



FICHERO HTML

Leer fichero html en busca de imágenes y urls.

Noelia López Rúa y Guillermo Martín Lechuga



Objetivos del trabajo

- Crear una aplicación que partiendo de un fichero de texto (html) identifique los enlaces e imágenes que hay en él.
- Almacenar en un fichero nuevo los urls encontrados.
- Mostrar por pantalla la relación de imágenes encontradas.
- Manejo de funciones de paso por referencia y paso por valor con estructuras y del uso de ficheros en el lenguaje de programación C.

Requisitos

- Que el programa sea capaz de leer un fichero de texto de tipo html en busca de enlaces e imágenes.
- Poder almacenar los enlaces encontrados en un fichero nuevo tipo txt.
- Poder mostrar por pantalla las imágenes encontradas.

Funciones

- Función menú: muestra por pantalla las opciones que incluye el programa

```
int menu()
{
    int opcion;
    do
    {
        printf("Menu:\n");
        printf("1.Leer\n");
        printf("2.Guardar\n");
        printf("3.Imagenes\n");
        printf("4.Salir\n");
        scanf("%d",&opcion);
    }
    while((opcion<1) || (opcion>4));

    return opcion;
}
```

- **Función Leer:** esta función llama al resto de funciones que se encargan de buscar en el archivo tipo html las urls que haya en él.

```
void funcion_leer(FILE*fentrada)
{
    NODO *lista; // Lista con las URL detectadas
    //llamar a las funciones de la maquina de estado
    lista = procesar_fichero (fentrada);
    fclose (fentrada);
    imprime_lista(lista);
}

void imprime_lista (NODO *p)
{
    printf ("URLs detectadas:\n");
    while (p!=NULL)
    {
        printf ("%s\n",p->cadena);
        p = p->siguiente;
    }

    return ;
}
```

```
NODO * procesar_fichero(FILE *fichero)
{
    char letra;
    int estado=OUT_QUOTES;
    char cadena[L],longitud;
    NODO * lista_url=NULL;

    fscanf (fichero,"%c",&letra);
    while (!feof(fichero))
    {
        switch (estado)
        {
            case OUT_QUOTES: if (letra == '"')
                            {
                                estado=IN_QUOTES;
                                longitud=0;
                            }
                            break;

            case IN_QUOTES: if (letra!='"')
                            {
                                cadena[longitud]=letra;
                                longitud++;
                            }
                            else
                            {
                                cadena[longitud]='\0';
                                lista_url= procesa_cadena (cadena,lista_url);
                                estado=OUT_QUOTES;
                            }
                            break;
        }
        fscanf (fichero,"%c",&letra);
    }
    return lista_url;
}
```

- **Función Guardar:** se encarga de crear el nuevo fichero con los urls encontrados.

```
FILE* funcion_guardar(FILE *entrada) //guarda todas las urls encontradas en un fichero nuevo
{
    FILE *nuevo;
    int cierre;
    NODO *lista_guardar;

    lista_guardar = procesar_fichero (entrada);
    fclose (entrada);
    nuevo = fopen("Encontradas.txt","w");

    if(nuevo==NULL)
        printf("No se ha podido abrir el fichero\n");
    else
    {
        imprime_guardar(lista_guardar,nuevo);
        fclose(nuevo);

        if(cierre == EOF)
            printf("No se ha podido cerrar el fichero\n");
        else
            printf("Fichero creado\n");
    }
    return nuevo;
}
```

- **Función Imágenes:** llama al resto de funciones que se encargan de buscar las imágenes que hay dentro del archivo html y las imprime por pantalla.

```
void funcion_imagenes(FILE *fichero)
{
    NODO *lista_imagenes;

    lista_imagenes = procesar_fichero_imagenes (fichero);
    fclose (fichero);
    imprime_lista_imagenes(lista_imagenes);
}
```

```
void imprime_lista_imagenes(NODO *p)
{
    printf ("Imagenes detectadas:\n");
    while (p!=NULL)
    {
        printf ("%s\n",p->cadena);
        p = p->siguiente;
    }
}
```

```
NODO * procesar_fichero_imagenes(FILE *fichero)
{
    char letra;
    int estado=OUT_QUOTES;
    char cadena[L],longitud;
    NODO * lista_imagenes=NULL;

    fscanf (fichero,"%c",&letra);
    while (!feof(fichero))
    {
        switch (estado)
        {
            case OUT_QUOTES: if (letra == '"')
            {
                estado=IN_QUOTES;
                longitud=0;
            }
            break;

            case IN_QUOTES: if (letra!='"')
            {
                cadena[longitud]=letra;
                longitud++;
            }
            else
            {
                cadena[longitud]='\0';
                lista_imagenes= procesa_cadena_imagenes(cadena,lista_imagenes);
                estado=OUT_QUOTES;
            }
            break;
        }
        fscanf (fichero,"%c",&letra);
    }
    return lista_imagenes;
}
```

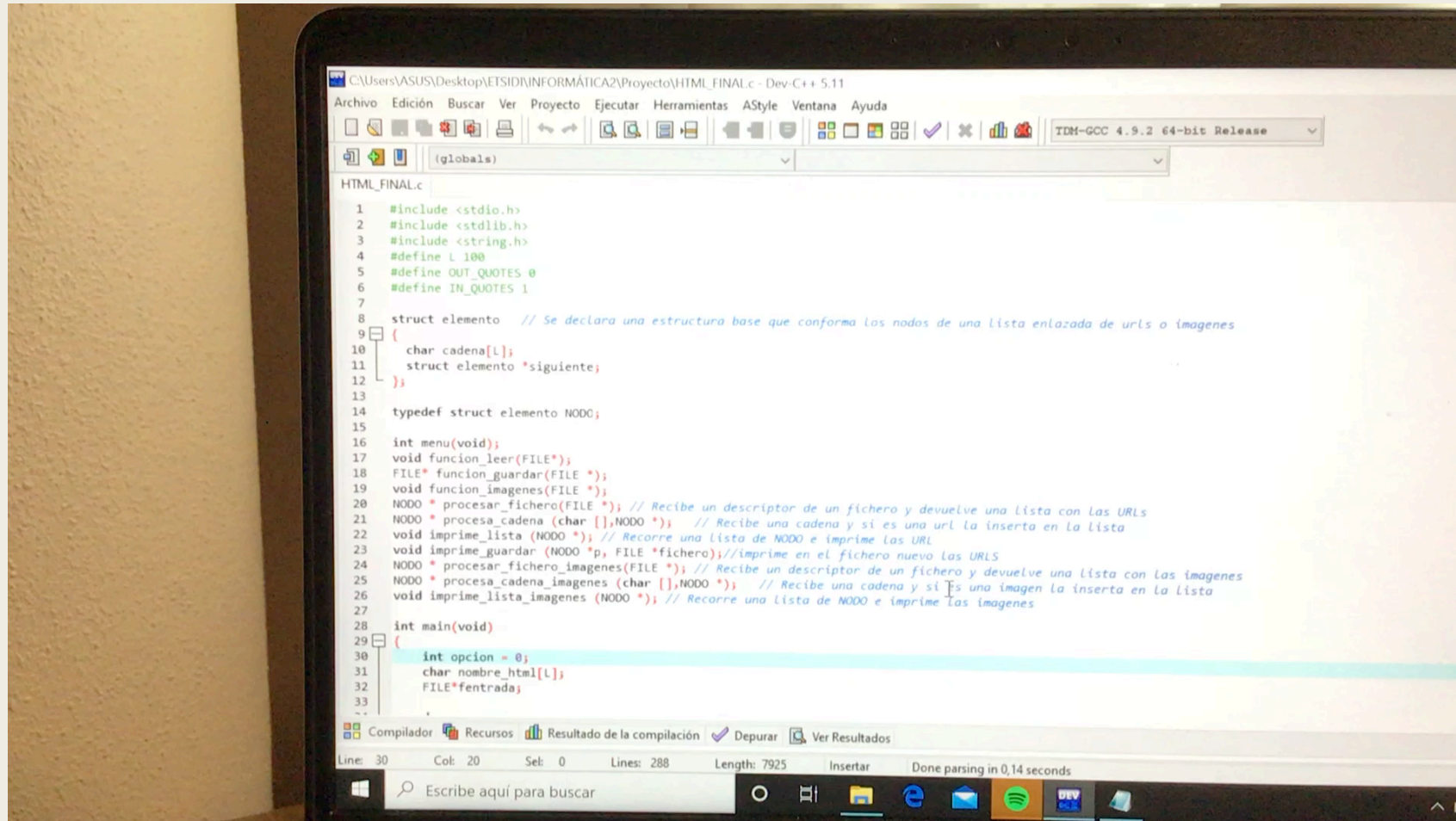
■ Función Comparar Cadena urls:

```
NODO * procesa_cadena (char cadena[],NODO *cabecera)
{
    NODO *p;
    if (strncmp (cadena,"http",4) == 0)
    {
        p = (NODO *) malloc (sizeof(NODO)); // Se crea un nuevo NODO
        if (p==NULL)
            printf ("Memoria insuficiente procesando URLs\n");
        else
        {
            strcpy (p->cadena,cadena);
            p->siguiente = cabecera;
            cabecera = p;
        }
    }
    return cabecera;
}
```

■ Función Comparar Cadena Imágenes:

```
NODO * procesa_cadena_imagenes (char cadena[],NODO *cabecera)
{
    NODO *p;
    if (strncmp (cadena,"images",6) == 0)
    {
        p = (NODO *) malloc (sizeof(NODO));
        if (p==NULL)
            printf ("Memoria insuficiente procesando URLs\n");
        else
        {
            strcpy (p->cadena,cadena);
            p->siguiente = cabecera;
            cabecera = p;
        }
    }
    return cabecera;
}
```


VÍDEO DEL FUNCIONAMIENTO DEL PROGRAMA



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #define L 100
5 #define OUT_QUOTES 0
6 #define IN_QUOTES 1
7
8 struct elemento // Se declara una estructura base que conforma los nodos de una lista enlazada de urls o imagenes
9 {
10     char cadena[L];
11     struct elemento *siguiente;
12 };
13
14 typedef struct elemento NODO;
15
16 int menu(void);
17 void funcion_leer(FILE*);
18 FILE* funcion_guardar(FILE *);
19 void funcion_imagenes(FILE *);
20 NODO * procesar_fichero(FILE *); // Recibe un descriptor de un fichero y devuelve una lista con las URLs
21 NODO * procesa_cadena (char [], NODO *); // Recibe una cadena y si es una url la inserta en la lista
22 void imprime_lista (NODO *); // Recorre una lista de NODO e imprime las URL
23 void imprime_guardar (NODO *p, FILE *fichero); // Imprime en el fichero nuevo las URLs
24 NODO * procesar_fichero_imagenes(FILE *); // Recibe un descriptor de un fichero y devuelve una lista con las imagenes
25 NODO * procesa_cadena_imagenes (char [], NODO *); // Recibe una cadena y si es una imagen la inserta en la lista
26 void imprime_lista_imagenes (NODO *); // Recorre una lista de NODO e imprime las imagenes
27
28 int main(void)
29 {
30     int opcion = 0;
31     char nombre_html[L];
32     FILE *fentrada;
33     ..
34 }
```