



Manual De Usuario Proyecto#02 - LFP

Descripción del lenguaje

Sección Claves

Lista de claves

Imagen de ejemplo de claves:

Sección Registros

Lista de registros

Imagen de ejemplo de registros:

Comentarios

Comentario Simple

Comentario Multilinea

Sección de Reportería

Función imprimir("cadena");

Función imprimirln("cadena");

Función Conteo();

Función promedio("campo");

Función contarsi("campo", valor);

Función datos();

Función sumar("campo");

Función max("campo");

Función min("campo");

Función exportarReporte("titulo");

Interfaz Gráfica

Ventana Principal

Centro de comandos

Cargar Archivo

Analizar Archivo

Reportes

Información

Salir

Descripción del lenguaje

Este proyecto tiene como entrada un lenguaje con estructura, mediante un archivo con extensión "lfp", dicho archivo cuenta con diferentes funciones y en esencia con la estructura del lenguaje, para una mejor visualización y entendimiento, se utilizará un archivo de entrada de prueba para ejemplificar cada parte del lenguaje.

Sección Claves

En esta sección se declaran los claves o campos por los que están contruidos los registros, su estructura está formada por la palabra reservada Claves, seguido de signo igual, corchete de apertura, lista de claves y corchete de cierre.

Lista de claves

está formada por cadenas de caracteres encerradas entre comillas y separadas por coma.

Imagen de ejemplo de claves:

```
Archivo  Edición  Formato  Ver  Ayuda
Claves = [
    "portero", "rating", "nacionalidad"
]
```

Sección Registros

En esta sección se detallan los registros que se quieren analizar y sigue la estructura dada por palabra reservada Registros, signo igual, corchete de apertura, lista de registros y corchete de cierre.

Lista de registros

Cada registro está encerrado entre llave de apertura y llave de cierre y sus valores están separados por comas, estos valores pueden ser cadenas de texto, enteros o decimales.

Imagen de ejemplo de registros:

```
Registros = [  
    {"Ter_Stegen", 99, "Alemania"}  
    {"Keylor_Navas", 88, "Costa_Rica"}  
    {"Claudio_Bravo", 79, "Chile"}  
    {"Dibu_Martinez", 96, "Argentina"}  
    {"Armani", 79, "Argentina"}  
]
```

Comentarios

Todo lenguaje debe tener la capacidad de comentar el código o la información, ya sea para explicar lo que sucede en esa parte o como recordatorio de la estructura de lo que se está indicando, en este caso el lenguaje cuenta con 2 opciones de comentarios, el simple y el multilinea.

Comentario Simple

Se representan con un numeral y finalizan con un salto de línea, a continuación un ejemplo.

```
#Un comentario corto
```

Comentario Multilinea

Inicia con tres comillas simples y finaliza con tres comillas simples, a continuación un ejemplo.

```
...  
Este comentario  
es un poco  
mas largo  
...
```

Sección de Reportería

Como todo lenguaje, los datos del archivo de entrada no tendrían sentido si no tuvieran la capacidad de generar resultados, por lo tanto, el lenguaje reconoce ciertas palabras reservadas, las cuales servirán para generar información importante como reporte.

Función imprimir("cadena");

Imprime por consola el valor dado por la cadena, a continuación un ejemplo.

```
imprimir("Reporte de ");  
imprimir("Porteros");
```

recalcar que en este caso no existe el salto de línea por lo tanto, en consola, aparecería un resultado como el siguiente ejemplo:

```
>>>Reporte de Porteros
```

Función imprimirln("cadena");

Imprime por consola el valor dado por la cadena, a continuación un ejemplo.

```
imprimirln("Porteros");  
imprimirln("Reporte de");  
imprimirln("Porteros");
```

recalcar que en este caso si existe el salto de línea, por lo tanto, la salida en consola se vería de la siguiente manera:

```
>>>Reporte de  
>>>Porteros
```

Función Conteo();

Imprime por consola la cantidad de registros en el arreglo de registros, como se explicó arriba para todas las funciones se utilizará como prueba el archivo de entrada de ejemplo que se ha venido utilizando, por lo tanto al realizar la función en dicho archivo, como salida en consola, mostraría lo siguiente:

```
conteo();  
-----  
>>>5
```

Recordando que en el archivo efectivamente, venían incluidos un total de 5 registros.

Función promedio("campo");

En esta función mediante un parámetro el cual es una clave en específico se calculará el promedio de datos, por lo tanto en consola mostraría un valor de tipo float.

```
promedio("rating");  
-----  
>>>88.2
```

Función contarsi("campo", valor);

Esta función recibe como parámetro, 2 valores, el primero es una de las claves antes precargadas, mientras que el segundo es un valor entero, esta función es la encargada de contar los datos de una cierta clave si se cumple la condición del segundo parámetro, a continuación un ejemplo:

```
contarsi("rating",99);  
contarsi("rating",79);  
contarsi("rating",96);
```

en este caso vienen 3 instrucciones de esta función, pero pueden venir "n" veces.

```

{"Ter_Stegen", 99, "Alemania"}
{"Keylor_Navas", 88, "Costa_Rica"}
{"Claudio_Bravo", 79, "Chile"}
{"Dibu_Martinez", 96, "Argentina"}
{"Armani", 79, "Argentina"}

```

recordando los registros previos, la consola debería contar registros si y solo si se cumple la condición del parámetro.

para el primer caso la condición es que contará si y solo si el valor de la clave rating es 99 y por lo que se observa, solo existe un valor, para el segundo caso la condición es que contará si y solo si el valor de la clave rating es 79 y por lo que se observa, existen dos valores, y por último el tercer caso contara si y solo si rating es igual a 96, y por lo que se observa, solo existe un caso, entonces la salida quedaría de la siguiente manera:

```

>>>1
>>>2
>>>1

```

Función datos();

Imprime por consola los registros leídos, a continuación se muestra un ejemplo:

```
datos();
```

lo que mostraría en consola:

```

>>>portero >>>rating >>>nacionalidad
>>>Ter_Stegen >>>99 >>>Alemania
>>>Keylor_Navas >>>88 >>>Costa_Rica
>>>Claudio_Bravo >>>79 >>>Chile
>>>Dibu_Martinez >>>96 >>>Argentina
>>>Armani >>>79 >>>Argentina

```

Función sumar("campo");

Imprime en consola la suma todos los valores del campo dado, a continuación un ejemplo:

```
sumar("rating");
```

```
>>>441
```

Función max("campo");

Imprime en consola el mayor valor del campo que viene de parámetro, a continuación un ejemplo:

```
max("rating");
```

```
>>>99
```

Función min("campo");

Imprime en consola el menor valor del campo que viene de parámetro, a continuación un ejemplo:

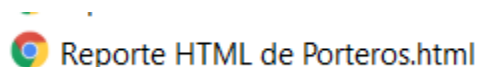
```
min("rating");
```

```
>>>79
```

Función exportarReporte("titulo");

Esta función genera un archivo html, el parámetro es una cadena, y esta servirá como título para el archivo que se generará, a continuación un ejemplo.

```
exportarReporte("Reporte HTML de Porteros");
```



Este reporte incluye todos los datos recolectados en el archivo de entrada, a continuación un ejemplo:

LFP_PROYECTO_2

Reporte HTML de Porteros

portero	rating	nacionalidad
Ter_Stegen	99	Alemania
Keylor_Navas	88	Costa_Rica
Claudio_Bravo	79	Chile
Dibu_Martinez	96	Argentina
Armani	79	Argentina

Interfaz Gráfica

Este proyecto, contiene una interfaz gráfica, la cuál tiene diferentes funciones y componentes, que a continuación se irán desarrollando uno a uno.

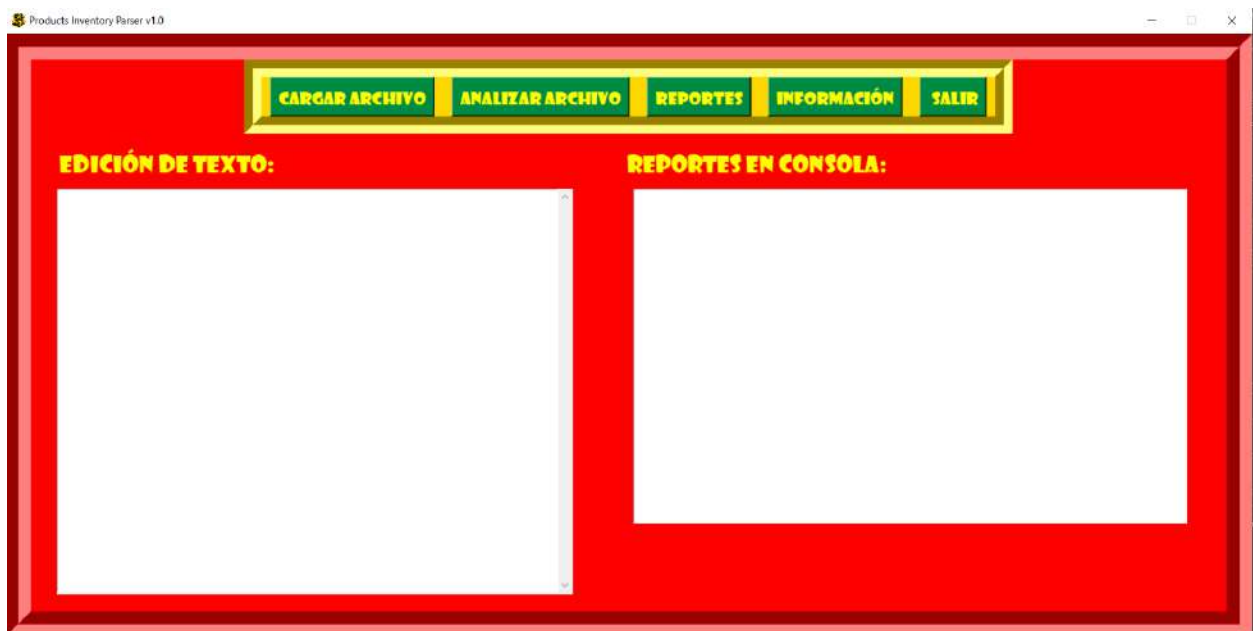
Ventana Principal

Al iniciar la aplicación, esta ventana es lo que usted visualizará, en este punto solo tiene 2 opciones, avanzar con el botón iniciar o puede cerrar la aplicación.



Centro de comandos

Aquí es donde todo ocurre, y donde el usuario puede interactuar con la aplicación.



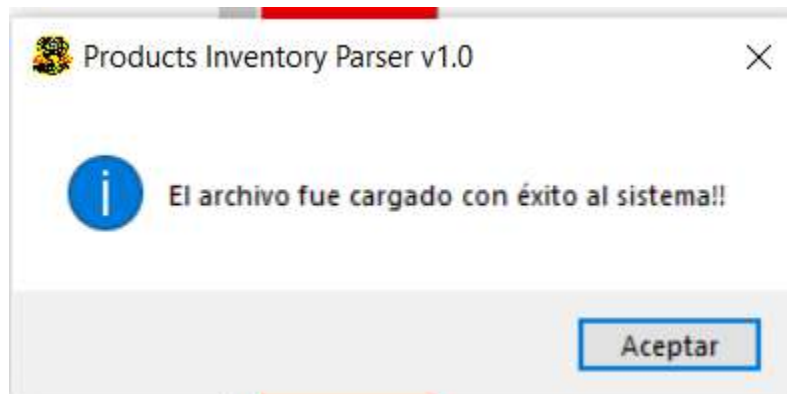
Cargar Archivo

Mediante este botón usted podrá cargar a memoria un archivo con extensión ".lfp", de lo contrario mostrará un mensaje de error, y se le pedirá un nuevo archivo.

En caso de no ser un archivo lfp:



De lo contrario, se le mostrará un mensaje de que el archivo se cargó con éxito:



Luego de que el archivo se cargue a memoria, automáticamente la parte de edición de texto se habilitará, y el usuario podrá tanto visualizar como editar el texto para un posterior análisis.

EDICIÓN DE TEXTO:

```
Claves = [  
    "portero", "rating", "nacionalidad"  
]  
  
Registros = [  
    {"Ter_Stegen", 99, "Alemania"}  
    {"Keylor_Navas", 88, "Costa_Rica"}  
    {"Claudio_Bravo", 79, "Chile"}  
    {"Dibu_Martinez", 96, "Argentina"}  
    {"Armani", 79, "Argentina"}  
]  
  
#Un comentario corto  
"  
Este comentario  
es un poco  
mas largo  
"  
  
imprimir("Reporte de ");  
imprimir("Porteros");  
imprimirln("Reporte de");  
imprimirln("Porteros");  
conteo();  
promedio("rating");  
contarsi("rating",99);  
contarsi("rating",79);
```

Analizar Archivo

Como su nombre lo indica, analiza el archivo, primero con un analizador léxico que recolecta los datos, tokens y errores léxicos, y por último con un analizador sintáctico en busca de errores sintácticos.

Automáticamente luego de analizar, la sección de reportes mostrará toda la información generada, recordando que solamente se puede visualizar, no es permitida la edición en esta sección.

REPORTES EN CONSOLA:

```
>>>Reporte de Porteros
>>>Reporte de
>>>Porteros
>>>5
>>>88.2
>>>1
>>>2
>>>1
>>>portero >>>rating >>>nacionalidad
>>>Ter_Stegen >>>99 >>>Alemania
>>>Keylor_Navas >>>88 >>>Costa_Rica
>>>Claudio_Bravo >>>79 >>>Chile
>>>Dibu_Martinez >>>96 >>>Argentina
>>>Armani >>>79 >>>Argentina
>>>99
>>>79
>>>441
```

Reportes

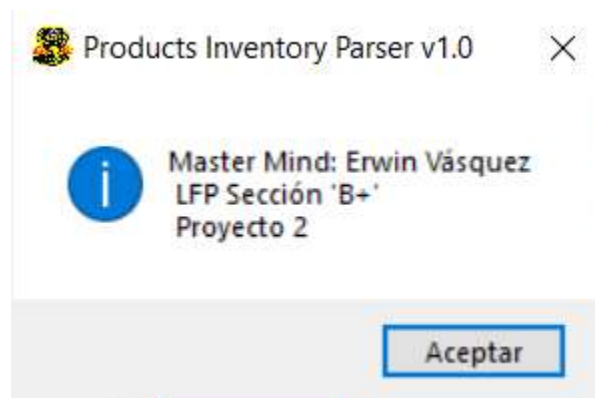
Este botón tiene como finalidad generar el reporte de tokens y el de errores.

REPORTE_DE_TOKENS

TOKEN	LEXEMA	FILA	COLUMNA
CLAVES	Claves	1	7
equals	=	1	8
open_square_bracket	[1	10
double_quotation_mark	"	2	2
Clave	portero	2	10
double_quotation_mark	"	2	11
comma	,	2	12
double_quotation_mark	"	2	14
Clave	rating	2	21
double_quotation_mark	"	2	22

Información

Con este botón se muestra un mensaje con información acerca de la aplicación así como información del fundador de la misma.



Salir

La última función disponible de la interfaz gráfica, y como su nombre lo indica al presionar este botón, se le dará el fin a la ejecución de la aplicación, sin antes mostrar este mensaje.

