

# DISEÑO DE COMPILADORES

## ANALIZADOR LÉXICO

CAMPUS QUERÉTARO  
agosto, 2019

---

Para la primera evaluación parcial, debes entregar el analizador léxico del lenguaje formal que elegiste. Recuerda que cada uno de ustedes tiene diferentes idiomas, con diferentes características y propósitos. Sin embargo, hay algunos requisitos mínimos que espero ver:

1. **Comentarios:** Tu lenguaje debe permitir comentarios. Por ejemplo, en C usas `/* */` o `//`.
2. **Estructuras anidadas:** Debe haber una manera de crear estructuras anidadas como estructuras condicionales, ciclos o jerarquías (como en XML). Por ejemplo, en C, puedes anidar tantos `if`'s como quieras.
3. **Variables y constantes:** Tu lenguaje debe permitir el uso de variables y constantes, tengan o no un tipo.
4. **Cadenas de caracteres (strings):** Tu lenguaje debe permitir el uso de cadenas como un token.
5. **Tipos de datos:** Si está usando un lenguaje de programación tipado, debe usar al menos 3 tipos (entero, booleano, cadena).
6. **Condicionales y ciclos:** Tu lenguaje de programación debe tener al menos una instrucción de ciclo y una instrucción condicional. Ambos deben admitir el anidamiento.

Debes crear los siguientes casos de prueba de unidad, todos debe definir:

1. Un programa sencillo con un comentario de una palabra.
2. Un programa sencillo con un comentario de una línea.
3. Un programa sencillo con la definición de una variable.
4. Un programa sencillo con la definición de una constante.
5. Un programa sencillo con cadenas.
6. Un programa sencillo con variables de todos los tipos de datos.
7. Un programa sencillo con un ciclo y una condicional.
8. Un programa sencillo usando las instrucciones de entrada y salida.
9. Un programa sencillo con todas las instrucciones que has definido.

También debes crear los siguientes casos de prueba, todos deben definir:

1. Un programa sencillo con la definición de una variable en el lugar incorrecto y en el orden incorrecto.
2. Un programa sencillo que utiliza una cadena, variable y constante en un lugar que no está permitido.
3. Un programa sencillo con un ciclo definido pero usando una gramática incorrecta.

Todos los tokens no definidos en la definición formal de tu lenguaje deberán generar un error. Todos los errores contenidos en una entrada deben estar presentes.

**Si alguno de estos requisitos no se aplica a tu lenguaje de programación, prepárate para explicar por qué.**