



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño



ARCHIVOS BINARIOS

PROGRAMACIÓN ESTRUCTURADA

PRÁCTICA 13

Ingeniero en Software y Tecnologías Emergentes

Brayan Ivan Perez Ventura

372781

19 de Noviembre del 2023 en Ensenada, Baja California

```

125 void menu()
126 {
127     int op;
128     Twrkr employee[MAX_REGISTERS];
129     int position = 0;
130     int ordFlag = 0;
131     int tempPstion = 0;
132     int flagFile = 0;
133     do
134     {
135         system("CLS");
136         op = msge_menu();
137         system("CLS");
138         switch (op)
139         {
140             case 1:
141                 position = addRegisters(employee, position, ordFlag);
142                 ordFlag = 0;
143                 break;
144             case 2:
145                 ordFlag = editRegister(employee, position, ordFlag);
146                 break;
147             case 3:
148                 deleteEmployee(employee, position, ordFlag);
149                 break;
150             case 4:
151                 searchEmployee(employee, position, ordFlag);
152                 break;
153             case 5:
154                 ordFlag = ordRegister(employee, position, ordFlag);
155                 break;
156             case 6:
157                 displayRegister(employee, position);
158                 break;
159             case 7:
160                 generateTXT(employee, position);
161                 break;
162             case 8:
163                 showTxTFile();
164                 break;
165             case 9:
166                 getBinaryFile(employee, position);
167                 break;
168             case 10:
169                 if (!flagFile)
170                 {
171                     tempPstion = position;
172                     position = readFileBinary(employee, position);
173                     if (tempPstion != position)
174                     {
175                         flagFile = 1;
176                     }
177                 }
178                 else
179                 {
180                     printf("El archivo ya ha sido cargado con anterioridad\n");
181                 }
182                 break;
183             case 11:
184                 showBinaryDltd();
185                 break;
186         }
187         system("PAUSE");
188     } while (op != 0);
189 }

```

```

107 int msge_menu()
108 {
109     printf("----- M E N U -----\\n");
110     printf("[1] Agregar 100 registros\\n");
111     printf("[2] Editar registro\\n");
112     printf("[3] Eliminar registro\\n");
113     printf("[4] Buscar\\n");
114     printf("[5] Ordenar\\n");
115     printf("[6] Imprimir\\n");
116     printf("[7] Generar archivo de texto\\n");
117     printf("[8] Mostrar archivo de texto\\n");
118     printf("[9] Crear archivo binario\\n");
119     printf("[10] Cargar archivo binario\\n");
120     printf("[11] Mostrar borrados\\n");
121     printf("[0] Salir\\n");
122     return valid("Selecciona una opcion: ", 0, 11);
123 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITL

```

----- M E N U -----
[1] Agregar 100 registros
[2] Editar registro
[3] Eliminar registro
[4] Buscar
[5] Ordenar
[6] Imprimir
[7] Generar archivo de texto
[8] Mostrar archivo de texto
[9] Crear archivo binario
[10] Cargar archivo binario
[11] Mostrar borrados
[0] Salir
Selecciona una opcion: 

```

```

191 int addRegisters(Twkr employee[], int position, int ordFlag)
192 {
193     int i;
194     if (position + GEN_REGISTERS < MAX_REGISTERS)
195     {
196
197         for (i = 0; i < GEN_REGISTERS; i++)
198         {
199             employee[position + i] = getOneEmployee(employee, position, ordFlag);
200             displayListEmp(employee[i], position + i);
201         }
202         position += GEN_REGISTERS;
203     }
204     else
205     {
206         printf("El registro no puede añadir %d de registros debido a que ha sobrepasado el limite.\n", GEN_REGISTERS);
207     }
208     return position;
209 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Se han aniadido 100 registros
 Presione una tecla para continuar . . .

```

211 int editRegister(Twrkr employee[], int position, int ordFlag)
212 {
213     int index;
214     Tkey num;
215     int enrollment[position];
216
217     getNumEnrollment(employee, enrollment, position);
218     num = valid("Ingresa la matricula de la persona para editar el registro: ", 300000, 399999);
219     if (ordFlag)
220     {
221         index = binarySearch(enrollment, 0, position, num);
222     }
223     else
224     {
225         index = existElem(enrollment, position, num);
226     }
227
228     if (index != -1)
229     {
230         if (employee[index].status)
231         {
232             ordFlag = menuRegister(employee, position, index, enrollment, ordFlag);
233         }
234         else
235         {
236             printf("El trabajador ha sido de baja con anterioridad\n");
237         }
238     }
239     return ordFlag;
240 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Ingresa la matricula de la persona para editar el registro: 318222

----- ENCONTRADO -----

Matricula: 318222
Nombre: ANTONIA
Ap. Paterno: BARRIOS
Ap. Materno: CERVANTES
Sexo: MUJER
Edad: 31
Posicion: MedGen
Num. Cel: 646-1003279
Lug. Nacim: HG

MODIFICAR ALGUN CAMPO

1.- Edad
2.- Lugar de nacimiento
3.- Num. Cel
4.- Posicion de trabajo
5.- Matricula
0.- Salir
Selecciona una opcion: █

PROBLEMS

TERMINAL

OUTPUT

PORTS

GIT LENS

COMMENTS

DEBUG CONSOLE

Ingrese la edad correcta: 30

Presione una tecla para continuar . . .

PROBLEMS

TERMINAL

OUTPUT

PORTS

GIT LENS

COMMENTS

----- ENCONTRADO -----

Matricula: 318222

Nombre: ANTONIA

Ap. Paterno: BARRIOS

Ap. Materno: CERVANTES

Sexo: MUJER

Edad: 30

Posicion: MedGen

Num. Cel: 646-1003279

Lug. Nacim: HG

MODIFICAR ALGUN CAMPO

1.- Edad

2.- Lugar de nacimiento

3.- Num. Cel

4.- Posicion de trabajo

5.- Matricula

0.- Salir

Selecciona una opcion:

```

242 int deleteEmployee(Twrkr employee[], int position, int ordFlag)
243 {
244     int index;
245     int enrollment[position];
246     Tkey num;
247     getNumEnrollment(employee, enrollment, position);
248     num = valid("Ingresa la matricula de la persona: ", 300000, 399999);
249     if (ordFlag)
250     {
251         index = binarySearch(enrollment, 0, position, num);
252     }
253     else
254     {
255         index = existElem(enrollment, position, num);
256     }
257
258     if (index != -1)
259     {
260         if (employee[index].status != 0)
261         {
262             printf("\n----- ENCONTRADO ----- \n");
263             displayRegEmp(employee[index]);
264             printf("----- \n");
265             if (valid("Deseas eliminar el registro (0.- No / 1.- Si): ", 0, 1))
266             {
267                 employee[index].status = 0;
268
269                 printf("El trabajador ha sido dado de baja correctamente.\n");
270                 return 0;
271             }
272         }
273         else
274         {
275             printf("El trabajador ya sido dado de baja con anterioridad.\n");
276         }
277     }
278     else
279     {
280         printf("La enrollment ingresada no pertecene a ningun trabajador\n");
281     }
282     return ordFlag;
283 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Ingresa la matricula de la persona: 309976

----- ENCONTRADO -----

Matricula: 309976
Nombre: JUAN
Ap. Paterno: CHACON
Ap. Materno: SOTO
Sexo: HOMBRE
Edad: 40
Posicion: EspLog
Num. Cel: 646-1026927
Lug. Nacim: CM

Deseas eliminar el registro (0.- No / 1.- Si): 1
El trabajador ha sido dado de baja correctamente.
Presione una tecla para continuar . . .

```

285 void searchEmployee(Twrkr employee[], int position, int ordFlag)
286 {
287     int num[position];
288     Tkey index;
289     getNumEnrollment(employee, num, position);
290     index = valid("Ingresa la matricula del trabajador a buscar: ", 300000, 399999);
291     if (ordFlag)
292     {
293         index = binarySearch(num, 0, position, index);
294     }
295     else
296     {
297         index = existElem(num, position, index);
298     }
299
300     if (index != -1)
301     {
302         if (employee[index].status)
303         {
304             printf("----- ENCONTRADO ----- \n");
305             displayRegEmp(employee[index]);
306             printf("----- \n");
307         }
308         else
309         {
310             printf("El trabajador ha sido de baja con anterioridad. \n");
311         }
312     }
313     else
314     {
315         printf("El trabajador con esa matricula no existe \n");
316     }
317 }

```

PROBLEMS

TERMINAL

OUTPUT

PORTS

GITLENS

COMMENTS

DEBUG CONSOLE

Ingresa la matricula del trabajador a buscar: 307790

----- ENCONTRADO -----

Matricula: 307790

Nombre: GUILLERMO

Ap. Paterno: CHACON

Ap. Materno: MENDEZ

Sexo: HOMBRE

Edad: 37

Posicion: EspLog

Num. Cel: 646-1031893

Lug. Nacim: NT

Presione una tecla para continuar . . .


```

319 int ordRegister(TWrkr employee[], int position, int ordFlag)
320 {
321     if (!ordFlag)
322     {
323         if (!(position == 0))
324         {
325             if (position < 600)
326             {
327                 bubbleSort(employee, position);
328             }
329             else
330             {
331                 quicksort(employee, 0, position);
332             }
333             printf("Se han ordenado por matricula de menor a mayor\n");
334             return 1;
335         }
336         else
337         {
338             printf("No hay nada en el vector.\n");
339         }
340     }
341     else
342     {
343         printf("Ya esta ordenado.\n");
344     }
345     return ordFlag;
346 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Se han ordenado por matricula de menor a mayor
 Presione una tecla para continuar . . .

```
348 void displayRegister(TWrkr employee[], int position)
349 {
350     int i;
351     int j = 0;
352     printf("%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-10s %-5s %-10s\n",
353         "No.",
354         "Matricula",
355         "Nombre",
356         "Ap. Paterno",
357         "Ao. Materno",
358         "Sexo",
359         "Posicion",
360         "Estado",
361         "Edad",
362         "Num. Cel");
363     for (i = 0; i < position; i++)
364     {
365         if (employee[i].status == 1)
366         {
367             displayListEmp(employee[i], j);
368             j++;
369         }
370     }
371 }
```

PROBLEMS	TERMINAL	OUTPUT	PORTS	GITLENS	COMMENTS	DEBUG CONSOLE				
No.	Matricula	Nombre	Ap. Paterno	Ao. Materno	Sexo	Posicion	Estado	Edad	Num. Cel	
1	300467	ISRAEL	VALENCIA	MENENDEZ	HOMBRE	AnlVent	JC	26	6461021137	
2	301240	PAULA	HERNANDEZ	CABRERA	MUJER	Contado	MN	28	6461016520	
3	301394	ENRIQUE	MEDINA	GOMEZ	HOMBRE	TrabSoc	BS	20	6461011172	
4	301722	MARTA	SOTO	ALVARADO	MUJER	GteProj	MS	39	6461011794	
5	301962	JOSE	ROLDAN	HERRERA	HOMBRE	EdCont	MC	20	6461031653	
6	302091	ROBERTO	FERNANDEZ	VALENCIA	HOMBRE	Traduct	DG	48	6461005111	
7	302213	SUSANA	BARRIOS	ALVAREZ	MUJER	Abogado	HG	20	6461025172	
8	302341	ANTONIO	SOTO	SUAREZ	HOMBRE	ProfPrim	DF	46	6461015663	
9	302387	NATALIA	DELGADO	PENA	MUJER	RepVInt	CM	43	6461017606	
10	302868	ISABEL	CALZADA	RAMOS	MUJER	EspRRHH	AG	49	6461032303	
11	303037	LAURA	ESPINOZA	LARA	MUJER	AnMktDg	NE	46	6461025987	
12	303046	FRANCISCO	PEREDES	SILVA	HOMBRE	AnlSist	BC	26	6461027823	
13	303049	JULIA	HERRERA	DURAN	MUJER	Arquitect	ZS	50	6461010674	
14	303348	GLORIA	SOLANO	DURAN	MUJER	IngRed	DG	42	6461019648	
15	303718	RAUL	LEON	IGLESIAS	HOMBRE	AnlVent	HG	18	6461031188	
16	303724	OSCAR	VALDES	SERRANO	HOMBRE	AsistAdm	MC	25	6461030117	
17	304303	CLARA	ESPINOSA	ESPINOSA	MUJER	Arquitect	TL	50	6461026428	
18	304441	SUSANA	FLORES	LEAL	MUJER	PsOrg	DF	23	6461023576	
19	304632	SOFIA	JIMENEZ	SOLANO	MUJER	RepVInt	TS	36	6461017541	
20	304724	LOURDES	CASILLAS	SILVA	MUJER	Traduct	OC	26	6461010959	
21	305050	FERNANDO	LEAL	PACHECO	HOMBRE	Abogado	JC	32	6461016170	
22	306376	AURORA	ROMAN	CASTILLO	MUJER	GteProj	NT	29	6461003799	
23	306499	LUIS	RAMOS	LOPEZ	HOMBRE	InvCien	NL	42	6461003365	
24	306583	RAQUEL	SOTO	VARGAS	MUJER	RepVInt	MN	32	6461007447	
25	306688	JUAN	VILLANUEVA	VARGAS	HOMBRE	ChefEje	MC	31	6461015159	
26	306693	ALBERTO	CHACON	VALENZUELA	HOMBRE	IngElec	NE	41	6461014278	
27	307016	CARMEN	SOSA	GONZALES	MUJER	ConsFin	NL	26	6461005331	
28	307790	GUILLERMO	CHACON	MENDEZ	HOMBRE	EspLog	NT	37	6461031893	
29	308035	ALEJANDRA	BAUTISTA	SUAREZ	MUJER	Contado	DF	34	6461011906	
30	308035	GUILLERMO	VARELA	ZAMORA	HOMBRE	AnlDatos	YN	34	6461030880	
31	308558	ADRIANA	VARELA	GARZA	MUJER	EspERen	YN	28	6461009558	
32	308636	ROSA	TORRES	VASQUEZ	MUJER	Abogado	GT	18	6461020800	
33	308693	ANTONIA	GUZMAN	TORRES	MUJER	IngRed	NL	40	6461019286	
34	308847	ALBERTO	ARIAS	LEAL	HOMBRE	EspSeg	JC	36	6461004163	
35	308942	MERCEDES	ROMERO	PENA	MUJER	EspSeg	VZ	26	6461025008	
36	309153	SUSANA	RIOS	CAMACHO	MUJER	AsistAdm	NE	18	6461030521	
37	309277	PAULA	VALENCIA	GONZALES	MUJER	ConsFin	CC	32	6461017757	
38	309730	LOURDES	LEON	CAMACHO	MUJER	InvCien	DG	42	6461022421	
39	309945	CARMEN	ROMERO	HERNANDEZ	MUJER	ProfPrim	CM	36	6461011206	
40	310547	NATALIA	LEAL	JIMENEZ	MUJER	ChefEje	CH	36	6461022847	
41	310646	MANUEL	GONZALES	VEGA	HOMBRE	ProfPrim	ZS	45	6461001877	
42	311780	ISRAEL	CASILLAS	GOMEZ	HOMBRE	AnlDatos	SR	22	6461017542	
43	311878	BEATRIZ	CALZADA	REYES	MUJER	Contado	JC	47	6461022294	
44	312031	FERNANDO	MENDOZA	BARRIOS	HOMBRE	EspSeg	SP	44	6461025115	
45	312549	EMILIO	BELTRAN	CASTILLO	HOMBRE	EspRRHH	CH	19	6461030613	
46	312749	ARMANDO	NUNEZ	CERVANTES	HOMBRE	ChefEje	AG	19	6461002629	
47	313059	BEATRIZ	LARA	VILLANUEVA	MUJER	Traduct	HG	32	6461001012	
48	313195	SILVIA	MACIEL	GONZALES	MUJER	TecSup	BC	22	6461010780	
49	313660	MARIO	ALVARADO	VASQUEZ	HOMBRE	TrabSoc	GR	21	6461019521	
50	314232	GABRIELA	VARGAS	ROMERO	MUJER	AsistAdm	MC	30	6461020381	
51	314513	PEDRO	FLORES	CASILLAS	HOMBRE	ProfPrim	YN	45	6461024841	
52	314701	RAQUEL	LARA	HERRERA	MUJER	TecSup	DG	49	6461003094	
53	314741	ARTURO	MOLINA	SALAZAR	HOMBRE	AnlSist	SR	37	6461005652	
54	315032	ARMANDO	VALENCIA	CASILLAS	HOMBRE	Traduct	GT	42	6461023086	
55	315100	OSCAR	SUAREZ	QUINTERO	HOMBRE	AnlDatos	ZS	39	6461005112	
56	315136	RAQUEL	ARIAS	MEDINA	MUJER	EspSeg	TL	25	6461011913	
57	315167	RAUL	ALVAREZ	GUZMAN	HOMBRE	TecSup	CC	48	6461025897	
58	315275	CARLOS	TOVAR	SOTO	HOMBRE	IngRed	PL	26	6461030514	
59	315546	GABRIELA	CAMACHO	CERVANTES	MUJER	AnlSist	NL	50	6461013559	
60	315613	SUSANA	PEREDES	VALENCIA	MUJER	ConsFin	OC	18	6461010099	
61	315749	PEDRO	MORA	MALDONADO	HOMBRE	RepVInt	BC	33	6461009969	
62	315780	ROSA	ALVARADO	SOLANO	MUJER	InvCien	JC	31	6461005837	
63	316936	SUSANA	MERCADO	FERNANDEZ	MUJER	EspLog	SR	41	6461013509	

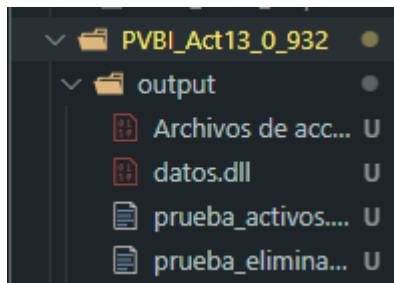
```

373 void generateTXT(TWrkr employee[], int position)
374 {
375     char fileName[30];
376     char temp[30];
377     if (position > 0)
378     {
379         do
380         {
381             ask("Ingresa el nombre con el que deseas guardarlo: ", fileName);
382             } while (alfaSpace(fileName) == -1 || strlen(fileName) > 30);
383             strcpy(temp, fileName);
384             strcat(fileName, "_activos.txt");
385             getTXT(employee, position, fileName, 1);
386             strcat(temp, "_eliminados.txt");
387             getTXT(employee, position, temp, 0);
388             printf("Archivos generados exitosamente!\n");
389         }
390     else
391     {
392         printf("El vector no tiene nada para guardar:\n");
393     }
394 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Ingresa el nombre con el que deseas guardarlo: prueba
Archivos generados exitosamente!
Presione una tecla para continuar . . .



```

396 void showTxTFile()
397 {
398     char fileName[30];
399     do
400     {
401         ask("Ingresa el nombre con el que deseas guardarlo: ", fileName);
402     } while (alfaSpace(fileName) == -1 || strlen(fileName) > 30);
403     strcat(fileName, ".txt");
404     loadFileEmployee(fileName);
405 }

```

PROBLEMSTERMINALOUTPUTPORTSGITLNSCOMMENTSDEBUG CONSOLE

Ingresa el nombre del archivo que deseas mostrar (txt): prueba_activos

No.	Matricula	Nombre	Ap. Paterno	Ao. Materno	Sexo	Posicion	Estado	Edad	Num. Cel
1	300467	ISRAEL	VALENCIA	MENENDEZ	HOMBRE	AnlVent	JC	26	6461021137
2	301240	PAULA	HERNANDEZ	CABRERA	MUJER	Contado	MN	28	6461016520
3	301394	ENRIQUE	MEDINA	GOMEZ	HOMBRE	TrabSoc	BS	20	6461011172
4	301722	MARTA	SOTO	ALVARADO	MUJER	GteProj	MS	39	6461011794
5	301962	JOSE	ROLDAN	HERRERA	HOMBRE	EdCont	MC	20	6461031653

```

407 void getBinaryFile(TWrkr employees[], int position)
408 {
409     FILE *fa;
410
411     char nomArchivo[11] = "datos.tmp";
412
413     rename("datos.tmp", "datos.dll");
414
415     fa = fopen(nomArchivo, "wb");
416
417     if (fa)
418     {
419         fwrite(employees, sizeof(TWrkr), position, fa);
420         fclose(fa);
421         printf("Archivo creado exitosamente.\n");
422     }
423     else
424     {
425         printf("No se pudo crear el archivo\n");
426     }
427 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

Archivo creado exitosamente.
 Presione una tecla para continuar . . .

output

- Archivos de acc... U
- datos.dll U
- datos.tmp U

```

429 int readFileBinary(TWrkr employees[], int position)
430 {
431     FILE *fa;
432     TWrkr reg;
433
434     fa = fopen("datos.dll", "rb");
435
436     if (fa)
437     {
438         while (fread(&reg, sizeof(TWrkr), 1, fa))
439         {
440             employees[position] = reg;
441             position++;
442         }
443         printf("El archivo ha sido cargado correctamente\n");
444         fclose(fa);
445     }
446     else
447     {
448         printf("No se pudo abrir\n");
449     }
450     return position;
451 }

```

PROBLEMS **TERMINAL** OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

El archivo ha sido cargado correctamente
Presione una tecla para continuar . . .

PROBLEMS	TERMINAL	OUTPUT	PORTS	GITLENS	COMMENTS	DEBUG CONSOLE			
No.	Matricula	Nombre	Ap. Paterno	Ao. Materno	Sexo	Posicion	Estado	Edad	Num. Cel
1	332456	VICTORIA	DELGADO	CHACON	MUJER	AnlDatos	SL	21	6461025577
2	304799	ELENA	NUNEZ	MIRANDA	MUJER	MedGen	BS	20	6461009827
3	302729	OSCAR	JIMENEZ	FERNANDEZ	HOMBRE	AnlVent	AG	19	6461004892
4	309948	VICTORIA	ARIAS	ESPINOSA	MUJER	Traduct	YN	37	6461018236
5	316772	BEATRIZ	ESTRELLA	ESTRELLA	MUJER	EspERen	GR	30	6461028995
6	311756	SUSANA	CRUZ	PAREDES	MUJER	AsServ	BC	36	6461020735
7	331848	JUAN	SOSA	ROJAS	HOMBRE	AsServ	PL	30	6461024465
8	312355	LAURA	SUAREZ	MEDINA	MUJER	EdCont	VZ	28	6461003406

```

453 void showBinaryDlted()
454 {
455     FILE *fa;
456     Twrkr reg;
457     int i = 0;
458     fa = fopen("datos.dll", "rb");
459     if (fa)
460     {
461         printf("%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-10s %-5s %-10s\n",
462             "No.",
463             "Matricula",
464             "Nombre",
465             "Ap. Paterno",
466             "Ao. Materno",
467             "Sexo",
468             "Posicion",
469             "Estado",
470             "Edad",
471             "Num. Cel");
472         while (fread(&reg, sizeof(Twrkr), 1, fa))
473         {
474             if (reg.status == 0)
475             {
476                 displayListEmp(reg, i);
477                 i++;
478             }
479         }
480         fclose(fa);
481     }
482     else
483     {
484         printf("El archivo no pudo cargarse\n");
485     }
486 }

```

PROBLEMS TERMINAL OUTPUT PORTS GITLENS COMMENTS DEBUG CONSOLE

No.	Matricula	Nombre	Ap. Paterno	Ao. Materno	Sexo	Posicion	Estado	Edad	Num. Cel
1	313511	ANTONIO	VEGA	AGUILAR	HOMBRE	EspRRHH	TC	24	6461015309
2	313557	FERNANDO	GONZALES	VILLANUEVA	HOMBRE	ConsFin	QT	21	6461029462

Presione una tecla para continuar . . .

FUNCIONES ADICIONALES:

```
489 Twrkr getOneEmployee(Twrkr employee[], int position, int ordFlag)
490 {
491     Twrkr tempEmployee;
492     int cellPhone[position];
493     int enrollment[position];
494
495     tempEmployee.age = numRandom(18, 50);
496     int i;
497     for (i = 0; i < position; i++)
498     {
499         cellPhone[i] = employee[i].cellPhone;
500         enrollment[i] = employee[i].enrollment;
501     }
502
503     do
504     {
505         tempEmployee.cellPhone = numRandom(1000000, 1999999);
506     } while (existElem(cellPhone, position, tempEmployee.cellPhone) != -1);
507
508     if (ordFlag)
509     {
510         do
511         {
512             tempEmployee.enrollment = numRandom(300000, 399999);
513             } while (binarySearch(enrollment, 0, position, tempEmployee.enrollment) != -1);
514         }
515     else
516     {
517         do
518         {
519             tempEmployee.enrollment = numRandom(300000, 399999);
520             } while (existElem(enrollment, position, tempEmployee.enrollment) != -1);
521         }
522
523     if (numRandom(0, 1))
524     {
525         nameMen(tempEmployee.name);
526         strcpy(tempEmployee.sex, "HOMBRE");
527     }
528     else
529     {
530         nameWomen(tempEmployee.name);
531         strcpy(tempEmployee.sex, "MUJER");
532     }
533
534     LastName(tempEmployee.LastName1);
535     LastName(tempEmployee.LastName2);
536     getState(tempEmployee.state);
537     getJobPositions(tempEmployee.JobPstion);
538     tempEmployee.status = 1;
539
540     return tempEmployee;
541 }
```

```
543 int existElem(int employee[], int longi, Tkey num)
544 {
545     int i;
546     for (i = 0; i < longi; i++)
547     {
548         if (employee[i] == num)
549         {
550             return i;
551         }
552     }
553     return -1;
554 }
555
556 int binarySearch(int employee[], int left, int right, Tkey number)
557 {
558     int medium;
559     while (left <= right)
560     {
561         medium = left + (right - left) / 2;
562
563         if (employee[medium] == number)
564         {
565             return medium;
566         }
567
568         if (employee[medium] < number)
569         {
570             left = medium + 1;
571         }
572         else
573         {
574             right = medium - 1;
575         }
576     }
577
578     return -1;
579 }
```

```

581 void displayRegEmp(TWrkr employee)
582 {
583     printf("Matricula:  %d\n", employee.enrollment);
584     printf("Nombre:     %s\n", employee.name);
585     printf("Ap. Paterno: %s\n", employee.LastName1);
586     printf("Ap. Materno: %s\n", employee.LastName2);
587     printf("Sexo:       %s\n", employee.sex);
588     printf("Edad:        %d\n", employee.age);
589     printf("Posicion:    %s\n", employee.JobPstion);
590     printf("Num. Cel:     646-%d\n", employee.cellPhone);
591     printf("Lug. Nacim:  %s\n", employee.state);
592 }
593
594 void displayListEmp(TWrkr employee, int i)
595 {
596     printf("%-10d %-12d %-15s %-20s %-15s %-10s %-10s %-10s %-5d 646%-10d\n",
597           i + 1,
598           employee.enrollment,
599           employee.name,
600           employee.LastName1,
601           employee.LastName2,
602           employee.sex,
603           employee.JobPstion,
604           employee.state,
605           employee.age,
606           employee.cellPhone);
607 }

```

```

609 void getNumEnrollment(TWrkr employees[], int num1[], int position)
610 {
611     int i;
612     for (i = 0; i < position; i++)
613     {
614         num1[i] = employees[i].enrollment;
615     }
616 }
617
618 void getNumCell(TWrkr employees[], int num1[], int position)
619 {
620     int i;
621     for (i = 0; i < position; i++)
622     {
623         num1[i] = employees[i].cellPhone;
624     }
625 }

```

```

627 int menuEditRegister()
628 {
629     printf("1.- Edad\n");
630     printf("2.- Lugar de nacimiento\n");
631     printf("3.- Num. Cel\n");
632     printf("4.- Posicion de trabajo\n");
633     printf("5.- Matricula\n");
634     printf("0.- Salir\n");
635     return valid("Selecciona una opcion: ", 0, 6);
636 }
637
638 int menuRegister(Twrkr employees[], int position, int index, int enrollments[], int ordFlag)
639 {
640     int op;
641     int cellPhone[position];
642     getNumCell(employees, cellPhone, position);
643
644     do
645     {
646         printf("\n----- ENCONTRADO ----- \n");
647         displayRegEmp(employees[index]);
648         printf("----- \n");
649         printf("MODIFICAR ALGUN CAMPO\n");
650         op = menuEditRegister();
651         system("CLS");
652         switch (op)
653         {
654             case 1:
655                 employees[index].age = valid("Ingrese la edad correcta: ", 18, 40);
656                 break;
657             case 2:
658                 displayStates(employees[index].state);
659                 break;
660             case 3:
661                 do
662                 {
663                     employees[index].cellPhone = valid("Ingrese el numero de telefono: 646-", 1000000, 1999999);
664                 } while (existElem(cellPhone, position, employees[index].cellPhone) != -1);
665                 break;
666             case 4:
667                 displayJobPositions(employees[index].JobPstion);
668                 break;
669             case 5:
670                 do
671                 {
672                     employees[index].enrollment = valid("Ingrese la nueva matricula: ", 300000, 399999);
673                 } while (existElem(enrollments, position, employees[index].enrollment) != -1);
674                 ordFlag = 0;
675                 break;
676             }
677         if (op != 0)
678         {
679             system("PAUSE");
680             system("CLS");
681         }
682     } while (op != 0);
683
684     return ordFlag;
685 }

```

```

687 void getTXT(Twrkr employee[], int position, char fileName[], int flag)
688 {
689     int i;
690     FILE *fa;
691     int j;
692     fa = fopen(fileName, "w");
693
694     if ((strcmp(fileName, "activos.txt") != 0) && (strcmp(fileName, "eliminados.txt") != 0))
695     {
696         fprintf(fa, "%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-8s %-5s %-10s\n",
697             "No.",
698             "Matricula",
699             "Nombre",
700             "Ap. Paterno",
701             "Ao. Materno",
702             "Sexo",
703             "Posicion",
704             "Estado",
705             "Edad",
706             "Num. Cel");
707     }
708     j = 0;
709     for (i = 0; i < position; i++)
710     {
711         if (employee[i].status == flag)
712         {
713             addOneEmployeeTxT(employee[i], j, fa);
714             j++;
715         }
716     }
717
718     fclose(fa);
719 }
720
721 void addOneEmployeeTxT(Twrkr employee, int position, FILE *fa)
722 {
723     fprintf(fa, "%-10d %-12d %-15s %-20s %-15s %-10s %-30s %-20s %-5d 646%-10d\n",
724         position + 1,
725         employee.enrollment,
726         employee.name,
727         employee.LastName1,
728         employee.LastName2,
729         employee.sex,
730         employee.JobPstion,
731         employee.state,
732         employee.age,
733         employee.cellPhone);
734 }

```

```

736 void loadFileEmployee(char fileName[])
737 {
738     int i;
739     FILE *fa;
740     TWkr reg;
741     int temp = 0;
742     int tempNo = 0;
743     char firstLine[200];
744     fa = fopen(fileName, "r");
745     i = 0;
746     if (fa)
747     {
748         fgets(firstLine, sizeof(firstLine), fa);
749         printf("%-10s %-12s %-15s %-20s %-15s %-10s %-10s %-10s %-5s %-10s\n",
750             "No.",
751             "Matricula",
752             "Nombre",
753             "Ap. Paterno",
754             "Ao. Materno",
755             "Sexo",
756             "Posicion",
757             "Estado",
758             "Edad",
759             "Num. Cel");
760         while (!feof(fa))
761         {
762             temp = fscanf(fa, "%d %d %s %s %s %s %s %s %d 646%d", &tempNo,
763                 &reg.enrollment,
764                 reg.name,
765                 reg.LastName1,
766                 reg.LastName2,
767                 reg.sex,
768                 reg.JobPstion,
769                 reg.state,
770                 &reg.age,
771                 &reg.cellPhone);
772
773             if (temp == 10)
774             {
775                 displayListEmp(reg, i);
776                 i++;
777             }
778         }
779     }
780     else
781     {
782         printf("El archivo no existe\n");
783     }
784     fclose(fa);
785 }

```

```

787 //***** ORDER FUNCTIONS *****/
788 void swap(TWrkr employees[], int i, int j)
789 {
790     TWrkr temp = employees[i];
791     employees[i] = employees[j];
792     employees[j] = temp;
793 }
794
795 int partition(TWrkr employees[], int low, int high)
796 {
797     TWrkr pivot;
798     pivot.enrollment = employees[high].enrollment;
799     int i = low - 1;
800
801     for (int j = low; j <= high - 1; j++)
802     {
803         if (employees[j].enrollment <= pivot.enrollment)
804         {
805             i++;
806             swap(employees, i, j);
807         }
808     }
809     swap(employees, i + 1, high);
810     return i + 1;
811 }
812
813 void quicksort(TWrkr employees[], int low, int high)
814 {
815     if (low < high)
816     {
817         int pi = partition(employees, low, high);
818
819         quicksort(employees, low, pi - 1);
820         quicksort(employees, pi + 1, high);
821     }
822 }
823
824 void bubbleSort(TWrkr employees[], int n)
825 {
826     int i, j;
827     TWrkr temp;
828     for (i = 0; i < n - 1; i++)
829     {
830         for (j = i + 1; j < n; j++)
831         {
832             if (employees[j].enrollment < employees[i].enrollment)
833             {
834                 temp = employees[i];
835                 employees[i] = employees[j];
836                 employees[j] = temp;
837             }
838         }
839     }
840 }

```