



MANUAL DEL PROGRAMADOR

ADOPTA

Programacion I

VELAZQUEZ HERNANDEZ/ LOPEZ LARA
179798/182239

INDICE

CONTENIDOS

| | |
|--|----|
| Introducción..... | 2 |
| Requisitos de instalación. | 2 |
| Bibliotecas y funciones declaradas | 2 |
| Gotoxy y Struct | 3 |
| Funcion Int Main | 4 |
| Funciones principales | 6 |
| Crear registro | 6 |
| Consulta Registro | 9 |
| • Imprimir por edad | 9 |
| • Imprimir por nombre..... | 10 |
| • Imprimir por esterilizado | 11 |
| • Imprimir por adoptado..... | 12 |
| • Imprimir por talla | 13 |
| Funcion Adoptar..... | 14 |
| Funcion Actualizar Registro | 18 |
| Funciones Validar Enteros y Validar Caracteres | 21 |
| Ficha de Adopcion..... | 24 |

Introducción

La aplicación ADOPTA es un programa diseñado con el propósito de facilitar la búsqueda de hogares para perros que se encuentran en situación de adopción. Esta aplicación cuenta con una interfaz intuitiva y fácil de usar que permite a los usuarios explorar perros disponibles y enviar solicitudes de adopción.

Requisitos de instalación.

- Compilador de C++
- Sistema operativo compatible
- Espacio en Disco Suficiente

Bibliotecas y funciones declaradas

```
#include<stdio.h>
#include<stdlib.h>
#include<windows.h>
#include<string.h>
#include<ctype.h>
#include <unistd.h>
#define BAR_SIZE 50
//Rafael Lopez Lara 182239
//Eric Leonardo Velazquez Hernandez 179798
//Proyecto Programacion I
void portada(void); //prototipo portada;
void bienvenida(void);
void menu();
int crear_Registro(struct datos_Registro *datos, int n);
void imprimir_por_edad(struct datos_Registro *datos, int *n);
void imprimir_por_nombre(struct datos_Registro *datos, int n);
void imprimir_por_esterilizado(struct datos_Registro *datos, int n);
void imprimir_por_adoptado(struct datos_Registro *datos, int n);
void imprimir_por_talla(struct datos_Registro *datos, int n, int opcTalla);
void adoptar(struct datos_Registro *datos, int n);
void actualiza_Registro(struct datos_Registro *datos, int n);
bool validarEnteros(char c[]);
bool validarCaracteres(char c[]);
void ficha_Adopcion(struct datos_Registro *datos, int id_Adopcion);
```

El código utiliza las bibliotecas estándar de C, como `stdio.h`, `stdlib.h`, `windows.h`, `string.h`, `ctype.h` y `unistd.h`. Hay una constante definida llamada `BAR_SIZE` que tiene un valor de 50. Se definen varias funciones, incluyendo `portada`, `bienvenida`, `menu`, `crear_Registro`, `imprimir_por_edad`, `imprimir_por_nombre`, `imprimir_por_esterilizado`, `imprimir_por_adoptado`, `imprimir_por_talla`, `adoptar`, `actualiza_Registro`, `validarEnteros`, `validarCaracteres` y `ficha_Adopcion`. La función `portada` muestra una portada del programa. La función `bienvenida` muestra un mensaje de bienvenida al usuario. La función `menu` muestra un menú de opciones para que el usuario interactúe con el programa. La función `crear_Registro` permite al usuario crear un registro de adopción. Las funciones `imprimir_por_edad`, `imprimir_por_nombre`, `imprimir_por_esterilizado` e `imprimir_por_adoptado` imprimen los registros de adopción según diferentes criterios (edad, nombre, esterilizado y adoptado). La función `imprimir_por_talla` imprime los registros de adopción según la talla del animal. La función `adoptar` permite al usuario marcar un registro de adopción como adoptado. La función `actualiza_Registro` permite al usuario actualizar la información de un registro de adopción. Las funciones `validarEnteros` y `validarCaracteres` validan si una cadena de caracteres contiene solo enteros o solo caracteres alfabéticos, respectivamente. La función `ficha_Adopcion` muestra la ficha de adopción de un animal en particular.

Gotoxy y Struct

```
// Función gotoxy
void gotoxy(int x,int y){
    COORD pos = {x, y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
    HANDLE hcon;
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x;
    dwPos.Y= y;
    SetConsoleCursorPosition(hcon,dwPos);
}

typedef struct datos_Registro{
    int id;
    char nombre[10];
    int edad;
    int talla;
    int dia;
    int mes;
    int anio;
    int esterilizado;
    int adoptado;
}datos;
```

La función gotoxy se utiliza para posicionar el cursor en una ubicación específica en la consola de Windows. Toma dos argumentos enteros, x e y, que representan las coordenadas de la posición a la que se desea mover el cursor. Utiliza funciones de la biblioteca windows.h para realizar esta operación, como SetConsoleCursorPosition y GetStdHandle.

La estructura datos_Registro se define utilizando la palabra clave typedef. Esta estructura tiene varios campos que representan diferentes atributos de un registro en el programa de adopción de animales. Los campos incluyen:

- id: un entero que representa el identificador del registro.
- nombre: una cadena de caracteres de longitud 10 que almacena el nombre del animal.
- edad: un entero que representa la edad del animal.
- talla: un entero que indica la talla del animal.
- dia, mes y anio: enteros que almacenan la fecha de adopción.
- esterilizado: un entero que indica si el animal está esterilizado o no.
- adoptado: un entero que indica si el animal ha sido adoptado o no.

Funcion Int Main

```
int main () { //Funcion Main

    datos_Registro datos[100];
    int numero_Registros = 0;
    int seleccion;
    bool salida;
    char sec[50];
    char dato[50];
    int n=0;
    int id_actual = 1;
    bool veri_Enterros;
    portada();
    printf("\n\n\n\n\n\n\n");
    system("pause");
    system("cls");
    bienvenida();
    system("pause");
    system("cls");
    do{
        int opc;
        salida=false;

        menu();
        do{
            veri_Enterros=false;
            fflush(stdin);
            scanf("%s", sec);
            veri_Enterros = validarEnterros(sec);
        }while(veri_Enterros==true);
        seleccion = atoi(sec);
        switch (seleccion){
            case 1:
                system("cls");
                n+= crear_Registro(datos, n);
                break;

            case 2:
                do {
                    system("cls");
                    printf("Seleccione una opcion:\n1.-Buscar por edad\n2.-Buscar por nombre\n3.-Buscar por esterilizacion\n4.-Buscar por adopcion\n5.-Regresar al menu principal\n");
                    do{
                        fflush(stdin);
                        scanf("%s", &dato);
                    }while(validarEnterros(dato) == true);
                    opc = atoi(dato);
                    switch (opc) {
                        case 1:
                            imprimir_por_edad(datos, &n);
                            system("pause");
                            break;
                        case 2:
                            imprimir_por_nombre(datos, n);
                            system("pause");
                            break;
                        case 3:
                            imprimir_por_esterilizado(datos, n);
                            system("pause");
                            break;
                        case 4:
                            imprimir_por_adoptado(datos, n);
                            system("pause");
                            break;
                        case 5:
                            break;
                        default:
                            printf("Opcion invalida. Por favor ingrese una opcion valida.\n");
                    }
                } while (opc != 5);
                break;
            case 3:
                system("cls");
                break;
        }
    } while (salida == false);
}
```

```

        actualiza_Registro(datos, n);
        break;
    case 4:
        adoptar(datos, n);
        break;
    case 5:
        system("cls");
        salida=true;
        break;
    default:
        printf("Opcion no valida ingrese su seleccion nuevamente\n");
    }
    system("cls");
}while(salida==false);
printf("Tenga un buen d%ca\n", 161);
return 0;
}

```

El código declara un arreglo de estructuras llamado "datos" para almacenar registros de adopción de animales. Utiliza variables para realizar un seguimiento del número de registros y otras variables para controlar la entrada del usuario. El programa muestra una portada y un mensaje de bienvenida, luego entra en un bucle principal donde se muestra un menú y se ejecutan diferentes acciones según la selección del usuario. Estas acciones incluyen crear nuevos registros, imprimir registros según diferentes criterios, actualizar registros y marcar registros como adoptados. El programa continúa en el bucle principal hasta que el usuario elija salir. Después, se muestra un mensaje de despedida y el programa termina.

Funciones principales

Crear registro

```
int crear_Registro(struct datos_Registro *datos, int n){
    int val=0, num;
    char dato[50];
    bool ver;
    printf("Ingrese la cantidad de perritos a registrar\n");
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    num = atoi(dato);
    val = n + num;
    for(int i=n; i<val; i++){
        datos[i].id = i+1;
        printf("Ingrese el nombre del perrito %d\n", i+1);
        int solo_letras;
        do {
            solo_letras = 1;
            scanf("%s", dato);
            for (int j = 0; dato[j] != '\0'; j++) {
                if (!isalpha(dato[j])) {
                    solo_letras = 0;
                    break;
                }
            }
            if (!solo_letras) {
                printf("El nombre solo debe de contener letras (sin acentos). Intentalo de nuevo.\n");
            }
            if(strlen(dato)>10){
                printf("El nombre no debe de contener mas de 10 caracteres\n");
                solo_letras = 0;
            }
        } while (!solo_letras);
        strcpy(datos[i].nombre, dato);
        printf("Ingrese la edad del perrito: "):
```

```

do{
    fflush(stdin);
    scanf("%s", &dato);
}while(validarEnteros(dato) == true);
datos[i].edad = atoi(dato);
printf("Ingrese la talla del perrito:  \n1.-Grande\n2.-Mediano\n3.-Chico\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[i].talla = atoi(dato);
    if(datos[i].talla != 1 && datos[i].talla !=2 && datos[i].talla !=3){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}while(ver==true);
printf("Ingrese la Fecha de registro (dd mm aaaa): \n");

do{
    ver=false;
    do{
        printf("D%ca: ", 161);
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[i].dia = atoi(dato);
    if(datos[i].dia<1 || datos[i].dia>31){
        printf("Ingrese un d%ca valido\n", 161);
        ver=true;
    }
}while(ver==true);
do{
    ver=false:

```



```

do{
    printf("\nMes: ");
    fflush(stdin);
    scanf("%s", &dato);
}while(validarEnteros(dato) == true);
datos[i].mes = atoi(dato);
if(datos[i].mes<1 || datos[i].mes>12){
    printf("Ingrese un mes valido\n");
    ver=true;
}
}while(ver==true);
do{
    ver=false;
    do{
        printf("\nA%co: ", 164);
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[i].anio = atoi(dato);
    if(datos[i].anio<1990 || datos[i].anio>2023){
        printf("Ingrese un a%co valido\n", 164);
        ver=true;
    }
}while(ver==true);
printf("Ingrese si el perrito esta esterilizado \n1.-Esta esterilizado\n2.-No esta esterilizado\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[i].esterilizado = atoi(dato);
    if(datos[i].esterilizado != 1 && datos[i].esterilizado !=2){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}

```

```

        ver=true;
    }
}while(ver==true);
printf("Ingrese si el perrito esta adoptado \n1.-Esta adoptado\n2.-No esta adoptado\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[i].adoptado = atoi(dato);
    if(datos[i].adoptado != 1 && datos[i].adoptado !=2){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}
}while(ver==true);
}
return num;
}

```

La función crear_Registro permite al usuario ingresar información para crear nuevos registros de perritos en el programa. Se solicita al usuario la cantidad de perritos a registrar y se validan las entradas para asegurarse de que sean números enteros. Luego, se utiliza un bucle para ingresar los datos de cada perrito, incluyendo su nombre, edad, talla, fecha de registro, esterilización y adopción. Se realizan validaciones adicionales en cada entrada para garantizar que cumplan con los criterios establecidos. Finalmente, la función devuelve la cantidad de perritos registrados.

Consulta Registro

- Imprimir por edad

```
void imprimir_por_edad(struct datos_Registro *datos, int *n) {
    int age, val1;
    bool ver2;
    char dato[50], nombreP[11];
    val1= *n;
    printf("Ingrese la edad del perrito a buscar: ");
    do{
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    age = atoi(dato);
    ver2=false;
    for (int i = 0; i < val1; i++) {
        char nombre[10];
        if ( datos[i].edad == age ){
            printf("ID: %d\n", datos[i].id);
            fflush(stdin);
            strcpy(nombreP, datos[i].nombre);
            printf("Nombre: %s\n", nombreP);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if(datos[i].talla == 1){
                printf("Grande\n");
            }
            if(datos[i].talla == 2){
                printf("Mediano\n");
            }
            if(datos[i].talla == 3){
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            printf("\n\n");
            ver2=true;
        }
    }
    if(ver2==false){
        printf("Perrito no encontrado\n");
    }
}
```

La función `imprimir_por_edad` recibe un arreglo de estructuras `datos_Registro` y un puntero `n` que indica el número de registros. El usuario ingresa la edad de un perro a buscar. Luego, se realiza un bucle para validar la entrada del usuario como un número entero. Después de convertir la entrada a un entero, se busca en el arreglo de registros por la coincidencia de la edad. Si se encuentra una coincidencia, se imprime la información del perro, incluyendo su ID, nombre, edad, talla, fecha de registro, estado de esterilización y estado de adopción. Si no se encuentra ninguna coincidencia, se muestra un mensaje indicando que el perro no fue encontrado.

- Imprimir por nombre

```
void imprimir_por_nombre(datos_Registro *datos, int n) {
    char nombre[20], dato[50], nombreP[11], nombreP2[11];
    bool ver2;
    printf("Ingrese el nombre del perrito a buscar: ");
    int solo_letras;
    do {
        solo_letras = 1;
        scanf("%s", dato);
        for (int j = 0; dato[j] != '\0'; j++) {
            if (!isalpha(dato[j])) {
                solo_letras = 0;
                break;
            }
        }
        if (!solo_letras) {
            printf("El nombre solo debe de contener letras (sin acentos). Intentalo de nuevo.\n");
        }
        if (strlen(dato) > 10) {
            printf("El nombre no debe de contener mas de 10 caracteres\n");
            solo_letras = 0;
        }
    } while (!solo_letras);
    for (int indice = 0; dato[indice] != '\0'; ++indice) {
        dato[indice] = tolower(dato[indice]);
    }
    strcpy(nombre, dato);
    for (int i = 0; i < n; i++) {
        strcpy(nombreP2, datos[i].nombre);
        for (int indice = 0; nombreP2[indice] != '\0'; ++indice) {
            nombreP2[indice] = tolower(nombreP2[indice]);
        }
        if (strcmp(nombre, nombreP2) == 0) {
            printf("ID: %d\n", datos[i].id);
            strcpy(nombreP, datos[i].nombre);
            printf("Nombre: %s \n", nombreP);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if (datos[i].talla == 1) {
                printf("Grande\n");
            }
            if (datos[i].talla == 2) {
                printf("Mediano\n");
            }
            if (datos[i].talla == 3) {
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            ver2 = true;
        }
    }
    if (ver2 == false) {
        printf("Perrito no encontrado\n");
    }
}
```

La función `imprimir_por_nombre` busca un perro en el arreglo de registros por su nombre. El usuario ingresa el nombre del perro a buscar y se valida que solo contenga letras y no exceda los 10 caracteres. Luego, se convierte el nombre ingresado y los nombres en los registros a minúsculas para realizar la comparación. Si se encuentra una coincidencia, se imprime la información del perro, incluyendo su ID, nombre, edad, talla, fecha de registro, estado de esterilización y estado de adopción. Si no se encuentra ninguna coincidencia, se muestra un mensaje indicando que el perro no fue encontrado.

- Imprimir por esterilizado

```
void imprimir_por_esterilizado(datos_Registro *datos, int n) {
    int esterilizado;
    bool ver2, ver;
    char nombreP[11], dato[50];
    printf("Ingrese 1 si quiere ver los perritos esterilizados o 2 si quiere ver los no esterilizados: ");
    do{
        ver=false;
        do{
            fflush(stdin);
            scanf("%s", &dato);
        }while(validarEnteros(dato) == true);
        esterilizado = atoi(dato);
        if(esterilizado!= 1 && esterilizado !=2){
            printf("Ingrese un dato valido\n");
            ver=true;
        }
    }while(ver==true);
    for (int i = 0; i < n; i++) {
        if (datos[i].esterilizado == esterilizado) {
            printf("ID: %d\n", datos[i].id);
            strcpy(nombreP, datos[i].nombre);
            printf("Nombre: %s\n", nombreP);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if(datos[i].talla == 1){
                printf("Grande\n");
            }
            if(datos[i].talla == 2){
                printf("Mediano\n");
            }
            if(datos[i].talla == 3){
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            ver2=true;
        }
    }
    if(ver2==false){
        printf("Perrito no encontrado\n");
    }
}
```

La función `imprimir_por_esterilizado` muestra los perros en el arreglo de registros según su estado de esterilización. El usuario ingresa 1 para ver los perros esterilizados o 2 para ver los perros no esterilizados. Se valida la entrada del usuario para asegurarse de que sea un valor válido. Luego, se recorre el arreglo de registros y se imprimen los perros que coinciden con el estado de esterilización seleccionado. Se muestra la información del perro, incluyendo su ID, nombre, edad, talla, fecha de registro, estado de esterilización y estado de adopción. Si no se encuentra ninguna coincidencia, se muestra un mensaje indicando que no se encontraron perros.

- Imprimir por adoptado

```
void imprimir_por_adoptado(datos_Registro *datos, int n) {
    int adoptado;
    bool ver2, ver;
    char dato[50], nombreP[11];
    printf("Ingrese 1 si quiere ver los perritos adoptados o 2 si quiere ver los no adoptados: ");
    do{
        ver=false;
        do{
            fflush(stdin);
            scanf("%s", &dato);
        }while(validarEnteros(dato) == true);
        adoptado = atoi(dato);
        if(adoptado!= 1 && adoptado !=2){
            printf("Ingrese un dato valido\n");
            ver=true;
        }
    }while(ver==true);
    for (int i = 0; i < n; i++) {
        if (datos[i].adoptado == adoptado) {
            printf("ID: %d\n", datos[i].id);
            strcpy(nombreP, datos[i].nombre);
            printf("Nombre: %s\n", nombreP);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if(datos[i].talla == 1){
                printf("Grande\n");
            }
            if(datos[i].talla == 2){
                printf("Mediano\n");
            }
            if(datos[i].talla == 3){
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            ver2=true;
        }
    }
    if(ver2==false){
        printf("Perrito no encontrado\n");
    }
}
```

La función `imprimir_por_adoptado` muestra los perros en el arreglo de registros según su estado de adopción. El usuario ingresa 1 para ver los perros adoptados o 2 para ver los perros no adoptados. Se valida la entrada del usuario para asegurarse de que sea un valor válido. Luego, se recorre el arreglo de registros y se imprimen los perros que coinciden con el estado de adopción seleccionado. Se muestra la información del perro, incluyendo su ID, nombre, edad, talla, fecha de registro, estado de esterilización y estado de adopción. Si no se encuentra ninguna coincidencia, se muestra un mensaje indicando que no se encontraron perros.

- Imprimir por talla

```
void imprimir_por_talla(datos_Registro *datos, int n, int opcTalla) {
    bool ver2;
    for (int i = 0; i < n; i++) {
        if (datos[i].talla == opcTalla) {
            printf("ID: %d\n", datos[i].id);
            printf("Nombre: %s\n", datos[i].nombre);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if(datos[i].talla == 1){
                printf("Grande\n");
            }
            if(datos[i].talla == 2){
                printf("Mediano\n");
            }
            if(datos[i].talla == 3){
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            ver2=true;
        }
    }
    if(ver2==false){
        printf("Perrito no encontrado\n");
    }
}
```

La función `imprimir_por_talla` muestra los perros en el arreglo de registros según su talla. Recibe como parámetros el arreglo de registros, el número de registros y la opción de talla seleccionada. Se recorre el arreglo de registros y se imprimen los perros que coinciden con la talla seleccionada. Se muestra la información del perro, incluyendo su ID, nombre, edad, talla, fecha de registro, estado de esterilización y estado de adopción. Si no se encuentra ninguna coincidencia, se muestra un mensaje indicando que no se encontraron perros.


```

void adoptar(datos_Registro *datos, int n){
    int opcAdopcion, continuar, id_Adopcion;
    bool salida, ver2;
    char dato[50];
    int opcTalla;
    do{
        system("cls");
        salida=false;
        printf("\n");
        printf("-----A D O P C I O N-----\n");
        printf("1.- Talla \n2.- Edad\n");
        printf("-----\n");
        do{
            ver2=false;
            scanf("%s", &dato);
            ver2=validarEnteros(dato);
            opcAdopcion = atoi(dato);
            if(opcAdopcion !=1 && opcAdopcion !=2){
                printf("Ingrese solamente una de las opciones anteriores\n");
                ver2=true;
            }
        }while( ver2 == true);
        switch(opcAdopcion){
            case 1:
                printf("\n");
                printf("-----A D O P C I O N-----\n");
                printf("1.- Grande\n2.- Mediano\n3.- Chico\n");
                printf("-----\n");
                do{
                    ver2 = false;
                    scanf("%s", &dato);
                    ver2=validarEnteros(dato);
                    opcTalla = atoi(dato);
                    if(opcTalla<1 && opcTalla>3){
                        ver2=true;
                    }
                }while( ver2 == true);
                imprimir_por_talla(datos, n, opcTalla);
                break;
            case 2:
                imprimir_por_edad(datos, &n);
                break;
            default:
                printf("Ingrese solamente una de las opciones anteriores\n");
        }
        printf("-----A D O P C I O N-----\n");
    }
}

```



```

printf("-----A D O P C I O N-----\n");
printf("1.- Continuar con la adopcion \n2.- Volver a consultar\n");
printf("-----\n");
do{
    scanf("%s", &dato);
}while(validarEnteros(dato) == true);
continuar = atoi(dato);
switch(continuar){
    case 1:
        system("cls");
        break;
    case 2:
        salida=true;
        break;
    default:
        printf("Ingrese solamente una de las opciones anteriores\n");
}
}while(salida==true);
bool ver3;
printf("Ingrese el ID del perrito a adoptar\n");
do{
    char salir[] = "R";
    ver3=true;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato)==true);
    id_Adopcion = atoi(dato);
    for(int i=0; i<n; i++){
        if(datos[i].id == id_Adopcion){
            ver3=false;
        }
    }
    if(ver3==true){
        printf("ID no encontrado\n");
    }
}while(ver3==true);
do{
    ver2=true;
    for (int i = 0; i < n; i++) {
        if (datos[i].id == id_Adopcion){
            printf("ID: %d\n", datos[i].id);
            printf("Nombre: %s\n", &datos[i].nombre);
            printf("Edad: %d\n", datos[i].edad);
            printf("Talla: ");
            if(datos[i].talla == 1){

```

```

                printf("Grande\n");
            }
            if(datos[i].talla == 2){
                printf("Mediano\n");
            }
            if(datos[i].talla == 3){
                printf("Chico\n");
            }
            printf("Fecha de registro: %02d/%02d/%d \n", datos[i].dia, datos[i].mes, datos[i].anio);
            printf("Esterilizado: %s\n", datos[i].esterilizado == 1 ? "Si" : "No");
            printf("Adoptado: %s\n", datos[i].adoptado == 1 ? "Si" : "No");
            ver2=false;
        }
    }
    if(ver2==true)
        printf("Ingrese un id valido\n");
}while(ver2==true);
system("pause");
ficha_Adopcion(datos, id_Adopcion);
}

```

La función adoptar permite realizar el proceso de adopción de perros. Se solicita al usuario que elija entre dos opciones: adopción por talla o adopción por edad. Dependiendo de la opción seleccionada, se muestra la lista de perros disponibles que cumplen con esa característica utilizando las funciones imprimir_por_talla o imprimir_por_edad. Luego, se le pide al usuario que ingrese el ID del perro que desea adoptar, verificando que el ID sea válido. Una vez seleccionado el perro, se muestra su información y se procede a generar la ficha de adopción utilizando la función ficha_Adopcion. El usuario tiene la opción de continuar con la adopción o volver a realizar una consulta.


```

void actualiza_Registro(datos_Registro *datos, int n){
    int num;
    char dato[50];
    bool ver, verId;
    do{
        verId=false;
        printf("Ingrese el ID a modificar\n");
        do{
            fflush(stdin);
            scanf("%s", &dato);
        }while(validarEnteros(dato) == true);
        num = atoi(dato);
        for(int i=0; i<n; i++){
            if(datos[i].id==num){
                verId = true;
            }
        }
        if(verId==false){
            printf("ID no encontrada vuelva a intentarlo\n");
        }
    }while(verId==false);
    printf("Ingrese el nombre del perrito %d\n", num);
    int solo_letras;
    do {
        solo_letras = 1;
        scanf("%s", dato);
        for (int j = 0; dato[j] != '\0'; j++) {
            if (!isalpha(dato[j])) {
                solo_letras = 0;
                break;
            }
        }
        if (!solo_letras) {
            printf("El nombre debe de contener solo letras(sin acentos). Intentalo de nuevo.\n");
        }
        if(strlen(dato)>10){
            printf("El nombre no debe de contener mas de 10 caracteres\n");
            solo_letras = 0;
        }
    } while (!solo_letras);
    strcpy(datos[num-1].nombre, dato);
    printf("Ingrese la edad del perrito: ");
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
}

```

```

}while(validarEnteros(dato) == true);
datos[num-1].edad = atoi(dato);
printf("Ingrese la talla del perrito: \n1.-Grande\n2.-Mediano\n3.-Chico\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[num-1].talla = atoi(dato);
    if(datos[num-1].talla != 1 && datos[num-1].talla !=2 && datos[num-1].talla !=3){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}while(ver==true);
printf("Ingrese la Fecha de registro (dd mm aaaa): \n");

do{
    ver=false;
    do{
        printf("D%ca: ", 161);
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[num-1].dia = atoi(dato);
    if(datos[num-1].dia<1 || datos[num-1].dia>31){
        printf("Ingrese un d%ca valido\n", 161);
        ver=true;
    }
}while(ver==true);
do{
    ver=false;
    do{
        printf("\nMes: ");
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[num-1].mes = atoi(dato);
    if(datos[num-1].mes<1 || datos[num-1].mes>12){
        printf("Ingrese un mes valido\n");
        ver=true;
    }
}while(ver==true);
do{
    ver=false;
    do{

```

```

do{
    printf("\nA%co: ", 164);
    fflush(stdin);
    scanf("%s", &dato);
}while(validarEnteros(dato) == true);
datos[num-1].anio = atoi(dato);
if(datos[num-1].anio<1990 || datos[num-1].anio>2023){
    printf("Ingrese un a%co valido\n", 164);
    ver=true;
}
}while(ver==true);
printf("Ingrese si el perrito esta esterilizado \n1.-Esta esterilizado\n2.-No esta esterilizado\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[num-1].esterilizado = atoi(dato);
    if(datos[num-1].esterilizado != 1 && datos[num-1].esterilizado !=2){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}while(ver==true);
printf("Ingrese si el perrito esta adoptado \n1.-Esta adoptado\n2.-No esta adoptado\n");
do{
    ver=false;
    do{
        fflush(stdin);
        scanf("%s", &dato);
    }while(validarEnteros(dato) == true);
    datos[num-1].adoptado = atoi(dato);
    if(datos[num-1].adoptado != 1 && datos[num-1].adoptado !=2){
        printf("Ingrese un dato valido\n");
        ver=true;
    }
}while(ver==true);
}
}

```

La función `actualiza_Registro` permite actualizar los datos de un perro en el registro. Primero, se solicita al usuario que ingrese el ID del perro que desea modificar, verificando que el ID exista en el registro. Luego, se solicitan los nuevos datos del perro, como el nombre, la edad, la talla, la fecha de registro, si está esterilizado y si está adoptado. Se realizan validaciones en cada dato ingresado para asegurar su formato y validez. Finalmente, los nuevos datos se asignan al perro correspondiente en el registro.

Funciones Validar Enteros y Validar Caracteres

```

bool validarEnteros(char c[]){
    bool ver;
    ver=false;
    for(int i=0; i<strlen(c); i++){
        if(isdigit(c[i])==0){
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
            printf("Error: valor no valido\n");
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
            ver=true;
            break;
        }
    }
    return ver;
}

```

La función `validarEnteros` se utiliza para validar si una cadena de caracteres contiene únicamente dígitos. Recorre cada carácter de la cadena y verifica si es un dígito utilizando la función `isdigit`. Si encuentra algún carácter que no es un dígito, muestra un mensaje de error y devuelve `true`. En caso contrario, devuelve `false`. Esta función puede ser utilizada para validar entradas que se esperan que sean números enteros.

```

bool validarCaracteres(char c[]) {
    bool ver;
    ver=false;
    for(int i=0; i<strlen(c); i++){
        if(isalpha(c[i]) == 0 && c[i] != ' '){
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 4);
            printf("Error: valor no valido\n");
            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);
            ver=true;
            break;
        }
    }
    return ver;
}

```

La función validarCaracteres se utiliza para validar si una cadena de caracteres contiene únicamente letras y espacios. Recorre cada carácter de la cadena y verifica si es una letra utilizando la función isalpha, o si es un espacio. Si encuentra algún carácter que no es una letra ni un espacio, muestra un mensaje de error y devuelve true. En caso contrario, devuelve false. Esta función puede ser utilizada para validar entradas que se esperan que sean nombres o descripciones que contengan solo letras y espacios.

Ficha de Adopcion

```
void ficha_Adopcion(datos_Registro *datos, int id_Adopcion){
    char nombre_Archivo[40], nombre[10], apellido[40], domicilio[100], dato[50];
    char tipoArchivo[] = ".txt";
    int edad;
    int solo_letras;
    bool ver;
    printf("Solicitud de adopcion aceptada\n");
    printf("Por favor ingrese los siguientes datos:\n");
    printf("Nombre: \n");
    do {
        solo_letras = 1;
        scanf("%s", nombre);
        for (int j = 0; nombre[j] != '\0'; j++) {
            if (!isalpha(nombre[j])) {
                solo_letras = 0;
                break;
            }
        }
        if (!solo_letras) {
            printf("Ingrese solo letras.\n");
        }
    } while (!solo_letras);
    printf("Apellido paterno: \n");
    do {
        solo_letras = 1;
        scanf("%s", apellido);
        for (int j = 0; apellido[j] != '\0'; j++) {
            if (!isalpha(apellido[j])) {
                solo_letras = 0;
                break;
            }
        }
        if (!solo_letras) {
            printf("Ingrese solo letras.\n");
        }
    } while (!solo_letras);
    printf("Edad: \n");
    do{
        ver=false;
        fflush(stdin);
        scanf("%s", &dato);
        ver = validarEnteros(dato);
        edad = atoi(dato);
        if(edad<18){
            printf("Debes ser mayor de edad para poder adoptar a un perrito\n");
        }
    }
```

```

        ver=true;
    }
}while( ver == true);
printf("Domicilio: \n");
scanf("%s", domicilio);
strcat(strcpy(nombre_Archivo,apellido), tipoArchivo);

FILE *fichero;
fichero = fopen(nombre_Archivo, "rt");
if(fichero == NULL){
    fichero = fopen(nombre_Archivo, "w");
}else{
    fichero = fopen(nombre_Archivo, "r+");
}
fprintf(fichero, "\n\n----FICHA DE ADOPCION----\n");
fprintf(fichero, "Nombre del perrito: %s\n", datos[id_Aopcion-1].nombre);
fprintf(fichero, "Edad: %d\n", datos[id_Aopcion-1].edad);
fprintf(fichero, "Talla: ");
if(datos[id_Aopcion-1].talla == 1){
    fprintf(fichero, "Grande\n");
}
if(datos[id_Aopcion-1].talla == 2){
    fprintf(fichero, "Mediano\n");
}
if(datos[id_Aopcion-1].talla == 3){
    fprintf(fichero, "Chico\n");
}
fprintf(fichero, "Esterilizado: %s\n", datos[id_Aopcion-1].esterilizado == 1 ? "Si" : "No");
fprintf(fichero, "Nombre completo del dueño: %s %s\n", nombre, apellido);
fprintf(fichero, "Edad: %d\n", edad);
fprintf(fichero, "Domicilio: %s\n", domicilio);
fclose(fichero);
system("cls");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 6);
gotoxy(25,6);
printf(" / \\_\n");
gotoxy(25,7);
printf(" ( @\\_\n");
gotoxy(35,7);
printf("WOOF!!");
gotoxy(25,8);
printf(" / 0\n");
gotoxy(25,9);
printf("/ (____/\n");
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 7);

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),7);
printf("Adopcion completada, consulte la ficha generada\n");
system("pause");
}

```

La función `ficha_Aopcion` se encarga de registrar los datos de adopción de un perro en un archivo de texto. Primero, solicita al usuario que ingrese su nombre, apellido, edad y domicilio, validando que el nombre y el apellido contengan solo letras. Luego, crea un nombre de archivo concatenando el apellido del adoptante con la extensión ".txt".

A continuación, se abre el archivo en modo lectura ("rt"). Si el archivo no existe, se crea en modo escritura ("w"). Si el archivo ya existe, se abre en modo lectura y escritura ("r+"). Luego, se escribe en el archivo la ficha de adopción, incluyendo el nombre del perro, edad, talla, esterilización y los datos del adoptante.

Finalmente, se muestra en pantalla una representación gráfica de un perro y se informa al usuario que la adopción se ha completado, solicitando que consulte la ficha generada. Se pausa la ejecución hasta que el usuario presione una tecla.