

## **Propuesta de Proyecto: Desarrollo de Software para la Validación de Expresiones Regulares**

### **Objetivo del Proyecto:**

Desarrollar un software capaz de validar expresiones regulares para un lenguaje dado, demostrando así la aplicación práctica de los conceptos de teoría de lenguajes formales y autómatas.

### **Descripción General:**

El proyecto consiste en crear una herramienta de software que permita a los usuarios ingresar una expresión regular y un conjunto de cadenas, y que valide si las cadenas cumplen con la expresión regular proporcionada. El software debe ser capaz de manejar diferentes operadores y construcciones de expresiones regulares y ofrecer resultados claros sobre si las cadenas son aceptadas o rechazadas.

### **Grupos:**

Cantidad máxima de estudiantes tres.

### **Componentes del Proyecto:**

#### **1. Interfaz de Usuario:**

- Entrada de Expresión Regular: Un campo donde los usuarios puedan ingresar una expresión regular.
- Entrada de Cadenas: Un campo o área donde se puedan ingresar múltiples cadenas para validar.
- Botón de Validación: Un botón que inicie el proceso de validación.
- Resultado: Un área para mostrar si cada cadena es aceptada o rechazada por la expresión regular.

#### **2. Módulo de Procesamiento:**

- Analizador de Expresiones Regulares: Convierte la expresión regular en un formato que pueda ser procesado por el software, por ejemplo, un autómata finito.
- Motor de Validación: Implementa el algoritmo de validación basado en el tipo de autómata o técnica seleccionada (por ejemplo, autómata finito determinista o no determinista).
- Manejo de Errores: Gestiona expresiones regulares inválidas o mal formadas y proporciona retroalimentación adecuada.

#### **3. Funcionalidades Adicionales (opcional):**

- Resaltado de Sintaxis: Colorear diferentes partes de la expresión regular para facilitar la lectura.
- Explicación de la Expresión Regular: Ofrecer una descripción de lo que hace la expresión regular en lenguaje sencillo.
- Pruebas Unitarias: Incluye pruebas automáticas para verificar que el software maneje correctamente una variedad de casos.

### **Requisitos Técnicos:**

- Lenguaje de Programación: Python.
- Documentación: Incluir manual de usuario, guía de instalación, y documentación técnica del código.

### **Metodología:**

1. Investigación: Estudiar la teoría detrás de las expresiones regulares y los autómatas finitos.

2. Diseño: Crear un diseño detallado de la interfaz de usuario y del módulo de procesamiento.
3. Desarrollo: Implementar el software siguiendo el diseño.
4. Pruebas: Verificar la funcionalidad del software con diferentes expresiones regulares y conjuntos de cadenas.
5. Documentación: Escribir documentación clara y útil para los usuarios y otros desarrolladores.

**Evaluación del Proyecto:**

- Funcionalidad: El software debe validar correctamente las expresiones regulares para diferentes casos de prueba.
- Usabilidad: La interfaz debe ser intuitiva y fácil de usar.
- Documentación: La documentación debe ser completa y útil para usuarios y desarrolladores.
- Código: El código debe estar bien organizado, comentado y seguir las mejores prácticas de programación.

Entregas:

Fecha de entrega parcial: 30 de octubre

Fecha de entrega final: 19 de noviembre.

### **Rúbrica de Evaluación del Proyecto**

#### **Funcionalidad (40 puntos), los cuales estarán contemplados de la siguiente manera:**

- Cumplimiento de Requisitos Básicos (20 puntos):
  - 20 puntos: La aplicación valida correctamente expresiones regulares y cadenas de acuerdo con los requisitos especificados. Maneja una amplia gama de casos de prueba, incluyendo casos límite y errores comunes.
  - 15 puntos: La aplicación valida la mayoría de las expresiones regulares y cadenas, pero puede tener algunos errores o casos que no maneja adecuadamente.
  - 10 puntos: La aplicación valida algunas expresiones regulares y cadenas, pero no cumple con varios requisitos especificados.}
  - 5 puntos: La aplicación tiene serios problemas de funcionalidad y no cumple con los requisitos básicos.
  - 0 puntos: La aplicación no valida expresiones regulares ni cadenas correctamente.
- Manejo de Errores (10 puntos):
  - 10 puntos: La aplicación maneja errores de entrada y expresiones regulares mal formadas de manera clara y efectiva, proporcionando mensajes de error útiles.
  - 7 puntos: La aplicación maneja algunos errores de entrada y expresiones regulares mal formadas, pero podría mejorar en claridad o utilidad de los mensajes de error.
  - puntos: La aplicación maneja errores de entrada y expresiones regulares mal formadas de manera limitada o confusa.
  - 0 puntos: La aplicación no maneja errores de entrada ni expresiones regulares mal formadas.
- Validación de Cadenas (10 puntos):
  - 10 puntos: La aplicación valida cadenas de manera precisa y eficiente contra la expresión regular proporcionada.
  - 7 puntos: La aplicación valida la mayoría de las cadenas correctamente, pero puede tener algunas imprecisiones o ineficiencias.
  - 4 puntos: La aplicación tiene problemas significativos en la validación de cadenas.
  - 0 puntos: La aplicación no valida las cadenas correctamente.

#### **Usabilidad y Diseño de la Interfaz (20 puntos), los cuales estarán contemplados de la siguiente manera:**

- Interfaz de Usuario (10 puntos):
  - 10 puntos: La interfaz es intuitiva, fácil de usar y estéticamente agradable. Los usuarios pueden ingresar expresiones regulares y cadenas sin dificultad.
  - 7 puntos: La interfaz es funcional pero podría mejorarse en términos de usabilidad o diseño.
  - 4 puntos: La interfaz es difícil de usar o poco clara, lo que dificulta la interacción con la aplicación.
  - 0 puntos: La interfaz es confusa o no permite interactuar correctamente con la aplicación.

- Experiencia del Usuario (10 puntos):
  - 10 puntos: La experiencia del usuario es fluida y eficiente. Los usuarios entienden claramente cómo usar la aplicación y obtener resultados.
  - 7 puntos: La experiencia del usuario es aceptable, pero podría mejorarse para una mayor claridad o eficiencia.
  - 4 puntos: La experiencia del usuario es deficiente y puede causar confusión o errores.
  - 0 puntos: La experiencia del usuario es insatisfactoria y no permite un uso efectivo de la aplicación.

**Calidad del Código (20 puntos), los cuales estarán contemplados de la siguiente manera:**

- Organización y Legibilidad (10 puntos):
  - 10 puntos: El código está bien organizado, comentado y es fácil de leer. Sigue las mejores prácticas de codificación.
  - 7 puntos: El código está organizado y comentado, pero podría mejorarse en términos de legibilidad o estructura.
  - 4 puntos: El código está desorganizado o tiene pocos comentarios, dificultando su comprensión.
  - 0 puntos: El código es desorganizado y difícil de entender.
- Eficiencia y Optimización (10 puntos):
  - 10 puntos: El código es eficiente y está optimizado para el rendimiento. Utiliza algoritmos y estructuras de datos adecuados.
  - 7 puntos: El código es generalmente eficiente, pero hay oportunidades para mejorar la optimización.
  - 4 puntos: El código presenta problemas significativos de eficiencia y optimización.
  - 0 puntos: El código es ineficiente y no está optimizado.

**Documentación (10 puntos), los cuales estarán contemplados de la siguiente manera:**

- Documentación del Usuario (5 puntos):
  - 5 puntos: La documentación del usuario es completa, clara y útil. Incluye instrucciones de instalación y uso.
  - 3 puntos: La documentación del usuario es adecuada pero podría mejorarse en claridad o detalle.
  - 1 punto: La documentación del usuario es mínima o poco clara.
  - 0 puntos: No se proporciona documentación del usuario.
- Documentación Técnica (5 puntos):
  - 5 puntos: La documentación técnica es detallada y útil para otros desarrolladores. Incluye descripciones de la arquitectura y el diseño del código.
  - 3 puntos: La documentación técnica es adecuada pero podría ser más completa.
  - 1 punto: La documentación técnica es limitada o poco clara.
  - 0 puntos: No se proporciona documentación técnica.

**Presentación y Entrega (10 puntos), los cuales estarán contemplados de la siguiente manera:**

- Cumplimiento de Plazos (5 puntos):
  - 5 puntos: El proyecto se entrega a tiempo, cumpliendo con todos los plazos establecidos.
  - 3 puntos: El proyecto se entrega con algunos retrasos menores.
  - 1 punto: El proyecto se entrega significativamente tarde.
  - 0 puntos: El proyecto no se entrega.

- Presentación del Proyecto (5 puntos):
  - 5 puntos: La presentación es clara, profesional y demuestra un entendimiento profundo del proyecto.
  - 3 puntos: La presentación es adecuada pero puede mejorar en claridad o profesionalismo.
  - 1 punto: La presentación es confusa o poco profesional.
  - 0 puntos: La presentación es inadecuada o no se realiza.