



Introducción

Zagnes es una compañía dedicada al desarrollo de software, con el objetivo de brindar soluciones prácticas, por medio de un sistema de información.

Miriam Stampanone

39.322.702

Micaela Zaghis

38.374.019

```

1. #include <stdlib.h>
2. #include <stdio.h>
3. #include <string.h>
4. #include <windows.h>
5. #include <locale.h>
6.
7. struct empleados{
8.     int cod;
9.     char DNI[9], cuil[15], nombres[40], apellidos[40];
10.
11. };
12. struct empleados_lista{
13.     int cod;
14.     char DNI[9], cuil[15], nombres[40], apellidos[40];
15.     struct empleados_lista *sig;
16. };
17.
18. struct empresas{
19.     int cod;
20.     char cuit[15], nombreE[40];
21. };
22. struct empresas_lista{
23.     int cod;
24.     char cuit[15], nombreE[40];
25.     struct empresas_lista *sig;
26. };
27.
28. struct datos{
29.     int dispositivo;
30.     int codigoEmpleado;
31.     char fecha[11];
32.     float costoH;
33.     float costoS;
34.     char nombreEmpresa[100];
35.     int codigoEmpresa;
36.     float precioFinal;
37. };
38. struct datos_lista{
39.     int dispositivo;
40.     int codigoEmpleado;
41.     char fecha[11];
42.     float costoH;
43.     float costoS;
44.     char nombreEmpresa[100];
45.     int codigoEmpresa;
46.     float precioFinal;
47.     struct datos_lista *sig;
48. }nodo;
49. //*****
50. struct empleados_lista *empleado_primer;
51. struct empleados_lista *empleado_ultimo;
52.
53. struct empresas_lista *emp_primer;
54. struct empresas_lista *emp_ultimo;
55.
56. struct datos_lista *primer;

```

```

57.     struct datos_lista *ultimo;
58.
59.     struct datos_lista *prim;
60.     struct datos_lista *ult;
61.     //*****
62.     /*PROTOTIPOS*/
63.     void cargarDatos();
64.     void cargarDatos2();
65.     void cargarDatos3();
66.     void menu();
67.
68.     void ingresarReparaciones();
69.     void mostrar_emp();
70.     void mostrar_empresa();
71.
72.     void buscarUNAreparacion();
73.     void buscarLAreparacion();
74.
75.     void empleadoespecifico();
76.     void reparacionesUNAempresa();
77.     void reparacionesUNAfecha();
78.
79.     void almacenarreparaciones();
80.     void almacenarreparaciones2();
81.
82.     void pregunta();
83.     void mensaje();
84.
85.     int contador=0;
86.     int cont2=0;
87.     //*****
88.     int main () {
89.         cargarDatos();
90.         cargarDatos2();
91.         cargarDatos3();
92.         menu();
93.     }
94.     //*****
95.     void menu() {
96.
97.         setlocale(LC_ALL, "");
98.         int opcion=1;
99.
100.        printf("\t\t**BIENVENIDO A ZAGNES**\n\n");
101.        printf("MENU\n");
102.        printf("1-INGRESAR REPARACIONES.\n");
103.        printf("2-BUSCAR UNA REPARACIÓN SEGUN TIPO DE
DISPOSITIVO.\n");
104.        printf("3-BUSCAR LA REPARACION CON EL MAYOR PRECIO
FINAL.\n");
105.        printf("4-MOSTRAR LAS REPARACIONES REALIZADAS POR UN
EMPLEADO ESPECIFICO.\n");
106.        printf("5-MOSTRAR LAS REPARACIONES CORRESPONDIENTES A UNA
SOLA EMPRESA.\n");
107.        printf("6-MOSTRAR LAS REPARACIONES REALIZADAS EN UNA
FECHA.\n");

```

```

108.     printf("7-ALMACENAR TODAS LAS REPARACIONES.\n");
109.     printf("8-SALIR DEL PROGRAMA.\n");
110.     printf("\nINGRESE UNA OPCION: ");
111.     scanf("%d",&opcion);
112.
113.     switch(opcion){
114.         case 1:{
115.             ingresarReparaciones();
116.             break;
117.         }
118.         case 2:{
119.             buscarUNAreparacion();
120.             break;
121.         }
122.         case 3:{
123.             buscarLAreparacion();
124.             break;
125.         }
126.         case 4:{
127.             empleadoespecifico();
128.             break;
129.         }
130.         case 5:{
131.             reparacionesUNAempresa();
132.             break;
133.         }
134.         case 6:{
135.             reparacionesUNAfecha();
136.             break;
137.         }
138.         case 7:{
139.             almacenarreparaciones();
140.             break;
141.         }
142.         case 8:{
143.             if(contador==1){
144.                 pregunta();
145.             }
146.             mensaje();
147.             break;
148.         }
149.     }
150. }
151. //*****
152. void cargarDatos(){
153.     FILE *ficherol=fopen("empleados.dat","rb");
154.     if(ficherol==NULL){
155.         printf("ERROR DE APERTURA...");
156.         exit(0);
157.     }
158.     else{
159.         empleado_primer=0;
160.         empleado_ultimo=0;
161.         struct empleados *empleado = malloc(11* sizeof(struct
162.             empleados));
163.         fread(empleado, sizeof(struct empleados), 11, fichero1);

```

```

163.         system("cls");
164.
165.         for(int i=0;i< 11;i++){
166.
167.             struct empleados_lista *nuevo;
168.             nuevo=malloc(sizeof(struct empleados_lista));
169.             nuevo->cod=empleado[i].cod;
170.             strcpy(nuevo->DNI,empleado[i].DNI);
171.             strcpy(nuevo->cuil,empleado[i].cuil);
172.             strcpy(nuevo->nombres,empleado[i].nombres);
173.             strcpy(nuevo->apellidos,empleado[i].apellidos);
174.
175.             if(empleado_primeros==NULL){
176.                 empleado_primeros=nuevo;
177.                 empleado_ultimo=nuevo;
178.                 empleado_ultimo->sig=NULL;
179.             }
180.             else{
181.                 empleado_ultimo->sig=nuevo;
182.                 empleado_ultimo=nuevo;
183.                 empleado_ultimo->sig=NULL;
184.             }
185.         }
186.         free(empleado);
187.     }
188.     fclose(fichero1);
189. }
190.
191. void cargarDatos2(){
192.     FILE *fichero2=fopen("empresas.dat","rb");
193.     if(fichero2==NULL){
194.         printf("ERROR DE APERTURA...");
195.         exit(0);
196.     }
197.     else{
198.         emp_primeros=NULL;
199.         emp_ultimo=NULL;
200.         struct empresas *empresa=malloc(10* sizeof(struct empresas));
201.         fread(empresa,sizeof(struct empresas),10,fichero2);
202.         system("cls");
203.         for(int i=0;i<10;i++){
204.
205.             struct empresas_lista *nuevo;
206.             nuevo=malloc(sizeof(struct empresas_lista));
207.             nuevo->cod=empresa[i].cod;
208.             strcpy(nuevo->cuit,empresa[i].cuit);
209.             strcpy(nuevo->nombreE,empresa[i].nombreE);
210.
211.             if(emp_primeros==NULL){
212.                 emp_primeros=nuevo;
213.                 emp_ultimo=nuevo;
214.                 emp_ultimo->sig=NULL;}
215.             else{
216.                 emp_ultimo->sig=nuevo;
217.                 emp_ultimo=nuevo;
218.                 emp_ultimo->sig=NULL;

```

```

219.         }
220.         }
221.         free(empresa);
222.     }
223.         fclose(fichero2);
224.     }
225. //*****
226. void ingresarReparaciones(){
227.     system("cls");
228.     struct datos_lista *nuevo;
229.     nuevo=malloc(sizeof(struct datos_lista));
230.     printf("TIPO DE DISPOSITIVO:\n1-ESCRITORIO\n2-NOTEBOOK\n3-
CELULAR\n");
231.     scanf("%d",&nuevo->dispositivo);
232.     mostrar_emp();
233.     printf("\\nINGRESE CODIGO DE EMPLEADO: \\n");
234.     scanf("%d",&nuevo->codigoEmpleado);
235.     printf("INGRESE FECHA: (DD/MM/AAAA) \\n");
236.     scanf("%s",nuevo->fecha);
237.     printf("INGRESE COSTO DE HARDWARE: \\n");
238.     scanf("%f",&nuevo->costoH);
239.     printf("INGRESE COSTO DE SOFTWARE: \\n");
240.     scanf("%f",&nuevo->costoS);
241.     mostrar_empresa();
242.     printf("\\n");
243.     printf("INGRESE NOMBRE DE EMPRESA: \\n");
244.     fflush(stdin);
245.     gets(nuevo->nombreEmpresa);
246.     printf("INGRESE CODIGO DE EMPRESA: \\n");
247.     scanf("%d",&nuevo->codigoEmpresa);
248.
249.     nuevo->precioFinal = nuevo->costoH + nuevo->costoS;
250.
251.     if(primerero==NULL){
252.         primero=nuevo;
253.         ultimo=nuevo;
254.         ultimo->sig=NULL;
255.     }
256.     else {
257.         ultimo->sig=nuevo;
258.         ultimo=nuevo;
259.         ultimo->sig=NULL;
260.     }
261.     contador=1;
262.     sleep(2);
263.     system("cls");
264.     menu();
265.
266. }
267. //*****
268. void mostrar_emp(){
269.     struct empleados_lista *puntero;
270.     puntero=empleado_primerero;
271.     while(empleado_primerero!=NULL){
272.         printf("COD: %d  -",empleado_primerero->cod);
273.         printf("DNI: %s  -",empleado_primerero->DNI);

```

```

274.         printf("CUIL: %s  -", empleado_primer->cuil);
275.         printf("NOMBRES: %s  -", empleado_primer->nombres);
276.         printf("APELLIDOS: %s\n", empleado_primer->apellidos);
277.         empleado_primer=empleado_primer->sig;
278.     }
279.     empleado_primer=puntero;
280. }
281.
282. void mostrar_empresa(){
283.     struct empresas_lista *puntero;
284.     puntero=emp_primer;
285.     while(emp_primer!=NULL){
286.         printf("COD: %d  -", emp_primer->cod);
287.         printf("CUIT: %s  -", emp_primer->cuit);
288.         printf("NOMBRE: %s\n", emp_primer->nombreE);
289.         emp_primer=emp_primer->sig;
290.     }
291.     emp_primer=puntero;
292. }
293. //*****
294. void buscarUNAreparacion(){
295.     system("cls");
296.     int opc;
297.     int c=0;
298.     printf("¿QUE TIPO DE DISPOSITIVO DESEA BUSCAR?\n");
299.     printf("1-ESCRITORIO.\n2-NOTEBOOK.\n3-CELULAR.\n");
300.     scanf("%d",&opc);
301.     switch(opc){
302.         case 1:
303.         case 2:
304.         case 3:
305.             {
306.                 struct datos_lista *p=primer;
307.                 while(primer!=NULL){
308.                     if(opc==primer->dispositivo){
309.                         printf("REPARACION/ES ENCONTRADA/S EN LA RAM:\n\n");
310.                         printf("DISPOSITIVO: %d\n", primer->dispositivo);
311.                         printf("CODIGO EMPLEADO: %d\n", primer->codigoEmpleado);
312.                         printf("FECHA: %s\n", primer->fecha);
313.                         printf("COSTO HARDWARE: %.1f\n", primer->costoH);
314.                         printf("COSTO SOFTWARE: %.1f\n", primer->costoS);
315.                         printf("NOMBRE EMPRESA: %s\n", primer->nombreEmpresa);
316.                         printf("CODIGO EMPRESA: %d\n", primer->codigoEmpresa);
317.                         printf("PRECIO FINAL: %.1f\n\n", primer->precioFinal);
318.
319.                         c=1;
320.                     }
321.                     primer=primer->sig;
322.                 }
323.                 primer=p;
324.                 struct datos_lista *pp=prim;
325.                 while(prim!=NULL){
326.                     if(opc==prim->dispositivo){
327.                         printf("REPARACION/ES ENCONTRADA/S EN EL FICHERO:\n\n");
328.                         printf("DISPOSITIVO: %d\n", prim->dispositivo);
329.                         printf("CODIGO EMPLEADO: %d\n", prim->codigoEmpleado);

```



```

330.         printf("FECHA: %s\n",prim->fecha);
331.         printf("COSTO HARDWARE: %.1f\n",prim->costoH);
332.         printf("COSTO SOFTWARE: %.1f\n",prim->costoS);
333.         printf("NOMBRE EMPRESA: %s\n",prim->nombreEmpresa);
334.         printf("CODIGO EMPRESA: %d\n",prim->codigoEmpresa);
335.         printf("PRECIO FINAL: %.1f\n\n",prim->precioFinal);
336.
337.         c=1;
338.     }
339.     prim=prim->sig;
340. }
341.     prim=pp;
342. }
343.     break;
344. }
345.     if(c!=1){
346.         printf("DISPOSITIVO NO ENCONTRADO.\n");
347.     }
348.     system("PAUSE");
349.     system("cls");
350.     menu();
351. }
352. //*****
353. void cargarDatos3() {
354.     FILE *fichero3=fopen("reparaciones.txt","r");
355.     if(fichero3==NULL) {
356.     }
357.     else{
358.         prim=NULL;
359.         ult=NULL;
360.         system("cls");
361.
362.         for(int i=0;i<50;i++){
363.
364.             struct datos_lista *nuevo;
365.             nuevo=malloc(sizeof(struct datos_lista));
366.             fscanf(fichero3,"%d",&nuevo->dispositivo);
367.             fscanf(fichero3,"%d",&nuevo->codigoEmpleado);
368.             fscanf(fichero3,"%s",&nuevo->fecha);
369.             fscanf(fichero3,"%f",&nuevo->costoH);
370.             fscanf(fichero3,"%f",&nuevo->costoS);
371.             fscanf(fichero3,"%s",&nuevo->nombreEmpresa);
372.             fscanf(fichero3,"%d",&nuevo->codigoEmpresa);
373.             fscanf(fichero3,"%f",&nuevo->precioFinal);
374.
375.             if(prim==NULL) {
376.                 prim=nuevo;
377.                 ult=nuevo;
378.                 ult->sig=NULL;}
379.             else{
380.                 ult->sig=nuevo;
381.                 ult=nuevo;
382.                 ult->sig=NULL;
383.             }
384.         }
385.     }

```

```

386.         fclose(fichero3);
387.     }
388.     //*****
389.     void buscarLAreparacion() {
390.         int c=0;
391.         float mayor=0;
392.         struct datos_lista *m;
393.         struct datos_lista *x=primero;
394.         while(x!=NULL) {
395.             if(x->precioFinal>mayor) {
396.                 mayor=x->precioFinal;
397.                 m=x;
398.                 c=1;
399.             }
400.             x=x->sig;
401.         }
402.         struct datos_lista *y=prim;
403.         while(y!=NULL) {
404.             if(y->precioFinal>mayor) {
405.                 mayor=y->precioFinal;
406.                 m=y;
407.
408.                 c=1;
409.             }
410.             y=y->sig;
411.         }
412.         printf("DISPOSITIVO ENCONTRADO\n");
413.         printf("ESPECIFICACIONES:\n");
414.         printf("DISPOSITIVO: %d\n",m->dispositivo);
415.         printf("EMPLEADO: %d\n",m->codigoEmpleado);
416.         printf("EMPRESA: %s\n",m->nombreEmpresa);
417.         printf("FECHA: %s\n",m->fecha);
418.         printf("COSTO HARDWARE: %f\n",m->costoH);
419.         printf("COSTO SOFTWARE: %f\n",m->costoS);
420.         printf("COSTO TOTAL: %f\n",m->precioFinal);
421.
422.         if(c!=1) {
423.             printf("DISPOSITIVO NO ENCONTRADO.\n");
424.         }
425.         system("PAUSE");
426.         system("cls");
427.         menu();
428.     }
429.     //*****
430.     void empleadoespecifico() {
431.         mostrar_emp();
432.         printf("\n");
433.         int z;
434.         int c=0;
435.         printf("INGRESE CODIGO DE EMPLEADO A BUSCAR: ");
436.         scanf("%d",&z);
437.
438.         struct datos_lista *cod_emp=primero;
439.         while(primero!=NULL) {
440.             if(z==primero->codigoEmpleado) {
441.                 printf("REPARACION ENCONTRADA EN LA RAM\n");

```

```

442.         printf("DISPOSITIVO: %d\n", primero->dispositivo);
443.         printf("FECHA: %s\n", primero->fecha);
444.         printf("COSTO HARDWARE: %.1f\n", primero->costoH);
445.         printf("COSTO SOFTWARE: %.1f\n", primero->costoS);
446.         printf("NOMBRE EMPRESA: %s\n", primero->nombreEmpresa);
447.         printf("PRECIO FINAL: %.1f\n\n", primero->precioFinal);
448.
449.         c=1;
450.     }
451.     primero=primero->sig;
452. }
453.     primero=cod_emp;
454.
455.     struct datos_lista *cod_emp2=prim;
456.     while(prim!=NULL) {
457.         if(z==prim->codigoEmpleado) {
458.             printf("REPARACION ENCONTRADA EN EL FICHERO:\n\n");
459.             printf("REPARACION/ES HECHA/S POR EL EMPLEADO ELEGIDO:\n");
460.             printf("DISPOSITIVO: %d\n", prim->dispositivo);
461.             printf("FECHA: %s\n", prim->fecha);
462.             printf("COSTO HARDWARE: %.1f\n", prim->costoH);
463.             printf("COSTO SOFTWARE: %.1f\n", prim->costoS);
464.             printf("NOMBRE EMPRESA: %s\n", prim->nombreEmpresa);
465.             printf("PRECIO FINAL: %.1f\n\n", prim->precioFinal);
466.
467.             c=1;
468.         }
469.         prim=prim->sig;
470.     }
471.     if(c!=1) {
472.         printf("EMPLEADO NO ENCONTRADO.\n");
473.     }
474.     prim=cod_emp2;
475.     system("PAUSE");
476.     system("cls");
477.     menu();
478. }
479. //*****
480. void reparacionesUNAempresa() {
481.     mostrar_empresa();
482.     printf("\n");
483.     int e;
484.     int c=0;
485.     printf("INGRESE CODIGO DE LA EMPRESA A BUSCAR: ");
486.     scanf("%d", &e);
487.
488.     struct datos_lista *cod_empresa=primero;
489.     while(primero!=NULL) {
490.         if(e==primero->codigoEmpresa) {
491.             printf("REPARACION ENCONTRADA EN LA RAM: \n");
492.             printf("EMPRESA: %s\n", primero->nombreEmpresa);
493.             printf("DISPOSITIVO: %d\n", primero->dispositivo);
494.             printf("FECHA: %s\n", primero->fecha);
495.             printf("PRECIO FINAL: %f\n", primero->precioFinal);
496.
497.             c=1;

```

```

498.         }
499.         primero=primero->sig;
500.     }
501.     primero=cod_empresa;
502.
503.     struct datos_lista *cod_empresa2=prim;
504.     while(prim!=NULL){
505.         if(e==prim->codigoEmpresa){
506.             printf("REPARACION ENCONTRADA EN EL FICHERO: \n");
507.             printf("EMPRESA: %s\n",prim->nombreEmpresa);
508.             printf("DISPOSITIVO: %d\n",prim->dispositivo);
509.             printf("FECHA: %s\n",prim->fecha);
510.             printf("PRECIO FINAL: %f\n",prim->precioFinal);
511.
512.             c=1;
513.         }
514.         prim=prim->sig;
515.     }
516.     if(c!=1){
517.         printf("EMPRESA NO ENCONTRADA.\n");
518.     }
519.     prim=cod_empresa2;
520.     system("PAUSE");
521.     system("cls");
522.     menu();
523. }
524. //*****
525. void reparacionesUNAfecha(){
526.     char x[11];
527.     int c=0;
528.     printf("INGRESE FECHA DE REPARACION A BUSCAR (DD/MM/AAAA): ");
529.     scanf("%s",x);
530.
531.     struct datos_lista *fechal=primero;
532.     while(primero!=NULL){
533.         if(strcmp(x,primero->fecha)==0){
534.             printf("REPARACION/ES ENCONTRADA/S EN LA RAM:\n ");
535.             printf("DISPOSITIVO: %d\n",primero->dispositivo);
536.             printf("COSTO HARDWARE: %.1f\n",primero->costoH);
537.             printf("COSTO SOFTWARE: %.1f\n",primero->costoS);
538.             printf("NOMBRE EMPRESA: %s\n",primero->nombreEmpresa);
539.             printf("CODIGO DE EMPLEADO: %d\n",primero->codigoEmpleado);
540.             printf("PRECIO FINAL: %.1f\n\n",primero->precioFinal);
541.             c=1;
542.         }
543.         primero=primero->sig;
544.     }
545.     primero=fechal;
546.
547.     struct datos_lista *fecha2=prim;
548.     while(prim!=NULL){
549.         if(strcmp(x,prim->fecha)==0){
550.
551.             printf("REPARACION/ES ENCONTRADA/S EN EL FICHERO:\n ");
552.             printf("DISPOSITIVO: %d\n",prim->dispositivo);
553.             printf("COSTO HARDWARE: %.1f\n",prim->costoH);

```

```

554.     printf("COSTO SOFTWARE: %.1f\n", prim->costoS);
555.     printf("NOMBRE EMPRESA: %s\n", prim->nombreEmpresa);
556.     printf("CODIGO DE EMPLEADO: %d\n", prim->codigoEmpleado);
557.     printf("PRECIO FINAL: %.1f\n\n", prim->precioFinal);
558.
559.         c=1;
560.         }
561.         prim=prim->sig;
562.     }
563.     if(c!=1){
564.         printf("REPARACION NO ENCONTRADA.\n");
565.     }
566.     prim=fecha2;
567.     system("PAUSE");
568.     system("cls");
569.     menu();
570. }
571. //*****
572. void almacenarreparaciones(){
573.     FILE *archivo;
574.     archivo=fopen("reparaciones.txt", "a");
575.     if(archivo==NULL){
576.         printf("ERROR EN EL ARCHIVO...");
577.         exit(1);
578.     }
579.     else{
580.         if(ftell(archivo)==0){
581.             printf("SE HA CREADO EL ARCHIVO CORRECTAMENTE...\n");
582.             getchar();
583.         }
584.         struct datos_lista *rep;
585.         rep=primero;
586.
587.         while(rep!=NULL){
588.
589.             fprintf(archivo, "%d ", rep->dispositivo);
590.             fprintf(archivo, "%d ", rep->codigoEmpleado);
591.             fprintf(archivo, "%s ", rep->fecha);
592.             fprintf(archivo, "%.0f ", rep->costoH);
593.             fprintf(archivo, "%.0f ", rep->costoS);
594.             fprintf(archivo, "%s ", rep->nombreEmpresa);
595.             fprintf(archivo, "%d ", rep->codigoEmpresa);
596.             fprintf(archivo, "%.0f", rep->precioFinal);
597.
598.             fprintf(archivo, "%s", "\n");
599.             rep=rep->sig;
600.         }
601.     }
602.     cont2=1;
603.     fclose(archivo);
604.     sleep(3);
605.     system("cls");
606.     menu();
607. }
608. //*****
609.

```

```

610.     void almacenarreparaciones2() {
611.         FILE *archivo;
612.         archivo=fopen("reparaciones.txt","a");
613.         if(archivo==NULL) {
614.             printf("ERROR EN EL ARCHIVO...");
615.             exit(1);
616.         }
617.         else{
618.             if(ftell(archivo)==0){
619.                 printf("CREACION DE ARCHIVO..");
620.                 getchar();
621.             }
622.             struct datos_lista *rep;
623.             rep=primero;
624.
625.             while(rep!=NULL) {
626.
627.                 fprintf(archivo,"%d ",rep->dispositivo);
628.                 fprintf(archivo,"%d ",rep->codigoEmpleado);
629.                 fprintf(archivo,"%s ",rep->fecha);
630.                 fprintf(archivo,"%0f ",rep->costoH);
631.                 fprintf(archivo,"%0f ",rep->costoS);
632.                 fprintf(archivo,"%s ",rep->nombreEmpresa);
633.                 fprintf(archivo,"%d ",rep->codigoEmpresa);
634.                 fprintf(archivo,"%0f",rep->precioFinal);
635.
636.                 fprintf(archivo,"%s","\n");
637.                 rep=rep->sig;
638.             }
639.         }
640.         fclose(archivo);
641.         system("cls");
642.     }
643.     //*****
644.     void mensaje() {
645.         printf("GRACIAS POR ELEGIR ZAGNES.\n");
646.         printf("HASTA LUEGO.");
647.     }
648.     //*****
649.     void pregunta() {
650.         if(cont2!=1) {
651.             int a;
652.             printf("¿DESEA ALMACENAR LAS REPARACIONES ANTES DE SALIR?\t1-
SI \t2-NO\n");
653.             scanf("%d",&a);
654.             if(a==1) {
655.                 almacenarreparaciones2();
656.             }
657.         }
658.     }

```