

## Intérprete y verificaciones dinámicas

Esta es la última fase del proyecto y consiste en *interpretar* programas escritos en nuestro lenguaje, además de hacer verificaciones de errores dinámicos, lo cual interpretaremos como ejecuciones incorrectas, éstas no pueden ser detectadas estáticamente.

Hay que recorrer el árbol sintáctico abstracto generado anteriormente ejecutando las instrucciones y evaluando las expresiones presentes en el programa. Esto lo logramos teniendo un procedimiento o una función *evaluar* para los árboles que presentan expresiones.

Si el programa está bien escrito y no presenta errores estáticos, el intérprete debe ser capaz de ejecutarlo, y además detectar los siguientes errores dinámicos:

- **Variables que no estén inicializadas:** no se debe permitir que el programador intente utilizar el valor de una variable antes de que la misma sea inicializada. Se debe abortar con un error en caso en que esto ocurra.
- **División por cero:** Se debe evitar que se intente realizar una división por cero, Se debe abortar con un error en caso en que esto ocurra.

El comportamiento esperado de su intérprete, en el caso que le sea pasado un programa mal escrito es idéntico al de la entrega pasada.

**Detalles de entrega:** Como se acostumbra en el curso, deben entregar un email que contenga:

- El código fuente en el lenguaje y la herramienta de su elección. Una documentación decente del mismo, sin embargo, con un *.txt* bien hecho es suficiente. El archivo debe tener una extensión *“.gusb”*.

**Fecha de entrega:** Martes de Semana 12 **sin prórroga!**