Semantic Tool

CPP Semantic Tool

Ricardo Calvo Mendez
Diego Felipe Lopez

Introducción

Necesidad identificada

Los participantes de ejercicios de programación competitiva no ven fácilmente algunos errores muy pequeños en fragmentos de código, de aquí la idea de construir una herramienta que permita identificarlos.

Antecedentes

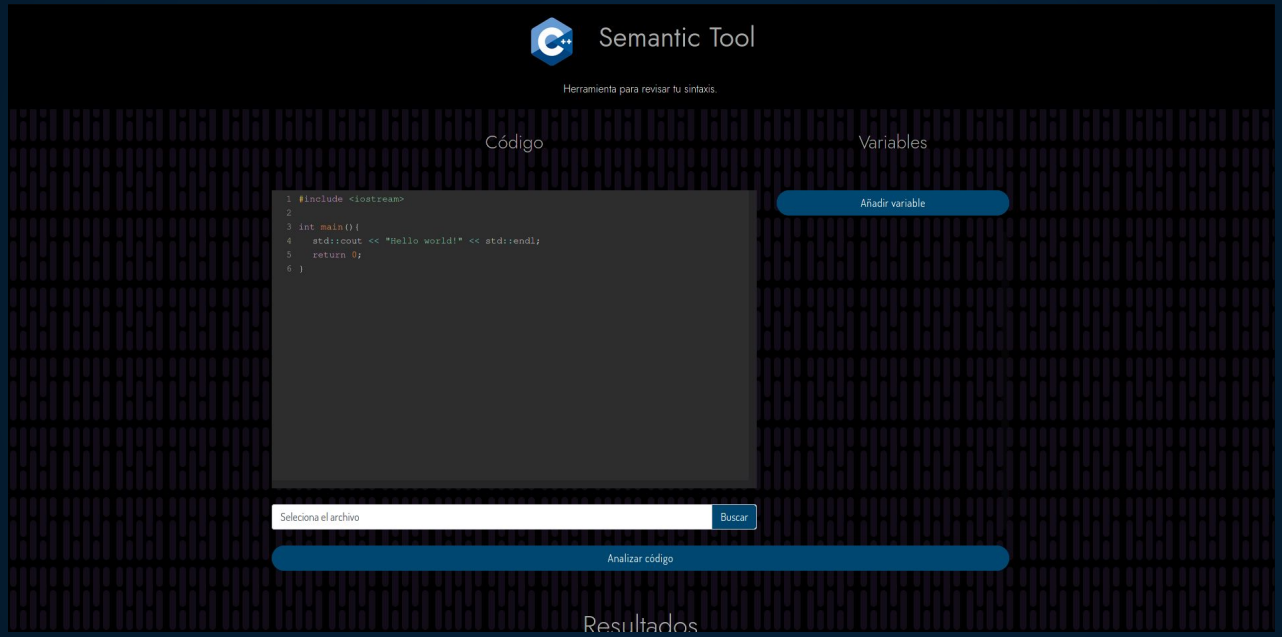
GRAMMATECH
CodeSonar

Motivación

¿Por qué C++?

- C++ tiene una librería conocida como Standard Template Library que facilita a los programadores hacer una programación eficiente y efectiva.
- C++ es muy rápido.
- C++ requiere menos líneas de código que lenguajes como Java.
- Lenguaje muy didáctico, gracias a este lenguaje puedes aprender muchos otros lenguajes con gran facilidad, como C#, Java, Visual Basic, Javascript, PHP, entre otros.
- Existen compiladores de C++ para diferentes sistemas operativos, lo cual representa una ventaja en cuestión de portabilidad.

Propuesta de solución



Página web para evaluar el código de C++

Implementación

Análisis de código

- El análisis de código se realiza usando la gramática de C++ que se puede encontrar en los repositorios de GitHub de ANTLR
- Se utilizan visitors para recorrer evaluar el código propuesto por el usuario.
- Se entrega resultados de análisis sintáctico propuesto por la misma gramática de C++
- Se pueden agregar variables para mostrar posibles rangos de fallos
- Identificar y alertar sobre posibles fallos ya sean intencionados o no intencionados.
- Evaluar teniendo en cuenta el valor de las variables de entrada.

Objetivos

CASOS OVERFLOW

CASOS CASTING

CASOS CORE DUMP

OVERFLOW

```
//-- Casos de overflows

// Cuando el valor de una constante excede el limite de su representación.
// - limite de enteros de 32 bits.
long long overflow_1 = 2147483648;
long long overflow_2 = -2147483649;

// - limite de enteros de 64 bits.
long long overflow_3 = 92233720368547758O8LL;
long long overflow_4 = -92233720368547758O9LL;

// - funciona con varios sistemas numéricos.
int overflow_5 = 0xffffffff; // Tiene una F de más
int overflow_6 = 0b11111111111111111111111111111111; // Tiene un 1 de más

// - Cuando el producto de valores excede el limte que representa su producto.
long long overflow_7 = 2147483647 * 2;
long long overflow_8 = 2147483647 * 2LL; // El producto se convierte en un valor de 64 bits -> No Error

// - Cuando la suma de valores excede el limite que representa su producto.
long long overflow_8 = 2147483645 + 1 + 1 + 1 + 1 + 1;

// - Se tienen en cuenta casting de datos.
long long overflow_10 = ((long long) 2147483647) * 2147483647; // -> No Error

// - Se usan variables personalizadas.
int overflow_11 = 2147483640 + A;
```

CASTING

```
//-- Casos de casting.
// - Alerta de división entera.
int casting_1 = 2 / 3;

// - Alerta en producto grande.
int casting_2 = 2 * 4 * 5 * 2147483647;

// - Alerta en producto grande.
int casting_3 = 2 * B * 5 * 10;

// - Alerta de módulo con decimales.
int casting_4 = 4.0 % 5.0;

// - Alerta en asignaciones.
short casting_5 = 1f;
int casting_6 = 1.0f;
double casting_7 = 100;
float casting_8 = 10;
long int casting_9 = 50000f;
long long casting_10 = 100.0;

// Se tiene en cuenta el operador ternario
int casting_11 = (C > 20 ? 10 : 5.0);
```

CORE DUMP

```
// Generación de cores en arreglos.
int arreglo[12];
arreglo[15] = 10;
```

Pruebas y validación

Probamos los errores de Overflow, casting y core dump directamente en compiladores de C++, y son equivalentes a los que la aplicación identifica, sin la necesidad de detener el programa.

Puede ingresarse un archivo .cpp que el programa reconocerá y posteriormente mostrará las advertencias. También es posible escribir el código directamente en la aplicación web, quien desde el FrontEnd enviará el código en una petición hacia el BackEnd y este nos responderá con las advertencias.

Pruebas y validación

Conclusión

C++ es un lenguaje muy completo, es por esto en este realizar análisis semántico no es tarea sencilla, es por eso que se analizan la mayor cantidad de casos posibles para así ayudar al programador a encontrar errores no tan visibles

VISITA NUESTRA PÁGINA

REFERENCIAS

- https://es.quora.com/Cu%C3%A1les-son-los-pros-y-los-contras-de-usar-C-para-programaci%C3%B3n-competitiva