Proyecto final

Fundamentos de informática

1. Ámbito e investigación documental:

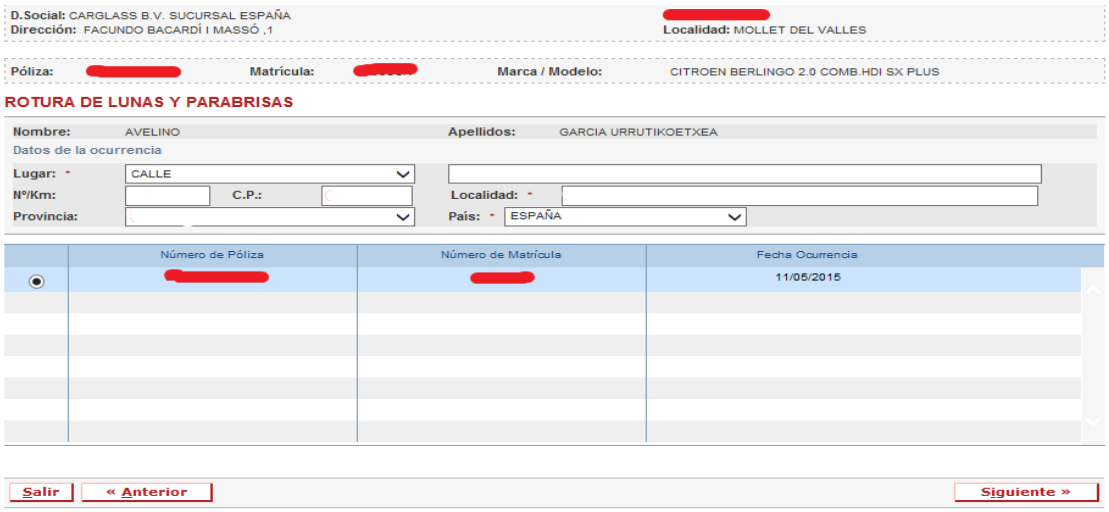
El ámbito que he escogido para el desarrollo de mi proyecto es la confección y el diseño de un programa de gestión y administración de siniestros o partes, basándome en las funciones de la empresa líder en talleres acristaladores de vehículos, Carglass, empresa en la que actualmente trabajo y, con lo cual, he tenido la posibilidad de obtener conocimientos del sector y de la gestión interna de toda la cadena de talleres.

1) ¿Cuál es la naturaleza de la información relevante del ámbito?

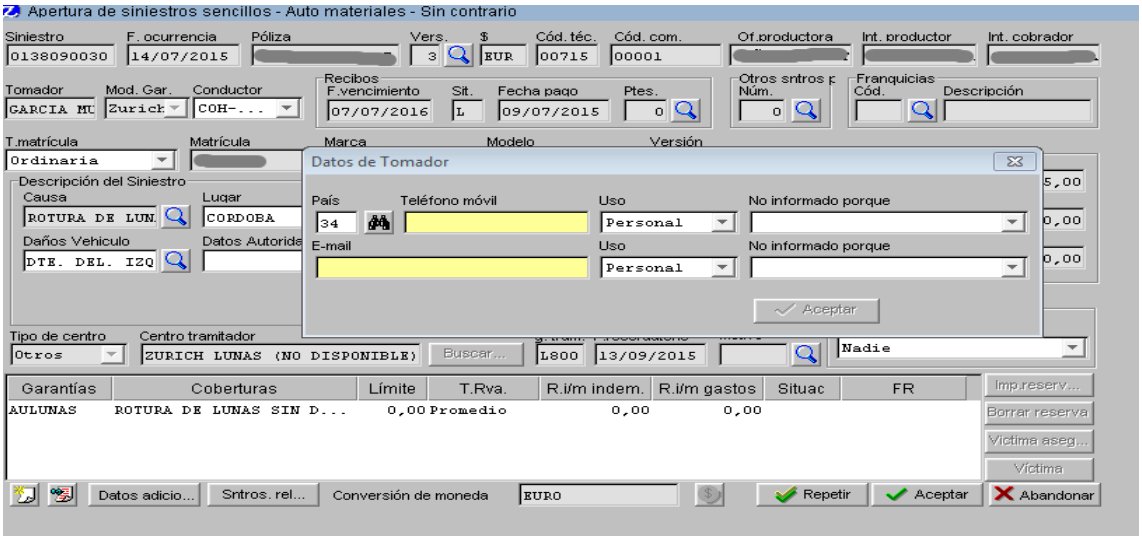
- La información obtenida proviene directamente de las formaciones impartidas en el ámbito laboral. Dicha información consta de documentos, imágenes y aplicativo que me permitirá asentar las bases de la idea principal de mi proyecto de manera muy específica y detallada.

2) ¿Cómo está la informática actualmente relacionada con el ámbito?

- Basándome en el aplicativo utilizado en la empresa, el diseño del programa principal de mi proyecto pretende sintetizar y simplificar la apertura de una serie de "fichas" de datos para poder registrar tanto vehículos, como conductores, y poder aperturar partes o siniestros con el objetivo de poder administrar esta información para su posterior facturación o gestión por parte del taller.



En la empresa, utilizamos diferentes aplicativos ya sea para aperturar un parte o para registrar una cita con nuestros talleres, y dependiendo de la compañía aseguradora, se utiliza un programa u otro.



Se pretende plasmar en lenguaje C++, un programa , mucho más básico (a su vez mas visual y rápido), con un *input* de datos mínimos para que la gestión sea mucho más ágil y fácil de modificar.

3) ¿Cuáles son los retos futuros del desarrollo de ese ámbito y cómo la informática pu ede ayudar a abordarlos?

Cada vez mas, los negocios están siendo informatizados en su totalidad, y un programa de gestión que sea manejable y efectivo, a la par , puede resultar de gran utilidad para poder agilizar el proceso de gestión, con el objetivo de que cualquier trabajador de una empresa sea capaz de gestionar cualquier conjunto grande de datos, minimizando, con la ayuda del programa los errores que se puedan cometer en dicho proceso.

Diseño de código del programa: Primeramente, el código del programa constará de una tabla principal, y de dos "subtablas", que podrán interconectarse mediante una variable común, como se aprecia en el siguiente esquema:

Tabla 1 – Partes:

Nº de siniestro	Matricula (Dato eliminado por corrección de la primera fase)	Daños	DNI Conductor (Dato eliminado por corrección de la primera fase)	Fecha

Tabla 2 – Vehículos:

Marca	Modelo	Matricula	Año del vehículo	Color	DNI Conductor	Nº de Siniestro

Tabla 3 – Conductores:

DNI Conductor	Nombre	Apellidos	Matricula	Nº de siniestro	Dirección	e-mail

***Las tablas y tuplas han sido ligeramente reestructuradas en función de la primera corrección. Las funcionalidades se detallan posteriormente.**

ESTRUCTURAS UTILIZADAS (CÓDIGO):

Estructuras principales:

- **Conductor**
- **Vehículo**
- **Parte**

```
struct Nombre{
    string Nombre,ape1,ape2;};
struct Fecha{
    int dia, mes, anyo;};
struct Direccion{
    int CP;
    string Poblacion, calle;};
struct Coche{
    string marca,modelo,color;
    string matcoche;
    string dnicon;
    int anyo,siniestro;
    Fecha fechasin;
    char danyo;
};
struct Conductor{
    string email;
    Direccion dicon;
    string matcoche;
    Nombre nomcon;
    string dnicon;
    int siniestro;};
```

Estructuras principales

```
typedef Conductor tCon[pmax];
typedef Coche tCoche[pmax];
```

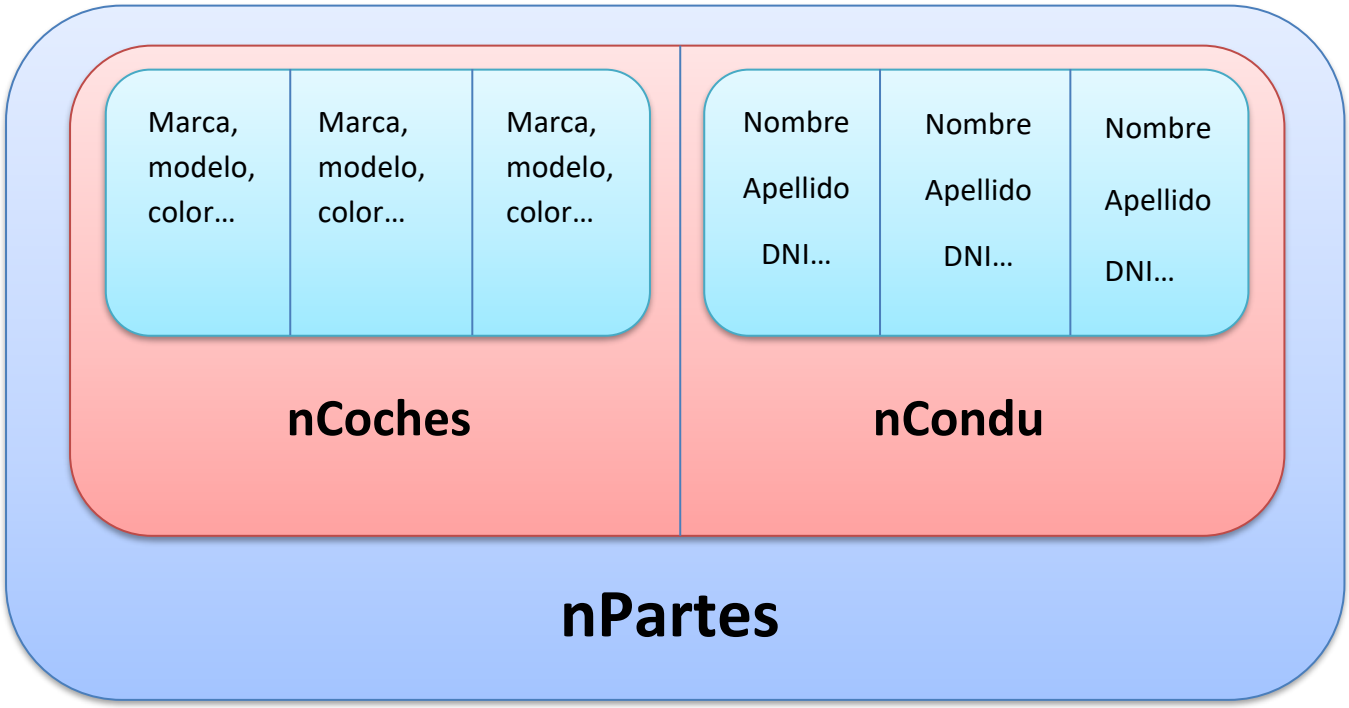
Tablas

```
struct tCondu{
  tCon coches;
  int nCondu;};
struct tCoch{
  tCoche coches;
  int nCoches;};
struct Partes{
    tCon Conductores;
    tCoch Coches;
    int nPartes;};
```

Tuplas que alojan tablas

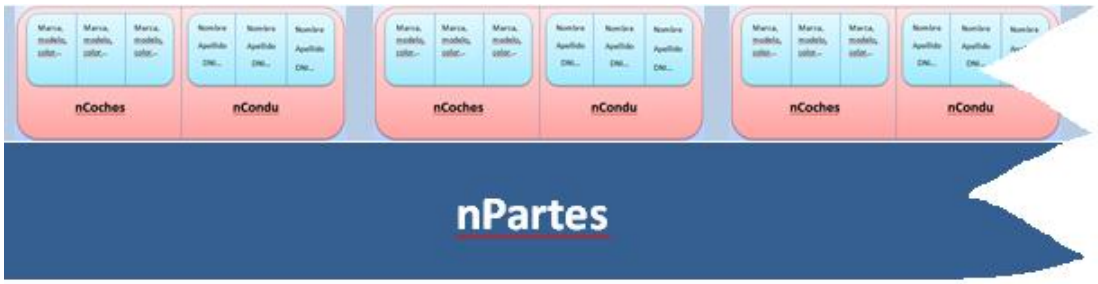
Se han utilizado estructuras complementarias para poder completar la información correctamente en las estructuras principales.

2. Diseño definitivo de estructuras (Representación grafica):



La estructura principal esta definida por la tupla "*Partes*" la cual a su vez, consta de una variable "*Conductores*" , una variable "*Coches*" que son tablas del tipo "*Conductor y coche*", y también tiene la variable nPartes que es el índice que controla las posiciones ocupadas por estas tablas.

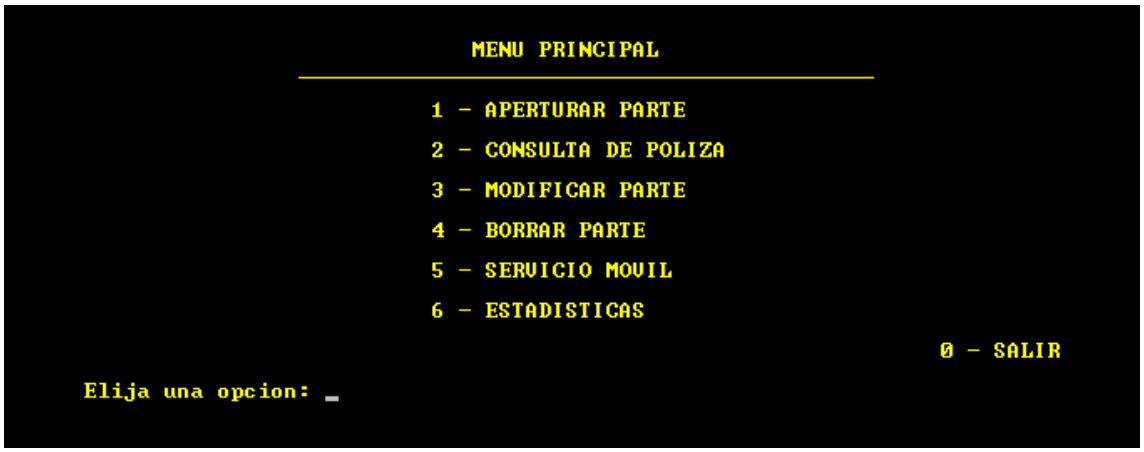
En la imagen de arriba se puede apreciar en detalle como se estructura esta tupla, y en la de abajo, podemos ver como nPartes actúa de índice de posiciones (cada vez que añado un conjunto de datos para rellenar las tablas incremento en 1 el valor del índice, de esta manera , tengo controlado cuantos valores hay en cada momento.



3. Funcionalidades básicas y específicas:

· Funcionalidades básicas

El programa pretende gestionar las fichas de cada persona que solicita un servicio en nuestros talleres, para ello, un nuevo cliente tiene que quedar registrado en el sistema, tanto los datos personales, como los datos del vehículo y de los daños que tiene. Por ello, el menú principal consta de:



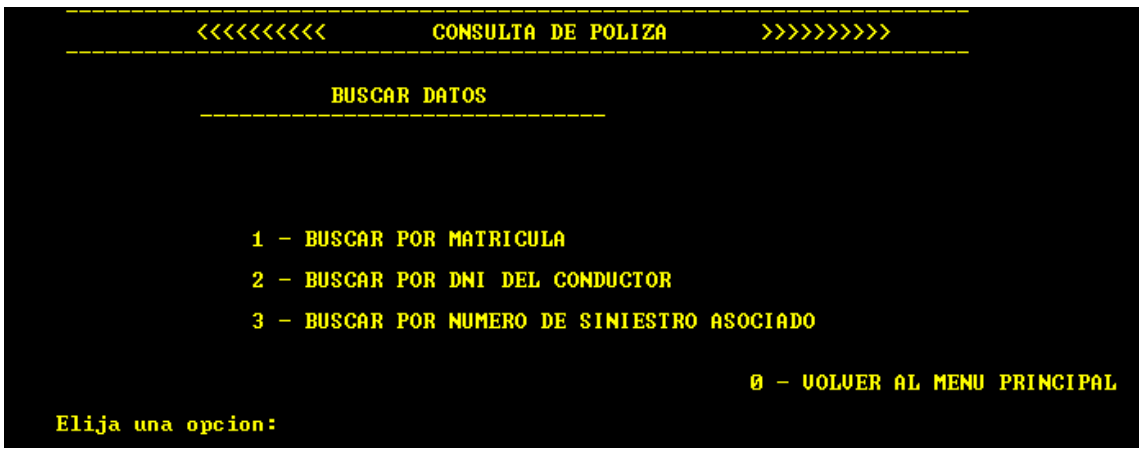
(Imagen 1)

- Se utiliza un switch con su respectivo menú de opciones para ello

En la primera opción (Imagen 1), abrimos un nuevo cliente (con sus datos personales, los datos de su vehículo y los datos del siniestro que se ha producido), de esta manera el cliente y sus datos queda incorporado en el sistema para una posterior gestión.

Se utilizan las acciones:

- leerinfo(part); → Lee la información que hay en el fichero
- introducir(part); → Pide los datos a introducir, y a su vez escribe los datos en el fichero



(Imagen 2)

En la opción 2 (Imagen 1), podremos consultar la póliza o datos del cliente que anteriormente hemos introducido, el submenú de esta opción (Imagen 2) , el sistema nos pregunta cómo queremos localizar al cliente que deseamos consultar, escogeremos la opción que más nos convenga.

Se utilizan las acciones:

- buscarfichaMAT(part,mat,trobat,i);→ Localiza la ficha que buscamos.
- mostrardatos(part,trobat,i);→ Muestra los datos por pantalla.

En la tercera opción, encontraremos un menú igual al submenú de la opción 2 (Imagen 2) , para poder localizar el cliente que deseamos modificar, y posteriormente el sistema nos pregunta si queremos modificar un dato en concreto o todos los datos.

Se utilizan las acciones:

- buscarfichaMAT(part,mat,trobat,i);→ Localiza la ficha que buscamos.
- mostrardatos(part,trobat,i);→ Muestra los datos por pantalla.
- modificardatos(part,trobat,i);→ Modifica los datos.

En la opción 4, podremos eliminar un cliente, mediante las acciones:

- buscarfichaMAT(part,mat,trobat,i);→ Localiza la ficha que buscamos.
- mostrardatos(part,trobat,i);→ Muestra los datos por pantalla.
- borrardatos(part,i);→ Elimina los datos.

· Funcionalidades específicas:

Constan en primer lugar de las acciones del submenú de la opción 2,3 y 4. Estas acciones permiten filtrar la búsqueda enlazando 2 tablas que son “Partes y Coches”, que a su vez enlazan con la tabla “Conductores”.

En la opción 5 tenemos la opción de gestionar las unidades móviles, que es el servicio que se da cuando el cliente desea que acudan a su domicilio a realizarle la reparación, para ello se utiliza un booleano en el momento de introducir los datos del siniestro si el cliente solicita este servicio.

En este menú, se nos mostrara una lista de matriculas de clientes que han solicitado este servicio, si el cliente ha solicitado el servicio y ya se ha quedado con el en un dia de pasada, nos aparecerá los siguiente:

```
0831CGG--> ASIGNADO !! Fecha: 5/6/2016
```

En cambio, si el cliente ha solicitado el servicio, pero todavía esta pendiente de que le indiquemos la fecha de pasada, nos aparecerá:

```
5060TTF--> NO ASIGNADO!
```

Este menú, esta diseñado para poder llevar al dia los servicios móviles que se tienen que desplazar y cuando, tenemos las siguientes opciones:

```
1 - ASIGNAR SERVICIO MOVIL
2 - MODIFICAR FECHA DE SERVICIO MOVIL
3 - BORRAR SERVICIO MOVIL
```

Y se utilizan las siguientes acciones para ello:

- leerinfo(part); → Lee la información que hay en el fichero.
- leerinfoSM(sermovil); → Lee la información que hay en el fichero (de la unidad móvil).
- mostrarSM(part,sermovil); → Muestra la lista de matriculas que solicitan el servicio y si se les ha asignado fecha o no
- SMasignado(part,sermovil); → Asigna una fecha a una matricula no asignada.
- modificarSM(sermovil); → Modifica una fecha asignada.
- borrarSM(part,sermovil); → Elimina una fecha asignada, y, da la opción de eliminar la solicitud del servicio o de quedar a la espera de una nueva fecha de pasada.

Finalmente, tenemos la opción de Estadisticas, que es una opción cuya utilidad es conocer de manera desglosada los tipos de daño que se han registrado y facturado en los talleres:

```
El total de siniestros registrados es : 10
Total de Parabrisas: 5
Total de Lunetas: 2
Total de Cristales laterales: 2
Total de Reparaciones de impacto: 1
Total de Techos solares: 0


       VALORES REPRESENTATIVOS

               PB:  #####
               Rep: ##
               Lun: ##
               Lat: #
               Tec:

Desea filtrar los datos por anyo? (S: Si, N: Regresar)
```

En este menú , el sistema nos da la opción de filtrar los daños por año de registro, de esta manera podemos controlar las reparaciones facturadas durante el periodo anual.

Este menú utiliza las siguientes acciones:

- leerinfo(part); → Lee la información que hay en el fichero.
- estadistica(part); → Muestra las estadísticas y pregunta si el usuario desea filtrar los datos mostrados por año.

· Funcionalidades no impartidas en clase:

- **void camcolor(int x)** → Funcion que cambia el color de la linea tanto de la letra como del fondo, se le introduce una variable *int* que utiliza para seleccionar el color, dentro de esta acción, llama a otra acción de la librería windows.h , "SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), x);" que realiza lo especificado anteriormente.

- **void logo()** → Es simplemente un conjunto de *cout* que, junto con la acción camcolor(); , muestra por pantalla un logo. Para los logos he utilizado un conversor de imagen a ASCII, que he tomado como base para dibujarlo.

- **void gotoxy(int x,int y)** → Lleva el punter a las coordenadas especificadas.
- **bool AjustarVentana(int Ancho, int Alto) {**

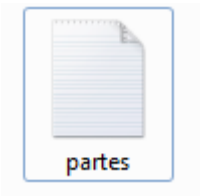
```
_COORD Coordenada;
Coordenada.X = Ancho;
Coordenada.Y = Alto;
_SMALL_RECT Rect;
Rect.Top = 0;
Rect.Left = 0;
Rect.Right = Ancho - 1;
Rect.Bottom = Alto - 1;
HANDLE hConsola = GetStdHandle(STD_OUTPUT_HANDLE);
SetConsoleScreenBufferSize(hConsola, Coordenada);
SetConsoleWindowInfo(hConsola, TRUE, &Rect);
return TRUE;
```

Este conjunto de funciones me permite cambiar el tamaño de la consola, pertenecen a la librería Windows.h , no se exactamente como funcionan , simplemente le paso la resolución que quiero (alto, ancho) y redimensiona la ventana.

- **SetConsoleTitle();** → Accion de la librería Windows.h que cambia el titulo de la consola.

4. Ejemplos de datos para utilizar el programa:

Desde el fichero partes.txt, se leen y escriben los datos, a continuación, se detallan algunos ejemplos de clientes (Matricula, siniestro o dni, para poder consultar,modificar o borrar):



- Matriculas: 0831CGG , 2030GPS , 6562TYJ.
- Siniestros: 1000003 , 1000010 , 1000003.
- DNI: 45489766T, 47596067M , 84561546T.

5. Errores y dificultades en el codigo:

Los errores que se han producido durante la programación de la aplicación han sido muchísimos durante el proceso. La mayoría de errores me han obligado a reestructurar en gran medida el código, los mas comunes han sido:

5.1 Bucles infinitos en whiles:

Uno de los errores más comunes estaba en las condiciones del while y en el momento de obtener el siguiente valor para llevar a cabo secuencias, al ser estructuras complejas, he tendido a confundirme, pero finalmente, paso a paso, he conseguido que funcione correctamente.

5.2 Errores de lectura desde el fichero:

Es muy importante que en el fichero estén exactamente los datos , tipos de datos, y posiciones que se van a leer, por culpa de ello el programa se colapsaba o no leía correctamente, me basto con controlar bien sobretodo el tipo y orden de datos que se encuentra en el fichero.

5.3 Errores en el momento de localizar las posiciones de las tuplas, errores en los índices de posición.

Este error está muy relacionado con los bucles infinitos, muchas veces, introducía datos o leía datos o mostraba datos en pantalla con la posición incorrecta y por lo tanto, no funcionaba bien.

Para solucionarlo, inicializaba correctamente los índices e incrementaba el valor cuando era necesario.

5.4 Errores de sobreescritura en el fichero.

Cuando modificaba datos, al sobrescribir en el fichero, se desordenaba todo, o simplemente, el programa escribía “0 0 0 0 0 0 0 0 0” .

Esto ocurría porque la posición no se borraba correctamente y el número de control para saber cuántas posiciones estaban ocupadas, no era correcto, lo solucione repasando paso por paso y localizando el momento exacto donde se perdía la posición.

5.5 Al introducir datos desde la consola, un error que me ha retrasado mucho ha sido, cuando introducía datos al azar (para probar si el sistema funcionaba correctamente), el programa se bloqueaba.

Este error se producía, porque introducía datos al azar que no eran del tipo correcto de la variable donde la estaba introduciendo, por ello el programa no lo podía guardar y aparecía el error.

Simplemente se ha de tener en cuenta que a la hora de introducir datos, el dato introducido tiene que ser del mismo tipo que la variable en la que lo quieres introducir, un error básico que ya nos enseñaron a principio de curso.

6. Conclusiones.

En resumen, puedo decir que ha sido un proyecto muy entretenido, frustrante en ocasiones cuando no conseguía que el programa hiciese lo que quería, pero que se compensaba con creces cuando lo solucionaba.

He consolidado los conocimientos obtenidos en clase y he aprendido mucho más de programación de lo que esperaba. He disfrutado dándole color al programa y cambiando la interfaz gráfica, pero me ha agradado más acabar el proyecto y ver que todo el tiempo empleado ha servido para que funcione bien.

En conclusión, me ha tomado más tiempo del que me esperaba acabarlo, y he tenido que consultar con la profesora varias veces pero finalmente, el resultado obtenido, es el deseado.