

# MEMORIA PRÁCTICA FINAL

Autores: Carlos Sánchez Mora y Jairo Pulido Garrido  
Grupo: 33 Viernes      Curso: 2019-20

Fecha de entrega: 8/1/2020



# **ÍNDICE**

1.Objetivos.....	5
2.Descomposición modular.....	5
3.Codigo Fuente.....	7
4.Pruebas de ejecución.....	39

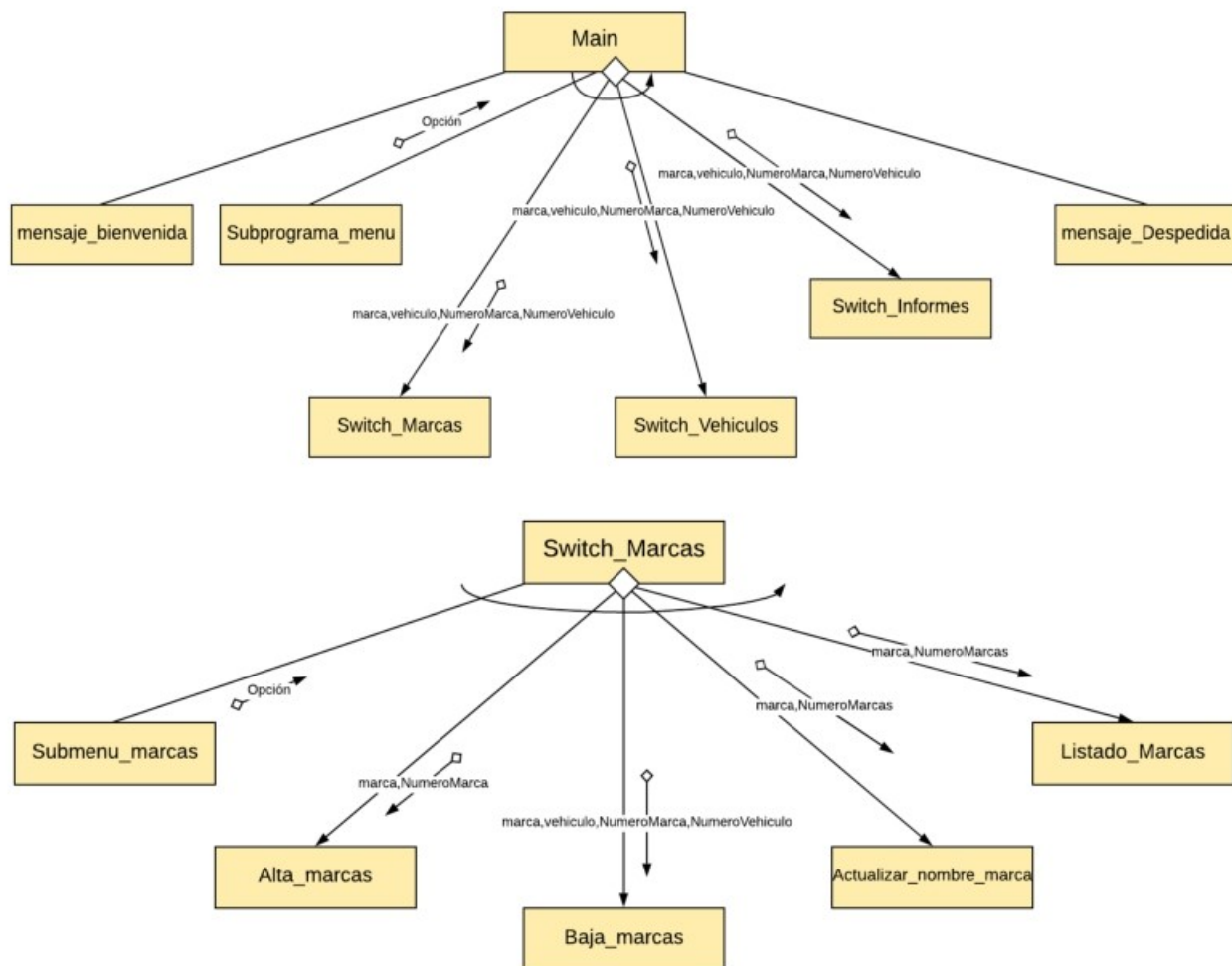


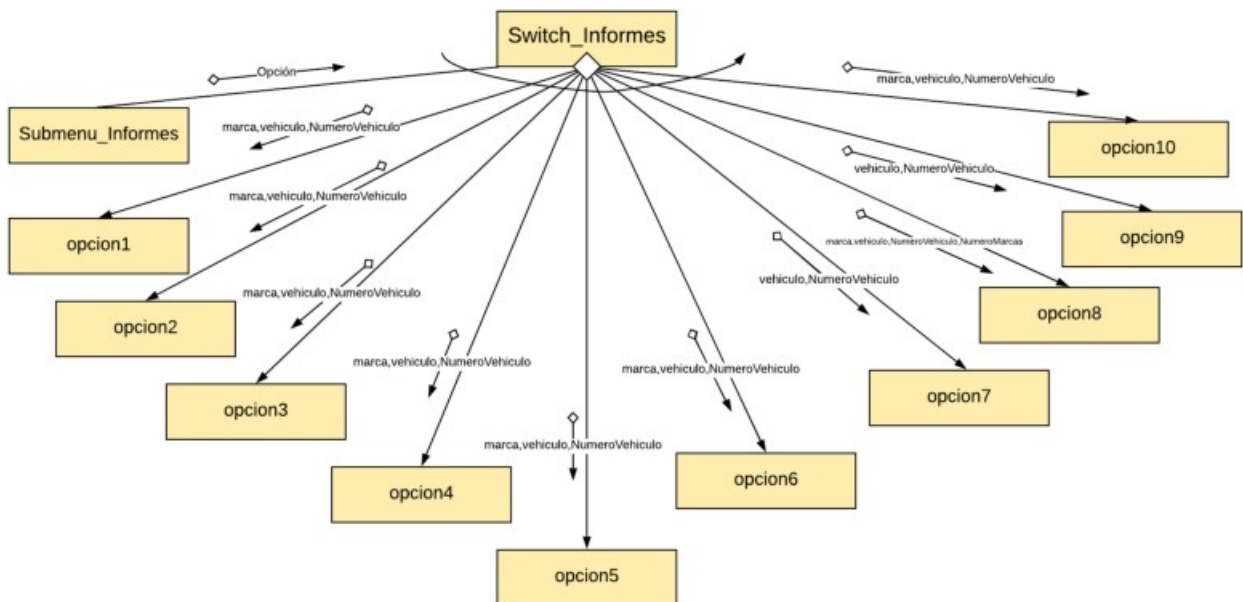
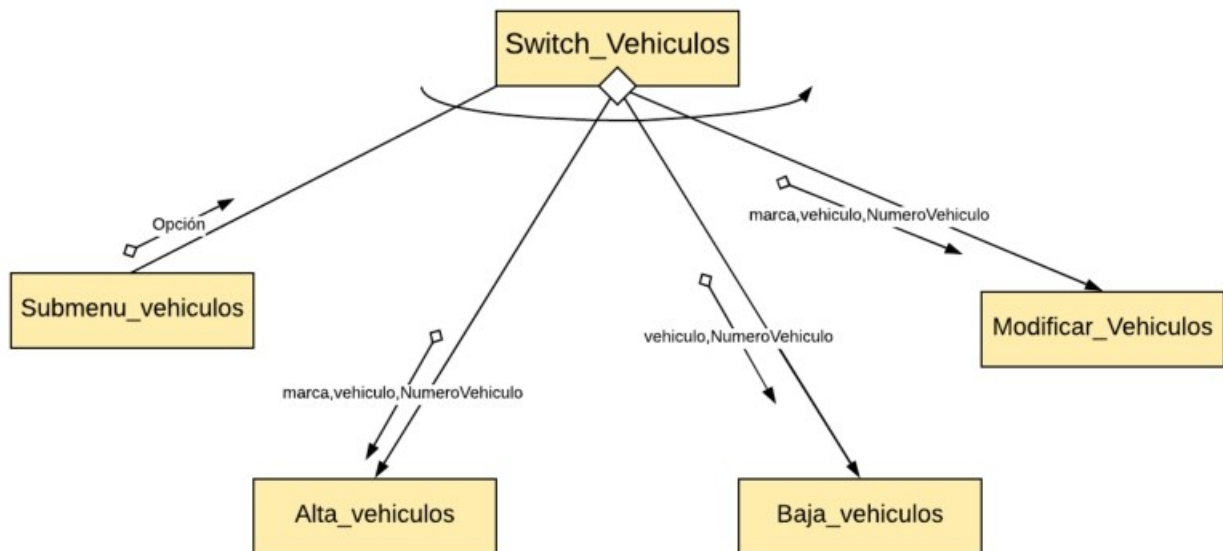
# OBJETIVOS

Ante la llegada del fin del año, una empresa especializada en la venta de vehículos decide lanzar una campaña con importantes descuentos con el fin de dar salida al stock de vehículos que tiene en su almacén. Se considera que la empresa puede comercializar un máximo de 5 marcas distintas (en este momento trabajan con las marcas Audi, BMW y Jeep, pero se prevee ampliar con otras marcas en el futuro). Así mismo, se considera que la empresa puede comercializar 10 tipos de vehículos diferentes, aunque pueden existir varias unidades de cada uno de ellos.

El programa debe permitir realizar alta / baja / modificación / consulta de las marcas y de los vehículos que vende la empresa, y además, proporcionar diferentes listados con información relevante.

## DIAGRAMA DE DESCOMPOSICIÓN MODULAR





# CÓDIGO FUENTE

/\*\*\*\*\*\*

APLICACIÓN GESTIAUTO

Alumno 1: Pulido Garrido, Jairo

Alumno 2: Sánchez Mora, Carlos

Grupo: 33

Fecha: 3/1/2020

\*\*\*\*\*/

//-----LIBRERIAS-----

#include <stdio.h>

#include <locale.h>

#include <time.h>

#include <math.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

#include <ctype.h>

#define SIZECODE 12

//-----ESTRUCTURAS DE DATOS----

typedef struct marcas{

    char marca[15];

    int NumeroMarca;

    int existencias;

}tRegMarcas;

typedef struct vehiculos {

    char codigo[SIZECODE];

    char gama[25];

    int marca;           // de la 0-4

    char modelo[20];

    int motor;           //(valor 1 - diésel, 2 - gasolina, 3 - híbrido, 4 - eléctrico)

    int potencia\_KW;

    int potencia\_CV;

    float consumo\_medio;

    float precio;

    int descuento;

    int unidades;       // =0 si no queda ninguna.

}tRegVehiculos;

//-----PROTOTIPOS DE PROGRAMA-----

int Subprograma\_menu();

void mensaje\_bienvenida();

void mensaje\_Despedida();

void subprograma\_Pausa();

int Submenu\_Marcas();

int Submenu\_Vehiculos();

void mostrarMotor(tRegVehiculos \*vehiculo,int n);

void baja(tRegVehiculos \*vehiculo, int n, int \*NumeroVehiculo);

void Alta\_marcas(tRegMarcas \*marca, int \*NumeroMarca);

void Baja\_marcas(tRegMarcas \*marca, tRegVehiculos \*vehiculo, int \*NumeroMarca,int \*NumeroVehiculo);

void Alta\_vehiculos(tRegVehiculos \*vehiculo, tRegMarcas \*marca,int \*NumeroVehiculo);

void Switch\_marcas(tRegMarcas \*marca, tRegVehiculos \*vehiculo, int \*NumeroMarca,int \*NumeroVehiculo);

void Switch\_Vehiculos(tRegMarcas \*marca, tRegVehiculos \*vehiculo, int \*NumeroVehiculo);

void Actualizar\_nombre\_marca(tRegMarcas \*marca, int NumeroMarca);

void Listado\_Marcas(tRegMarcas \*marca, int NumeroMarca);

void Baja\_vehiculos(tRegVehiculos \*vehiculo,int \*NumeroVehiculo);

void Modificar\_Vehiculos(tRegVehiculos \*vehiculo, tRegMarcas \*marca,int NumeroVehiculo);

int Submenu\_Informes();

void Switch\_informes(tRegVehiculos \*vehiculo,tRegMarcas \*marca, int NumeroVehiculo, int NumeroMarca);

```
void Listado_opcion1(tRegVehiculos *vehiculo,tRegMarcas *marca,int NumeroVehiculo);
void Listado_opcion3(tRegVehiculos *vehiculo,tRegMarcas *marca,int NumeroVehiculo);
void Listado_opcion7(tRegVehiculos *vehiculo,int NumeroVehiculo);
void Listado_opcion2(tRegVehiculos *vehiculo,tRegMarcas *marca,int NumeroVehiculo);
void Listado_opcion4(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo);
void Listado_opcion9(tRegVehiculos *vehiculo, int NumeroVehiculo);
void Listado_opcion6(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo);
void Listado_opcion8(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo,int NumeroMarca);
void Listado_opcion10(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo);
void Listado_opcion5(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo);
void Inicializar(tRegVehiculos *vehiculo,tRegMarcas *marca);
//-----MAIN O PRINCIPAL-----
int main(){

    int opcion;

    int NumeroVehiculo = 3,NumeroMarca = 3; // esta variable es para contar el numero de vehiculos dados de alta.
    tRegVehiculos vehiculo[10] = {"BMW X1 SDR110","X1",0,"sDrive18d",1,110,150,5.5,36700,10,3},
{"BMW SE630195","Serie 6-Gran Turismo",0,"630d",1,195,265,6.8,79569,12,2},
{"JEE WRSP0203","Wrangler",1,"Sport 2.0",2,203,270,11.25,42562,15,1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1},{""",-1,-1,-1,-1,-1,-1,-1};
    tRegMarcas marca[5] = {"BMW",0,5},{"Jeep",1,1},{"Audi",2,0},{"" ,3,-1},{"" ,4,-1});
    mensaje_bienvenida();
    setlocale(LC_CTYPE,"Spanish");

do{
    opcion=Subprograma_menu();
    system("cls");

    switch(opcion){

        case 1: Switch_marcas(marca,vehiculo,&NumeroMarca,&NumeroVehiculo);
            break;

        case 2: Switch_Vehiculos(marca,vehiculo,&NumeroVehiculo);
            break;

        case 3: Switch_informes(vehiculo,marca,NumeroVehiculo,NumeroMarca);
            break;

        case 0: mensaje_Despedida();
            break;

        default: printf("\n ERROR, introduce un numero para ejecutar una opción \n");
    }
}while(opcion!=0);

return 0;
} // fin del main

//-----
/* Subprograma:          Subprograma_menu                      */
/* Tipo:                Función                                */
/* Parámetros de entrada: ----                               */
/* Parámetros de Salida: Un entero(la opción elegida)         */
/* Prerrequisitos:      ----                                  */
/* Objetivo:             Presenta en pantalla un listado de opciones del*/
/*                       que el usuario elegira una.           */
/*-----*/
int Subprograma_menu(){

    int opcion;

    printf("\n\n\n\t\t\t MENU PRINCIPAL \n");
```



```

printf("\t\t\t\t\t----- \n");
printf("\n\t\t\t\t\t 1.- Marcas \n\n");
printf("\t\t\t\t\t 2.- Vehiculos\n\n");
printf("\t\t\t\t\t 3.- Informes \n\n");
printf("\t\t\t\t\t 0.- Salir \n\n");
printf("\n\n\t\t\t\t\t Elija opción: ");
scanf("%i",&opcion);
return(opcion);
}

//-----
/*****
/* Subprograma:      mensaje_bienvenida      */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  ----              */
/* Parámetros de Salida:  ----              */
/* Prerrequisitos:      ----                */
/* Objetivo:          Presenta en pantalla un mensaje de bienvenida */
/*                   al usuario.              */
*****/

void mensaje_bienvenida (){

printf(" \n\n\n\n\n \t\t\t\t\t----- \n");
printf(" \n\n\n\n\n \t\t\t\t\t *****BIENVENIDO A GESTIAUTO***** ");
printf(" \n\n\n\n\n \t\t\t\t\t-----\n");
subprograma_Pausa();
system("cls");
}

//-----
/*****
/* Subprograma:      mensaje_Despedida      */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  ----              */
/* Parámetros de Salida:  ----              */
/* Prerrequisitos:      ----                */
/* Objetivo:          Presenta en pantalla un mensaje de despedida al*/
/*                   usuario.                */
*****/

void mensaje_Despedida(){

system("cls");
printf(" \n\n\t\t\t\t\tGRACIAS POR UTILIZAR NUESTRO PROGRAMA \n\n");
}

//-----
/*****
/* Subprograma:      subprograma_Pausa      */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  ----              */
/* Parámetros de Salida:  ----              */
/* Prerrequisitos:      ----                */
/* Objetivo:          Presenta en pantalla un mensaje para que el u- */
/*                   suario pulse Intro y espera a ello.              */
*****/

void subprograma_Pausa(){

printf("\n\n\n\n\n\t\t\t\t\tPulse <Intro> para continuar...");
fflush(stdin);
getchar();
}

//-----
/*****
/* Subprograma:      subprograma_baja      */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  tRegVehiculos *vehiculo, int n, int *NumeroVehiculo
/* Parámetros de Salida:  ----              */

```

```

/* Prerrequisitos:      ---- */
/* Objetivo:            decrementa el contador del numero de vehiculos
                        cuando se efectua la baja y pone las variables con el campo centinela -1*/
/***** */
void baja(tRegVehiculos *vehiculo, int n, int *NumeroVehiculo){

    vehiculo[n] = vehiculo[*NumeroVehiculo-1];

    strcpy(vehiculo[*NumeroVehiculo-1].codigo,"");
    vehiculo[*NumeroVehiculo-1].unidades = -1;
    vehiculo[*NumeroVehiculo-1].marca = -1;
    vehiculo[*NumeroVehiculo-1].potencia_KW = -1;
    vehiculo[*NumeroVehiculo-1].precio = -1;
    *NumeroVehiculo = *NumeroVehiculo-1;
}
//-----
/***** */
/* Subprograma:          subprograma_mostrar_motor */
/* Tipo:                 Procedimiento */
/* Parámetros de entrada: ---- */
/* Parámetros de Salida: ---- */
/* Prerrequisitos:      ---- */
/* Objetivo:             efectua la conversion de tipo entero (tipo de motor)
                        a caracteres letra */
/***** */
void mostrarMotor(tRegVehiculos *vehiculo, int n){
    switch(vehiculo[n].motor){
        case 1: printf(" Diésel");
            break;
        case 2: printf(" Gasolina");
            break;
        case 3: printf(" Híbrido");
            break;
        case 4: printf("Eléctrico");
            break;
    }
}
//-----
/***** */
/* Subprograma:          subprograma_Pausa */
/* Tipo:                 Procedimiento */
/* Parámetros de entrada: ---- */
/* Parámetros de Salida: ---- */
/* Prerrequisitos:      ---- */
/* Objetivo:             Presenta en pantalla un mensaje para que el u- */
/*                        suario pulse Intro y espera a ello. */
/***** */
int Submenu_Marcas()
{

    int opcion;

    printf("\n\n \t\t\tMENU MARCAS");
    printf("\n\t\t\t----- \n\n");
    printf("\t\t\t1.- Alta \n\n");
    printf("\t\t\t2.- Baja \n\n");
    printf("\t\t\t3.- Actualizar nombre de la marca \n\n");
    printf("\t\t\t4.- Listado de marcas \n\n");
    printf("\t\t\t0.- Volver al Menu Principal.\n\n");
    printf("\t\t\tElija una opción: ");
    fflush(stdin);
    scanf("%i",&opcion);

    return(opcion);
}

```

```

}
//-----
/*****
/* Subprograma:          Switch_marcas          */
/* Tipo:                Procedimiento          */
/* Parámetros de entrada: Arrays con los datos de los vehiculos y marcas,*/
/*                      además de dos enteros.          */
/* Parámetros de Salida: ----          */
/* Prerrequisitos:      tener marcas y vehiculos registrados */
/* Objetivo:            Pide al usuario un opcion por medio de */
/*                      Submenu_Marcas y ejecuta otro subprograma */
/*                      dependiendo de la opción elegida.          */
*****/

```

```

void Switch_marcas(tRegMarcas *marca, tRegVehiculos *vehiculo, int *NumeroMarca,int *NumeroVehiculo){

```

```

    int opcion;

```

```

    do{

```

```

        opcion=Submenu_Marcas();

```

```

        system("cls");

```

```

        switch(opcion){

```

```

            case 1: Alta_marcas(marca,NumeroMarca);

```

```

                subprograma_Pausa();

```

```

                system("cls");

```

```

                break;

```

```

            case 2: Baja_marcas(marca,vehiculo,NumeroMarca,NumeroVehiculo);

```

```

                subprograma_Pausa();

```

```

                system("cls");

```

```

                break;

```

```

            case 3: Actualizar_nombre_marca(marca,*NumeroMarca);

```

```

                subprograma_Pausa();

```

```

                system("cls");

```

```

                break;

```

```

            case 4: Listado_Marcas(marca,*NumeroMarca);

```

```

                subprograma_Pausa();

```

```

                system("cls");

```

```

                break;

```

```

            case 0: system("cls");

```

```

                break;

```

```

            default : printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR
FAVOR, INGRESE UN NUMERO VALIDO.\n ");

```

```

        } // fin del switch

```

```

    }while (opcion!=0);

```

```

}

```

```

//-----
/*****
/* Subprograma:          Submenu_Vehiculos          */
/* Tipo:                Función          */
/* Parámetros de entrada: ----          */
/* Parámetros de Salida: Un entero(la opción elegida) */
/* Prerrequisitos:      ----          */
/* Objetivo:            Presenta en pantalla un listado de opciones y */
/*                      pide al usuario que elija una          */
*****/

```

```

int Submenu_Vehiculos()
{
    int opcion;

    printf("\n\n\t\t\tMENU VEHÍCULOS");
    printf("\n\t\t\t----- \n\n");
    printf("\t\t\t1.- Alta \n\n");
    printf("\t\t\t2.- Baja \n\n");
    printf("\t\t\t3.- Modificación \n\n");
    printf("\t\t\t0.- Volver al Menú Principal.\n\n");
    printf("\t\t\tElija una opción: ");
    fflush(stdin);
    scanf("%i",&opcion);

    return(opcion);
}

//-----
/*****
/* Subprograma:          Switch_Vehiculos          */
/* Tipo:                Procedimiento              */
/* Parámetros de entrada: Arrays con los datos de los vehiculos y marcas,*/
/*                      además de dos enteros.      */
/* Parámetros de Salida: ----                      */
/* Prerrequisitos:      Tener marcas y vehiculos registrados */
/* Objetivo:            Pide al usuario un opcion por medio de */
/*                      Submenu_Vehiculos y ejecuta otro subprograma */
/*                      dependiendo de la opción elegida.      */
*****/
void Switch_Vehiculos(tRegMarcas *marca, tRegVehiculos *vehiculo, int *NumeroVehiculo){

    int opcion;

    do{
        opcion = Submenu_Vehiculos();
        system("cls");

        switch(opcion){

            case 1: Alta_vehiculos(vehiculo,marca,NumeroVehiculo);
                    subprograma_Pausa();
                    system("cls");
                    break;

            case 2: Baja_vehiculos(vehiculo,NumeroVehiculo);
                    subprograma_Pausa();
                    system("cls");
                    break;

            case 3: Modificar_Vehiculos(vehiculo,marca,*NumeroVehiculo);
                    subprograma_Pausa();
                    system("cls");
                    break;

            case 0: system("cls");
                    break;

            default : printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR
FAVOR, INGRESE UN NUMERO VALIDO.\n ");

        } // fin del switch
    }while (opcion!=0);
}

//-----
/*****

```

```

/* Subprograma: Alta_marcas */
/* Tipo: Función */
/* Parámetros de entrada: Array con los datos de las marcas y un entero */
/* Parámetros de Salida: - */
/* Prerrequisitos: No tener 5 marcas registradas */
/* Objetivo: Pide al usuario el nombre de la marca a dar de alta y comprueba si esta ya registrada, en caso negativo, la dara de alta con 0 unidades. */
/*****/
void Alta_marcas(tRegMarcas *marca, int *NumeroMarca){

    char marcaAux[15];
    int aux = *NumeroMarca, seguir = 1;

    if(*NumeroMarca == 5){ // SI EL NUMERO DE MARCAS YA ES 5, DIRECTAMENTE NO SE PUEDEN ALMACENAR MAS.
        printf("\n\t\t\t\t\tATENCIÓN, no se pueden almacenar mas marcas \n");
    }else{
        printf("\n\n\t\t\t\t\tSERVICIO DE ALTA DE NUEVA MARCA");
        printf("\n\t\t\t\t\t----- \n\n");
        printf("\t\t\t\t\tPor favor, introduce la marca de vehículos que deseas introducir en el sistema: ");
        fflush(stdin);
        gets(marcaAux);
        fflush(stdin);

        for(int i = 0; i < aux; i++)
        {
            if(marca[i].existencias == -1)
            {
                aux++;
            }
            if(strcmp(marca[i].marca,marcaAux) == 0)
            {
                printf("\n\t\t\t\t\tERROR LA MARCA YA ESTA REGISTRADA");
                seguir = 0;
            }
        }
        if(seguir == 1){
            strcpy(marca[*NumeroMarca].marca,marcaAux);
            marca[*NumeroMarca].existencias = 0;
            marca[*NumeroMarca].NumeroMarca = *NumeroMarca;
            *NumeroMarca = *NumeroMarca + 1;
            printf("\n\n\t\t\t\t\tMarca %s registrada",marca[*NumeroMarca - 1].marca);
        }
    }
}

//-----
/*****/
/* Subprograma: Baja_marcas */
/* Tipo: Función */
/* Parámetros de entrada: Arrays con los datos de marcas y vehiculos, a- demás de dos enteros. */
/* Parámetros de Salida: - */
/* Prerrequisitos: Tener marcas registradas */
/* Objetivo: Pide al usuario el nombre de la marca a dar de baja(si hay varias), comprueba si tiene existencias, en caso negativo da la marca de baja y a todos los vehiculos asociados. */
/*****/
void Baja_marcas(tRegMarcas *marca, tRegVehiculos *vehiculo, int *NumeroMarca,int *NumeroVehiculo){

    char marcaAux[15];
    char respuesta;

```

```

for (int i=0; i<aux; i++){
    if(marca[i].existencias == -1)
    {
        aux++;
    }
    if(strcmp(marca[i].marca, marcaAux)==0)
    {
        seguir = 0;
        if (marca[i].existencias!=0){
            printf("\n\t\t\t\t *** ERROR, AÚN QUEDAN EXISTENCIAS EN EL ALMACÉN *** \n");
        }else{
            printf("\n\n\t\t\t\t ***MARCA ENCONTRADA***, ¿ Desea eliminarla ? (S/N)");
            scanf("%c",&respuesta);
            if(toupper(respuesta)=='S'){
                strcpy(marca[i].marca,"");
                marca[i].existencias = -1;
                *NumeroMarca= *NumeroMarca-1;
                for(int j = 0;j < *NumeroVehiculo;j++)
                {
                    if(vehiculo[j].marca == marca[i].NumeroMarca)
                    {
                        baja(vehiculo,j,NumeroVehiculo);
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    printf("\n\t\t\t\t\t *** MARCA ELIMINADA *** \n");
} else if (toupper(respuesta) == 'N') {
    printf("\n\t\t\t\t\t *** PROCESO ABORTADO *** \n");
} else {
    printf("\n\t\t\t\t\t ERROR: OPCION NO VÁLIDA");
}
}
}
}
if (seguir == 1) {
    printf("\n\t\t\t\t\t *** NO SE ENCUENTRA LA MARCA INTRODUCIDA, inténtelo de nuevo *** \n");
    subprograma_Pausa();
    system("cls");
    Baja_marcas(marca, vehiculo, NumeroMarca, NumeroVehiculo);
}
}
} // fin subprograma baja.

```

```

//-----
/*****
/* Subprograma:      Alta_vehiculos          */
/* Tipo:            Función                  */
/* Parámetros de entrada:  Arrays con los datos de los vehiculos y marcas, */
/*                    además de un entero          */
/* Parámetros de Salida:   -                  */
/* Prerrequisitos:       No tener 10 vehiculos registrados          */
/* Objetivo:            Pide al usuario todos los datos a registrar, */
/*                    genera el codigo y verifica si ya existe, en ca- */
/*                    so negativo, da de alta el vehiculo y actualiza */
/*                    las existencias de la marca asociada.          */
*****/
void Alta_vehiculos(tRegVehiculos *vehiculo, tRegMarcas *marca, int *NumeroVehiculo) {

```

```

    tRegVehiculos vehiculoaux;
    char a, b, c, respuesta;
    int seguir = 1;

    printf("\n\n \t\t\t\t\t SERVICIO DE ALTA DE UN VEHÍCULO");
    printf("\n\t\t\t\t\t----- \n\n");
    if (*NumeroVehiculo == 10)
    {
        printf("ERROR: NO SE PUEDEN DAR MÁS VEHICULOS DE ALTA");
    } else {
        printf("\n\t\t\t\t\t Introduzca los siguientes datos del vehículo :\n");

        printf("\n\t\t\t\t\t Marca:");
        for (int i = 0; i < 5; i++)
        {
            if (marca[i].existencias != -1) {
                printf("%i .- %s ", (i), marca[i].marca);
            }
        }
        do {
            fflush(stdin);
            scanf("%i", &vehiculoaux.marca);
            if ((vehiculoaux.marca > 0) || (vehiculoaux.marca < 4)) {
                if (marca[vehiculoaux.marca].existencias == -1) {
                    printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
                    vehiculoaux.marca = -1;
                }
            } else {

```

```

        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
    }
}while((vehiculoaux.marca < 0)||((vehiculoaux.marca > 4)));
fflush(stdin);
printf("\n\t\t\t\t\tGama: ");
gets(vehiculoaux.gama);
fflush(stdin);
printf("\n\t\t\t\t\tModelo: ");
gets(vehiculoaux.modelo);
fflush(stdin);

do{
    printf("\n\t\t\t\t\tPotencia del motor (Kw): ");
    scanf("%i",&vehiculoaux.potencia_KW);
    fflush(stdin);
    if(vehiculoaux.potencia_KW < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while(vehiculoaux.potencia_KW < 0);

do{
    printf("\n\t\t\t\t\tCombustible: (1.-Diésel, 2.-Gasolina, 3.-Híbrido, 4.-Eléctrico) ");
    scanf("%i",&vehiculoaux.motor);
    fflush(stdin);
    if((vehiculoaux.motor < 0)||((vehiculoaux.motor > 4))
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while((vehiculoaux.motor < 0)||((vehiculoaux.motor > 4)));

do{
    printf("\n\t\t\t\t\tPotencia del motor (CV): ");
    scanf("%i",&vehiculoaux.potencia_CV);
    fflush(stdin);
    if(vehiculoaux.potencia_CV < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while(vehiculoaux.potencia_CV < 0);

do{
    printf("\n\t\t\t\t\tConsumo medio (L/100): ");
    scanf("%f",&vehiculoaux.consumo_medio);
    fflush(stdin);
    if(vehiculoaux.consumo_medio < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while(vehiculoaux.consumo_medio < 0);

do{
    printf("\n\t\t\t\t\tPrecio (PVP): ");
    scanf("%f",&vehiculoaux.precio);
    fflush(stdin);
    if(vehiculoaux.precio < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while(vehiculoaux.precio < 0);

do{
    printf("\n\t\t\t\t\tDescuento (sobre PVP): ");
    scanf("%i",&vehiculoaux.descuento);
    fflush(stdin);
    if(vehiculoaux.descuento < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}while(vehiculoaux.descuento < 0);

do{
    printf("\n\t\t\t\t\tUnidades existentes: ");
    scanf("%i",&vehiculoaux.unidades);
    fflush(stdin);
    if(vehiculoaux.unidades < 0)
        printf("\n\t\t\t\t\tERROR, VALOR NO VÁLIDO");
}

```



```

}while(vehiculoaux.unidades < 0);

printf("\n\t\t\t\t ¿ *** LOS DATOS SON CORRECTOS (S/N) ***? ");
scanf("%c",&respuesta);

respuesta = toupper(respuesta);
a=marca[vehiculoaux.marca].marca[0];
b=marca[vehiculoaux.marca].marca[1];
c=marca[vehiculoaux.marca].marca[2];

sprintf(vehiculoaux.codigo,"%c%c%c%c%c%c%c%c%c%c%
%i",toupper(a),toupper(b),toupper(c),toupper(vehiculoaux.gama[0]),toupper(vehiculoaux.gama[1]),toupper(vehiculoaux.modelo[0]),toupper(vehiculoaux.modelo[1]),toupper(vehiculoaux.modelo[2]),vehiculoaux.potencia_KW);

if(respuesta == 'N'){
    printf("\n\t\t\t\t Repita el proceso: ");
    subprograma_Pausa();
    system("cls");
    Alta_vehiculos(vehiculo,marca,NumeroVehiculo);
}else if (respuesta == 'S'){
    for(int i = 0; i < *NumeroVehiculo; i++)
    {
        if(strcmp(vehiculo[i].codigo,vehiculoaux.codigo) == 0)
        {
            printf("\n\t\t\t\t ERROR: VEHICULO YA EXISTENTE");
            seguir = 0;
        }
    }
    if(seguir == 1){
        marca[vehiculoaux.marca].existencias = marca[vehiculoaux.marca].existencias+vehiculoaux.unidades;
        fflush(stdin);
        vehiculo[*NumeroVehiculo] = vehiculoaux;
        *NumeroVehiculo = *NumeroVehiculo+1;
    }
}

printf("\n\t\t\t\t ¿ Desea dar de alta un nuevo vehiculo (S/N) ? ");
scanf("%c",&respuesta);
fflush(stdin);
respuesta = toupper(respuesta);// transformamos a mayúscula si no lo está para evitar fallos input.

if(respuesta == 'S'){
    system("cls");
    Alta_vehiculos(vehiculo,marca,NumeroVehiculo);
}
}
// fin subprograma

//-----
/* Subprograma: Actualizar_nombre_marca */
/* Tipo: Procedimiento */
/* Parámetros de entrada: Un array con datos de las marcas y un entero */
/* Parámetros de Salida: ---- */
/* Prerrequisitos: Tener marcas registradas */
/* Objetivo: Pide al usuario el nombre de la marca a modifi-*/
/* car(en caso de haber varias) y el nuevo nombre,*/
/* tras esto, los intercambia. */
/*-----*/
void Actualizar_nombre_marca(tRegMarcas *marca, int NumeroMarca){

```

```

char marcaAux[10];

if(NumeroMarca == 0)
{
    printf("\n\t\t\t\t\tNO HAY MARCAS QUE MODIFICAR");
}
else if(NumeroMarca == 1){
    for(int i=0; i<5; i++){
        if(marca[i].existencias != -1)
        {
            printf("\n\t\t\t\t\tIntroduce un nuevo nombre para la marca %s: ",marca[i].marca);
            fflush(stdin);
            gets(marcaAux);
            strcpy(marca[i].marca,marcaAux);
            printf("\n\t\t\t\t\t***NOMBRE CAMBIADO CORRECTAMENTE***\n");
            break;
        }
    }
}
else{
    printf("\n\n \t\t\t\t\tSERVICIO DE MODIFICACIÓN DE MARCA");
    printf("\n\t\t\t\t\t----- \n\n");
    printf("\t\t\t\t\tPor favor, introduce la marca de vehículos que deseas modificar: ");
    fflush(stdin);
    gets(marcaAux);

    for(int i=0; i<5; i++){
        if(strcmp(marca[i].marca,marcaAux)== 0)
        {
            printf("\n\t\t\t\t\tIntroduce un nuevo nombre para la marca %s: ",marca[i].marca);
            fflush(stdin);
            gets(marcaAux);
            strcpy(marca[i].marca,marcaAux);
            printf("\n\t\t\t\t\t***NOMBRE CAMBIADO CORRECTAMENTE***\n");
            break;
        }
        else if(i==4){
            printf(" \n\t\t\t\t\t***IMPOSIBLE MODIFICAR***, la marca introducida no se encuentra en el sistema \n ");
        }
    }
}
}
}
//-----
/*****
/* Subprograma:          Listado_Marcas          */
/* Tipo:                Procedimiento          */
/* Parámetros de entrada: Recibe un array que almacena los datos de todas*/
/*                      las marcas y un entero          */
/* Parámetros de Salida:  ----          */
/* Prerrequisitos:       Debe haber marcas registradas          */
/* Objetivo:             Presenta en pantalla un listado de el nombre de*/
/*                      todas las marcas con su numero asociado.          */
*****/

void Listado_Marcas(tRegMarcas *marca, int NumeroMarca){

    int aux = 0;
    if(NumeroMarca == 0)
    {
        printf("\n\t\t\t\t\tNO HAY MARCAS REGISTRADAS");
    }
    else{
        printf("\n\n \t\t\t\t\tLISTADO GENERAL DE MARCAS");
        printf("\n\t\t\t\t\t-----\n\n");

        for (int i=0; i<NumeroMarca+aux; i++){
            if(marca[i].existencias != -1)
                printf("\t\t\t\t\tla marca numero %i es %s \n ",marca[i].NumeroMarca,marca[i].marca);
            else

```

```

        aux++;
    }
}
}
//-----
/*****
/* Subprograma:      Baja_veihculos          */
/* Tipo:             Función                  */
/* Parámetros de entrada:  Array con los datos de los vehiculos y entero */
/* Parámetros de Salida:   -                  */
/* Prerrequisitos:       Tener vehiculos registrados */
/* Objetivo:           Pide el codigo del vehiculo a dar de baja al */
/*                     usuario(si hay varios) y comprueba si hay exis-*/
/*                     tencias,en caso negativo, lo da de baja.      */
/*****
void Baja_vehiculos(tRegVehiculos *vehiculo,int *NumeroVehiculo){

    char Auxcodigo[15];
    char Baja;
    int seguir = 1;

    printf("\n\n \t\t\t\t BAJA DE UN VEHICULO");
    printf("\n\n\t\t\t\t-----\n\n");

    printf("\n\n\t\t\t\t Introduzca el código del vehiculo a dar de baja ");
    fflush(stdin);
    gets(Auxcodigo);

    for (int i=0; i<*NumeroVehiculo; i++){
        if(strcmp(vehiculo[i].codigo,Auxcodigo)==0)
        { // vehiculo encontrado

            seguir = 0;
            printf("\n\n\t\t\t\t Vehículo encontrado ,¿Desea darlo de baja con todos sus datos ?(S/N) ");
            scanf("%c",&Baja);
            fflush(stdin);
            Baja = toupper(Baja);

            if(vehiculo[i].unidades==0){ // las unidades pueden estar a -1(inicializadas) o agotadas (a 0)
                if (Baja == 'N'){
                    printf("\n\n\t\t\t\t *** PROCESO DE BAJA ABORTADO*** \n");
                }else{
                    baja(vehiculo,i,NumeroVehiculo);
                    printf("\n\n\t\t\t\t *** VEHÍCULO ELIMINADO CORRECTAMENTE*** \n");
                }
            }else {
                printf("\n\n\t\t\t\t ***NO SE PUEDE ELIMINAR , AÚN QUEDAN EXISTENCIAS EN EL ALMACÉN*** \n");
            }
        }
    }
    if(seguir == 1){
        printf("\n\n\t\t\t\t ***NO SE ENCUENTRA EL VEHÍCULO INTRODUCIDO*** \n");
        subprograma_Pausa();
        system("cls");
        Baja_vehiculos(vehiculo,NumeroVehiculo);
    }
}

//-----
/*****
/* Subprograma:      Modificar_Vehiculos          */
/* Tipo:             Función                  */
/* Parámetros de entrada:  Recibe 2 arrays que almacenan los datos de los */
/*                     vehiculos y marcas,además de un entero.      */

```

[illegible]

```

printf("\t\t\t\t\t9.- Unidades existentes \n");
printf("\t\t\t\t\t10.- Motor \n");
printf("\t\t\t\t\t0.- Salir de Modificación ");
printf("\n\t\t\t\t\t Elije una opción: ");
fflush(stdin);
scanf("%i",&opcion);

system("cls");
switch(opcion){
    case 1: marca[vehiculo[i].marca].existencias = marca[vehiculo[i].marca].existencias -
vehiculo[i].unidades;
        printf("\n\n\t\t\t\t\t Nueva Marca (valor entero 0-4): ");
        scanf("%i",&aux);
        if(marca[aux].existencias != -1)
        {
            vehiculo[i].marca=aux;
            printf("\n\n\t\t\t\t\t ----- \n");
            printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
            printf("\t\t\t\t\t ----- \n");
            marca[vehiculo[i].marca].existencias = marca[vehiculo[i].marca].existencias +
vehiculo[i].unidades;
        }else{
            printf("\n\t\t\t\t\t ERROR: NINGUNA MARCA ASOCIADA AL NUMERO");
        }

        break;

    case 2: printf("\n\n\t\t\t\t\t Nueva gama: ");
        gets(vehiculo[i].gama);
        printf("\n\n\t\t\t\t\t ----- \n");
        printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
        printf("\t\t\t\t\t ----- \n");
        break;

    case 3: printf("\n\n\t\t\t\t\t Nuevo Modelo: ");
        gets(vehiculo[i].modelo);
        printf("\n\n\t\t\t\t\t ----- \n");
        printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
        printf("\t\t\t\t\t ----- \n");
        break;

    case 4: do{
        printf("\n\t\t\t\t\t Introduzca la nueva potencia(Kw): ");
        scanf("%i",&vehiculo[i].potencia_KW);
        fflush(stdin);
        if(vehiculo[i].potencia_KW < 0)
            printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
    }while(vehiculo[i].potencia_KW < 0);

    printf("\n\n\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");
    break;

    case 5: do{
        printf("\n\t\t\t\t\t Introduzca la nueva potencia(CV): ");
        scanf("%i",&vehiculo[i].potencia_CV);
        fflush(stdin);
        if(vehiculo[i].potencia_CV < 0)
            printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
    }while(vehiculo[i].potencia_CV < 0);

    printf("\n\n\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");

```

break;

```
case 6: do{
    printf("\n\t\t\t\t\t Introduzca el nuevo consumo: ");
    scanf("%f",&vehiculo[i].consumo_medio);
    fflush(stdin);
    if(vehiculo[i].consumo_medio < 0)
        printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
}while(vehiculo[i].consumo_medio < 0);

    printf("\n\n\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");
    break;
```

```
case 7: do{
    printf("\n\t\t\t\t\t Introduzca el nuevo precio (PVP): ");
    scanf("%f",&vehiculo[i].precio);
    fflush(stdin);
    if(vehiculo[i].precio < 0)
        printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
}while(vehiculo[i].precio < 0);
    printf("\n\n\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");
    break;
```

```
case 8: do{
    printf("\n\t\t\t\t\t Introduzca el nuevo descuento: ");
    scanf("%i",&vehiculo[i].descuento);
    fflush(stdin);
    if(vehiculo[i].descuento < 0)
        printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
}while(vehiculo[i].descuento < 0);
    printf("\n\n\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");
    break;
```

```
case 9: marca[vehiculo[i].marca].existencias = marca[vehiculo[i].marca].existencias -
vehiculo[i].unidades;
do{
    printf("\n\t\t\t\t\t Introduzca la nueva cantidad de unidades: ");
    scanf("%i",&vehiculo[i].unidades);
    fflush(stdin);
    if(vehiculo[i].unidades < 0)
        printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
}while(vehiculo[i].unidades < 0);
    marca[vehiculo[i].marca].existencias = marca[vehiculo[i].marca].existencias +
vehiculo[i].unidades;
    printf("\t\t\t\t\t ----- \n");
    printf("\t\t\t\t\t *** OK. DATOS GUARDADOS *** \n");
    printf("\t\t\t\t\t ----- \n");
    break;
```

```
case 10: do{
    printf("\n\n\t\t\t\t\t Combustible motor (1.-Diésel, 2.-Gasolina, 3.-Híbrido, 4.-Eléctrico) : ");
    scanf("%i",&vehiculo[i].motor);
    fflush(stdin);
    if((vehiculo[i].motor < 0)|| (vehiculo[i].motor > 4))
        printf("\n\t\t\t\t\t ERROR, VALOR NO VÁLIDO");
}while((vehiculo[i].motor < 0)|| (vehiculo[i].motor > 4));
    break;
```

```
case 0: break; // salir del subprograma
```

```

        default:printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR
FAVOR, INGRESE UN NUMERO VALIDO.\n ");

```

```

    }
    system("cls");
}while(opcion!=0);
}
}
if(opcion != 0){
    printf(" ***NO SE ENCUENTRA EL VEHÍCULO INTRODUCIDO , inténtelo de nuevo.*** \n");
    subprograma_Pausa();
    system("cls");
    Modificar_Vehiculos(vehiculo,marca,NumeroVehiculo);
}
}
}
//-----
/*****

```

```

/* Subprograma:      Submenu_Informes      */
/* Tipo:             Función                */
/* Parámetros de entrada:  ----            */
/* Parámetros de Salida:   Un entero(la opción elegida)      */
/* Prerrequisitos:       ----              */
/* Objetivo:           Presenta en pantalla un listado de opciones y */
/*                     pide al usuario que elija una            */
/*****

```

```

int Submenu_Informes(){

```

```

    int opcion;

```

```

    printf("\n\n \t\t\tMENU LISTADOS");
    printf("\n\n\t\t\t----- \n\n");
    printf("\t\t\t1.- Listado General de vehículos(completo) \n\n");
    printf("\t\t\t2.- Listado General de vehículos(resumido)\n\n");
    printf("\t\t\t3.- Listado de datos de un vehículo \n\n");
    printf("\t\t\t4.- Listado de vehículos (de una sola marca)\n\n");
    printf("\t\t\t5.- Listado de vehículos (ordenado por rango de potencias -Kw) \n\n");
    printf("\t\t\t6.- Listado de vehículos (ordenado por precio en oferta)\n\n");
    printf("\t\t\t7.- Importe total a cobrar(Todos los vehículos) \n\n");
    printf("\t\t\t8.- Importe total a cobrar (por marca)\n\n");
    printf("\t\t\t9.- Valores Medios (potencia,descuento,precio)\n\n");
    printf("\t\t\t10.- Comparar dos vehículos \n\n");
    printf("\t\t\t0.- Volver al Menu Principal.\n\n");
    printf("\t\t\tElija una opción: ");
    scanf("%i",&opcion);
    fflush(stdin);

```

```

    return(opcion);

```

```

}
//-----
/*****
/* Subprograma:      Switch_informes      */
/* Tipo:             Procedimiento        */
/* Parámetros de entrada:  Arrays con los datos de los vehiculos y marcas,*/
/*                     además de dos enteros.      */
/* Parámetros de Salida:   ----            */
/* Prerrequisitos:       tener marcas y vehiculos registrados      */
/* Objetivo:           Pide al usuario un opcion por medio de      */
/*                     Submenu_informes y ejecuta otro subprograma */
/*                     dependiendo de la opción elegida.          */
/*****

```

```

void Switch_informes(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo, int NumeroMarca){

```

```

int opcion;
do{
    opcion=Submenu_Informes();
    system("cls");

    switch(opcion){

        case 1: Listado_opcion1(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 2: Listado_opcion2(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 3: Listado_opcion3(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 4: Listado_opcion4(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 5: Listado_opcion5(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 6: Listado_opcion6(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 7: Listado_opcion7(vehiculo,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 8: Listado_opcion8(vehiculo,marca,NumeroVehiculo,NumeroMarca);
            subprograma_Pausa();
            system("cls");
            break;

        case 9: Listado_opcion9(vehiculo,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 10: Listado_opcion10(vehiculo,marca,NumeroVehiculo);
            subprograma_Pausa();
            system("cls");
            break;

        case 0: system("cls");
            break;

        default : printf("ERROR: EL NUMERO INTRODUCIDO NO CORRESPONDE CON NINGUNA OPCION, POR
FAVOR, INGRESE UN NUMERO VALIDO.\n ");

    } // fin del switch

```



```
}while (opcion!=0);  
}  
  
//-----  
/*****/  
/* Subprograma:          Listado_opción1                */  
/* Tipo:                 Procedimiento                  */  
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */  
/*                        vehiculos y marcas y un entero.      */  
/* Parámetros de Salida:   ----                      */  
/* Prerrequisitos:        Debe haber vehiculos y marcas registradas */  
/* Objetivo:              Presenta en pantalla un listado completo de to-*/  
/*                        dos los datos de todos los vehiculos.    */  
/*****/  
void Listado_opcion1(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){  
  
system("cls");  
float descuento,oferta;  
for (int j = 0; j < 5; j++){  
    for(int i = 0; i < NumeroVehiculo; i++){  
        if(j == vehiculo[i].marca){  
            printf("\n\t\t\tCódigo:\t\t\t\t\t%20s",vehiculo[i].codigo);  
            printf("\n\t\t\tMarca:\t\t\t\t\t%20s",marca[vehiculo[i].marca].marca);  
            printf("\n\t\t\tGama:\t\t\t\t\t%20s",vehiculo[i].gama);  
            printf("\n\t\t\tModelo:\t\t\t\t\t%20s",vehiculo[i].modelo);  
  
            printf("\n\t\t\tMotor:\t\t\t\t\t ");  
mostrarMotor(vehiculo,i);  
            printf("\n\t\t\tPotencia(en Kw):\t\t\t%17i kw",vehiculo[i].potencia_KW);  
            printf("\n\t\t\tPotencia(en CV):\t\t\t%17i cv",vehiculo[i].potencia_CV);  
            printf("\n\t\t\tConsumo:\t\t\t\t\t%12.2f L/100km",vehiculo[i].consumo_medio);  
            printf("\n\t\t\tPVP:\t\t\t\t\t%14.2f euros",vehiculo[i].precio);  
descuento=((vehiculo[i].descuento) * (vehiculo[i].precio))/100;  
            printf("\n\t\t\tDescuento(%%%):\t\t\t\t\t%18i %%",vehiculo[i].descuento);  
            printf("\n\t\t\tDescuento Aplicable:\t\t\t%14.2f euros",descuento);  
oferta = (vehiculo[i].precio) - (descuento);  
            printf("\n\t\t\tPrecio en oferta:\t\t\t%14.2f euros",oferta);  
            printf("\n\t\t\tUnidades existentes:\t\t\t%20i",vehiculo[i].unidades);  
printf("\n-----");  
        }  
    }  
}  
}  
  
//-----  
/*****/  
/* Subprograma:          Listado_opción3                */  
/* Tipo:                 Función                          */  
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */  
/*                        vehiculos y marcas, además de un entero      */  
/* Parámetros de Salida:   -                            */  
/* Prerrequisitos:        Debe haber marcas y vehiculos registrados */  
/* Objetivo:              Pide al usuario el codigo de un vehiculo, tras */  
/*                        ello, muestra un listado de todos sus datos.  */  
/*****/  
void Listado_opcion3(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){  
  
char Auxcodigo[15];  
float descuento,oferta;  
int seguir = 1;  
  
printf("\n\t\t\tIntroduzca el código del vehiculo ");  
gets(Auxcodigo);  
fflush(stdin);
```

```

system("cls");

for (int i=0;i< NumeroVehiculo; i++){
    if (strcmp(vehiculo[i].codigo,Auxcodigo)==0){
        printf("\n\t\t\t\t\t DATOS EXISTENTES DEL VEHÍCULO %s",vehiculo[i].codigo);
        printf("\n\t\t\t\t\t-----\n\n");
        printf("\t\t\t\t\t Código:           %s\n",vehiculo[i].codigo);
        printf("\t\t\t\t\t Marca:           %s\n",marca[vehiculo[i].marca].marca);
        printf("\t\t\t\t\t Gama:           %s\n",vehiculo[i].gama);
        printf("\t\t\t\t\t Modelo:          %s\n",vehiculo[i].modelo);
        printf("\t\t\t\t\t Motor:           ");
        mostrarMotor(vehiculo,i);
        printf("\n\t\t\t\t\t Potencia del motor (KW):   %d kw \n",vehiculo[i].potencia_KW);
        printf("\t\t\t\t\t Potencia del motor (CV):    %d cv \n",vehiculo[i].potencia_CV);
        printf("\t\t\t\t\t CONSUMO PROMEDIO (L/100km):  %.2f litros \n",vehiculo[i].consumo_medio);
        printf("\t\t\t\t\t Precio original (PVP):      %.2f euros \n",vehiculo[i].precio);
        descuento=((vehiculo[i].descuento) * (vehiculo[i].precio))/100;
        printf("\t\t\t\t\t Descuento (%%):           %i \n",vehiculo[i].descuento);
        printf("\t\t\t\t\t Descuento aplicable:       %.2f\n",descuento);
        oferta = (vehiculo[i].precio) - (descuento);
        printf("\t\t\t\t\t Precio en oferta:         %.2f euros \n",oferta);
        printf("\t\t\t\t\t Unidades existentes:       %i\n",vehiculo[i].unidades);
        printf("\t\t\t\t\t-----\n");
        subprograma_Pausa();
        seguir = 0;
    }
}

if(seguir == 1){
    printf(" ***EL CÓDIGO NO ESTÁ REGISTRADO EN EL SISTEMA*** \n");
    subprograma_Pausa();
    system("cls");
}

}

//-----
/*****
/* Subprograma:      Listado_opción2          */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  Recibe un array que almacena los datos de los */
/*                     vehiculos y un entero.          */
/* Parámetros de Salida:   ----              */
/* Prerrequisitos:       Debe haber vehiculos registrados          */
/* Objetivo:            Presenta en pantalla la suma de dinero que pro-*/
/*                     duciria la venta de todos los vehículos.    */
*****/

void Listado_opcion7(tRegVehiculos *vehiculo, int NumeroVehiculo){

    float total = 0,descuento = 0,oferta;

    for (int i=0; i<NumeroVehiculo; i++){
        total = total + vehiculo[i].precio * vehiculo[i].unidades;
        descuento = descuento + ((vehiculo[i].descuento * vehiculo[i].precio * vehiculo[i].unidades)/100);
    }
    oferta = total-descuento;
    printf("\n\n\n\n\n\t\t\t\t\t IMPORTE TOTAL A COBRAR (todos los vehículos) \n");
    printf("\t\t\t\t\t -----\n");
    printf("\t\t\t\t\t el precio total a cobrar es: %.2f euros \n",oferta);
}

//-----
/*****
/* Subprograma:      Listado_opción2          */
/* Tipo:             Procedimiento            */
/* Parámetros de entrada:  Recibe 2 arrays que almacenan los datos de los */

```

```

/* vehiculos y marcas,además de un entero. */
/* Parámetros de Salida: ---- */
/* Prerrequisitos: Debe haber vehiculos y marcas registradas */
/* Objetivo: Presenta en pantalla un listado resumido de los */
/* datos de todos los vehiculos. */
/*****/

void Listado_opcion2(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){

    system("cls");
    float precio;
    printf("\n Código| Marca Gama Modelo Motor PotenciaKw PotenciaCV Consumo
PVP Descuento(%%) Cantidad Precio");
    printf("\n
-----");
    for(int j = 0; j < 5; j++)
    {
        for(int i = 0; i < NumeroVehiculo; i++){
            if(j == vehiculo[i].marca){
                printf("\n%15s|",vehiculo[i].codigo);
                printf("%10s ",marca[vehiculo[i].marca].marca);
                printf("%20s ",vehiculo[i].gama);
                printf("%20s ",vehiculo[i].modelo);
                mostrarMotor(vehiculo,i);
                printf("%12i",vehiculo[i].potencia_KW);
                printf("%12i",vehiculo[i].potencia_CV);
                printf("%9.2f ",vehiculo[i].consumo_medio);
                printf("%8.2f ",vehiculo[i].precio);
                printf("%6i",vehiculo[i].descuento);
                printf("%9i ",vehiculo[i].unidades);
                precio = vehiculo[i].precio - vehiculo[i].precio*vehiculo[i].descuento/100;
                printf("%8.2f",precio);
            }
        }
    }
}

//-----
/*****/
/* Subprograma: Listado_opción4 */
/* Tipo: Procedimiento */
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */
/* vehiculos y marcas,además de un entero. */
/* Parámetros de Salida: ---- */
/* Prerrequisitos: Debe haber vehiculos y marcas registradas */
/* Objetivo: Pide al usuario la marca que quiere ver, tras */
/* ello,muestra un listado resumido de los vehicu-*/
/* los asociados a esta marca. */
/*****/

void Listado_opcion4(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){

    int maarca;
    float precio;
    int aux = 0;

    printf("\n\t\t\t\tElija una marca: ");
    for(int i = 0; i < 5; i++){
        if(marca[i].existencias != -1)
            printf(" %i.- %s |",i,marca[i].marca);
    }
    printf("\n\n\t\t\t\t");
    fflush(stdin);
    do{

```

```

scanf("%i",&maarca);
if((maarca > 4)|| (maarca < 0))
{
    printf("ERROR: MARCA NO EXISTENTE");
    maarca = -1;
}else if(marca[maarca].existencias == -1){
    printf("ERROR: MARCA NO EXISTENTE");
    maarca = -1;
}
}while(maarca < 0);

for(int i = 0;i < NumeroVehiculo; i++)
{
    if(vehiculo[i].marca == maarca)
        aux = 1;
}
if(aux == 1){
    system("cls");
    printf("\n      Código|   Marca      Gama      Modelo      Motor PotenciaKw PotenciaCV Consumo
PVP Descuento(%%) Cantidad Precio");
    printf("\n
-----");

```

```

for(int i = 0; i < NumeroVehiculo; i++)
{
    if(marca[maarca].NumeroMarca == vehiculo[i].marca)
    {
        printf("\n%15s|",vehiculo[i].codigo);
        printf("%10s ",marca[vehiculo[i].marca].marca);
        printf("%20s ",vehiculo[i].gama);
        printf("%20s ",vehiculo[i].modelo);
        mostrarMotor(vehiculo,i);
        printf("%12i",vehiculo[i].potencia_KW);
        printf("%12i",vehiculo[i].potencia_CV);
        printf("%9.2f  ",vehiculo[i].consumo_medio);
        printf("%8.2f  ",vehiculo[i].precio);
        printf("%6i",vehiculo[i].descuento);
        printf("%9i  ",vehiculo[i].unidades);
        precio = vehiculo[i].precio - vehiculo[i].precio*vehiculo[i].descuento/100;
        printf("%8.2f",precio);
    }
}
}else{
    printf("\n\t\t\t\tNingún vehiculo ha sido registrado bajo esa marca");
}
}
//-----
/*****
/* Subprograma:      Listado_opción9      */
/* Tipo:            Procedimiento          */
/* Parámetros de entrada:  Recibe un array que almacena los datos de todos*/
/*                  los vehiculos, además de un entero.      */
/* Parámetros de Salida:   ----            */
/* Prerrequisitos:        Debe haber vehiculos registrados    */
/* Objetivo:             Presenta en pantalla un la media aritmética de */
/*                  la potencia en KW, precio y descuento.    */
/*****/

```

```

void Listado_opcion9(tRegVehiculos *vehiculo, int NumeroVehiculo){

```

```

    system("cls");
    float precio = 0,potencia = 0,descuentos = 0;

```

```
printf("\n\t\t\t\t\tVALORES MEDIOS");
printf("\n\t\t\t\t\t-----");

for(int i = 0; i < NumeroVehiculo; i++)
{
    precio = precio + vehiculo[i].precio;
    potencia = potencia + vehiculo[i].potencia_KW;
    descuentos = descuentos + vehiculo[i].descuento;
}
precio = precio / NumeroVehiculo;
descuentos = descuentos / NumeroVehiculo;
potencia = potencia / NumeroVehiculo;

printf("\n\n\t\t\t\t\tPotencia Media   => %.2f Kw",potencia);
printf("\n\t\t\t\t\tDescuento Medio   =>  %.2f %%",descuentos);
printf("\n\t\t\t\t\tPrecio Medio Oferta=> %.2f eur \n\n",precio);
}
//-----
/*****
/* Subprograma:          Listado_opción5                */
/* Tipo:                 Procedimiento                  */
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */
/*                        vehiculos y marcas,además de un entero.        */
/* Parámetros de Salida:  ----                          */
/* Prerrequisitos:       Debe haber vehiculos y marcas registradas      */
/* Objetivo:             Pide al usuario la potencia máxima y mínima del*/
/*                       rango y muestra los datos resumidos de los      */
/*                       vehiculos que entran en el rango                */
*****/

void Listado_opcion5(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){

int min,max;
float precio;
printf("\n\t\t\t\t\tDATOS DE VEHICULO EN RANGO DE POTENCIAS\n");
printf("\t\t\t\t\t-----\n");
do{
    printf("\n\t\t\t\t\tPotencia mínima: ");
    fflush(stdin);
    scanf("%i",&min);
    if(min <= 0)
        printf("\n\t\t\t\t\tVALOR INCORRECTO");
}while(min <= 0);
do{
    printf("\n\t\t\t\t\tPotencia máxima: ");
    fflush(stdin);
    scanf("%i",&max);
    if(max < min)
        printf("\n\t\t\t\t\tVALOR INCORRECTO");
}while(max < min);

system("cls");
printf("\n\t\t\t\t\tDATOS DE VEHICULOS(en rango de potencia de %i a %i Kw)\n",min,max);
printf("\n\t\t\t\t\tCódigo|  Marca\t\t\t\t\tGama\t\t\t\t\tModelo\t\t\t\t\tMotor\t\t\t\t\tPotenciaKw\t\t\t\t\tPotenciaCV\t\t\t\t\tConsumo\t\t\t\t\tPVP\t\t\t\t\tDescuento(%%)\t\t\t\t\tCantidad\t\t\t\t\tPrecio");
printf("\n\t\t\t\t\t-----");
printf("\n\t\t\t\t\t-----");

for(int i = 0; i < NumeroVehiculo; i++){
    if((vehiculo[i].potencia_KW <= max)&(vehiculo[i].potencia_KW >= min))
    {
        printf("\n%15s|",vehiculo[i].codigo);
        printf("%10s ",marca[vehiculo[i].marca].marca);
```

```

        printf("%20s ",vehiculo[i].gama);
        printf("%20s ",vehiculo[i].modelo);
        mostrarMotor(vehiculo,i);
        printf("%12i",vehiculo[i].potencia_KW);
        printf("%12i",vehiculo[i].potencia_CV);
        printf("%9.2f ",vehiculo[i].consumo_medio);
        printf("%8.2f ",vehiculo[i].precio);
        printf("%6i",vehiculo[i].descuento);
        printf("%9i ",vehiculo[i].unidades);
        precio = vehiculo[i].precio - vehiculo[i].precio*vehiculo[i].descuento/100;
        printf("%8.2f",precio);
    }
}
}
//-----
/*****
/* Subprograma:      Listado_opción6          */
/* Tipo:            Procedimiento              */
/* Parámetros de entrada:  Recibe 2 arrays que almacenan los datos de los */
/*                    vehiculos y marcas,además de un entero.          */
/* Parámetros de Salida:  ----              */
/* Prerrequisitos:      Debe haber vehiculos y marcas registradas      */
/* Objetivo:            Presenta en pantalla un listado resumido de los*/
/*                    datos de todos los vehiculos, ordenados por su */
/*                    precio en oferta.              */
*****/
void Listado_opcion6(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){

```

```

    float precios[NumeroVehiculo];
    float temp=0;

    for(int i = 0; i < NumeroVehiculo; i++){
        precios[i]=vehiculo[i].precio - vehiculo[i].precio*vehiculo[i].descuento/100;
    }

    for (int i = 1;i< NumeroVehiculo;i++)
    {
        for (int j = 0; j < (NumeroVehiculo)-i ;j++)
        {
            if (precios[j] > precios[j+1])
            {
                temp=precios[j];
                precios[j]=precios[j+1];
                precios[j+1]=temp;
            }
        }
    }

    system("cls");
    printf("\n      Código|  Marca          Gama      Modelo      Motor  PotenciaKw  PotenciaCV  Consumo
PVP Descuento(%%)  Cantidad Precio");
    printf("\n
-----");

```

```

    int j = 0;
    for(int i = 0; i < NumeroVehiculo; i++)
    {
        temp = vehiculo[i].precio - vehiculo[i].precio*vehiculo[i].descuento/100;
        if((temp == precios[j])&&(vehiculo[i].unidades > 0)){
            printf("\n%15s",vehiculo[i].codigo);
            printf("%10s ",marca[vehiculo[i].marca].marca);
            printf("%20s ",vehiculo[i].gama);
            printf("%20s ",vehiculo[i].modelo);

```

```

        mostrarMotor(vehiculo[i]);
        printf("%12i",vehiculo[i].potencia_KW);
        printf("%12i",vehiculo[i].potencia_CV);
        printf("%9.2f  ",vehiculo[i].consumo_medio);
        printf("%8.2f  ",vehiculo[i].precio);
        printf("%6i",vehiculo[i].descuento);
        printf("%9i  ",vehiculo[i].unidades);
        printf("%8.2f",precios[j]);
        i=0;
        j++;
    }
}
}
//-----
/*****
/* Subprograma:          Listado_opción8          */
/* Tipo:                Procedimiento              */
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */
/*                      vehiculos y marcas,además de dos enteros.      */
/* Parámetros de Salida:  ----                      */
/* Prerrequisitos:       Debe haber vehiculos y marcas registradas    */
/* Objetivo:             Presenta en pantalla la suma de dinero que pro-*/
/*                      duciria la venta de todos los vehículos separa-*/
/*                      dos por marcas.                      */
*****/
void Listado_opcion8(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo, int NumeroMarca){

    printf("\n\t\t\t\t\tIMPORTE POR MARCAS");
    printf("\n\t\t\t\t\t-----");

    float importe[5] = {0,0,0,0,0};

    for(int i = 0; i < NumeroMarca; i++)
    {
        importe[vehiculo[i].marca] = importe[vehiculo[i].marca] + (vehiculo[i].precio -
(vehiculo[i].precio*vehiculo[i].descuento)/100)*vehiculo[i].unidades;
    }

    for(int i = 0; i < NumeroMarca ; i++)
    {
        printf("\n\n\t\t\t\t\t%s => %.2f eur",marca[i].marca,importe[i]);
    }
}
//-----
/*****
/* Subprograma:          Listado_opción10          */
/* Tipo:                Procedimiento              */
/* Parámetros de entrada: Recibe 2 arrays que almacenan los datos de los */
/*                      vehiculos y marcas, asdemás de un entero          */
/* Parámetros de Salida:  ----                      */
/* Prerrequisitos:       Debe haber marcas y vehiculos registrados    */
/* Objetivo:             Presenta en pantalla un listado resumido de los*/
/*                      vehiculos y pide al usuario que elija 2 distin-*/
/*                      tos,de los que mostrara una tabla comparandolos*/
*****/
void Listado_opcion10(tRegVehiculos *vehiculo,tRegMarcas *marca, int NumeroVehiculo){

    printf("\n\t\t\t\t\tLISTADO RESUMIDO");
    printf("\n\n\t\tVehiculo Marca          Gama          Modelo");
    printf("\n\t\t-----");

    int opcion1, opcion2;

```

[illegible]



# PRUEBAS DE EJCUCIÓN

-----  
\*\*\*\*\*BIENVENIDO A GESTIAUTO\*\*\*\*\*

-----  
Pulse <Intro> para continuar...

## MENU PRINCIPAL

- 
- 1.- MARCAS
  - 2.- VEHICULOS
  - 3.- INFORMES
  - 0.- SALIR

ELIJA OPCIÓN:

## MENU MARCAS

- 
- 1.- Alta
  - 2.- Baja
  - 3.- Actualizar nombre de la marca
  - 4.- Listado de marcas
  - 0.- Volver al Menu Principal.

Elija una opción:

SERVICIO DE ALTA DE NUEVA MARCA

-----  
Por favor, introduce la marca de vehículos que deseas introducir en el sistema: Volkswagen

Marca Volkswagen registrada

Pulse <Intro> para continuar...

SERVICIO DE ALTA DE NUEVA MARCA

-----  
Por favor, introduce la marca de vehículos que deseas introducir en el sistema: BMW

ERROR LA MARCA YA ESTA REGISTRADA

Pulse <Intro> para continuar...

SERVICIO DE BAJA DE UNA MARCA

-----  
Por favor, introduce la marca de vehículos que deseas dar de baja en el sistema: BMW

\*\*\* ERROR, AÚN QUEDAN EXISTENCIAS EN EL ALMACÉN \*\*\*

Pulse <Intro> para continuar...

SERVICIO DE BAJA DE UNA MARCA

-----  
Por favor, introduce la marca de vehículos que deseas dar de baja en el sistema: Volkswagen

\*\*\*MARCA ENCONTRADA\*\*\*, ¿ Desea eliminarla ? (S/N)S

\*\*\* MARCA ELIMINADA\*\*\*

Pulse <Intro> para continuar...

SERVICIO DE MODIFICACIÓN DE MARCA

-----  
Por favor, introduce la marca de vehículos que deseas modificar: Jeep

Introduce un nuevo nombre para la marca Jeep: JEEP

\*\*\*NOMBRE CAMBIADO CORRECTAMENTE\*\*\*

Pulse <Intro> para continuar...

## LISTADO GENERAL DE MARCAS

-----

la marca numero 0 es BMW  
la marca numero 1 es JEEP  
la marca numero 2 es Audi

Pulse <Intro> para continuar...

## MENU VEHÍCULOS

-----

- 1.- Alta
  - 2.- Baja
  - 3.- Modificación
  - 0.- Volver al Menú Principal.
- Elija una opción:

## SERVICIO DE ALTA DE UN VEHÍCULO

-----

Introduzca los siguientes datos del vehículo :

Marca:0 .- BMW 1 .- JEEP 2 .- Audi 2

GAMA: Sedan

MODELO: TFSI

POTENCIA DEL MOTOR (Kw): 221

COMBUSTIBLE: (1.-Diésel, 2.-Gasolina, 3.-Híbrido, 4.-Eléctrico) 2

POTENCIA DEL MOTOR (CV): 300

CONSUMO MEDIO DEL VEHÍCULO (L/100): 8.1

PRECIO ORIGINAL (PVP): 54050

DESCUENTO A APLICAR ( SOBRE PVP): 12

Nº UNIDADES EXISTENTES: 0

¿\*\*\* LOS DATOS SON CORRECTOS (S/N) \*\*\*? S

¿ Desea dar de alta un nuevo vehiculo (S/N) ? N

Pulse <Intro> para continuar...

#### BAJA DE UN VEHICULO

---

Introduzca el código del vehiculo a dar de baja BMWX1SDR110

Vehículo encontrado ,¿Desea darlo de baja con todos sus datos ?(S/N) S

\*\*\*NO SE PUEDE ELIMINAR , AÚN QUEDAN EXISTENCIAS EN EL ALMACÉN\*\*\*

Pulse <Intro> para continuar...

#### BAJA DE UN VEHICULO

---

Introduzca el código del vehiculo a dar de baja AUDSETFS221

Vehículo encontrado ,¿Desea darlo de baja con todos sus datos ?(S/N) S

\*\*\* VEHÍCULO ELIMINADO CORRECTAMENTE\*\*\*

Pulse <Intro> para continuar...

#### MODIFICACIÓN DE UN VEHICULO

---

Introduzca el código del vehiculo para modificar: BMWX1SDR110

## DATOS ACTUALES DEL VEHÍCULO

-----

Código:	BMW X1 SDR110
Marca:	BMW
Gama:	X1
Modelo:	sDrive18d
Motor:	Diésel
Potencia del motor (KW):	110 KW
Potencia del motor (CV):	150 CV
CONSUMO PROMEDIO (L/100km):	5.50 litros
Precio original (PVP):	36700.00 euros
Descuento (sobre PVP):	10
Descuento aplicable:	3670.00
Precio en oferta:	33030.00 euros
Unidades existentes:	3

-----

¿Qué dato desea modificar?

- 1.- Marca
- 2.- Gama
- 3.- Modelo
- 4.- Potencia (Kw)
- 5.- Potencia (CV)
- 6.- Consumo Promedio
- 7.- Precio original
- 8.- Descuento
- 9.- Unidades existentes
- 10.- Motor
- 0.- Salir de Modificación

Nuevo Descuento: 11

-----  
\*\*\* OK. DATOS GUARDADOS \*\*\*  
-----

Elije una opción:

Pulse <Intro> para continuar...

## DATOS ACTUALES DEL VEHÍCULO

Código:	BMW15DR110
Marca:	BMW
Gama:	X1
Modelo:	sDrive18d
Motor:	Diésel
Potencia del motor (KW):	110 KW
Potencia del motor (CV):	150 CV
CONSUMO PROMEDIO (L/100km):	5.50 litros
Precio original (PVP):	36700.00 euros
Descuento (sobre PVP):	11
Descuento aplicable:	4037.00
Precio en oferta:	32663.00 euros
Unidades existentes:	3

¿Qué dato desea modificar?

- 1.- Marca
- 2.- Gama
- 3.- Modelo
- 4.- Potencia (Kw)
- 5.- Potencia (CV)
- 6.- Consumo Promedio
- 7.- Precio original
- 8.- Descuento
- 9.- Unidades existentes
- 10.- Motor
- 0.- Salir de Modificación

Elije una opción:

## MENU LISTADOS

- 1.- Listado General de vehículos(completo)
- 2.- Listado General de vehículos(resumido)
- 3.- Listado de datos de un vehículo
- 4.- Listado de vehículos (de una sola marca)
- 5.- Listado de vehículos (ordenado por rango de potencias -Kw)
- 6.- Listado de vehículos (ordenado por precio en oferta)
- 7.- Importe total a cobrar(Todos los vehículos)
- 8.- Importe total a cobrar (por marca)
- 9.- Valores Medios (potencia,descuento,precio)
- 10.- Comparar dos vehículos
- 0.- Volver al Menu Principal.

Elija una opción:

Código: BMWX1SDR110  
Marca: BMW  
Gama: X1  
Modelo: sDrive18d  
Motor: Diésel  
Potencia(en Kw): 110 kw  
Potencia(en CV): 150 cv  
Consumo: 5.50 L/100km  
PVP: 36700.00 euros  
Descuento(%): 11 %  
Descuento Aplicable: 4037.00 euros  
Precio en oferta: 32663.00 euros  
Unidades existentes: 3

Código: BMWSE630195  
Marca: BMW  
Gama: Serie 6-Gran Turismo  
Modelo: 630d  
Motor: Diésel  
Potencia(en Kw): 195 kw  
Potencia(en CV): 265 cv  
Consumo: 6.80 L/100km  
PVP: 79569.00 euros  
Descuento(%): 12 %  
Descuento Aplicable: 9548.28 euros  
Precio en oferta: 70020.72 euros  
Unidades existentes: 2

Código: JEEWRSP0203  
Marca: Jeep  
Gama: Wrangler  
Modelo: Sport 2.0  
Motor: Gasolina  
Potencia(en Kw): 203 kw  
Potencia(en CV): 270 cv  
Consumo: 11.25 L/100km  
PVP: 42562.00 euros  
Descuento(%): 15 %  
Descuento Aplicable: 6384.30 euros  
Precio en oferta: 36177.70 euros  
Unidades existentes: 1

Pulse <Intro> para continuar...

Código	Marca	Gama	Modelo	Motor	PotenciaKw	PotenciaCV	Consumo	PVP	Descuento(%)	Cantidad	Precio
BMWX1SDR110	BMW	X1	sDrive18d	Diésel	110	150	5.50	36700.00	11	3	32663.00
BMWSE630195	BMW	Serie 6-Gran Turismo	630d	Diésel	195	265	6.80	79569.00	12	2	70020.72
JEEWRSP0203	Jeep	Wrangler	Sport 2.0	Gasolina	203	270	11.25	42562.00	15	1	36177.70

Pulse <Intro> para continuar...

Introduzca el código del vehículo BMWX1SDR110



## DATOS EXISTENTES DEL VEHÍCULO BMWX1SDR110

-----

Código: BMWX1SDR110  
 Marca: BMW  
 Gama: X1  
 Modelo: sDrive18d  
 Motor: Diésel  
 Potencia del motor (KW): 110 kw  
 Potencia del motor (CV): 150 cv  
 CONSUMO PROMEDIO (L/100km): 5.50 litros  
 Precio original (PVP): 36700.00 euros  
 Descuento (%): 11  
 Descuento aplicable: 4037.00  
 Precio en oferta: 32663.00 euros  
 Unidades existentes: 3

-----

Pulse <Intro> para continuar...

Elija una marca: 0 .- BMW 1 .- Jeep 2 .- Audi

0

Código	Marca	Gama	Modelo	Motor	PotenciaKw	PotenciaCV	Consumo	PVP	Descuento(%)	Cantidad	Precio
BMWX1SDR110	BMW	X1	sDrive18d	Diésel	110	150	5.50	36700.00	11	3	32663.00
BMWSE630195	BMW Serie 6-Gran Turismo		630d	Diésel	195	265	6.80	79569.00	12	2	70020.72

Pulse <Intro> para continuar...

## DATOS DE VEHICULO EN RANGO DE POTENCIAS

Potencia mínima: 110

Potencia máxima: 109

VALOR INCORRECTO

Potencia máxima: 220

## DATOS DE VEHICULOS(en rango de potencia de 110 a 220 Kw)

Código	Marca	Gama	Modelo	Motor	PotenciaKw	PotenciaCV	Consumo	PVP	Descuento(%)	Cantidad	Precio
BMWX1SDR110	BMW	X1	sDrive18d	Diésel	110	150	5.50	36700.00	11	3	32663.00
BMWSE630195	BMW Serie 6-Gran Turismo		630d	Diésel	195	265	6.80	79569.00	12	2	70020.72
JEENRSP0203	Jeep	Wrangler	Sport 2.0	Gasolina	203	270	11.25	42562.00	15	1	36177.70

Pulse <Intro> para continuar...

Código	Marca	Gama	Modelo	Motor	PotenciaKw	PotenciaCV	Consumo	PVP	Descuento(%)	Cantidad	Precio
BMWX1SDR110	BMW	X1	sDrive18d	Diésel	110	150	5.50	36700.00	11	3	32663.00
JEENRSP0203	Jeep	Wrangler	Sport 2.0	Gasolina	203	270	11.25	42562.00	15	1	36177.70
BMWSE630195	BMW Serie 6-Gran Turismo		630d	Diésel	195	265	6.80	79569.00	12	2	70020.72

Pulse <Intro> para continuar...



IMPORTE TOTAL A COBRAR (todos los vehículos)

-----  
el precio total a cobrar es: 274208.13 euros

Pulse <Intro> para continuar...

IMPORTE POR MARCAS

VALORES MEDIOS

-----

BMW => 238030.44 eur

Jeep => 36177.70 eur

Audi => 0.00 eur

Potencia Media => 169.33 Kw  
Descuento Medio => 12.67 %  
Precio Medio Oferta => 52943.67 eur

Pulse <Intro> para continuar...

Pulse <Intro> para continuar...

#### LISTADO RESUMIDO

Vehiculo	Marca	Gama	Modelo
1	BMW	X1	sDrive18d
2	BMW	Serie 6-Gran Turismo	630d
3	Jeep	Wrangler	Sport 2.0

Elija el primer vehiculo a comparar: 1

Elija el segundo vehiculo a comparar: 1

ERROR: OPCION INCORRECTA

Elija el segundo vehiculo a comparar: 2

#### TABLA COMPARATIVA

	BMW X1 SDR110	BMW Serie 6-Gran Turismo 630d	
Marca	BMW	BMW	
Gama	X1	Serie 6-Gran Turismo	
Modelo	sDrive18d	630d	
Motor	1	1	Diferencias encontradas
Potencia motor(Kw)	110	195	-85 Kw
Potencia motor(CV)	150	265	-115 CV
Consumo promedio(100 km)	5.50	6.80	-1.30 litros
PVP(euros)	36700.00	79569.00	-42869.00 eur
PVP oferta(euros)	33030.00	70020.72	-36990.72 eur
Descuento(%)	10	12	-2 %
Unidades en el almacén	3	2	

Pulse <Intro> para continuar...

GRACIAS POR UTILIZAR NUESTRO PROGRAMA