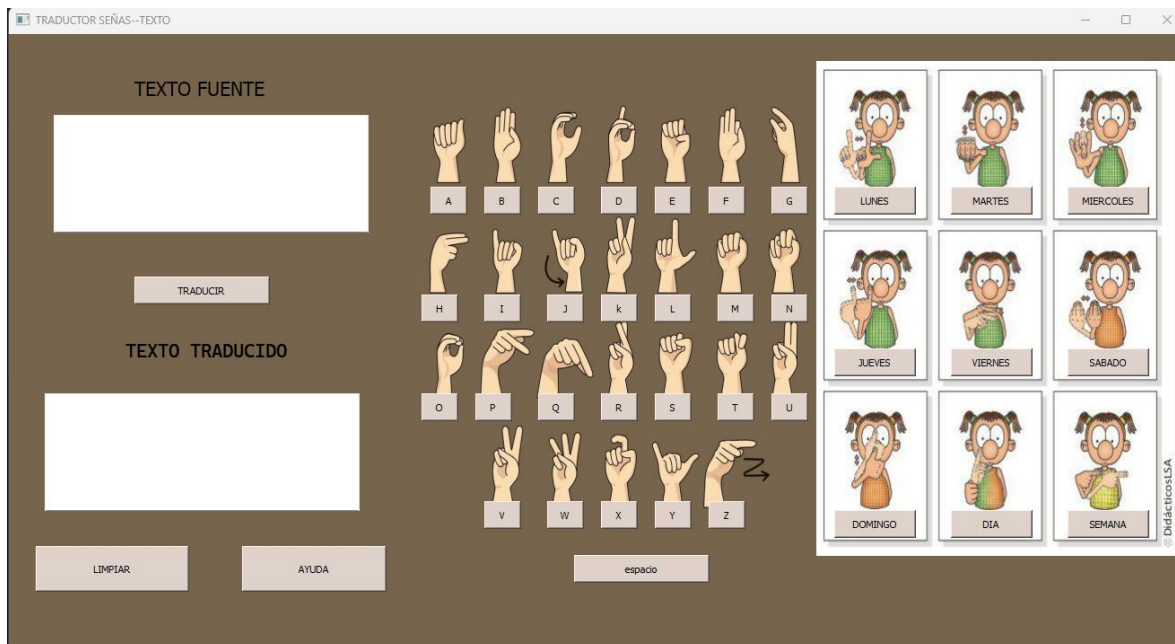


Archivos de Prueba y Validación

PROYECTO: TRADUCTOR DE LENGUA DE SEÑAS A TEXTO

- Universidad: Universidad Autónoma de Tamaulipas
- Facultad: Facultad de Ingeniería Tampico
- Materia: programación de sistemas base 2
- Semestre: 2025-3
- Profesor: Muñoz Quintero Dante Adolfo
- Integrantes del Equipo:
 - Felipe García Juan Diego
 - Jiménez Ramírez Leonel
 - Saldierna Segura Alberto Ángel



1. Casos Válidos (examples/valid/)

Archivo: caso_simple.txt

Propósito: Probar la traducción básica de letras y separadores.

18.25.22.11

TEXTO FUENTE

18.25.22.11

TRADUCIR

TEXTO TRADUCIDO

h o l a

LIMPIAR

AYUDA

FASE 1: ANÁLISIS SEMÁNTICO (Tabla de Símbolos)			
POS	TOKEN (LEXEMA)	TIPO	VALOR
1	18	CHAR	h
2	25	CHAR	o
3	22	CHAR	l
4	11	CHAR	a

FASE 2: GENERACIÓN CÓDIGO INTERMEDIO	
LINEA	INSTRUCCIÓN
1	T1 = DECODE(18)
2	T2 = DECODE(25)
3	T3 = DECODE(22)
4	T4 = DECODE(11)

FASE 3: OPTIMIZACIÓN (Código Redundante)		
LINEA	ACCION	DETALLE
1	[NUEVO]	T1 = DECODE(18)
2	[NUEVO]	T2 = DECODE(25)
3	[NUEVO]	T3 = DECODE(22)
4	[NUEVO]	T4 = DECODE(11)

RESULTADO FINAL	
>> h o l a	

Archivo: caso_optimizacion.txt

Propósito: Probar la fase de Optimización (Unidad 4). Contiene la misma seña repetida tres veces para forzar la reutilización de temporales.

11.11.11

TEXTU FUENTE

11.11.11

TRADUCIR

TEXTU TRADUCIDO

a a a

LIMPIAR

AYUDA

INICIO DE COMPILACIÓN

FASE 1: ANÁLISIS SEMÁNTICO (Tabla de Símbolos)

POS	TOKEN (LEXEMA)	TIPO	VALOR
1	11	CHAR	a
2	11	CHAR	a
3	11	CHAR	a

FASE 2: GENERACIÓN CÓDIGO INTERMEDIO

LINEA	INSTRUCCIÓN
1	T1 = DECODE(11)
2	T2 = DECODE(11)
3	T3 = DECODE(11)

FASE 3: OPTIMIZACIÓN (Código Redundante)

LINEA	ACCION	DETALLE
1	[NUEVO]	T1 = DECODE(11)
2	[OPTIMIZADO]	REUSE T1 (Ya calculado)
3	[OPTIMIZADO]	REUSE T1 (Ya calculado)

RESULTADO FINAL

>> a a a

Archivo: caso_semantico.txt

Propósito: Probar la fase Semántica (Unidad 1). Mezcla una palabra reservada (Lunes) con una letra simple para verificar la detección de tipos STRING y CHAR.

2231241529.11

TEXTO FUENTE

2231241529.11

TRADUCIR

TEXTO TRADUCIDO

LUNES a

LIMPIAR

AYUDA

```
##### INICIO DE COMPILACIÓN #####
##### FASE 1: ANÁLISIS SEMÁNTICO (Tabla de Símbolos) #####
| POS | TOKEN (LEXEMA) | TIPO | VALOR |
|-----|-----|-----|-----|
| 1 | 2231241529 | STRING | LUNES |
| 2 | 11 | CHAR | a |
##### FASE 2: GENERACIÓN CÓDIGO INTERMEDIO #####
LINEA INSTRUCCIÓN
1 T1 = DECODE(2231241529)
2 T2 = DECODE(11)
##### FASE 3: OPTIMIZACIÓN (Código Redundante) #####
LINEA ACCION DETALLE
1 [NUEVO] T1 = DECODE(2231241529)
2 [NUEVO] T2 = DECODE(11)
##### RESULTADO FINAL #####
>> LUNES a
```

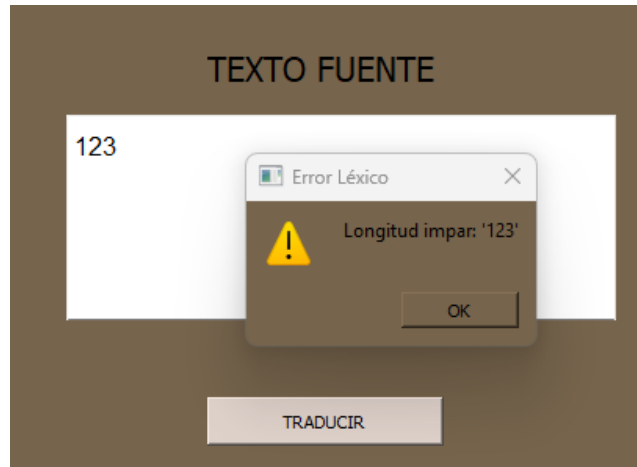
2. Casos Inválidos (examples/invalid/)

Estos archivos contienen errores intencionales para verificar el manejo de excepciones y robustez del sistema.

Archivo: error_lexico_impar.txt

Propósito: Forzar un "Error Léxico". El código tiene longitud 3 (impar), lo cual viola las reglas del lenguaje.

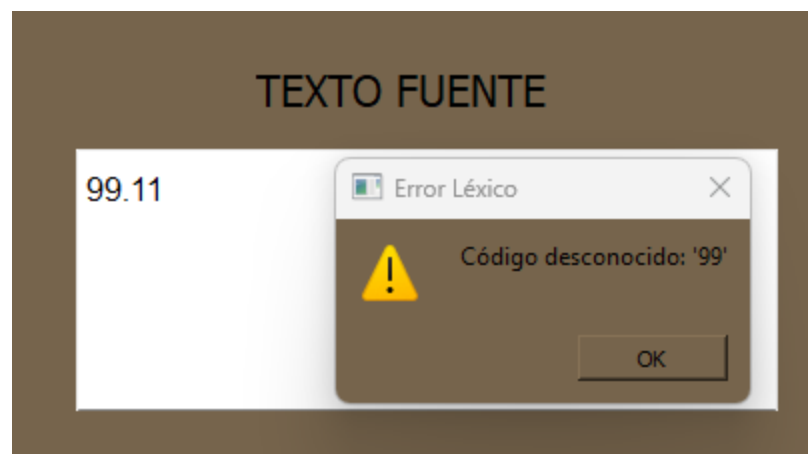
123



Archivo: error_lexico_desconocido.txt

Propósito: Probar la validación de diccionario. El código 99 no existe en la base de datos de señas.

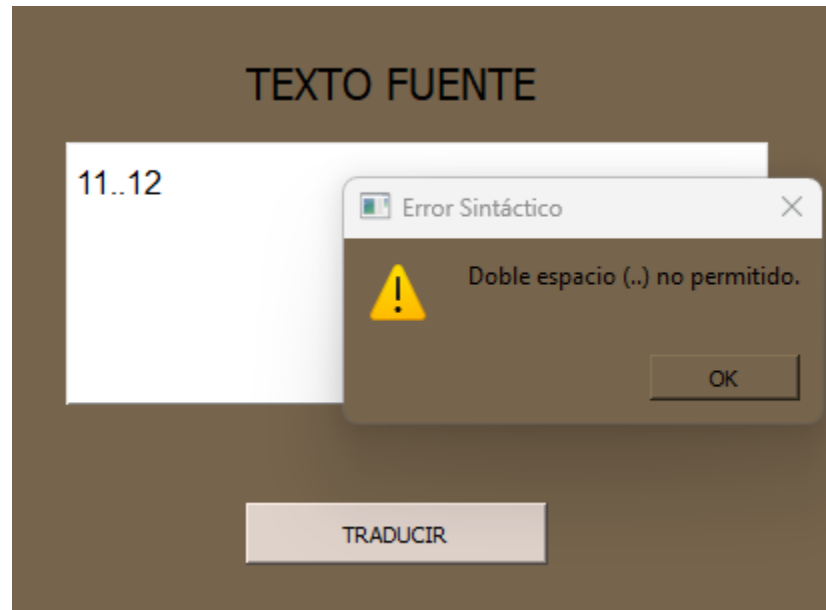
99.11



Archivo: error_sintactico.txt

Propósito: Forzar un "Error Sintáctico". Contiene dos puntos consecutivos (..), violando la estructura de separación.

11..12



3. Salidas Esperadas

Contenido que debería generar el compilador al procesar los casos válidos.

Salida para caso_simple.txt

h o l a

Salida para caso_optimizacion.txt

a a a

Salida para caso_semantico.txt

lunes a