



# **COMPILADORES REPORTE #7 HOC6 -- Ciclo for**

---

**3 de Diciembre del 2018**

PROFESOR: TECLA PARRA ROBERTO

ALUMNO: SALDAÑA AGUILAR ANDRÉS

GRUPO: 3CM8

## Conceptos

El ciclo For es una de las instrucciones más sencillas de aprender, y consiste en utilizar mayormente “rangos” en los cuales se define el número de iteraciones que se pueden hacer en este ciclo.

**La sintaxis es la siguiente:**

```
for(inicio;mientras;incremento)
{
    //Código a ejecutar
}
```

Donde el inicio es la declaración de una variable que funciona como un “contador” mientras ejecutamos el ciclo. Continuamente el “mientras” especifica los valores o el rango de valores que puede tomar el contador de “inicio” y el “incremento” específico cuánto se va a incrementar el contador con cada iteración, lo que indicaría que eventualmente el contador saldría de su posible rango de valores y terminará el ciclo.

## Desarrollo

En esta práctica se hizo la implementación del ciclo for en el HOC5 que había implementado en java, donde en esta práctica, agregamos código al archivo de la clase máquina, a la gramatical y léxica del programa, primeramente analizaremos la gramática:

Se agregó una nueva producción en la regla de statement, donde se define que para la declaración de un ciclo for necesitaremos de la palabra reservada **‘for’**, seguida de un paréntesis de apertura, una **variable de control** del ciclo seguida de un punto y coma, la **condición de paro** seguida de un punto y coma y el **incremento** de la variable de control. Las acciones gramaticales tienen como objetivo colocar las instrucciones de los elementos previamente mencionados en el lugar adecuado para simular el funcionamiento de un ciclo for.

```
stmt: ...
    | for '(' inicio ';' cond ';' inc ')' stmt end { maq.getProg().setElementAt($3.obj, (int) $1.obj + 1);
maq.getProg().setElementAt($5.obj, (int) $1.obj + 2);
maq.getProg().setElementAt($7.obj, (int) $1.obj + 3);
maq.getProg().setElementAt($9.obj, (int) $1.obj + 4);
maq.getProg().setElementAt($10.obj, (int) $1.obj + 5);
}
```

También se agregó una nueva regla gramatical para el símbolo terminal **'for'**, que tiene como objetivo agregar en la RAM el código para ejecución del for que después ejecuta la máquina virtual, y los respectivos stops.

```
for: FOR { int numI = maq.code("forCode");
        maq.code("STOP"); maq.code("STOP"); maq.code("STOP"); maq.code("STOP");
maq.code("STOP");
        $$ = new ParserVal(new Integer(numI));
    }
;
```

Por último, este es el código de la función que consigue los parámetros necesarios para ejecutar el ciclo for que el usuario programó:

```
void forCode(){
    boolean d;
    int savepc = pc;
    execute(((Integer)prog.elementAt(savepc)).intValue()); // Init of var
    execute(((Integer)prog.elementAt(savepc + 1)).intValue()); /* condition */
    d = ((Boolean)pila.pop()).booleanValue();

    while (d) {
        execute(((Integer)prog.elementAt(savepc + 3)).intValue()); // statement
        execute(((Integer)prog.elementAt(savepc + 2)).intValue()); // increment
        execute(((Integer)prog.elementAt(savepc + 1)).intValue()); /* condition */

        d = ((Boolean)pila.pop()).booleanValue();
    }

    pc = ((Integer)prog.elementAt(savepc + 4)).intValue();
}
```

## Resultados

Para probar el funcionamiento de nuestro programa, se simuló con un ciclo for la exponenciación del número complejo  $1+1i$  a la novena potencia, para hacer esto declare una variable  $numa = 0+0i$  que estará incrementándose al término de cada iteración, mientras este número sea menor a  $9+0i$ , por lo tanto se repetirá nueve veces este ciclo, dentro de la declaración, se está multiplicando a  $numb = 1+1i$  por  $1+1i$  nueve veces e imprimiendo su valor más actual, dando como resultado el siguiente código de salida:

```
Expression: for(numb = 1+1i ; ( numa < 9+0i ) ; numa = numa + 1+0i ){ numb = numb*1+1i print(numb) }
ProgBase(): 0
Result:
0.0 + 2.0 i
Result:
-2.0 + 2.0 i
Result:
-4.0 + 0.0 i
Result:
-4.0 -4.0 i
Result:
0.0 -8.0 i
Result:
8.0 -8.0 i
Result:
16.0 + 0.0 i
Result:
16.0 + 16.0 i
Result:
0.0 + 32.0 i
```

## Conclusión

El desarrollo de esta práctica fue más sencillo de lo habitual ya que solamente agregamos la funcionalidad de un ciclo for al programa, el cual es muy útil para quizás el llenado de arreglos o matrices.