



# Trabajo Práctico: Diseño e Implementación de un Lenguaje

72.39 - Autómatas, Teoría de Lenguajes y Compiladores

Primer Cuatrimestre 2025

*Primera Entrega*

**Especificación**

# 1. Equipo

Nombre	Apellido	Legajo	E-mail
Ana	Negre	63074	<a href="mailto:anegre@itba.edu.ar">anegre@itba.edu.ar</a>
Matías	Leporini	62872	<a href="mailto:mleporinikogan@itba.edu.ar">mleporinikogan@itba.edu.ar</a>

# 2. Repositorio

La solución y su documentación serán versionadas en: [TP-TLA-72.39](#).

# 3. Dominio

Desarrollar un lenguaje que permita crear presentaciones mediante componentes simples como *Slide*, *Textblock*, *Image* y más. Así, permitirá definir filminas, establecer su contenido, color, fuente y otras propiedades. También brindará la posibilidad de determinar una disposición de los objetos en cada filmina, utilizando para esto operadores binarios propios del lenguaje. Por último, deberá permitir crear animaciones mediante una conjunción de estructuras de control y otro set de operadores.

La salida del lenguaje será un archivo HTML, simulando así una presentación de PowerPoint sin la complejidad agregada del manejo de directorios propios de ese programa. Sin embargo, las ventajas no sólo estarían presentes del lado de los desarrolladores; la implementación de este lenguaje permitirá construir presentaciones livianas que puedan accederse desde cualquier navegador web de manera offline.

# 4. Construcciones

El lenguaje desarrollado debería ofrecer las siguientes construcciones, prestaciones y funcionalidades:

- (I). La totalidad del código estará englobada dentro de un objeto *Presentation*.
- (II). El código se dividirá en tres secciones: objetos, estructura y animaciones.
- (III). Se podrán crear una o varias filminas dentro de una *Presentation*.
- (IV). Se podrán modificar las propiedades de una *Presentation*.
- (V). Se podrá crear una disposición de los objetos personalizada en cada filmina.
- (VI). Se podrán agregar animaciones a los objetos dentro de las filminas.
- (VII). Se podrán agregar animaciones entre las filminas.
- (VIII). Las variables podrán ser de tipo *Slide*, *Image*, *Textblock*, *Animation*, *integer* o *string*.
- (IX). Se contará con palabras reservadas para las propiedades de cada objeto, determinables mediante el símbolo = (igual).

- (X). Se proveerá la opción de cambiar el valor de una *Image* o *Textblock* con el operador *with*.
- (XI). Se proveerán operaciones básicas como *add* y *define*.
- (XII). Se proveerán operadores de transformación como *flip* y *mirror*.
- (XIII). Se proveerán operadores de posición como *above* y *below*.
- (XIV). Se proveerán operadores de animación como *rotate*, *appear* y *dissappear*.
- (XV). Se proveerán estructuras de control de tipo REPEAT y START-THEN-END.

## 5. Casos de Prueba

Se proponen los siguientes casos iniciales de prueba de **aceptación**:

- (I). Un programa que cree una presentación vacía.
- (II). Un programa que cree una presentación con una única slide en blanco.
- (III). Un programa que cuente con una filmina con un *Textblock* y una *Image*.
- (IV). Un programa que cuente con dos filminas, con una animación entre ellas.
- (V). Un programa con una filmina que tenga un *Textblock*, con el color del texto modificado.
- (VI). Un programa con una filmina que tenga una *Image* de un tamaño determinado.
- (VII). Un programa con una filmina que establezca la posición relativa entre dos *Textblocks*.
- (VIII). Un programa con una filmina, animada para que aparezca un *Textblock*.
- (IX). Un programa con una filmina, animada para que una imagen rote varias veces.
- (X). Un programa con una filmina que defina y use dos *Textblocks* distintos.

Además, los siguientes casos de prueba de **rechazo**:

- (I). Un programa malformado.
- (II). Un programa que añada un objeto a una filmina inexistente.
- (III). Un programa que defina un objeto con base en uno inexistente.
- (IV). Un programa que le asigne un valor a una propiedad inexistente.
- (V). Un programa que le asigne un valor de tipo inapropiado a una propiedad.
- (VI). Un programa que coloque una animación inexistente entre dos diapositivas.
- (VII). Un programa con una animación que no tenga los tres elementos de START-THEN-END.

## 6. Ejemplos

La sección de **objects**, con definiciones de objetos reutilizables a lo largo del código:

```
objects {  
  define slide1 Slide {  
    background-color = "red";  
  }  
  define heading Textblock {  
    font-size = 14;  
  }  
}
```

```
}
```

La sección de **structure**, con la agrupación de los objetos dentro de cada slide y sus respectivas ubicaciones:

```
structure {  
  Slide slide1 {  
    add heading;  
    add image1 with "local/image/path";  
    add heading1 with "hola";  
    add heading2 with "subtítulo";  
    heading above image1;  
    heading2 below left heading1;  
  }  
}
```

La sección **animations**, transcurriendo sucesivamente al hacer click:

```
animations {  
  slide1 start heading appear  
    then image1 appear  
    then repeat 10 image1 rotate  
    then image1 disappear  
    end;  
  slide1 dissolve_into slide2;  
}
```

Un archivo completo, entonces, luciría:

```
Presentation pres_name {  
  objects { /* object definition */}  
  structure { /* object position and grouping */}  
  animations { /* object animations */}  
}
```