

Casos de uso semánticos

1. El usuario intenta realizar operaciones con una variable que aún tiene valor nulo.

Al momento de realizar una operación se revisará que las variables no tengan valor nulo, y debido a que nulo se lee como un vacío y no puede realizar operaciones con ese valor, más que solo operaciones lógicas para tomar decisiones. de modo que se revisarán los valores y comprobarán que no son nulos de ser el caso se mostrará el siguiente mensaje al usuario "ERROR SEMÁNTICO: La operación no se pudo realizar, las siguientes variables tienen valor nulo: "

Ejemplo:

```
ent x;  
ent y;  
x = y + 2;
```

2. El usuario desea imprimir una variable con valor nulo.

Al igual que el anterior si una variable tiene un valor nulo es decir que está vacío por lo cual no podrá imprimirse, de forma que debemos comprobar que efectivamente la variable tenga un valor de lo contrario mostrar al usuario el siguiente mensaje "ERROR SEMÁNTICO: Las siguientes variables tienen valor nulo por lo cual no se pueden imprimir: "

Ejemplo:

```
cad abc  
imprime(abc);
```

3. Comparar dos variables de diferente tipo.

Deberá revisarse que las variables a comparar sean del mismo tipo de dato, Los casos condicionales necesita que sus dos operandos sean del tipo entero, no se acepta comparar cadenas. de lo contrario mostrará el siguiente mensaje al usuario "ERROR SEMÁNTICO: imposible comparar, tipos diferentes de datos: "

Ejemplo:

```
cad x;  
x = "una cadena";  
if ( x < 55){  
    /*bloque de código*/  
}
```

4. Realizar operación con variables de diferente tipo.

Se deberán revisar los tipos de datos de los operandos comprobando que estos sean compatibles, de lo contrario no se podrá realizar la operación y mostrará el siguiente

mensaje al usuario "ERROR SEMÁNTICO: Las siguientes variables son de dif tipo, no se puede realizar la operación: ", solo se admiten operaciones aritméticas de valores numéricos.

Ejemplo:

```
cad abc;  
ent x;  
abc = "una cadena";  
x = 2 + abc;
```

5. Asignar un valor de tipo cadena a una variable de tipo entero. (errores de tipo)

Se revisará toda la tabla de símbolos, y primero comparará que los lexemas en la tabla de símbolo sean iguales a las variables declaradas, posteriormente deberá revisar que los id sean referentes a los de la palabra reservada ent y que enseguida de ellos haya un "=", en caso de que estemos definiendo un ent se revisará que lo que se le está asignando sea un entero, un número, de no ser así se guarda el lexema en una variable y lo mismo para el de ent, comprobará que se le esté asignando un número de lo contrario guardará el lexema en una variable y posteriormente se mostrará junto con el mensaje "ERROR SEMÁNTICO:

Las siguientes variables son de dif tipo al declarar: "

Ejemplo:

```
ent x;  
x = "una cadena";
```

6. Asignar un valor de tipo entero a una variable de tipo cadena. (errores de tipo)

Se revisará toda la tabla de símbolos, y primero comparará que los lexemas en la tabla de símbolo sean iguales a las variables declaradas, posteriormente deberá revisar que los id sean referentes a los de las palabras reservadas cad y que enseguida de ellos haya un "=", , en caso de que estemos definiendo un cad se revisará que lo que se le está asignando sea una cadena, de no ser así se guarda el lexema en una variable y lo mismo para el de ent, comprobará que se le esté asignando un número de lo contrario guardará el lexema en una variable y posteriormente se mostrará junto con el mensaje "ERROR SEMÁNTICO:

Las siguientes variables son de dif tipo al declarar: "

Ejemplo:

```
cad abc;  
abc = 123;
```

7. Hacer uso de una variable no declarada.

Se realiza una revisión en toda la tabla de símbolos, revisando especialmente a los identificadores, de forma que comparará todos los identificadores de la tabla de símbolos con las variables declaradas, de dicha forma sabrá si fue declarada o no, de no haber sido declarada la guardará en una variable y mostrará un mensaje de error al usuario notificando

lo siguiente "ERROR SEMÁNTICO: Las siguientes variables no están declaradas: mostrando las variables que faltan declarar.

Ejemplo:

```
ent x;  
x = y + 2;
```

8. No se puede declarar dos o más variables con el mismo nombre.

Se realiza una revisión para que no exista más de una variable con el mismo nombre, revisará las variables declaradas comparándolas con las demás, de ser encontrado algún error de este tipo guarda la variable declarada en un String y mostrará un mensaje de error al usuario con la siguiente notificación "ERROR SEMÁNTICO: Se repiten variables " indicando la variable donde se detectó el error.

Ejemplo:

```
ent x;  
x = 2;  
ent x;
```

9. No se puede declarar dos o más variables con el mismo nombre y que sean de diferente tipo

Se realiza una revisión para que no exista más de una variable con el mismo nombre, revisará las variables declaradas comparándolas con las demás, además de que revisa el tipo de estas variables, de ser encontrado algún error de este tipo guarda la variable declarada en un String y mostrará un mensaje de error al usuario con la siguiente notificación "ERROR SEMÁNTICO: Se repiten variables " indicando la variable donde se detectó el error.

Ejemplo:

```
ent x;  
x = 2;  
cad x;  
x = "una cadena";
```

10. División de un valor entero entre cero.

Revisará la tabla de símbolos, y cuando encuentre que se realiza una división, revisará los operadores si el último es 0, mostrará el siguiente mensaje al usuario "ERROR SEMÁNTICO: DIVISIÓN ENTRE CERO"

Ejemplo:

```

1.
ent y;
y = 8 / 0;
2.
ent x;
ent z;
z = 0;
x = 10 / z;

```

11. Asignar un valor numérico que exceda el límite.

Se definió en nuestro lenguaje que al usar un entero este tiene que estar dentro del rango, el cual es: -2^{15} (-32768) y $2^{15}-1$ (32767). En caso de sobrepasar algún límite mostrará el siguiente mensaje al usuario "Las siguientes variables numéricas exceden el límite:"

Ejemplo:

```

ent x;
x = 32769;

```

12. Declarar una variable y no utilizarla en ninguna parte del código.

Revisará toda la tabla de símbolos comparando las variables declaradas con los lexemas en la tabla de símbolos, de esta forma sabrá si todas las variables que fueron declaradas realmente están siendo utilizadas y cuáles no, en caso de encontrar alguna que no está siendo utilizada se guardará en una variable para mostrarla al usuario junto con el siguiente mensaje "Las siguientes variables se declaran pero no se usan: "

Ejemplo (y nunca se usa):

```

ent x;
ent y;
x = 2;
x = x * 2;

```

13. Ámbito de las variables.

Si una variable es declarada dentro de un bloque de código, por ejemplo, un SI puede ser usada sólo dentro de este bloque, de tal manera que si se excede se marca el error de una variable saliendo de su ámbito.

Ejemplo:

```

ent y;
y = 2;
si(y < 3){
    ent x;
    x = 2;
}
x = 3;

```