferm
Num. Cel:
            646-1020168
Lug. Nacim: CC
Presione una tecla para continuar . . .
```

ELIMINAR ESTUDIANTE:

```
void deleteRegister(int position, int max_registers, int ord)
           int index;
           TIndexStrct indexFile[max registers];
           TWrkr regTemp;
           index = valid("Ingrese la matricula del estudiante a eliminar: ", 300000, 399999);
           fillIndexRegister(indexFile);
           if (ord)
Makefile
               index = binarySearch(indexFile, 0, position, index);
               index = existElem(indexFile, position, index);
           if (index != -1)
               fa = fopen("datos.dat", "rb");
               fseek(fa, index * sizeof(TWrkr), SEEK_SET);
               fread(&regTemp, sizeof(TWrkr), 1, fa);
               fclose(fa);
               if (regTemp.status)
                   displayRegEmp(regTemp);
                   if (valid("Desea eliminarlo? (1.- Si, 0.- No): ", 0, 1))
                       regTemp.status = 0;
                       fa = fopen("datos.dat", "rb+");
                       fseek(fa, index * sizeof(TWrkr), SEEK_SET);
                       fwrite(&regTemp, sizeof(TWrkr), 1, fa);
                       fclose(fa);
                       printf("Eliminado con exito\n");
                   printf("Alumno eliminado con anterioridad\n");
                                                                PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS
               printf("El alumno no existe\n");
                                                                Ingrese la matricula del estudiante a eliminar: 319601
                                                                Matricula: 319601
                                                                Nombre:
                                                                             LUIS
                                                                Ap. Paterno: ROJAS
                                                                Ap. Materno: ESTRELLA
                                                                Sexo:
                                                                             HOMBRE
                                                                Edad:
                                                                             34
                                                                             Abogado
                                                                Posicion:
                                                                Num. Cel:
                                                                             646-1012697
                                                                Lug. Nacim: MS
                                                                Desea eliminarlo? (1.- Si, 0.- No): 1
                                                                Eliminado con exito
                                                                Presione una tecla para continuar . . .
```

ORDENAR

```
int OrderIndex(int position, int flag)
{

IIndexStrct Index[position];

FILE *fa;

if (flag)
{
    printf("El vector ya ha sido ordenado con anterioridad");
}
else
{
    fillIndexRegister(Index);
    bubbleSort(Index, position);
    fa = fopen("datos_index.dat", "wb");
    fwrite(Index, sizeof(TWrkr), 1, fa);
    fclose(fa);
    printf("El vector ha sido ordenado\n");
    return 1;
}
return flag;

313
}
```

```
PROBLEMS TERMINAL OUTPUT PORTS GITLENS C

El vector ha sido ordenado

Presione una tecla para continuar . . .
```

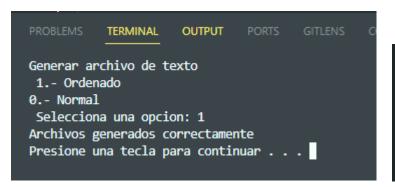
IMPRIMIR REGISTRO

```
int displayReg(int position, int flag)
   FILE *fa;
   TWrkr temp;
   int i = 0;
   fa = fopen("datos.dat", "rb");
   if (valid("Imprimir datos\n 1.- Ordenado\n0.- Normal\n Selecciona una opcion: ", 0, 1))
       TIndexStrct Indexs[position];
       fillIndexRegister(Indexs);
       if (!flag)
           bubbleSort(Indexs, position);
           flag = 1;
       for (i = 0; i < position; i++)
           fseek(fa, Indexs[i].index * sizeof(TWrkr), SEEK_SET);
           fread(&temp, sizeof(TWrkr), 1, fa);
           if (temp.status)
               displayListEmp(temp, i);
       fclose(fa);
       system("CLS");
       if (fa)
           while (fread(&temp, sizeof(TWrkr), 1, fa))
               if (temp.status)
                   displayListEmp(temp, i);
                                                      PROBLEMS
                                                                   TERMINAL
                                                      Imprimir datos
           fclose(fa);
                                                       1.- Ordenado
                                                      0.- Normal
   return flag;
                                                       Selecciona una opcion:
```

PROBLEMS	TERMINAL	OUTPUT PORTS	GITLENS COMMENTS	DEBUG CONSOLE					
1009	309459	HECTOR	CERVANTES	IGLESIAS	HOMBRE	AnlVent	CM	46	6461030224
1010	309465	EDUARDO	MACIEL	ROMAN	HOMBRE	AnMktDg	NT	21	6461030704
1011	309466	RAQUEL	MACIEL	DURAN	MUJER	EspERen	MN	50	6461028287
1012	309467	LUIS	REYES	DELGADO	HOMBRE	Traduct	SR	46	6461026132
1013	309476	ARMANDO	CASILLAS	CAMACHO	HOMBRE	TecSup	AG	39	6461013823
1014	309485	JORGE	RAMOS	ROJAS	HOMBRE	GteProj	OC	36	6461010533
1015	309495	GUILLERMO	ESTRELLA	CAMACHO	HOMBRE	DisUX	NE	46	6461024759
1016	309498	DANIEL	ALVARADO	RUBIO	HOMBRE	EdCont	YN	33	6461013563
1017	309499	ERNESTO	GUERRERO	URIBE	HOMBRE	IngRed	YN	28	6461029439
1018	309510	JULIA	VARELA	CERVANTES	MUJER	ChefEje	SL	50	6461015169
1019	309520	HECTOR	GUERRERO	VALENZUELA	HOMBRE	MedGen	NL	29	6461007931
1020	309532	MARiA	ROJAS	ESPINOSA	MUJER	GteProj	CH	35	6461001447
1021	309540	GLORIA	AGUILAR	RUBIO	MUJER	IngCivil	VZ	48	6461015660
1022	309549	LAURA	MERCADO	BARRIOS	MUJER	ChefEje	SL	44	6461012076
1023	309552	PILAR	CALZADA	VALENZUELA	MUJER	AsistAdm	VZ.	24	6461028689
1024	309576	ERNESTO	LARA	CERVANTES	HOMBRE	Abogado	MS	47	6461021959
1025	309610	GABRIELA	SOLIS	CORDERO	MUJER	AnMktDg	VZ	38	6461005544
1026	309627	ANA	CRUZ	VALENZUELA	MUJER	DevSoft	MC	28	6461010491
1027	309637	Marta	ROJAS	BELTRAN	MUJER	Contado	TS	28	6461014655
1028	309638	RAQUEL	RAMIREZ	MORA	MUJER	AsServ	NL.	20	6461027396
1029	309655	ANTONIA	SILVA	LARA	MUJER	Arquitect	DG	44	6461019876
1030	309656	ROSA	MORENO	ARIAS	MUJER	ChefEje	NE	38	6461014881
1031	309657	Marta	ARIAS	TORRES	MUJER	TrabSoc	TL	36	6461012917
1032	309660	MIGUEL	Castaneda	MACIEL	HOMBRE	Ps0rg	HG	28	6461002085
1033	309661	ANTONIA	CASILLAS	PENA	MUJER	IngRed	PL	32	6461019665
1034	309670	JULIA	SOTO	BARRIOS	MUJER	RepVInt	NE	21	6461022133
1035	309678	ALEJANDRO	LOPEZ	OCHOA	HOMBRE	Arquitect	CH	49	6461014007
1036	309684	ROBERTO	ESPINOSA	MACIAS	HOMBRE	ChefEje	BC	45	6461026190
1037	309686	ALBERTO	BAUTISTA	DIAZ	HOMBRE	DisGraf	SL	38	6461010124
1038	309688	NATALIA	ALVAREZ	CERVANTES	MUJER	Contado	NE	24	6461021312
1039	309695	PAULA	LEAL	FERNANDEZ	MUJER	EspERen	QT	36	6461024448
1040	309709	GLORIA	CASTILLO	ESTRELLA	MUJER	EspLog	NT	39	6461017370
1041	309741	HECTOR	ROJAS	VALENCIA	HOMBRE	ProfPrim	NL	24	6461010552

GENERAR ARCHIVO DE TEXTO

```
void genTxT(int position, int flag)
    TWrkr temp;
    if (valid("Generar archivo de texto\n 1.- Ordenado\n0.- Normal\n Selecciona una opcion: ", 0, 1))
        TIndexStrct Indexs[position];
        fillIndexRegister(Indexs);
        if (!flag)
            bubbleSort(Indexs, position);
            flag = 1;
        getTXT(Indexs, position, "datos_activos.txt", 1);
getTXT(Indexs, position, "datos_inactivos.txt", 0);
        printf("Archivos generados correctamente\n");
        FILE *fa, *pa, *ma;
        int i = 0, j = 0;
        fa = fopen("datos.dat", "rb");
        pa = fopen("datos_activos.txt", "w");
        ma = fopen("datos_inactivos.txt", "w");
        if (fa)
            while (fread(&temp, sizeof(TWrkr), 1, fa))
                 if (temp.status)
                     addOneEmplyeeTxT(temp, i, pa);
                     addOneEmplyeeTxT(temp, j, ma);
            fclose(fa);
            printf("Archivos generados correctamente\n");
        fclose(pa);
        fclose(ma);
```





EMPAQUETAR

```
void packageFile(int position)
{

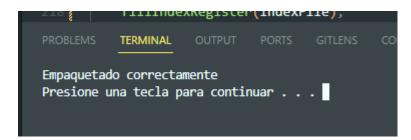
FILE *fa;
TWrkr vectTemp[position];
TWrkr vectTempActive[position];
int j = 0;
fillRegister(vectTemp);
rename("datos.dat", "datos.bak");
fa = fopen("datos.dat", "wb");
if (fa)

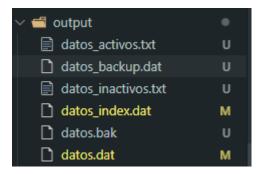
for (int i = 0; i < position; i++)

{
    if (vectTemp[i].status)
    {
        vectTempActive[j++] = vectTemp[i];
        }
}

fwrite(vectTempActive, sizeof(TWrkr), j, fa);
fclose(fa);
printf("Empaquetado correctamente\n");
}

429
}</pre>
```





FUNCIONES EXTRAS:

```
TWrkr generateRegister()
          TWrkr temp;
          temp.age = numRandom(18, 40);
          if (numRandom(0, 1))
              nameMen(temp.name);
              strcpy(temp.sex, "HOMBRE");
          else
              nameWomen(temp.name);
              strcpy(temp.sex, "MUJER");
          LastName(temp.LastName1);
          LastName(temp.LastName2);
          getState(temp.state);
          getJobPositions(temp.JobPstion);
454
          temp.cellPhone = numRandom(1000000, 1999999);
455
          temp.status = 1;
          return temp;
```

```
int fillRegister(TWrkr employess[])

fillE *fa;
fa = fopen("datos.dat", "rb");
int i = 0;
if (fa)

fillRegister(TWrkr employess[])

fa = fopen("datos.dat", "rb");
int i = 0;
if (fa)

fillRegister(TWrkr employess[])

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

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

fillRegister(TWrkr employess[])

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

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

fa = fopen("datos.dat", "rb");
int i = 0;
if (fa)

fillRegister(TWrkr employess[])

fa = fopen("datos.dat", "rb");
int i = 0;
if (fa)

fa = fopen("datos.dat", "rb");
fa = fopen("datos.dat", "rb");
int i = 0;
if (fa)

fa = fopen("datos.dat", "rb");
fa = fopen("datos.dat
```

```
int existElem(TIndexStrct employee[], int longi, Tkey num)

{
    int i;
    for (i = 0; i < longi; i++)

    {
        if (employee[i].enrollment == num)

        {
            return employee[i].index;
        }

        }

return -1;
}</pre>
```

```
void addRegisterData(TWrkr employee)

{

FILE *fa;

fa = fopen("datos.dat", "ab");

fseek(fa, 0, SEEK_END);

fwrite(&employee, sizeof(TWrkr), 1, fa);

fclose(fa);
}
```

```
void addRegisterIndex(TIndexStrct employeeIndx)

{

FILE *fa;

fa = fopen("datos_index.dat", "ab");

fseek(fa, 0, SEEK_END);

fwrite(&employeeIndx, sizeof(TIndexStrct), 1, fa);

fclose(fa);
}
```

```
int binarySearch(TIndexStrct employee[], int left, int right, Tkey number)

{
    int medium;
    while (left <= right)
    {
        medium = left + (right - left) / 2;

        if (employee[medium].enrollment == number)
        {
            return employee[medium].index;
        }

        if (employee[medium].enrollment < number)
        {
            left = medium + 1;
        }
        else
        {
            right = medium - 1;
        }
}</pre>
```

```
void getTXT(TIndexStrct Indexs[], int position, char fileName[], int flag)
    int i;
   FILE *fa;
   FILE *pa;
   TWrkr temp;
    int j
   fa = fopen(fileName, "w");
   pa = fopen("datos.dat", "rb");
    fprintf(fa, "%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-8s %-5s %-10s\n",
            "No.",
            "Matricula",
            "Nombre",
            "Ap. Paterno",
            "Ao. Materno",
            "Sexo",
            "Posicion",
            "Estado",
            "Edad",
            "Num. Cel");
   j = 0;
    for (i = 0; i < position; i++)
        fseek(pa, Indexs[i].index * sizeof(TWrkr), SEEK_SET);
        fread(&temp, sizeof(TWrkr), 1, pa);
        if (temp.status == flag)
            addOneEmplyeeTxT(temp, j, fa);
            j++;
    fclose(pa);
    fclose(fa);
```

```
void swap(TIndexStrct employees[], int i, int j)
    TIndexStrct temp = employees[i];
    employees[i] = employees[j];
    employees[j] = temp;
int partition(TIndexStrct employees[], int low, int high)
    TIndexStrct pivot;
    pivot.enrollment = employees[high].enrollment;
    int i = low - 1;
    for (int j = low; j \leftarrow high - 1; j++)
        if (employees[j] enrollment <= pivot enrollment)</pre>
            i++;
            swap(employees, i, j);
    swap(employees, i + 1, high);
    return i + 1;
void quicksort(TIndexStrct employees[], int low, int high)
    if (low < high)
        int pi = partition(employees, low, high);
        quicksort(employees, low, pi - 1);
        quicksort(employees, pi + 1, high);
void bubbleSort(TIndexStrct employees[], int n)
    int i, j;
   TIndexStrct temp;
    for (i = 0; i < n - 1; i++)
        for (j = i + 1; j < n; j++)
            if (employees[j].enrollment < employees[i].enrollment)</pre>
                temp = employees[i];
                employees[i] = employees[j];
                employees[j] = temp;
```