

PROYECTO 2

MANUAL DE USUARIO



Allan Roynell González Hernández

202003708

PROYECTO 2

OBJETIVOS DEL SISTEMA

- Utilizar un lenguaje de programación para establecer funciones que puedan ejecutarse en una interfaz gráfica
- Establecer una gramática para los archivos que nuestra interfaz pueda reconocer y a su vez poder ejecutar
- Practicar sobre las teorías vistas en clases, expresiones regulares, método de árbol, reducciones de DFA, Gramática de tipo 2.

INFORMACIÓN DEL SISTEMA

Nuestro software esta diseñado para poder solucionar problemas referidos a un simulador de Sistema de Bases de Datos, quien recibe una cantidad de claves, de registros y una cantidad de funciones. Estos los definirá la persona quien use el software, esto es con el objetivo de poder poner en práctica y poder establecer las gramáticas vistas en el curso del área Magistral.

El sistema tiene un espacio de edición donde podemos editar el texto y este se verá reflejado en los cambios del archivo, a su vez cuenta con una parte de salida donde se ejecutarán las funciones que nosotros escribamos en el documento, dentro de las opciones de los botones podemos “Compilar”, “Abrir el archivo”, “Generar reporte de errores”, “Graphviz” que nos mostrara el árbol de derivación del documento.

REQUISITOS DEL SISTEMA

Procesador:

- Intel Superior a la 4ta gen
- AMD Superior a las versiones de Opteron

Memoria RAM:

- Superior a las 2GB de RAM, en versiones superiores a DDR3

Espacio de Disco:

- Tener libre 30MB

Sistema Operativo:

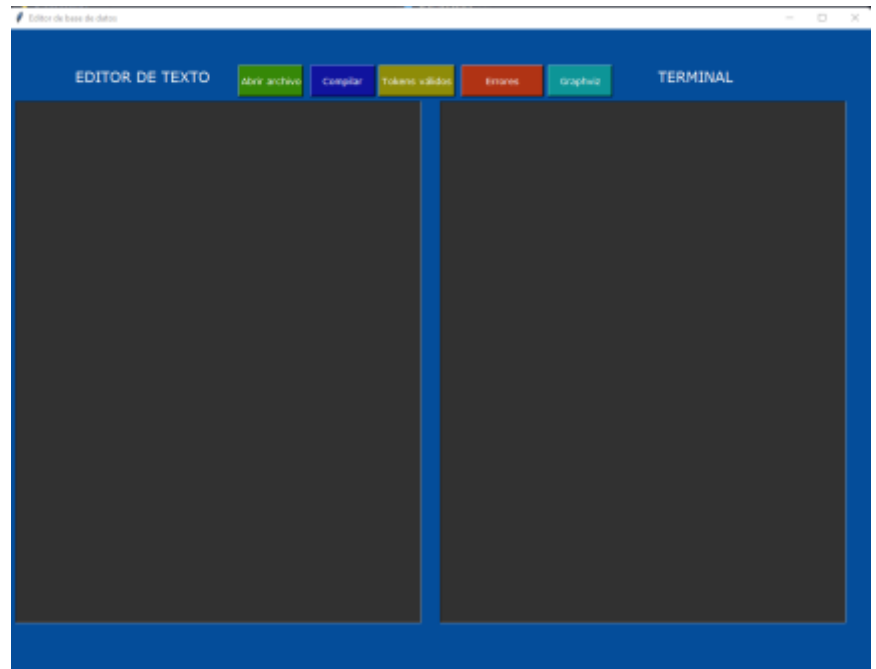
- MacOS, Windows, Distribuciones de Linux. (Compatibles con Python)

Resolución Gráfica:

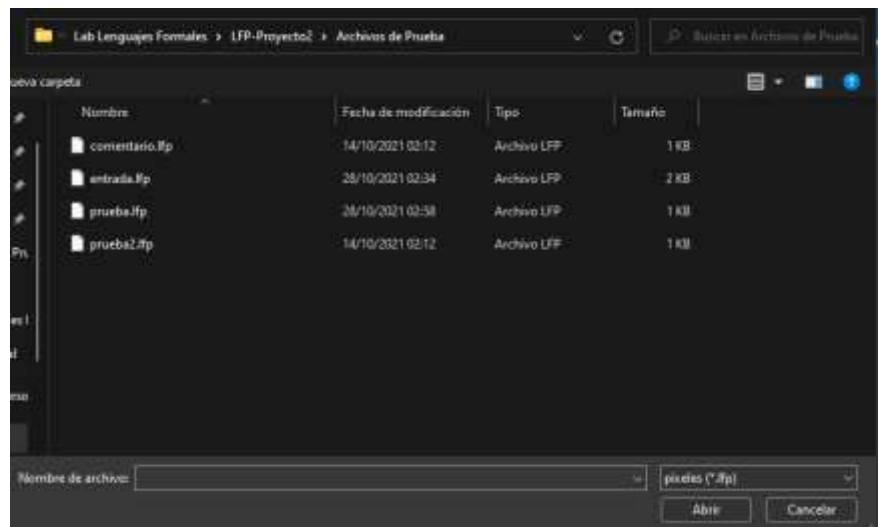
- No es necesario una tarjeta gráfica o un chip de video adicional para ejecutar el programa.

INTERFAZ GRÁFICA Y FLUJO DE LAS FUNCIONALIDADES


Ejecutamos nuestro archivo GUI.py haciendo doble click. Tendremos la siguiente interfaz:



Si nos percatamos tenemos 3 botones podemos empezar a trabajar usando la opción de abrir archivo.



Una vez elegido el archivo nos podemos percatar que se nos abre en una parte de edición donde podemos hacer ciertas modificaciones a nuestro archivo.



```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_PRODUCTOS 10
#define MAX_STOCK 100

typedef struct {
    int codigo;
    char producto[MAX_PRODUCTOS];
    float precio_compra;
    float precio_venta;
    int stock;
} Producto;

Producto productos[MAX_PRODUCTOS];

void inicializarProductos();
void mostrarProductos();
void actualizarProducto(int codigo, float precio_compra, float precio_venta, int stock);

int main() {
    inicializarProductos();
    mostrarProductos();
    actualizarProducto(1, 15.00, 18.00, 5);
    return 0;
}


void inicializarProductos() {
    productos[0] = {"Barranco", 10.00, 12.00, 5};
    productos[1] = {"Salta", 15.00, 18.00, 7};
    productos[2] = {"Buenos Aires", 20.00, 25.00, 10};
    productos[3] = {"Rosario", 12.00, 15.00, 8};
}

void mostrarProductos() {
    printf("Reporte de productos\n");
    printf("-----\n");
    printf("%-5s %-20s %-10s %-10s %-5s\n", "codigo", "producto", "precio_compra", "precio_venta", "stock");
    for (int i = 0; i < MAX_PRODUCTOS; i++) {
        printf("%-5d %-20s %-10.2f %-10.2f %-5d\n", productos[i].codigo, productos[i].producto, productos[i].precio_compra, productos[i].precio_venta, productos[i].stock);
    }
}

void actualizarProducto(int codigo, float precio_compra, float precio_venta, int stock) {
    printf("Actualizando producto\n");
    printf("Codigo: %d\n", codigo);
    printf("Precio compra: %.2f\n", precio_compra);
    printf("Precio venta: %.2f\n", precio_venta);
    printf("Stock: %d\n", stock);
}

```

Al hacer click al botón de compilar las funciones que estén en el archivo se mostrarán en nuestra terminal, si tenemos algún error este mostrará el tipo de error, columna, fila y una posible solución.



```

Reporte de productos
-----
codigo  producto          precio_compra  precio_venta  stock
1       Barranco      10.00         12.00         5
2       Salta         15.00         18.00         7
3       Buenos Aires  20.00         25.00        10
4       Rosario      12.00         15.00         8

```

Los botones restantes como Tokens Válidos nos generará un reporte en HTML para que podamos ver todos los tokens ingresados en el documento.

El botón de Errores nos mostrará el error que hayamos tenido si en dado caso lo tenemos.

El botón de Graphviz nos generará un árbol de derivación del archivo de entrada, respecto a la gramática establecida.

