

## Actividad 4

### Implementación de los conceptos de la teoría de los lenguajes formales

#### OBJETIVO:

- Comprender e Implementar los conceptos de la teoría de los lenguajes formales en un programa de computadoras.
- Mejorar las competencias en el desarrollo de software.

#### TAREA:

El estudiante deberá construir o desarrollar un programa usando el lenguaje Python.

Este programa debe tener los siguientes requisitos funcionales:

- **Sobre los alfabetos:**
  - a. Debe permitir que el usuario ingrese como mínimo dos alfabetos. La interfaz de usuario mínimo debería estar por línea de comandos.
  - b. **Calcular la unión de alfabetos.**
  - c. **Calcular la diferencia entre alfabetos.**
  - d. **Calcular la intersección entre alfabetos.**
  - e. **Debe calcular la cerradura de estrella del alfabeto**, como es un conjunto infinito debe permitirle al usuario ingresar el número de símbolos que generará la cerradura de estrella. Estas palabras se generarán de forma aleatoria.
- **Sobre los lenguajes:**
  - a. A partir de los alfabetos se generarán palabras aleatorias que conforman dos lenguajes. El número de palabras de los dos lenguajes será determinado por el usuario mediante una interfaz.
  - b. **Calcular la unión entre lenguajes.**
  - c. **Calcular la diferencia entre lenguajes.**
  - d. **Calcular la intersección entre lenguajes.**
  - e. **Calcular la concatenación entre lenguajes.**
  - f. **Calcular la potencia de un lenguaje. El usuario determinará la potencia del lenguaje.**
  - g. **Calcular la inversa de un lenguaje.**
  - h. **Calcular la cardinalidad de un lenguaje.**
- **Requisito No funcional:**
  - a. Debe reutilizar una pieza de código que permita calcular la unión, la diferencia y la intersección.
  - b. La interfaz de usuario debe permitir escribir el conjunto de forma literal. Ejemplo: **NombrePrograma** (Parámetros)  $A=\{a,b,cd\}$   $B=\{rf, ac, de\}$

**Nota 2: El código fuente del programa debe publicarlo en un sistema de control versiones como por ejemplo GIT. En el sistema debe registrarse el código que aportó cada programador).No acepta reportar un solo push.**

**Grupo: Máximo de 3 estudiantes.**

**FECHA MÁXIMA DE ENTREGA:**

El 12 septiembre entrega del trabajo.

**PROCESO:**

El programa debe construirse usando el lenguajes Python. El estudiante deberá entregar el código fuente del programa en alguna plataforma de control de código fuente como GitHub entre otras.

**RECURSOS:**

- Internet.
- Lenguaje Python última versión.
- Diapositiva del curso.
- Sistemas de control de código fuente.

**EVALUACIÓN:**

**Sobresaliente:70 puntos**

- El programa funciona correctamente.
- Entrega el trabajo en la fecha establecida.
- El programa tiene interfaz de consola. Debe cumplir con su requisito no funcional.
- El código debe estar en Inglés.
- Aplicar la escritura Snake case para código Python.
- Reutiliza el código para la unión, intersección y la diferencia.
- Se publica en una plataforma de control de versiones. (Git) y cada programador debe tener un registro de su código. Deben existir varios reportes en Git.
- **La interfaz de usuario es fácil e intuitiva.**
- El estudiante es capaz de modificar el programa fuente de acuerdo a lo que le pida el docente. El docente determinará las veces que crea necesaria para asegurarse que el estudiante fue realmente quien programó el software. Asegurarse de llevar el IDE donde construyó el programa para que pueda hacerles las modificaciones.

**Destacado:50 puntos.**

- El programa funciona correctamente.
- Se publica en una plataforma de control de versiones. (Git) y cada programador debe tener un registro de su código. Deben existir varios reportes en Git.
- Entrega el trabajo en la fecha establecida.
- Se publica en una plataforma de control de versiones. (Git) y cada programador debe tener un registro de su código. Deben existir varios reportes en Git.
- El estudiante es capaz de modificar el programa fuente de acuerdo a lo que le pida el docente. El docente determinará las veces que crea necesaria para asegurarse que el estudiante fue realmente quien programó el software. Asegurarse de llevar el IDE donde construyó el programa para que pueda hacerles las modificaciones. .

**Satisfactorio: 35 puntos.**

- Algunas funcionalidades están correctas.
- Entrega el trabajo en la fecha establecida.
- El programa tiene una interfaz de usuario.

**Básico: 25 puntos.**

- Algunas funcionalidades están correctas.
- El programa tiene una interfaz de usuario.

**No cumplimiento: 5 puntos.**

- No funciona.
- No entrega el trabajo en la fecha establecida.
- El estudiante no es capaz de modificar el programa fuente de acuerdo a lo que le pida el docente. El docente determinará las veces que crea necesaria para asegurarse que el estudiante fue realmente quien programó el software.