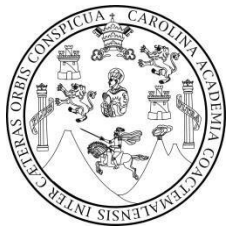
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACIÓN



Sección	Catedrático	Tutor Académico
P	Ing. Kevin Adiel Lajpop Ajpacaja	Elder Anibal Pum Rojas

Enunciado Proyecto #1

Objetivos:

- Implementar un software en Python que permitan plasmar los conocimientos sobre lenguajes regulares, adquiridos en la unidad 3 del programa del curso de lenguajes formales y de programación.
- Dar continuidad a la implementación de soluciones de software empleando paradigmas de programación.
- Utilizar la herramienta de Graphviz para la creación de autómatas.

Descripción:

Con el fin de entender y profundizar en el tema de gramáticas regulares, se le solicita que realice un programa en Python en el cual se puedan definir Gramáticas Regulares (GR),

generar Autómatas Finitos Deterministas (AFD) y también Autómatas Finitos No Deterministas (AFN), así como la evaluación de cadenas válidas para las gramáticas definidas, para esto se le pide que utilice paradigmas de programación.

El programa deberá ser desarrollado de forma visual utilizando la librería Tkinter para la creación de las distintas ventanas las cuales se podrá acceder dependiendo de lo que el usuario desee realizar y que serán descritos más adelante, también se tendrá una sección de reportes en la cual se mostrarán todos los detalles de las gramáticas generadas, el programa será capaz realizar grafos utilizando la herramienta Graphviz para un mejor entendimiento de lo que se realizó.

Una gramática regular nos sirve para validar cadenas y sus producciones las podemos llevar a un AFD y también a un AFN, estos procesos se detallarán más adelante.

Las gramáticas podrán ser creadas desde la aplicación o bien, por medio de una archivo de texto con extensión .grm que contendrá las producciones, los terminales, los no terminales y la producción inicial. La estructura del archivo se detallará en el apartado de carga masiva. Al igual que en la práctica no será necesario el uso de un analizador para leer el archivo pues este vendrá sin errores de sintaxis y con un patrón definido.

Tomar en cuenta que se realizará una revisión de código para verificar los paradigmas utilizados los cuales deberá ser capaz de explicar para comprobar la autoría de su aplicación.

El programa podrá empezar tanto por la creación de una gramática, para generar su respectivo AFD/AFN, así como por la creación de un AFD/AFN, y generar su gramática correspondiente. La notación para AFD/AFN y GR será detallado más adelante, recordar que debe de incluir todos los elementos de estos para cuando realice los reportes.

CARACTERÍSTICAS DEL PROGRAMA

Pantalla Inicial:

En esta pantalla deberá de mostrar el nombre del curso, sección, carné y nombre. También deberá de mostrar un menú de botones donde se puedan seleccionar las siguientes funciones:

- **AFN:** Mostrará el módulo de AFN (Se detallará más adelante).
- **AFD**: Mostrará el módulo de AFD (Se detallará más adelante).
- **OE**: Mostrará el módulo de Optimización de Estados (Se detallará más adelante).
- **Cargar Archivos**: Mostrará el módulo para cargar archivos (Se detallará más adelante).
- **Salir**: Detendrá el flujo de la aplicación.

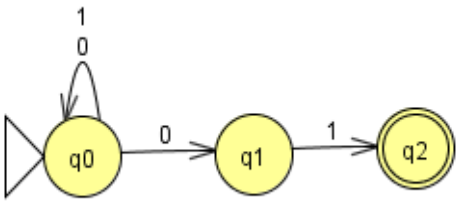
Módulo AFN

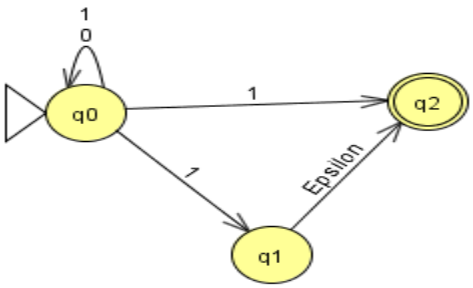
En esta pantalla deberá de mostrar el nombre del curso, sección, carné y nombre. También deberá de mostrar un menú de botones donde se puedan seleccionar las siguientes funciones:

- **Crear AFN:** Al seleccionar esta opción deberá de solicitar los siguientes datos para la creación de este.
    - **Nombre:** Será utilizado para diferenciarlo de los demás AFNs.
    - **Estados:** Se le solicitará los estados que desea crear, estos serán ingresados de la siguiente forma: A;B;C;D (Separados por punto y coma).
    - **Alfabeto:** Se solicitará el alfabeto del lenguaje, cuidando que no se repitan símbolos y que no se utilicen símbolos que fueron declarados **Estados**.
    - **Estado Inicial:** Se solicitará el estado inicial, este debe de existir en los estados declarados, de lo contrario se deberá de indicar el error y solicitar corrección.
    - **Estados de Aceptación:** Se solicitarán los estados de aceptación, serán ingresados de la siguiente forma: A;B;C (Separados por punto y coma), estos deben de existir en los estados declarados, de lo contrario deberá de indicar el error y solicitar corrección.

- **Transiciones:** Deberá de específicar cada una de las transiciones del AFN a través del siguiente formato: Estado Origen, Símbolo de Entrada, Estado Destino; Estado Origen, Símbolo de Entrada, Estado Destino… (Sucesivamente).

Ejemplo de Transiciones:





Consideraciones:

Se deberá verificar que las transiciones ingresadas correspondan a un AFN, por ejemplo, permitir transiciones con Epsilon entre los diferentes estados. Tomar en cuenta las reglas de creación del AFN para tener mayor claridad a la hora de registrar un nuevo AFN.

Deberá de utilizar el **MÉTODO DE THOMPSON** en la lógica de su proyecto a la hora de crear un AFN.

Al presionar el botón “Aceptar” se deberá de guardar el AFN ingresado tomando en cuenta los puntos detallados anteriormente. Deberá de permanecer en la misma pantalla e indicar si se agregó correctamente.

Esta pantalla deberá de tener la lógica para regresar al menú anterior.

- **Evaluar Cadena:** Al seleccionar esta opción se le solicitará al usuario que seleccione los autómatas disponibles y seguidamente mostrar el menú de opciones en las cuales podrá seleccionar las siguientes:
  - **Solo Validar:** Se le solicitará y se dará un mensaje con cadena válida o inválida, según sea el caso.
  - **Ruta:** Se solicitará una cadena y de ser válida deberá de mostrar la ruta que se siguió para poder validarla, si no es válida mostrará error.

Consideraciones:
Al finalizar la evaluación debe retornar al menú del módulo AFN.

La forma en la que se deberá de mostrar la ruta tomada por el analizador para la cadena deberá de ser mostrada en formato pdf utilizando la herramienta graphviz.

- **Generar Reporte AFN:** Esta opción deberá mostrar en pantalla todos los AFN's que se encuentran actualmente en el sistema, el usuario deberá seleccionar uno para generar el reporte. El reporte deberá contener los siguientes datos:
    1. Nombre
    2. Estados
    3. Alfabeto
    4. Estados de Aceptación
    5. Estado Inicial
    6. Transiciones
    7. Una cadena mínima válida
    8. Grafo del AFN generado con graphviz

Nombre: AFN de ejemplo

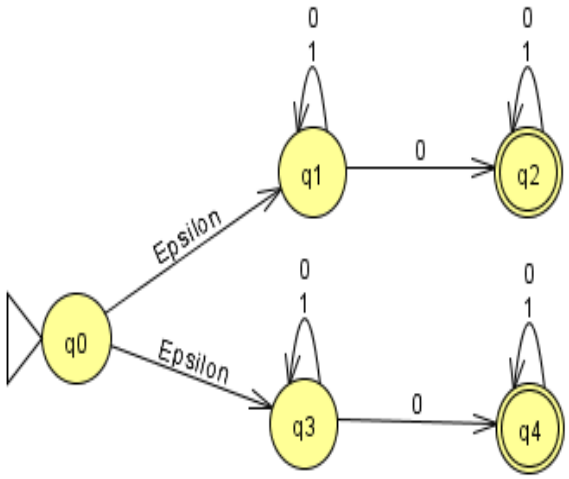
Estados: `q0,q1,q2,q3,q4`
Alfabeto: `0,1`
Estados de aceptación: `q2,q4`

Estado inicial: q0

Transiciones:
q0,Epsilon;q1
q0,Epsilon;q4
q1,1;q1
q1,0;q1
q1,0;q2
q2,1;q2
q2,0;q2
q3,1;q3
q3,0;q3
q3,0;q4
q4,1;q4
q4,0;q4

Cadena válida de ejemplo: 0101

Grafo:



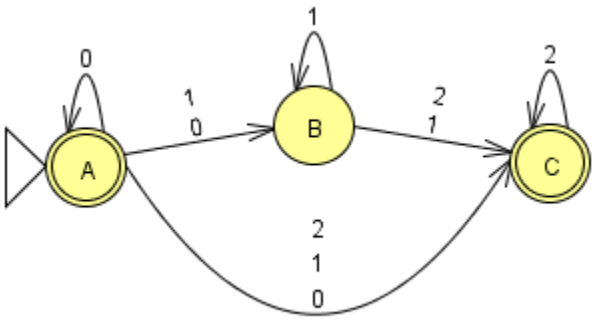
Consideraciones:
La cadena debe ser una cadena válida, en caso de que se puedan generar varias cadenas mínimas de un mismo tamaño queda a discreción del estudiante cuál presentar.
La disposición de los elementos del ejemplo de reporte es solamente una sugerencia, queda a discreción del alumno su presentación, el reporte debe ser legible y el grafo debe permitir la identificación de los distintos componentes que conforman el AFN.
Al finalizar la generación del reporte, el flujo del programa debe retornar al menú del módulo AFN.

- **Ayuda:** Mostrará una pantalla explicando qué es un AFN y dar como mínimo un ejemplo de AFN.

  Ejemplo:

Nombre: AFN2
Estados: A, B, C
Alfabeto: 0, 1
Estados de Aceptación: A, C
Estado Inicial: A
Transiciones:

A,0;A	A,2;C
A,0;B	B,1;B
A,1;B	B,1;C
A,0;C	B,2;C
A,1;C	C,2;C



Módulo AFD:

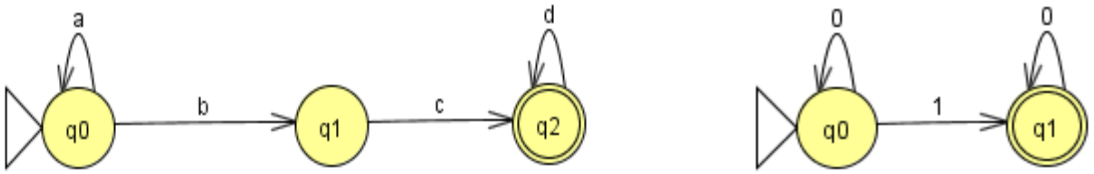
Al ingresar a esta pantalla se deberá de mostrar una serie de botones donde se podrán seleccionar las siguientes funciones:

- **Crear AFD:** Al seleccionar esta opción deberá de solicitar los siguientes datos para la creación de este.

  - **Nombre**: Será utilizado para diferenciarlo de los demás AFDs.
  - **Estados**: Se le solicitarán los estados que desea crear, estos serán ingresados de la siguiente forma: A;B;C;D (Separados por punto y coma).
  - **Alfabeto:** Se solicitará el alfabeto del lenguaje, cuidando que no se repitan símbolos y que no se utilicen símbolos que fueron declarados **Estados**.

- **Estado Inicial**: Se solicitará el estado inicial, este debe de existir en los estados declarados, de lo contrario se deberá de indicar el error y solicitar corrección.
- **Estados de Aceptación:** Se solicitarán los estados de aceptación, serán ingresados de la siguiente forma: A;B;C (Separados por punto y coma), estos deben de existir en los estados declarados, de lo contrario deberá de indicar el error y solicitar corrección.
- **Transiciones:** Deberá de especificar cada una de las transiciones del AFD a través del siguiente formato:
  Estado origen, símbolo de entrada, estado destino; Estado origen, símbolo entrada, estado destino… (Sucesivamente)

Ejemplos de transiciones:



Consideraciones:

Se deberá verificar que las transiciones ingresadas correspondan a un AFD, por ejemplo, si se intenta hacer más de una transición de un estado con en el mismo símbolo, se debe mostrar error.

Deberá de utilizar el **MÉTODO DE SUBCONJUNTOS** en la lógica de su proyecto a la hora de crear un AFD.

Al presionar el botón “Aceptar” se deberá de guardar el AFD ingresado tomando en cuenta los puntos detallados anteriormente. Deberá de permanecer en la misma pantalla e indicar si se agregó correctamente.

Esta pantalla deberá de tener la lógica para regresar al menú anterior.

- **Evaluar Cadena**: Al seleccionar esta opción se le solicitará al usuario que seleccione los autómatas disponibles y seguidamente mostrar el menú de opciones en las cuales podrá seleccionar las siguientes:

    - **Solo Validar:** Se solicitará una cadena y se dará un mensaje con cadena válida o inválida según sea el caso.
    - **Ruta:** Se solicitará una cadena y ser válida deberá de mostrar la ruta que se siguió para poder validarla, si no es válida mostrará error.

Consideraciones:
Al finalizar la evaluación debe retornar al menú del módulo AFD.

La forma en la que se deberá de mostrar la ruta tomada por el analizador para la cadena deberá de ser mostrada en formato pdf utilizando la herramienta graphviz.

- **Generar reporte AFD:** Esta opción deberá mostrar en pantalla todos los AFD's que se encuentran actualmente en el sistema, el usuario deberá seleccionar uno para generar el reporte. El reporte deberá contener los siguientes datos:

    1. Nombre
    2. Estados
    3. Alfabeto
    4. Estados de aceptación
    5. Estado inicial
    6. Transiciones
    7. Una cadena mínima válida
    8. Grafo del AFD generado con graphviz

Ejemplo:

Nombre: AFN de ejemplo

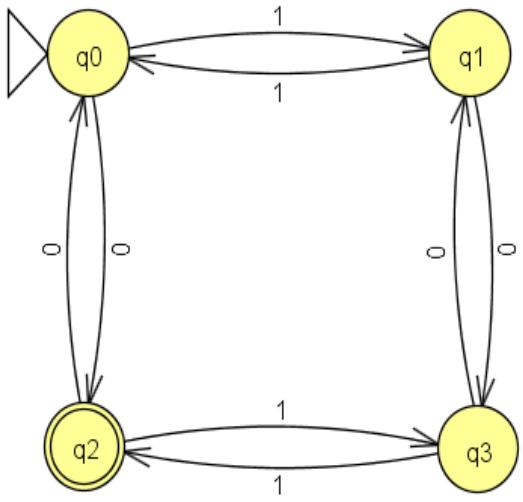
Estados: `q0,q1,q2,q3`
Alfabeto: `0,1`
Estados de aceptación: `q2`

Estado inicial: A

Transiciones:
q0,1;q1
q0,0;q2
q1,1;q0
q1,0;q3
q2,1;q3
q2,0;q0
q3,1;q2
q3,0;q1

Cadena válida de ejemplo: 011

Grafo:

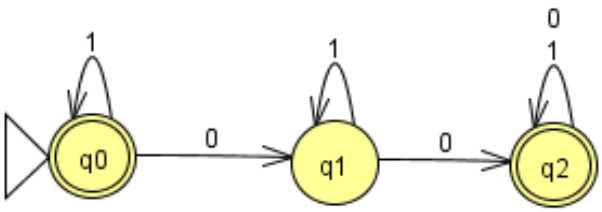


Consideraciones:
La cadena debe ser una cadena válida, en caso de que se puedan generar varias cadenas mínimas de un mismo tamaño queda a discreción del estudiante cuál presentar.
La disposición de los elementos del ejemplo de reporte es solamente una sugerencia, queda a discreción del alumno su presentación, el reporte debe ser legible y el grafo debe permitir la identificación de los distintos componentes que conforman el AFD.
Al finalizar la generación del reporte, el flujo del programa debe retornar al menú del módulo AFD.

- **Ayuda:** Mostrará una pantalla explicando qué es un AFD y dar como mínimo un ejemplo de AFD.

  Ejemplo:

Nombre: AFD2
Estados: A, B, C
Alfabeto: 0, 1
Estados de Aceptación: A, C
Estado Inicial: A
Transiciones:
A,1;A
A,0;B
B,1;B
C,1;C
C,0;C
B,0;C



Módulo OE:

Al ingresar a esta pantalla se deberá de mostrar una serie de botones donde se podrán seleccionar las siguientes funciones:

- **Seleccionar AFD:** Al seleccionar esta opción deberá de solicitar los siguientes datos para la creación del nuevo AFD optimizado.
    - **Nombre AFD a Optimizar**: Será utilizado para seleccionar un AFD creado previamente.
    - **Nuevo Nombre para AFD Optimizado**: Será el nombre del nuevo AFD optimizado a partir del AFD original. Si no se ingresa nombre, debería por defecto dar el nombre del AFD original seguido por un apostrofe. Ejemplo: (AutómataOriginal')

Consideraciones:
La optimización de estados es únicamente aplicable a **Autómatas Finitos Deterministas**, ya que para que funcione para un AFN este tendría que pasarse a AFD.
Si en dado caso se trata de seleccionar un AFN, debería de mostrar error de que este módulo únicamente es aplicable a AFD válidos.

Al finalizar la optimización del AFD, el flujo del programa debe retornar al menú del módulo OE, además de mostrar un mensaje de éxito.

- **Generar reporte OE:** Esta opción deberá mostrar en pantalla todos los AFD's optimizados que se encuentran actualmente en el sistema, el usuario deberá seleccionar uno para generar el reporte. El reporte deberá contener los siguientes datos:

1. Nombre
2. No terminales
3. Terminales
4. No Terminal Inicial
5. Producciones utilizando la notación que se detalla a continuación
6. Generar un AFD a partir de la Gramática y mostrar su grafo
7. Comparativa entre el AFD original y el AFD optimizado

Ejemplo:

Nombre: Gramática de ejemplo

No terminales: `A,B,C,D`

Terminales: `0,1`

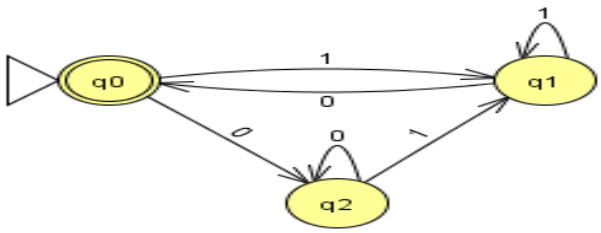
No terminal inicial: `A`

Producciones:

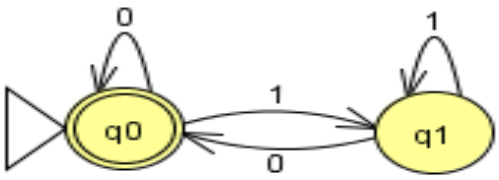
```
q0 > 0 q0
   | 1 q1
   | $
q1 > 0 q0
   | 1 q1
```

Cadena de ejemplo: 011

Grafo antes de Minimizar:



Grafo después de Minimizar:

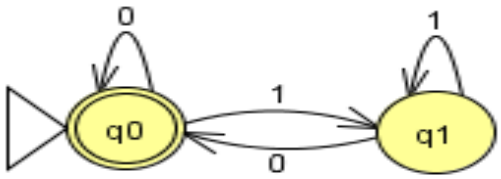


Consideraciones:
La cadena debe ser una cadena válida, en caso de que se puedan generar varias cadenas mínimas de un mismo tamaño queda a discreción del estudiante cuál utilizar.
La disposición de los elementos del ejemplo de reporte es solamente una sugerencia, queda a discreción del alumno su presentación, el reporte debe ser legible y el grafo debe permitir la identificación de los distintos componentes que conforman el AFD. La sección de Producciones debe de tener las producciones del **AFD MINIMIZADO,** no del AFD original.
Al finalizar la generación del reporte, el flujo del programa debe retornar al menú del módulo OE.

- **Ayuda:** Mostrará una pantalla explicando qué es un OE y dar como mínimo un ejemplo de OE.

  Ejemplo:

Nombre: AFD2
Estados: q0, q1
Alfabeto: 0, 1
Estados de Aceptación: q0
Estado Inicial: q0
Transiciones:
q0,0;q0
q0,1;q1
q1,0;q0
q1,1;q1



Un AFD minimizado es la versión simplificada de lo que fue un AFD con estados redundantes entre sí. Utilizando algoritmos es posible encontrar estados equivalentes para formar un AFD mínimo.

Módulo Cargar Archivo

Al ingresar a esta pantalla se deberán de mostrar las opciones para qué tipo de archivo se desea cargar. Las opciones disponibles son las siguientes:

- **Cargar AFD:** Esta opción permitirá cargar 1 o más AFDs. El archivo será de texto plano con extensión afd, el contenido del archivo estará estructurado de la siguiente manera:

Ubicación	*Valor*	*Ejemplo de archivo*
Línea 0	Nombre AFD	`AFD_1`
Línea 1	Estados	`A,B,C,D`
Línea 2	Alfabeto	`0,1`
Línea 3	Estado inicial	`A`
Línea 4	Estados de aceptación	`C`
Línea 5	Transición 1	`A,1;B`
Línea 6	Transición 2	`A,0;C`
Línea 7	Transición 3	`B,1;A`
Línea 8	Transición 4	`B,0;D`
Línea 9	Transición 5	`C,1;D`
Línea 10	Transición 6	`C,0;A`
Línea 11	Transición 7	`D,1;C`
Línea 12	Transición 8	`D,0;B`
Línea 13	*Transición 9*	`A,1;B`
Línea 14	%	
Línea 15	Nombre AFD	`AFD_2`
Línea 16	Estados	`A,x;C`
…	…	…

Consideraciones:

Los archivos de entrada estarán estructurados de la forma descrita anteriormente. En ningún momento tendrán errores de sintaxis.
Las líneas de: Estados, Alfabeto y Estados de aceptación tendrán sus valores separados por comas cuando sea más de un dato.
Los datos de los AFD vendrán siempre en el mismo orden.
La línea con el símbolo **%** indicará el final de los datos de un AFD/AFN e inmediatamente iniciarán los datosdel siguiente AFD.
Cada AFD dentro del sistema tomará el nombre que se indica al inicio de su definición.
Al finalizar la carga del archivo el sistema debe retornar al menú principal.

- **Cargar AFN:** Esta opción permitirá cargar 1 o más AFNs. El archivo será de texto plano con extensión afn, el contenido del archivo estará estructurado de la siguiente manera:

Ubicación	*Valor*	*Ejemplo de archivo*
Línea 0	Nombre AFN	AFN_1
Línea 1	Estados	A,B,C,D
Línea 2	Alfabeto	0,1
Línea 3	Estado inicial	A
Línea 4	Estados de aceptación	C
Línea 5	Transición 1	A,1;B
Línea 6	Transición 2	A,0;C
Línea 7	Transición 3	B,1;A
Línea 8	Transición 4	B,0;D
Línea 9	Transición 5	C,1;D
Línea 10	Transición 6	C,0;A
Línea 11	Transición 7	D,1;C
Línea 12	Transición 8	D,0;B
Línea 13	*Transición 9*	A,1;B
Línea 14	%	
Línea 15	Nombre AFD	AFD_2
Línea 16	Estados	A,x;C
…	…	…

Consideraciones:

Los archivos de entrada estarán estructurados de la forma descrita anteriormente. En ningún momento tendrán errores de sintaxis.
Las líneas de: Estados, Alfabeto y Estados de aceptación tendrán sus valores separados por comas cuando sea más de un dato.
Los datos de los AFN vendrán siempre en el mismo orden.
La línea con el símbolo **%** indicará el final de los datos de un AFN e inmediatamente iniciarán los datos del siguiente AFN.
Cada AFN dentro del sistema tomará el nombre que se indica al inicio de su definición.
Al finalizar la carga del archivo el sistema debe retornar al menú principal.

Notas Importantes:

- El proyecto deberá de desarrollarse de forma individual.
- El proyecto deberá de ser desarrollado utilizando Python y Graphviz
- La entrega se realizará en la plataforma UEDI.
- Deberá de crear un repositorio en Github.com agregando al usuario **elderpum**
- No será permitida la modificación de archivos de entrada durante la calificación.
- La calificación deberá de ser en línea y se estará grabando la reunión para tener un respaldo de la forma en que se procedió.
- La calificación del proyecto será personal y durará como máximo 30 minutos, en un horario que será establecido posteriormente.
- No se dará prórroga para la entrega del proyecto.
- **Copia parcial o total del proyecto tendrá una nota de 0 puntos, y se notificará a la Escuela de Sistemas para que se apliquen las sanciones correspondientes.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por el cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 16 de junio de 2023 antes de las 23:59.