**Teoría de la Computación**

**Actividad 19.**

**Objetivo**. El estudiante es capaz de generar el pseudocódigo aplicando el método de descenso recursivo a diversos diagramas de sintaxis.

**Desarrollo**: En cada uno de los siguientes ejercicios obtener el pseudocódigo aplicando el método de descenso recursivo.

Ejercicio 1.

< P > < B >

A

**< A > < C >**

C

B

**< P >**

**P( )**

{

if (next == a) {

avanza( )

A( )

If next == b {

avanza()

}

else { Error (b) }

}

else if( next == c) {

avanza()

} else {error (a, c)}

}

**<A>**

A ( )

{

B( )

C( )

avanza ()

if next == ; {

avanza( )

}

else if ( next == , ) {

C( )

}

else {error (; ,}

}

**<B>**

B( )

If next == b {

avanza ( )

if next == ; {

avanza( )

} else {error ( ; ) }

} else { error (b) }

**<C>**

C( )

If next == c {

avanza ( )

if next == ; {

avanza( )

} else {error ( ; ) }

} else { error (c) }

Ejercicio 2.

< VAR >

<VAR>

Var()

{

If (next == int || next == double || next == String) {

avanza ( )

if (next == id) {

ID( )

avanza( )

if (next == ; ) {

avanza()

} else { error( ; ) }

} else { error( id ) }

} else {error ( int, double, String )}

}

<ID>

ID( ){

if (next == id) {

Avanza ( )

If (next == [ ){

COR( )

} else {

Avanza( )

}

If (next == ,) {

ID( )

}

}

<COR>

COR ( )

{

If (next == [ ){

Avanza ( )

If (next == ] ){

If (next == [ ){

COR ( )

} else { avanza( ) }

} else { error( ] ) }

} else { error( [ ) }

}

Ejercicio 3.

< X >

Y

Y

< Y >

X

< X >

X( )

{

A( )

If (next == a) { A( ) }

else {

avanza ( )

}

B( )

If (next == b) {

B( )

}

else if ( next == c {

Avanza( )

If (next == ; ) {

avanza( )

}

Else { error( ; ) }

}

else { error(c, a) }

}

A( )

{

If (next == a) {

avanza ( )

Y( )

} else { error(a) }

}

B( )

{

If (next == b) {

avanza ( )

Y( )

} else { error(b) }

}