

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes
Programación Estructurada

Actividad 10. FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA ESTRUCTURAS Y LIBRERIAS

ALUMNO: Fernando Haro Calvo

MATRICULA: 372106

GRUPO: 932

PROFESOR: Pedro Núñez Yépiz

14 de octubre del 2023

Funciones Auxiliares (Librería):

```
27 // Valida la entrada del usuario en un rango de numeros enteros.
28 int valiNum(int ri, int rf)
29 {
30     char cadena[50];
31     int num;
32
33     do
34     {
35         fflush(stdin);
36         gets(cadena);
37         num = atoi(cadena);
38     } while (num < ri || num > rf);
39
40     return num;
41 }
```

```
76 // Valida que la cadena solo tenga caracteres alfabeticos
77 int valiAlfa(char cadena[])
78 {
79     int i = 0;
80
81     while (cadena[i] != '\0')
82     {
83         if (cadena[i] != ' ')
84         {
85             if (cadena[i] < 'A' || cadena[i] > 'Z')
86             {
87                 return 0; // La cadena contiene un carácter no válido
88             }
89         }
90
91         i++;
92     }
93
94     return 1;
95 }
```

```

43 // Valida que una cadena no inicie o termine con espacios, o contenga dobles espacios.
44 int valiEspacios(char cadena[])
45 {
46     int i = 0;
47
48     // Inicia con espacio
49     if (cadena[0] == ' ') // no puede iniciar con espacios
50     {
51         return 0;
52     }
53
54     // Termina con espacio
55     if (cadena[largoCadena(cadena)] == ' ')
56     {
57         return 0;
58     }
59
60     // 2 espacios seguidos
61     while (cadena[i] != '\0')
62     {
63         if (cadena[i] == ' ')
64         {
65             if (cadena[i + 1] == ' ')
66             {
67                 return 0;
68             }
69         }
70         i++;
71     }
72
73     return 1;
74 }

```

```

97 // Valida que no haya caracteres especiales en una cadena.
98 int valiCad(char cadena[])
99 {
100     mayus(cadena);
101
102     if (valiAlfa(cadena) == 0)
103     {
104         return 0;
105     }
106
107     if (valiEspacios(cadena) == 0)
108     {
109         return 0;
110     }
111
112     return 1;
113 }

```

```
268 // Convierte la cadena a mayusculas.
269 void mayus(char cadena[])
270 {
271     // Variables Locales
272     int i;
273
274     // Programa
275     for (i = 0; cadena[i] != '\0'; i++)
276     {
277         char character = cadena[i];
278
279         if (character >= 'a' && character <= 'z')
280         {
281             character -= 32; // Convertir a mayúscula (ASCII)
282         }
283
284         cadena[i] = character;
285     }
286 }
```

Ejercicio 1:

```
85     case 1:
86         if ((alumnos + 10) <= N)
87         {
88             for (i = 0; i < 10; i++)
89             {
90                 alum = genAlumAlea();
91
92                 while (busqSeqMatri(ingenieros, alumnos, alum.matricula) != -1) // Valida matricula unica
93                 {
94                     alum.matricula = numAleatorio(300000, 399999);
95                 }
96
97                 ingenieros[alumnos] = alum;
98                 alumnos++;
99             }
100             band = false;
101         }
102         else
103         {
104             printf("Ha llegado al maximo de alumnos\n");
105         }
106
107         break;

```

```
283 // Genera n cantidad de registros de alumnos aleatoriamente dentro del vector de alumnos especificado.
284 Talum genAlumAlea()
285 {
286     Talum alum;
287
288     alum.status = 1;
289     alum.matricula = numAleatorio(300000, 399999);
290     strcpy(alum.apPat, apellidos[numAleatorio(0, 39)]);
291     strcpy(alum.apMat, apellidos[numAleatorio(0, 39)]);
292     alum.edad = numAleatorio(18, 30);
293     alum.sexo = numAleatorio(1, 2);
294
295     if (alum.sexo == 1)
296     {
297         strcpy(alum.nombre, nombresH[numAleatorio(0, 19)]);
298     }
299     else
300     {
301         strcpy(alum.nombre, nombresM[numAleatorio(0, 19)]);
302     }
303
304     return alum;
305 }
```

Salida 1:

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

Imprimir Registros:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
323092	CLARA	GUERRERO	ZAPATA	26	MUJER
322420	MONICA	TORRES	FUENTES	20	MUJER
312780	FRANCISCO	FERNANDEZ	SILVA	21	HOMBRE
327326	LAURA	MARTINEZ	VEGA	19	MUJER
332090	LUCIA	FERNANDEZ	NAVARRO	21	MUJER
317376	VALENTINA	ZAPATA	LOPEZ	30	MUJER
324439	ISABEL	ROJAS	REYES	29	MUJER
317878	ALEJANDRO	SANCHEZ	RAMIREZ	20	HOMBRE
319064	PATRICIA	DIAZ	HERNANDEZ	25	MUJER
317421	CARMEN	GOMEZ	YEPIZ	22	MUJER

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

Vector lleno:

```
Ha llegado al maximo de alumnos
```

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

Ejercicio 2:

```
109 case 2:
110     if ((alumnos + 1) <= N)
111     {
112         alum = genAlumMan();
113
114         while (busqSeqMatri(ingenieros, alumnos, alum.matricula) != -1) // Valida matricula unica
115         {
116             system("CLS");
117             printf("Matricula repetida, ingrese una nueva: ");
118             alum.matricula = valiNum(300000, 399999);
119         }
120
121         ingenieros[alumnos] = alum;
122         alumnos++;
123         band = false;
124     }
125     else
126     {
127         printf("Ha llegado al maximo de alumnos\n");
128     }
129
130     break;
```

```
307 Talum genAlumMan()
308 {
309     Talum alum;
310
311     printf("Ingresa el status (0- No Activo, 1- Activo): ");
312     alum.status = valiNum(0, 1);
313
314     system("CLS");
315     printf("Ingresa una matricula (300000 - 399999): ");
316     alum.matricula = valiNum(300000, 399999);
317
318     do
319     {
320         system("CLS");
321         printf("Ingresa el apellido paterno: ");
322         fflush(stdin);
323         gets(alum.apPat);
324     } while (valiCad(alum.apPat) == 0);
325
326     do
327     {
328         system("CLS");
329         printf("Ingresa el apellido materno: ");
330         fflush(stdin);
331         gets(alum.apMat);
332     } while (valiCad(alum.apMat) == 0);
333
334     do
335     {
336         system("CLS");
337         printf("Ingresa el/los nombre/s: ");
338         fflush(stdin);
339         gets(alum.nombre);
340     } while (valiCad(alum.nombre) == 0);
341
342     system("CLS");
343     printf("Ingresa la edad: ");
344     alum.edad = valiNum(17, 100);
345
346     system("CLS");
347     printf("Ingresa el sexo (1- Hombre, 2- Mujer): ");
348     alum.sexo = valiNum(1, 2);
349
350     return alum;
351 }
```


Salida 1:

```
Ingresar el status (0- No Activo, 1- Activo): 1
```

✓

```
Ingresar una matricula (300000 - 399999): -1
```

X

```
Ingresar una matricula (300000 - 399999): -1  
372106
```

✓

```
Ingresar el apellido paterno: h4ro
```

X

```
Ingresar el apellido paterno: h@ro
```

X

```
Ingresar el apellido paterno: háro
```

X

```
Ingresar el apellido paterno: haro calvo
```

X

```
Ingresar el apellido paterno: haro
```

✓

Ingresar el apellido materno: Calvo

✓

Ingresar el/los nombre/s: Luis FERNANDO

✓

Ingresar la edad: 19

✓

Ingresar el sexo (1- Hombre, 2- Mujer): 0

X

Ingresar el sexo (1- Hombre, 2- Mujer): 0
1

Presione una tecla para continuar . . .

✓

Imprimir Registros:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
372106	LUIS FERNANDO	HARO	CALVO	19	HOMBRE
Presione una tecla para continuar . . .					

Vector lleno:

Ha llegado al maximo de alumnos

Presione una tecla para continuar . . .

Ejercicio 3:

```
132     case 3:
133         printf("Ingrese la matricula del estudiante que desea eliminar: ");
134         num = valiNum(300000, 399999);
135
136         system("CLS");
137         i = busqMatri(ingenieros, alumnos, num, band);
138
139         if (i != -1)
140         {
141             if (ingenieros[i].status != 0)
142             {
143                 ingenieros[i].status = 0;
144                 printf("Matricula eliminada con exito\n");
145             }
146             else
147             {
148                 printf("El alumno ya se encuentra inactivo\n");
149             }
150         }
151         else
152         {
153             printf("Matricula no encontrada\n");
154         }
155
156         break;
```

Registros:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
326348	SOFIA	HERNANDEZ	GUTIERREZ	21	MUJER
312574	LUIS	ROMERO	FUENTES	27	HOMBRE
332376	CLARA	MORALES	GOMEZ	20	MUJER
317545	LUISA	FLORES	MORALES	19	MUJER
309495	LOURDES	GONZALEZ	VEGA	20	MUJER
327430	MARIA	CRUZ	RAMIREZ	20	MUJER
309636	LOURDES	SILVA	FUENTES	25	MUJER
328201	ANTONIO	SANCHEZ	ALVAREZ	21	HOMBRE
320354	LUCIA	MORENO	REYES	27	MUJER
304867	ELENA	ACOSTA	FLORES	20	MUJER

Presione una tecla para continuar . . .

Salida 1:

```
Ingrese la matricula del estudiante que desea eliminar: 372106
```

```
Buscando con busqueda secuencial...
```

```
Matricula no encontrada
```

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

Salida 2:

```
Ingrese la matricula del estudiante que desea eliminar: 304867
```

```
Buscando con busqueda secuencial...
```

```
Matricula eliminada con exito
```

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

Registros Actualizados:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
326348	SOFIA	HERNANDEZ	GUTIERREZ	21	MUJER
312574	LUIS	ROMERO	FUENTES	27	HOMBRE
332376	CLARA	MORALES	GOMEZ	20	MUJER
317545	LUISA	FLORES	MORALES	19	MUJER
309495	LOURDES	GONZALEZ	VEGA	20	MUJER
327430	MARIA	CRUZ	RAMIREZ	20	MUJER
309636	LOURDES	SILVA	FUENTES	25	MUJER
328201	ANTONIO	SANCHEZ	ALVAREZ	21	HOMBRE
320354	LUCIA	MORENO	REYES	27	MUJER
Presione una tecla para continuar . . .					

Ejercicio 4:

```
158 case 4:
159     printf("Ingrese la matricula del estudiante que desea buscar: ");
160     num = valiNum(300000, 399999);
161     system("CLS");
162
163     // Verifica si está ordenado el vector
164     i = busqMatri(ingenieros, alumnos, num, band);
165
166     if (i != -1)
167     {
168         printf("La matricula %d SI se encuentra en el vector\n", num);
169     }
170     else
171     {
172         printf("La matricula %d NO se encuentra en el vector\n", num);
173     }
174
175     break;
```

```
243 // Busca una matricula en un arreglo usando el algoritmo optimo
244 int busqMatri(Talum vect[], int n, int matri, bool band)
245 {
246     int i;
247     if (band)
248     {
249         printf("Buscando con busqueda binaria...\n\n");
250         i = busqBinMatri(vect, n, matri);
251     }
252     else
253     {
254         printf("Buscando con busqueda secuencial...\n\n");
255         i = busqSeqMatri(vect, n, matri);
256     }
257
258     return i;
259 }
```

```
211 // Busca una matricula en un arreglo de alumnos ORDENADO.
212 int busqBinMatri(Talum vect[], int n, int matri)
213 {
214     int med;
215     int ri = 0;
216     int rf = n;
217
218     while (ri <= rf)
219     {
220         med = ri + (rf - ri) / 2;
221
222         if (vect[med].matricula == matri)
223         {
224             return med; // Valor encontrado en indice med
225         }
226         else
227         {
228             // Verifica si es menor o mayor
229             if (matri < vect[med].matricula)
230             {
231                 rf = med--;
232             }
233             else
234             {
235                 ri = med++;
236             }
237         }
238     }
239
240     return -1; // No se encontró
241 }
```

```

195 // Busca una matricula en un arreglo de alumnos NO ORDENADO.
196 int busqSeqMatri(Talum vect[], int n, int matri)
197 {
198     int i;
199
200     for (i = 0; i < n; i++)
201     {
202         if (vect[i].matricula == matri)
203         {
204             return i; // Retorna índice
205         }
206     }
207
208     return -1; // No se encontró
209 }

```

Registros:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
326348	SOFIA	HERNANDEZ	GUTIERREZ	21	MUJER
312574	LUIS	ROMERO	FUENTES	27	HOMBRE
332376	CLARA	MORALES	GOMEZ	20	MUJER
317545	LUISA	FLORES	MORALES	19	MUJER
309495	LOURDES	GONZALEZ	VEGA	20	MUJER
327430	MARIA	CRUZ	RAMIREZ	20	MUJER
309636	LOURDES	SILVA	FUENTES	25	MUJER
328201	ANTONIO	SANCHEZ	ALVAREZ	21	HOMBRE
320354	LUCIA	MORENO	REYES	27	MUJER

Presione una tecla para continuar . . .

Salida 1:

```
Ingrese la matricula del estudiante que desea buscar: 372106
```

```
Buscando con busqueda secuencial...
```

```
La matricula 372106 NO se encuentra en el vector
```

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

Salida 2:

```
Ingrese la matricula del estudiante que desea buscar: 327430
```

```
Buscando con busqueda secuencial...
```

```
La matricula 327430 SI se encuentra en el vector
```

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


Ejercicio 5:

```
177         case 5:
178             if (band == false)
179             {
180                 band = ordMatri(ingenieros, alumnos);
181                 printf("El vector ha sido ordenado\n");
182             }
183             else
184             {
185                 printf("El vector ya estaba ordenado\n");
186             }
187             break;
```

```
261 // Ordena el vector de alumnos usando una mezcla de bubble e insertion sort.
262 bool ordMatri(Talum vect[], int n)
263 {
264     int i, j;
265     Talum temp;
266
267     for (i = 0; i < n - 1; i++)
268     {
269         for (j = i + 1; j < n; j++)
270         {
271             if (vect[j].matricula <= vect[i].matricula)
272             {
273                 temp = vect[i];
274                 vect[i] = vect[j];
275                 vect[j] = temp;
276             }
277         }
278     }
279
280     return true;
281 }
```

Salida 1:

```
El vector ha sido ordenado
```

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

Registros (Ordenados por matricula)

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
309495	LOURDES	GONZALEZ	VEGA	20	MUJER
309636	LOURDES	SILVA	FUENTES	25	MUJER
312574	LUIS	ROMERO	FUENTES	27	HOMBRE
317545	LUISA	FLORES	MORALES	19	MUJER
320354	LUCIA	MORENO	REYES	27	MUJER
326348	SOFIA	HERNANDEZ	GUTIERREZ	21	MUJER
327430	MARIA	CRUZ	RAMIREZ	20	MUJER
328201	ANTONIO	SANCHEZ	ALVAREZ	21	HOMBRE
332376	CLARA	MORALES	GOMEZ	20	MUJER

Presione una tecla para continuar . . .

Búsqueda binaria con el vector ordenado:

Ingrese la matricula del estudiante que desea buscar: 332376

Buscando con busqueda binaria...

La matricula 332376 SI se encuentra en el vector

Presione una tecla para continuar . . .

Salida 2 (Vector ordenado):

El vector ya estaba ordenado

Presione una tecla para continuar . . .

Ejercicio 6:

```
189
190
191 case 6:
    imprAlumnos(ingenieros, alumnos);
    break;
```

```
360 // Imprime todos los alumnos en un vector dado.
361 void imprAlumnos(Talum vect[], int n)
362 {
363     int i;
364
365     printf("MATRICULA   NOMBRE                               APPAT   APMAT   EDAD   SEXO   \n\n");
366     for (i = 0; i < n; i++)
367     {
368         if (vect[i].status != 0)
369         {
370             printf("%-9d   %-30s   %-30s   %-30s   %-4d   ", vect[i].matricula, vect[i].nombre, vect[i].apPat, vect[i].apMat, vect[i].edad);
371
372             if (vect[i].sexo == 1)
373             {
374                 printf("HOMBRE\n");
375             }
376             else
377             {
378                 printf("MUJER\n");
379             }
380         }
381     }
382 }
```

Salida:

MATRICULA	NOMBRE	APPAT	APMAT	EDAD	SEXO
309495	LOURDES	GONZALEZ	VEGA	20	MUJER
309636	LOURDES	SILVA	FUENTES	25	MUJER
312574	LUIS	ROMERO	FUENTES	27	HOMBRE
317545	LUISA	FLORES	MORALES	19	MUJER
320354	LUCIA	MORENO	REYES	27	MUJER
326348	SOFIA	HERNANDEZ	GUTIERREZ	21	MUJER
327430	MARIA	CRUZ	RAMIREZ	20	MUJER
328201	ANTONIO	SANCHEZ	ALVAREZ	21	HOMBRE
332376	CLARA	MORALES	GOMEZ	20	MUJER

Presione una tecla para continuar . . .