



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño



PROGRAMACIÓN ESTRUCTURADA

**FUNCIONES, METODOS DE ORDENACION Y
BUSQUEDA**

ANEXO - PRÁCTICA 10

Ingeniero en Software y Tecnologías Emergentes

Brayan Ivan Perez Ventura

372781

15 de Octubre del 2023 en Ensenada, Baja California

```

80 void menu()
81 {
82     int op, position; // op will save user's input (menu);           position will be controlling the max siz
83     int flag;
84     flag = 1; // We will use this flag to know if it's ordered (use a different search method) or if it's no
85     position = 0;
86     Tstdnt studentArray[STDNTSIZE];
87     do
88     {
89         op = msge_menu();
90         system("CLS");
91         switch (op)
92         {
93             case 1:
94                 if (position + 10 >= STDNTSIZE)
95                 {
96                     printf("Has llegado a la maxima cantidad de registros permitidos\n");
97                 }
98                 else
99                 {
100                     fillReg(studentArray, position);
101                     position += 10;
102                     flag = 0;
103                     printf("\t\t\t10 registros creados correctamente!\n");
104                 }
105                 system("PAUSE");
106                 break;
107             case 2:
108                 if (position + 1 >= STDNTSIZE)
109                 {
110                     printf("Has llegado a la maxima cantidad de registros permitidos\n");
111                 }
112                 else
113                 {
114                     studentArray[position] = newStdnt(studentArray, position);
115                     flag = 0;
116                     position += 1;
117                 }
118                 system("PAUSE");
119                 break;
120             case 3:
121                 deleteStdnt(studentArray, position);
122                 flag = 0;
123                 system("PAUSE");
124                 break;
125             case 4:
126                 searchStdnt(studentArray, position, flag);
127                 system("PAUSE");
128                 break;
129             case 5:
130                 if (flag == 0)
131                 {
132                     bubbleSort(studentArray, position);
133                     flag = 1;
134                     printf("Los registros han sido ordenados correctamente por matricula de menor a mayor\n");
135                 }
136                 else
137                 {
138                     printf("Los registros ya han sido ordenados con anterioridad.\n");
139                 }
140                 system("PAUSE");
141                 break;
142             case 6:
143                 displayRegActive(studentArray, position);
144                 system("PAUSE");
145                 break;
146         }
147     } while (op != 0);
148     printf("Saliendo del programa\n");
149 }

```

```
151 int msge_menu()
152 {
153     int op;
154     system("CLS");
155     printArr("MANEJO Y CONTROL DE ESTUDIANTES\n");
156     printArr("1.- Agregar 10 registros automatico\n");
157     printArr("2.- Agregar manual\n");
158     printArr("3.- Eliminar registro\n");
159     printArr("4.- Buscar\n");
160     printArr("5.- Ordenar\n");
161     printArr("6.- Imprimir\n");
162     printArr("0.- Salir\n");
163     op = valid("Selecciona una opcion: ", 0, 6);
164     return op;
165 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
MANEJO Y CONTROL DE ESTUDIANTES
1.- Agregar 10 registros automatico
2.- Agregar manual
3.- Eliminar registro
4.- Buscar
5.- Ordenar
6.- Imprimir
0.- Salir
Selecciona una opcion: █
```

```

167 //***** MAIN SWITCH FUNCTIONS DEVELOPMENT *****
168 void fillReg(Tstdnt studentArray[], int position)
169 {
170     int i;
171     for (i = position; i < position + 10; i++)
172     {
173         studentArray[i] = genDataReg(studentArray, position);
174     }
175     displayReg(studentArray, position + 10);
176 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

No.	Matricula	Status	Ap. Paterno	Ap. Materno	Nombre	Edad	Sexo
1	303922	1	ROJAS	DURAN	GLORIA	31	M
2	328593	1	BARRIOS	ALVAREZ	EDUARDO	26	H
3	302921	1	LOPEZ	AGUILAR	JAVIER	25	H
4	326883	1	CHACON	VASQUEZ	MAR-IA	40	M
5	318318	1	SOSA	NUNEZ	RAQUEL	30	M
6	317299	1	VEGA	LEAL	SOFIA	38	M
7	324499	1	ESPINOZA	SOTO	EMILIO	39	H
8	315595	1	VALENCIA	CORDERO	ERNESTO	19	H
9	306074	1	ALVARADO	RAMIREZ	ANTONIA	31	M
10	313971	1	MENDEZ	AGUILAR	BEATRIZ	23	M

10 registros creados correctamente!

Presione una tecla para continuar . . .

```

178 | Tstdnt newStdnt(Tstdnt studentArray[], int position)
179 | {
180 |     Tstdnt tempStudentArray;
181 |
182 |     char tempSentence[30];
183 |
184 |     tempStudentArray.status = 1;
185 |
186 |     do
187 |     {
188 |         tempStudentArray.matricula = valid("Por favor, introduce la matricula del estudiante: ", 300000, 399999); // We validate user's input in a range between the
189 |     } while (existElem(studentArray, position, tempStudentArray.matricula) != -1); // If it already exists, it will be looping until t
190 |
191 |     do /*** GET LAST NAME 1 ***/
192 |     {
193 |         printf("Por favor, introduce el apellido paterno del estudiante en mayusculas: ");
194 |         ask(tempSentence);
195 |     } while (alfaSpace(tempSentence) == -1);
196 |
197 |     strcpy(tempStudentArray.LastName1, tempSentence); // Copy input to studentArray
198 |
199 |     do // **** GET LAST NAME 2 ****
200 |     {
201 |         printf("Por favor, introduce el apellido materno del estudiante en mayusculas: ");
202 |         ask(tempSentence);
203 |     } while (alfaSpace(tempSentence) == -1);
204 |
205 |     strcpy(tempStudentArray.Lastname2, tempSentence); // Copy input to studentArray
206 |
207 |     do /*** GET NAME *****
208 |     {
209 |         printf("Por favor, introduce el nombre del estudiante en mayusculas: ");
210 |         ask(tempSentence);
211 |     } while (alfaSpace(tempSentence) == -1);
212 |
213 |     strcpy(tempStudentArray.name, tempSentence); // Copy input to studentArray
214 |     /*** GET STUDENT AGE ***
215 |     tempStudentArray.age = valid("Introduce la edad del estudiante: ", 18, 40);
216 |     /*** GET STUDENT SEX*****
217 |     printf("0.- Mujer\t1.- Hombre\n");
218 |     tempStudentArray.sex = valid("Introduce el sexo del estudiante: ", 0, 1);
219 |
220 |     printf("\n\tINFORMACION DEL ESTUDIANTE AGREGADO RECIENTEMENTE\n");
221 |     printf("| %-3s | %-10s | %-6s | %-12s | %-12s | %-4s | %-3s |\n", "No.", "Matricula", "Status", "Ap. Paterno", "Ap. Materno", "Nombre", "Edad", "Sexo");
222 |     displayOneStdnt(tempStudentArray, position);
223 |
224 |     return tempStudentArray;
225 | }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```

Por favor, introduce la matricula del estudiante: 312321
Por favor, introduce el apellido paterno del estudiante en mayusculas: PÉREZ
Por favor, introduce el apellido paterno del estudiante en mayusculas: PEREZ
Por favor, introduce el apellido materno del estudiante en mayusculas: ventura
Por favor, introduce el apellido materno del estudiante en mayusculas: VENTURA
Por favor, introduce el nombre del estudiante en mayusculas: BRAYAN IVAN
Por favor, introduce el nombre del estudiante en mayusculas: BRAYAN IVAN
Por favor, introduce el nombre del estudiante en mayusculas: BRAYAN IVAN
Introduce la edad del estudiante: 19
0.- Mujer      1.- Hombre
Introduce el sexo del estudiante: 1

```

INFORMACION DEL ESTUDIANTE AGREGADO RECIENTEMENTE

No.	Matricula	Status	Ap. Paterno	Ap. Materno	Nombre	Edad	Sexo
11	312321	1	PEREZ	VENTURA	BRAYAN IVAN	19	H

Presione una tecla para continuar . . .

```

227 void deleteStdnt(Tstdnt studentArray[], int position)
228 {
229     int num, index;
230     num = valid("Ingrese la matricula del estudiante que desea eliminar: ", 300000, 399999);
231     index = existElem(studentArray, position, num);
232     if (index != -1)
233     {
234         if (studentArray[index].status != 0) // Student has been found and also it's actived (
235         {
236             studentArray[index].status = 0;
237             printf("El estudiante ha sido dado de baja correctamente.\n");
238         }
239         else // Student has been found, but it's already inactived (0)
240         {
241             printf("El estudiante ya estaba dado de baja con anterioridad.\n");
242         }
243     }
244     else // Student wasn't found
245     {
246         printf("La matricula ingresada no pertecene a ningun estudiante\n");
247     }
248 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```

Ingrese la matricula del estudiante que desea eliminar: 312321
El estudiante ha sido dado de baja correctamente.
Presione una tecla para continuar . . . █

```

```

250 void searchStdnt(Tstdnt studentArray[], int position, int flag)
251 {
252     int num, index;
253     num = valid("Ingrese la matricula del estudiante que desea buscar: ", 300000, 399999);
254     if (flag == 0) // If it's not ordered from smallest to largest
255     {
256         index = existElem(studentArray, position, num);
257     }
258     else // if so
259     {
260         index = binarySearch(studentArray, 0, position - 1, num);
261     }
262     if (index != -1) // Check what function returned
263     {
264         printf("El estudiante ha sido encontrado con exito\n");
265         printf("| %-3s | %-10s | %-6s | %-12s | %-12s | %-12s | %-4s | %-3s |\n", "No.", "Matricula", "Status",
266         "Ap. Paterno", "Ap. Materno", "Nombre", "Edad", "Sexo");
267         displayOneStdnt(studentArray[index], index);
268     }
269     else
270     {
271         printf("El estudiante no ha sido encontrado\n");
272     }
273 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

Ingrese la matricula del estudiante que desea buscar: 312321

El estudiante ha sido encontrado con exito

No.	Matricula	Status	Ap. Paterno	Ap. Materno	Nombre	Edad	Sexo
11	312321	0	PEREZ	VENTURA	BRAYAN IVAN	19	H

Presione una tecla para continuar . . .

```
274 void bubbleSort(Tstdnt studentArray[], int n)
275 {
276     int i, j;
277     Tstdnt temp;
278     for (i = 0; i < n - 1; i++)
279     {
280         for (j = i + 1; j < n; j++)
281         {
282             if (studentArray[j].matricula < studentArray[i].matricula)
283             {
284                 temp = studentArray[i];
285                 studentArray[i] = studentArray[j];
286                 studentArray[j] = temp;
287             }
288         }
289     }
290 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

Los registros han sido ordenados correctamente por matricula de menor a mayor
Presione una tecla para continuar . . .


```

293 void displayRegActive(Tstdnt studentArray[], int position)
294 {
295     int i;
296     printf("| %-3s | %-10s | %-6s | %-12s | %-12s | %-12s | %-4s | %-3s |\n", "No.",
297         "Matricula", "Status", "Ap. Paterno", "Ap. Materno", "Nombre", "Edad", "Sexo");
298     for (i = 0; i < position; i++)
299     {
300         if (studentArray[i].status == 1)
301         {
302             displayOneStdnt(studentArray[i], i);
303         }
304     }
305 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

DESPLEGANDO LISTA DE ALUMNOS ACTIVOS ACTUALMENTE

No.	Matricula	Status	Ap. Paterno	Ap. Materno	Nombre	Edad	Sexo
1	300501	1	TORRES	SOTO	FRANCISCO	31	H
2	302887	1	ALVAREZ	ESCOBAR	BEATRIZ	37	M
3	302941	1	RAMIREZ	RIOS	OSCAR	38	H
4	310935	1	ALVARADO	GUERRERO	ALBERTO	20	H
5	310972	1	CHACON	SOLANO	EDUARDO	34	H
6	312228	1	RUBIO	LARA	BEATRIZ	25	M
7	314848	1	RAMOS	ALVARADO	PATRICIA	37	M
8	317702	1	MENDEZ	GOMEZ	JUAN	29	H
10	324801	1	SALAZAR	CAMACHO	RAUL	28	H
11	329657	1	IGLESIAS	MUNGUIA	AURORA	19	M

Presione una tecla para continuar . . .

```

306 Tstdnt genDataReg(Tstdnt studentArray[], int position)
307 {
308     Tstdnt tempStudentArray;
309     // *** GET LAST NAME ***
310     char tempName[50];
311
312     //*** COPY LAST NAME 1 AND 2 ***
313     // LAST NAME 1
314     LastName(tempName);
315     strcpy(tempStudentArray.LastName1, tempName);
316
317     // LAST NAME 2
318     LastName(tempName);
319     strcpy(tempStudentArray.Lastname2, tempName);
320
321     //*** STATUS ***
322     tempStudentArray.status = 1;
323     You, yesterday • modified: PVBI_Act10_0_932/Frijoles.h ...
324     //**** STUDENT REGISTRATION ****
325     do
326     {
327         tempStudentArray.matricula = numRandom(300000, 399999);
328     } while (existElem(studentArray, position, tempStudentArray.matricula) != -1);
329
330     //**** GENERATE SEX AND NAME ****
331     if (numRandom(0, 1) == 1)
332     {
333         nameMen(tempName);
334         strcpy(tempStudentArray.name, tempName);
335         tempStudentArray.sex = 1; // MEN = 1
336     }
337     else
338     {
339         nameWomen(tempName);
340         strcpy(tempStudentArray.name, tempName);
341         tempStudentArray.sex = 0; // WOMEN = 0
342     }
343
344     //**** GENERATE AGE ****
345     tempStudentArray.age = numRandom(18, 40);
346
347     return tempStudentArray;
348 }

```

```

350 void displayReg(Tstdnt studentArray[], int position)
351 {
352     int i;
353     printf("| %-3s | %-10s | %-6s | %-12s | %-12s | %-12s | %-4s | %-3s |\n", "No.",
354     "Matricula", "Status", "Ap. Paterno", "Ap. Materno", "Nombre", "Edad", "Sexo");
355
356     for (i = 0; i < position; i++)
357     {
358         printf("| %-3d | %-10d | %-6d | %-12s | %-12s | %-12s | %-4d |", i + 1,
359         studentArray[i].matricula, studentArray[i].status, studentArray[i].LastName1,
360         studentArray[i].Lastname2, studentArray[i].name, studentArray[i].age);
361         if (studentArray[i].sex == 0)
362         {
363             printf(" M   |\n");
364         }
365
366         else
367         {
368             printf(" H   |\n");
369         }
370     }
371     printf("\n");
372 }

```

```

374 ✓ int existElem(Tstdnt studentArray[], int longi, int num)
375 {
376     int i;
377     ✓ for (i = 0; i < longi; i++)
378     {
379     ✓         if (studentArray[i].matricula == num)
380             {
381                 return i;
382             }
383     }
384     return -1;
385 }

```

```

387 ✓ void displayOneStdnt(Tstdnt studentArray, int position)
388 {
389     printf("| %-3d | %-10d | %-6d | %-12s | %-12s | %-12s | %-4d |", position + 1,
390     studentArray.matricula, studentArray.status,
391     studentArray.LastName1, studentArray.Lastname2, studentArray.name, studentArray.age);
392     ✓ if (studentArray.sex == 0)
393     {
394         printf(" M   |\n");
395     }
396     ✓ else
397     {
398         printf(" H   |\n");
399     }
400 }

```

```

430 void nameMen(char tempName[])
431 {
432     char nameMen1[30][30] = {"JUAN", "PEDRO", "LUIS",
433     "MIGUEL", "CARLOS", "JAVIER", "MANUEL", "JOSE",
434     "FERNANDO", "ALBERTO", "RICARDO", "ENRIQUE", "ANTONIO",
435     "ALEJANDRO", "EMILIO", "ARTURO", "JORGE", "EDUARDO", "HECTOR", "FRANCISCO",
436     "RAUL", "ROBERTO", "ERNESTO", "GUILLERMO", "ARMANDO", "MARIO", "DANIEL", "OSCAR",
437     "ISRAEL", "SALVADOR"};
438     strcpy(tempName, nameMen1[numRandom(0, 29)]);
439 }
440
441 void nameWomen(char tempName[])
442 {
443     char nameWomen1[30][30] = {"MARÍA", "ANA", "LAURA", "PATRICIA",
444     "CARMEN", "SOFIA", "ISABEL", "PAULA", "BEATRIZ", "ELENA", "GABRIELA",
445     "ROSA", "CLARA", "VICTORIA", "LOURDES", "ADRIANA", "NATALIA", "SUSANA",
446     "MARTA", "PILAR", "SONIA", "ALEJANDRA", "JULIA", "RAQUEL", "ANTONIA", "GLORIA",
447     "SILVIA", "AURORA", "CONSUELO", "MERCEDES"};
448     strcpy(tempName, nameWomen1[numRandom(0, 29)]);
449 }
450
451 void LastName(char tempLastName[])
452 {
453     char LastName1[120][60] = {"RAMOS", "MORENO", "FERNANDEZ", "TORRES",
454     "RAMIREZ", "JIMENEZ", "NUNEZ", "VEGA", "ROJAS",
455     "IGLESIAS", "PACHECO", "VALENCIA", "MORA", "SILVA",
456     "GOMEZ", "CORDERO", "SERRANO", "MEDINA", "ALVAREZ", "SOTO", "LARA",
457     "HERRERA", "GUERRERO", "ORTIZ", "PARDO", "ROLDAN", "SUAREZ", "SALAZAR",
458     "CASTILLO", "AGUILAR", "ROMAN", "ZAMORA", "DIAZ", "CASTANEDA", "VARGAS",
459     "QUINTERO", "MOLINA", "CABRERA", "GONZALES", "CRUZ", "DELGADO", "VILLANUEVA", "RIOS",
460     "REYES", "FLORES", "ROJAS", "ARIAS", "LOPEZ", "MALDONADO", "MENDEZ", "CERVANTES", "ESPINOSA",
461     "CHACON", "SOLIS", "TOVAR", "SANDOVAL", "VALENCIA", "RIVERA", "ROSALES", "DURAN", "RUBIO",
462     "MENDOZA", "BAUTISTA", "VASQUEZ", "ROMERO", "MERCADO", "ESPINOSA", "ESCOBAR", "SOSA",
463     "BARRIOS", "LEON", "SOTO", "OCHOA", "CONTRERAS", "CERVANTES", "VALENZUELA", "MIRANDA",
464     "PAREDES", "BELTRAN", "ESPINOZA", "PENNA", "CASILLAS", "VARELA", "HIDALGO", "GUZMAN",
465     "MENENDEZ", "ALVARADO", "ESTRELLA", "HERNANDEZ", "OSORIO", "MACIAS", "URIBE", "GARZA",
466     "VALDES", "CAMACHO", "LEAL", "MUNGUIA", "SOLANO", "MACIEL", "CALZADA", "ALCALA", "BARAJAS",
467     "PARRA", "ELIZONDO", "PALACIOS", "CASAS", "VENTURA", "MONTES", "MUNOZ", "LUGO", "SOSA", "OJEDA",
468     "FRANCO", "ARELLANO", "PIZARRO", "CHAVEZ", "ROBLES", "CARRANZA", "URBINA"};
469     strcpy(tempLastName, LastName1[numRandom(0, 99)]);
470 }

```

```
402 int binarySearch(Tstdnt studentArray[], int left, int right, int number)
403 {
404     int medium;
405     while (left <= right)
406     {
407         medium = left + (right - left) / 2;
408
409         // Check if number is present at mid
410         if (studentArray[medium].matricula == number)
411         {
412             return medium;
413         }
414
415         if (studentArray[medium].matricula < number) // If number greater, ignore left half
416         {
417             left = medium + 1;
418         }
419         else // If number is smaller, ignore right half
420         {
421             right = medium - 1;
422         }
423     }
424
425     // If we reach here, then element was not present
426     return -1;
427 }
```