

Sección	Catedrático	Tutor académico
A +	Ing. Otto Amilcar Rodríguez Acosta	Fernando Feliciano Chajón del Cid
B +	Ing. David Estuardo Morales Ajcot	Bryan Gustavo López Echeverría
A -	Inga. Damaris Campos de López	César David Juárez González
B -	Inga. Zulma Karina Aguirre Ordoñez	Douglas Omar Arreola Martínez

PROYECTO 2

Objetivos

- Que el estudiante:
- ➔ Desarrolle una solución de software implementando un analizador léxico mediante autómatas.
 - ➔ Desarrolle una solución de software implementando un analizador sintáctico a partir de la construcción de una gramática libre del contexto.
 - ➔ Aplique los conocimientos adquiridos en el laboratorio, del lenguaje de programación Python.
 - ➔ Desarrolle una interfaz gráfica utilizando el lenguaje Python.

Descripción

Cierto medio de comunicación de prestigio posee un [archivo csv](#) con la información de todos los partidos de La Liga, torneo español de fútbol, en este archivo cada línea representa un partido, de cada partido se indica la fecha en que se jugó, la temporada a la que pertenece, la jornada, el equipo local, el equipo visitante, los goles anotados por el equipo local y los goles anotados por el equipo visitante.

El problema que tiene este medio de comunicación es que son más de 14,000 registros en dicho archivo. Por lo cual se le solicita a usted que desarrolle una herramienta para obtener la información del archivo de una manera más sencilla y amigable a la vista. Se ha propuesto la realización de un bot que interactúe con el usuario a través de comandos específicos y muestre en pantalla o a través de archivos HTML los datos que el usuario solicita. El nombre de este proyecto será **La Liga Bot**.

Comandos del lenguaje de La Liga Bot:

*Consideraciones:

1. Las palabras resaltadas con rojo corresponden a palabras reservadas del lenguaje.
2. El lenguaje es case sensitive.
3. Las partes encerradas en corchetes son partes opcionales en el comando, los corchetes NO son parte del lenguaje.
4. En todos los comandos que se mencionan archivos HTML, dichos archivos deben ser abiertos automáticamente al ejecutar el comando.

1. **Resultado de un partido:** Muestra el resultado de un partido, se especifican los equipos y la temporada:

RESULTADO equipo VS equipo TEMPORADA <YYYY-YYYY>

Dónde:

- equipo es una cadena de texto (Encerrada en un par de comillas). Representa el nombre de un equipo de La Liga.
- YYYY es un número entero de 4 dígitos que representa un año.

Ejemplos de uso del comando:

- RESULTADO "Real Madrid" VS "Villarreal" TEMPORADA <2019-2020>
- RESULTADO "Levante" VS "Español" TEMPORADA <2017-2018>

El bot debe responder con el resultado del partido, por ejemplo:

El resultado de este partido fue: Real Madrid 2 - Villarreal 1

2. **Resultados de una jornada:** Muestra en un archivo HTML el resultado de todos los partidos disputados en una jornada, se especifican la jornada y la temporada:

JORNADA número TEMPORADA <YYYY-YYYY> [-f archivo]

Dónde:

- número es un número entero de dos dígitos como máximo.
- YYYY es un número entero de 4 dígitos que representa un año.
- -f es una bandera que indica que a continuación se presenta el nombre del archivo a exportar.
- archivo es un identificador que será el nombre del archivo generado debe iniciar con una letra y contener únicamente letras, números y guiones bajos. No será una ruta.

El nombre del archivo por defecto será jornada.html

Ejemplos de uso del comando:

- JORNADA 12 TEMPORADA <2019-2020>
- JORNADA 1 TEMPORADA <1996-1997> -f jornada1Reporte

El bot debe responder con el siguiente mensaje y luego abrir el archivo:

Generando archivo de resultados jornada 12 temporada 2019-2020

El archivo debe tener todos los partidos de la jornada y por cada partido: Los equipos que se enfrentaron y el resultado final.

3. **Total de goles en una temporada:** Muestra el total de goles anotados por un equipo, se especifica la condición de los goles, el equipo y la temporada:

GOLES condición equipo **TEMPORADA** <YYYY-YYYY>

Dónde:

- **condición** puede tomar tres valores:
 - ◆ **LOCAL** cuenta únicamente los goles anotados en condición de local.
 - ◆ **VISITANTE** cuenta únicamente los goles anotados en condición de visitante.
 - ◆ **TOTAL** cuenta los goles anotados en condición de visitante y de local.
- **equipo** es una cadena de texto (Encerrada en un par de comillas). Representa el nombre de un equipo de La Liga.
- **YYYY** es un número entero de 4 dígitos que representa un año.

Ejemplos de uso del comando:

- GOLES TOTAL "Valencia" TEMPORADA <1998-1999>
- GOLES LOCAL "Zaragoza" TEMPORADA <1998-1999>

El bot debe responder con un mensaje adecuado, por ejemplo:

Los goles anotados por el Valencia en total en la temporada 1998 1999 fueron 34

4. **Tabla general de temporada:** Muestra la clasificación de La Liga (ordenamiento respecto a puntos), se especifica la temporada:

TABLA TEMPORADA <YYYY-YYYY> [-f archivo]

Dónde:

- **YYYY** es un número entero de 4 dígitos que representa un año.
- **-f** es una bandera que indica que a continuación se presenta el nombre del archivo a exportar
- **archivo** es un identificador que será el nombre del archivo generado debe iniciar con una letra y contener únicamente letras, números y guiones bajos. No será una ruta.

El nombre del archivo por defecto será temporada.html

Ejemplos de uso del comando:

- TABLA TEMPORADA <2018-2019>
- TABLA TEMPORADA <1996-1997> -f reporteGlobal1

El bot debe responder con el siguiente mensaje y luego abrir el archivo:

Generando archivo de clasificación de temporada 2018-2019

El archivo HTML debe contener una tabla con todos los equipos y los puntos que consiguieron en la temporada (Cada victoria suma 3 puntos, un empate supone 1 punto y una derrota no otorga puntos).

5. **Temporada de un equipo:** Muestra todos los resultados de un equipo durante una temporada de La Liga, se especifica el equipo, la temporada y opcionalmente un rango de jornadas:

```
PARTIDOS equipo TEMPORADA <YYYY-YYYY> [ -f archivo ] [ -ji número ]  
[ -jf número ]
```

Dónde:

- **equipo** es una cadena de texto (Encerrada en un par de comillas). Representa el nombre de un equipo de La Liga.
- **YYYY** es un número entero de 4 dígitos que representa un año.
- **-f** es una bandera que indica que a continuación se presenta el nombre del archivo a exportar
- **archivo** es un identificador que será el nombre del archivo generado debe iniciar con una letra y contener únicamente letras, números y guiones bajos. No será una ruta.
- **-ji** es una bandera que indica que se incluirán en el archivo los partidos a partir de la jornada indicada (Inclusivo).
- **-jf** es una bandera que indica que se incluirán en el archivo los partidos hasta la jornada indicada (Inclusivo).
- **número** es un número entero de dos dígitos como máximo, representa la jornada.

Si no se incluyen las banderas -ji y -jf se deben incluir todas las jornadas. El nombre del archivo por defecto será partidos.html

Ejemplos de uso del comando:

- PARTIDOS "Real Madrid" TEMPORADA <1999-2000> -ji 1 -jf 18
- PARTIDOS "Español" TEMPORADA <1999-2000> -f reporteEspanol

El bot debe responder con el siguiente mensaje y luego abrir el archivo:
Generando archivo de resultados de temporada 1999-2000 del Real Madrid

El archivo HTML debe contener una tabla con todos los partidos disputados por el equipo, la jornada en que se disputó y el resultado.

6. **TOP de equipos:** Muestra el top (superior o inferior) de los equipos clasificados según los puntos conseguidos.

```
TOP condición TEMPORADA <YYYY-YYYY> [ -n número ]
```

Dónde:

- **condición** puede tomar tres valores:
 - ◆ **SUPERIOR** muestra los mejores clasificados.
 - ◆ **INFERIOR** muestra los peores clasificados.
- **YYYY** es un número entero de 4 dígitos que representa un año.
- **-n** es una bandera que indica que se incluirán en el top los mejores o peores clasificados indicados por el número.

→ **número** es un número entero de dos dígitos como máximo, representa el número de equipos a mostrar.

Si la bandera -n no viene, se muestran 5 equipos por defecto.

Ejemplos de uso del comando:

- TOP SUPERIOR TEMPORADA <2000-2001>
- TOP INFERIOR TEMPORADA <1999-2000> -n 5

El bot debe responder con un mensaje adecuado, por ejemplo:

El top superior de la temporada 2000-2001 fue:

1. Equipo 1
2. Equipo 2
3. Equipo 3
4. Equipo 4
5. Equipo 5

7. **Salida del bot:** Finaliza la ejecución del programa:

ADIOS

Ejemplos de uso del comando:

- ADIOS

El bot debe responder con un mensaje y terminar la ejecución del programa:

ADIOS

Funciones del sistema

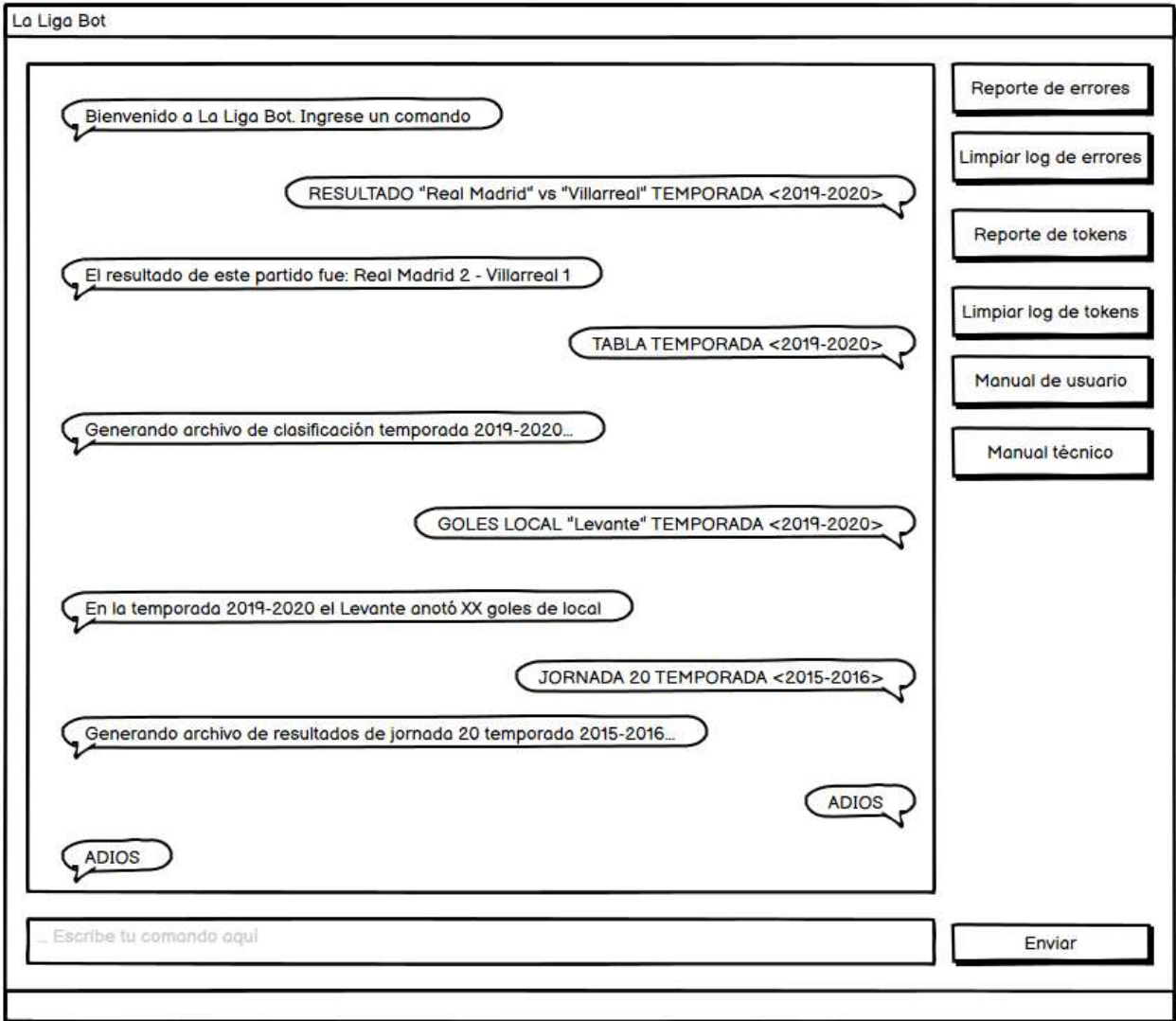
La aplicación cuenta con un interfaz gráfica, en la cual se poseen las siguientes opciones:

1. **Enviar:** Analiza léxicamente el texto escrito en la caja de texto del bot mediante un **autómata definido e implementado por el estudiante** y luego hace el análisis sintáctico utilizando como base una **gramática libre de contexto definida e implementada por el estudiante**, si todo está escrito con la sintaxis correcta, despliega los resultados o mensajes correspondientes en el área de consola de la aplicación. Todos los tokens y errores encontrados en un comando deben de ir acumulándose para ser mostrados en los reportes.

Si un comando posee un error el bot debe de intentar hacer la ejecución del mismo a menos que sea un error del cual no se pueda recuperar.
2. **Reporte de errores:** Se debe de escribir en un archivo HTML los errores léxicos y sintácticos encontrados durante la ejecución del programa, la manera en que los datos son mostrados deben de ser agradables al usuario. Los reportes deben abrirse automáticamente al seleccionar esta opción.
3. **Limpiar log de errores:** Elimina todos los errores encontrados en la aplicación hasta el momento de ser seleccionada esta opción.

- 4. **Reporte de tokens:** Se debe de escribir en un archivo HTML los tokens encontrados durante la ejecución del programa, la manera en que los datos son mostrados deben de ser agradables al usuario. Los reportes deben abrirse automáticamente al seleccionar esta opción.
- 5. **Limpiar log de tokens:** Elimina todos los tokens encontrados en la aplicación hasta el momento de ser seleccionada esta opción.
- 6. **Manual de usuario:** Abre automáticamente el manual de usuario de la aplicación.
- 7. **Manual técnico:** Abre automáticamente el manual técnico de la aplicación.

Interfaz sugerida



Especificación de los reportes

- La aplicación debe generar reportes en formato HTML, los reportes son:
- 1. Reporte de tokens: Se debe generar una tabla con los tokens que se identificaron. Se debe indicar el token, lexema y columna del token leído.
 - 2. Reporte de errores léxicos: Se debe generar una tabla con los errores léxicos detectados durante el análisis, indicando la columna y caracter que provocó el error.
 - 3. Reporte de errores sintácticos: Se debe generar una tabla con los errores sintácticos detectados durante el análisis, indicando la columna y token que provocó el error.

Entregables

- Manual de usuario.
- Manual técnico, en este se debe incluir además del contenido habitual lo siguiente:
 - ◆ Tabla de tokens, en la cual se debe indicar el patrón (Expresión regular) que define a cada token.
 - ◆ Autómata finito determinista utilizado en el análisis léxico.
 - ◆ Gramática libre de contexto utilizada para el análisis sintáctico.
- Código fuente.

Consideraciones importantes

- El proyecto debe desarrollarse individualmente.
- Se debe utilizar el lenguaje de programación Python
- Se deberá crear un repositorio en Github con el nombre **LFP_PY2_carnet**. Deberán agregar al auxiliar de su sección de laboratorio como colaborador del mismo.
- La entrega se realizará en la plataforma UEDI, se deberá entregar el enlace del repositorio y como respaldo un archivo comprimido zip con el siguiente nombre: **[LFP]Proyecto2_<<Carnet>>.zip**. Tomar en cuenta que el único medio de entrega es la plataforma UEDI.
- La calificación se realizará en línea y se grabará, esto para que quede constancia de la forma en que se calificó y como soporte en la toma de decisiones en reclamos por parte del alumno si se presenta el caso.
- La calificación es personal con una duración máxima de 30 minutos, en el horario posteriormente convenido. Durante la calificación el estudiante no podrá modificar el archivo de entrada ni el código fuente de su aplicación.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación para ceder su lugar a otro estudiante, en caso contrario el estudiante solo obtendrá el 80% de su nota obtenida.
- No se dará prórroga para la entrega del proyecto.
- **COPIA PARCIAL O TOTAL DE LA PRÁCTICA TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ AL CATEDRÁTICO DEL CURSO Y POSTERIORMENTE SI SE REQUIERE A LA ESCUELA DE SISTEMAS PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES.**
- **Cualquier librería que se desee usar debe ser consultada previamente con los auxiliares**

Fecha de entrega del proyecto: 28 de abril de 2022 antes de las 23:59 horas.