

---

Page 1 of 5

```

73     do
74     {
75         reg.ci=0;
76         //Inicializando Registro
77         reg.ci=0;
78         for(k=0;k<20;k++) reg.nombre[k]='\0';
79         for(k=0;k<20;k++) reg.appat[k]='\0';
80         for(k=0;k<35;k++) reg.apmat[k]='\0';
81         reg.borrado=0;
82         clrscr();
83         _txt(10,3,"  A D I C I O N A N D O      R E G I S T R O  ");
84         _txt(10,5,"Introduzca Nro. CI      : "); cin>>reg.ci;
85         _txt(10,7,"Introduzca Nombre      : "); gets(reg.nombre);
86         _txt(10,8,"Introduzca AP. Paterno : "); gets(reg.appat);
87         _txt(10,9,"Introduzca AP. Materno : "); gets(reg.apmat);
88         reg.borrado=1;
89         //Calculando la Direccion usando Funcion HASH
90         direccion=reg.ci % divisor;
91         if(direccion>124) direccion=0; //Hmmm. Si es > al rango, manda el puntero al inicio
92         rewind(fp);
93         fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
94         fread(&aux,sizeof(alumno),1,fp);
95         if(aux.ci==reg.ci&&reg.borrado==1)
96         {
97             _txt(22,22,"ERROR:: Registro Repetido"); getch();
98         }
99     else
100     {
101         if(aux.ci==0) //Hmmm, Direccion Libre
102         {
103             fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
104             fwrite(&reg,sizeof(alumno),1,fp);
105         }
106     else
107     {
108         if(aux.ci==reg.ci) //Hmmm, Registro ya Borrado, Sobrecribir
109         {
110             fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
111             fwrite(&reg,sizeof(alumno),1,fp);
112         }
113     else //Ooohhh, Colision, Resuelta por Sondeo Lineal
114     {
115         ban=0;
116         origen=direccion;
117         do{
118             if(direccion==124)
119             {
120                 rewind(fp); direccion=-1;
121             }
122             fread(&aux,sizeof(alumno),1,fp);
123             if(aux.ci==reg.ci)
124             {
125                 _txt(22,22,"ERROR:: Registro Repetido"); getch();
126             }
127             direccion++;
128             if(aux.ci==0) //Un Registro Libre
129             {
130                 fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
131                 fwrite(&reg,sizeof(alumno),1,fp);
132                 ban=1;
133             }
134             if(direccion==origen) ban=1; //Oooooohhhh, Archivo Lleno
135         }while(ban!=1);
136     } //Fin de Sondeo Lineal
137 }
138 }
139 _txt(22,22,"Desea Adicionar mas registro (S/N) : ");
140 opcion=getche();
141 while(toupper(opcion)=='S');
142 fclose(fp);
143 }
144 //Hacemos una Consulta a Pantalla

```

```

145 void consultas()
146 {
147     FILE *fp;
148     unsigned long buscar;
149     char opcion,opcion2;
150     alumno reg;
151     int direccion,mostrar,ban,origen;
152     // Abrir Archivo para Lectura
153     fp=fopen("alumno.dat","rb");
154     if(fp==NULL)
155     {
156         _txt(22,22,"ERROR:: Apertura de Archivo"); getch();
157         exit(0);
158     }
159     do{
160         clrscr();
161         _txt(10,5,"  C O N S U L T A  ");
162         _txt(10,7,"Introduzca Nro. ci: ");
163         cin>>buscar; direccion=buscar % divisor;
164         if(direccion>124) direccion=0;
165         rewind(fp);
166         fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
167         fread(&reg,sizeof(alumno),1,fp);
168         mostrar=0;
169         if(buscar==reg.ci&&reg.borrado==1)
170         {
171             mostrar=1;
172         }
173         else //Hmmm. Buscando ci por Sondeo Lineal
174         {
175             ban=0;
176             origen=direccion;
177             do{
178                 if(direccion==124)
179                 {
180                     rewind(fp); direccion=-1;
181                 }
182                 fread(&reg,sizeof(alumno),1,fp);
183                 if(buscar==reg.ci&&reg.borrado==1)
184                 {
185                     mostrar=1; //Hmmmm. Registro Encontrado
186                 }
187                 direccion++;
188                 if(reg.ci==0&&reg.borrado==0) //Hmmm, Registro Libre
189                 {
190                     ban=1; //Oooohhh. Registro No Encontrado
191                 }
192                 if(direccion==origen) ban=1; //Oooohhh, Archivo Lleno, No hay Registro
193             }while(ban!=1&&mostrar!=1);
194         }
195         if(mostrar==1)
196         {
197             clrscr();
198             _txt(35,5,"  C O N S U L T A S  ");
199             _txt(27,7,"(P) Consulta a Pantalla");
200             _txt(27,8,"(I) Consulta a Impresora");
201             _txt(32,21,"Registro encontrado, y tu opcion : ");
202             opcion2=getche();
203             opcion2=toupper(opcion2);
204             switch(opcion2)
205             {
206                 case 'P':
207                     clrscr();
208                     _txt(15,2,"  C O N S U L T A S  ");
209                     gotoxy(10,5); printf("Numero CI      : %ld", reg.ci);
210                     gotoxy(10,7); printf("Nombre          : %s",reg.nombre);
211                     gotoxy(10,8); printf("Apellido Paterno : %s",reg.appat);
212                     gotoxy(10,9); printf("Apellido Materno : %s",reg.apmat);
213                     break;
214                 case 'I':
215                     clrscr();
216                     _txt(10,2,"  C O N S U L T A S  ");

```

```

217         txt(10,15,"Mandando Registro a la cola de Impresion !!!");
218         fprintf(stdprn,"* Registro *\n\r");
219         fprintf(stdprn,"Numero CI          : %ld\n\r",reg.ci);
220         fprintf(stdprn,"Nombre           : %s\n\r",reg.nombre);
221         fprintf(stdprn,"Apellido Paterno : %s\n\r",reg.appat);
222         fprintf(stdprn,"Apellido Materno : %s\n\r",reg.apmat);
223         break;
224     }
225 }
226 else
227 {
228     _txt(10,14,"No se encuentra el registro");
229 }
230     _txt(10,22,"Deseas consultar otro registro (S/N) : ");
231     opcion=getche();
232     while(toupper(opcion)=='S');
233     fclose(fp);
234 }
235 //Baja logica de un registro
236 void bajas()
237 {
238     FILE *fp;
239     char resp,opcion;
240     unsigned long baja;
241     int direccion,encontrado,ban,origen;
242     alumno reg;
243     // Abrir Archivo para Lectura
244     fp=fopen("alumno.dat","rb+");
245     if(fp==NULL)
246     {
247         _txt(22,22,"ERROR:: Apertura de Archivo"); getch();
248         exit(0);
249     }
250     do{
251         clrscr();
252         _txt(5,2,"  B A J A S  ");
253         _txt(5,5,"Registro a dar de Baja, CI : ");
254         cin>>baja;
255         rewind(fp); //Hmmmm, colocando a Inicio
256         direccion=baja % divisor;
257         if(direccion>124) direccion=0;
258         rewind(fp);
259         fseek(fp,sizeof(alumno)*direccion,SEEK_SET);
260         fread(&reg,sizeof(alumno),1,fp);
261         encontrado=0;
262         if(baja==reg.ci&&reg.borrado==1)
263         {
264             encontrado=1;
265         }
266         else //Hmmmm, Buscando por Sondeo Lineal (ci)
267         {
268             ban=0;
269             origen=direccion;
270             do{
271                 if(direccion==124)
272                 {
273                     rewind(fp);
274                     direccion=-1;
275                 }
276                 fread(&reg,sizeof(alumno),1,fp);
277                 if(baja==reg.ci&&reg.borrado==1)
278                 {
279                     encontrado=1; //Hmmmm, Encontramos Registro
280                 }
281                 direccion++;
282                 if(reg.ci==0&&reg.borrado==0) //Registro Libre, para la busqueda
283                 {
284                     ban=1; //Ooohhh, No se Encontro
285                 }
286                 if(direccion==origen) ban=1; //Ooohhh, Archivo Lleno, No hay Registro
287             }while(ban!=1&&encontrado!=1);
288         }

```

```

289     if(encontrado==1)
290     {
291         fseek(fp, (direccion)*sizeof(alumno), SEEK_SET);
292         fread(&reg, sizeof(alumno), 1, fp);
293         gotoxy(10,7); printf("Numero CI      : %ld", reg.ci);
294         gotoxy(10,9); printf("Nombre       : %s", reg.nombre);
295         gotoxy(10,10); printf("Apellido Paterno : %s", reg.appat);
296         gotoxy(10,11); printf("Apellido Materno : %s", reg.apmat);
297         _txt(10,17,"Ok, Desea Eliminarlo ? (S/N) : ");
298         resp=getch();
299         if(toupper(resp)=='S')
300         {
301             reg.borrado=0;
302             fseek(fp, (direccion)*sizeof(alumno), SEEK_SET);
303             fwrite(&reg, sizeof(alumno), 1, fp);
304             _txt(10,20,"Registro Eliminado (Logico)");
305         }
306         else
307         {
308             _txt(10,13,"El Registro No Existe");
309         }
310         _txt(10,22,"Desea Eliminar Otro Registro (S/N)? :");
311         opcion=getch();
312         while(toupper(opcion)!='N');
313     }
314     fclose(fp);
315 }
316 //Acerca De
317 AcercaD()
318 {
319     int x=55,y=20;
320     clrscr();
321     textcolor(RED); _txt(x,y, "      exesoft@lycos.es  ");
322     textcolor(YELLOW); _txt(x,y+1, "      David Nina      ");
323     textcolor(GREEN); _txt(x,y+2, " EXEsoft  Bolivia 2003 ");
324     getch();textcolor(15);
325 }
326 //Programa principal
327 void main()
328 {
329     char opcion;
330     do{
331         clrscr();
332         _txt(20,3, "      M E N U      P R I N C I P A L      ");
333         _txt(27,7,"(A) Alta de registros");
334         _txt(27,8,"(B) Baja de registros");
335         _txt(27,9,"(C) Consulta individual");
336         _txt(27,10,"(D) Acerca De");
337         _txt(27,11,"(S) Salir");
338         _txt(27,21,"Elige una opcion : ");
339         opcion=getche();
340         opcion=toupper(opcion);
341         switch(opcion)
342         {
343             case 'A':altas();break;
344             case 'B':bajas();break;
345             case 'C':consultas();break;
346             case 'D':AcercaD();break;
347         }
348     }while(opcion!='S');
349     clrscr();
350 }

```