Conceptos y Paradigmas de Lenguajes de Programación - 2025 - Primer Parcial 1ra Fecha. T2
25/04/2025

Realice el parcial con lapicera de otra forma se desaprobará el/los ejercicio/s.
Se considera presentismo cuando se realiza completamente un ejercicio.

Legajo: 14350/7						Corrigió:
Apellido y Nombres: Carballo Pedro						SEBASTIAN
1	a B	b B				
2	a B	b B	c B	d B		
3	a B	b B				
4	a B	b B	c B			
				Resultado Final		A

Ejercicio 1

a) (Pts.25) Defina, utilizando EBNF, la gramática para la sentencia FOR de lenguaje C. Puede ser un ejemplo como:

```
for (expresionInicio:; condicionFin; ExpresionLoop)
{       instrucciones
}
```

b) (Pts. 5) Realice el diagrama sintáctico de la gramática correspondiente.

Ejercicio 2

Sea el siguiente código en Ada, indique para todos los identificadores

a) (Pts. 5) Su tipo de ligadura con l-valor. **b)** (Pts. 5) Su r-valor al momento de declaración (inicialización).

c) (Pts. 5) Tiempo de vida . **d)** (Pts. 5) Alcance → Se toma desde la línea siguiente a la declaración (como en la práctica)

```
1.   with text_io; use text_io;
2.   Procedure Main is;
3.   type arreglo_int is array(integer range <>);
4.   c, o, q:integer;
5.   vec:arreglo(1..100);
6.   t: constant integer:=5;
7.   Procedure Proc is;
7.1.     type pointer is access integer;
7.2.     vec2:arreglo(0..t);
7.3.     q: pointer;
7.4.     begin
7.4.1.       o:=4;
7.4.2.       vec2(o):= vec2(1) + vec(4);
7.4.3.       q:= new pointer;
7.4.4.       free q;
7.5.     end;
8.   begin
9.   o:=5;
10.  Proc;
11.  c:= o + 2;
12.  end.
```

Realice este ejercicio sobre esta misma hoja.

Identif	*L-value*	*R-Value*	*Alcance*	*T.V.*
main	-	-	3-12	2-12
c	autom.	basura	5-12	2-12
o	autom.	basura	5-12	2-12
q (linea 4)	autom.	basura	5-7.3 8-12	2-12
vec	autom.	basura	6-12	2-12
t	autom.	basura	7-12	2-12
Proc	-	-	7.1-~~12~~ 7.5	7-7.5
vec 2	semidin.	basura	7.3-7.5	7-7.5
q	autom.	null	7.4-7.5	7-7.5
q^	dinámica	basura	7.4-7.5	7.4.3-7.4.4

(linea 7.3)

Conceptos y Paradigmas de Lenguajes de Programación - 2025 - Primer Parcial 1ra Fecha. T2
25/04/2025

Realice el parcial con lapicera de otra forma se desaprobará el/los ejercicio/s.
Se considera presentismo cuando se realiza completamente un ejercicio.

Ejercicio 3

Sea el siguiente programa escrito en Pascal like, realice la ejecución del programa presentado siguiendo:
a) (Pts. 20) Cadena dinámica. b) (Pts. 5) Indique cuáles son los campos que componen un registro de activación y para qué sirve cada uno.

. El lenguaje evalua expresiones de izq. a der.

```
Program MiPrograma
var i: integer;
var x: integer;
var a: array[1..6] of integer;

Procedure Tres
 begin
  i := i + 2;
  x := x - 1;
 end

Procedure Dos
 var x: integer;
 Function Fun: integer;
  begin
   i := i - 1;
   return x + 1;
  end
 begin
  x := 7;
  a[i] := Fun();
 end
```

```
Procedure Uno
 var i: integer;
 begin
  for i:=1 to 6 begin
    a[i] := i * 2;
  end
  i := 6
  Dos();
  Tres();
  write(i);
 end

begin
 i := 3;
 x := 5;
 Uno();
 write(i);
 write(x);
 for i := 1 to 6 begin
  write(a[i]);
 end
end
```

Ejercicio 4

a) (Pts. 10) Describa a qué se denomina variable local y a qué se denomina variable global
b) (Pts. 10) Indique la diferencia entre un error sintáctico y uno semántico. Puede usar ejemplos.
c) (Pts. 5) Alcanza con definir una gramática EBNF, para la construcción de números romanos. Justifique su respuesta

EJERCICIO 1

PEDRO CARBALLO
143507

a)

$G = (N, T, S, P)$

$N = \{$ <for>, <expresionInicio>, <condicionFin>, <expresionLoop>, <instruccion>, <palabra>, <letra>, <digito>, <numero>, <varias>, <while>, <if>, ... <struct>, <tipo>, <variable>, <signo> $\}$

$T = \{$ '{', '}', ':', ';', a...z, A...Z, 0...9, 'for', '(', ')', '=', '>', '<', '++', '-' $\}$

$S = \{$ <for> $\}$

$P = \{$

<for> ::= 'for' '(' <expresionInicio> ':' ';' <condicionFin> ';' <expresionLoop> ')' '{' {<instruccion>}* '}'

<expresionInicio> ::= <TIPO> <variable> ':' '=' <numero>

<TIPO> ::= ('int' | 'float' | ...) // se asumen tipos ya declarados

<variables> ::= <letra> {(<letra> | <digito>)}*

<numero> ::= {<digito>}+

<letra> ::= (a | ... | z | A | ... | Z)

<digito> ::= (0 | ... | 9)

<condicionFin> ::= <variable> <signo> <numero>

<signo> ::= ('>' | '<' | '=')

<expresionLoop> ::= <variable> ('++' | '--')

<instrucciones> ::= { <varios> }+ NO HACE FALTA PORQUE YA LO DEFINIS ABAJO

<varios> ::= (<letras> | <palabras> | <estruct> | <numeros> | ...)+

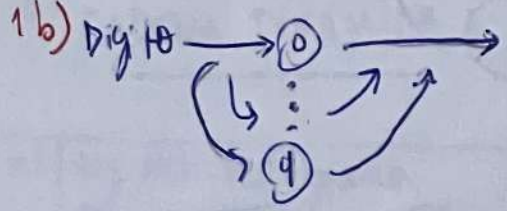
<estruct> ::= (<for> | <while> | <if> | ....) // se asume que el resto de las estruct. de control están definidas

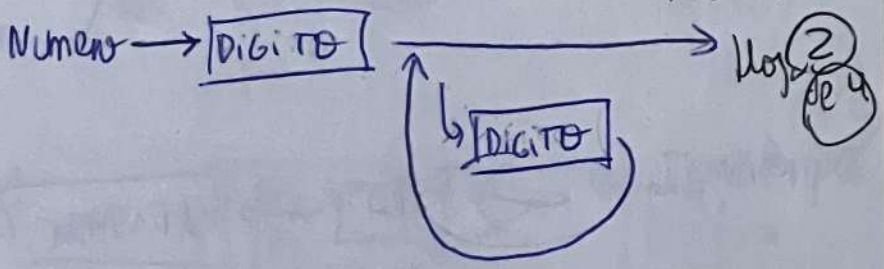
<palabras> ::= { <letra> }+

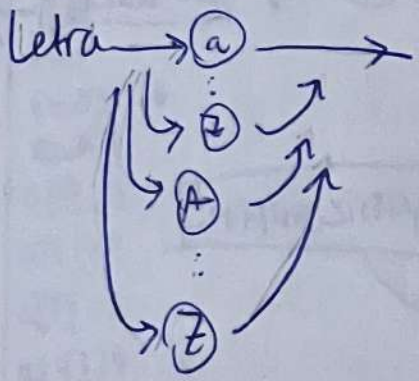
EJERCICIO 1 b

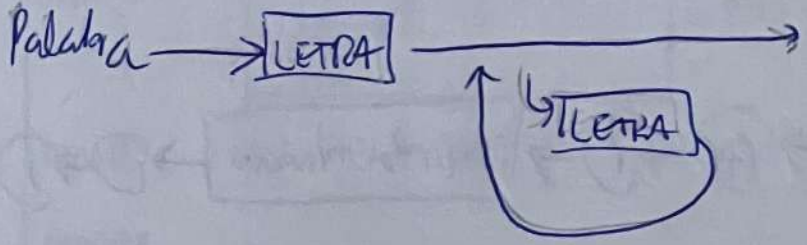
PEDRO CARBALLO 143017

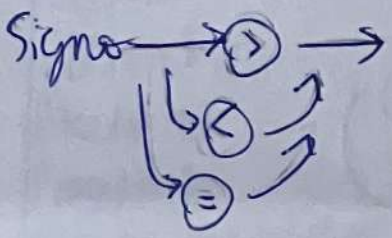
1b)

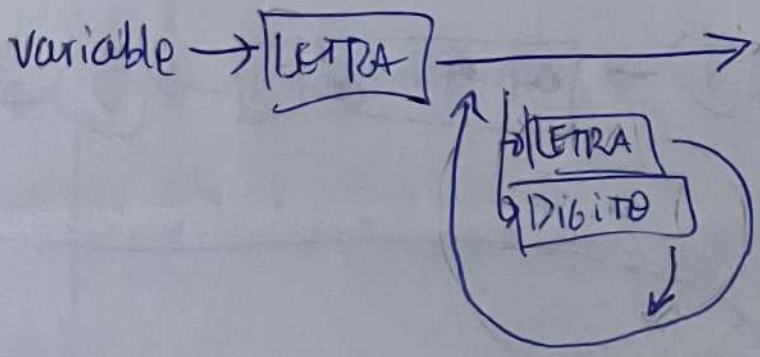


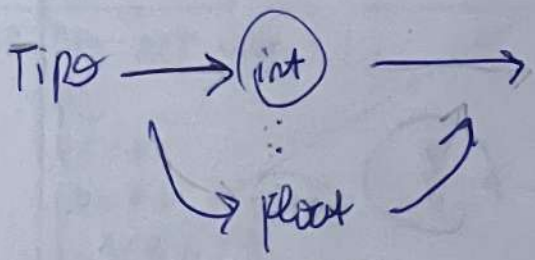


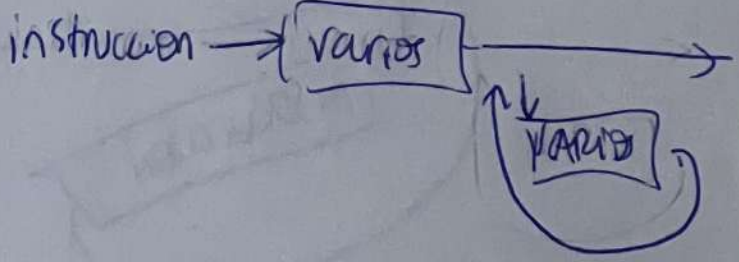


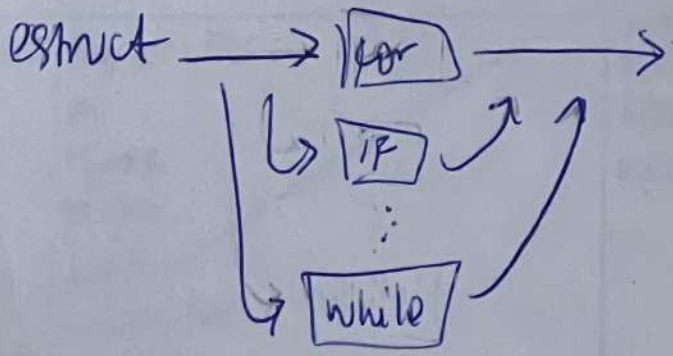


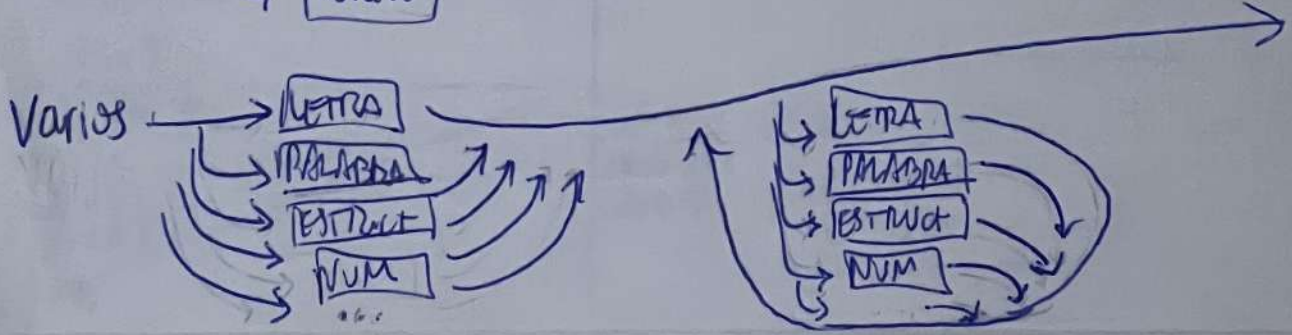


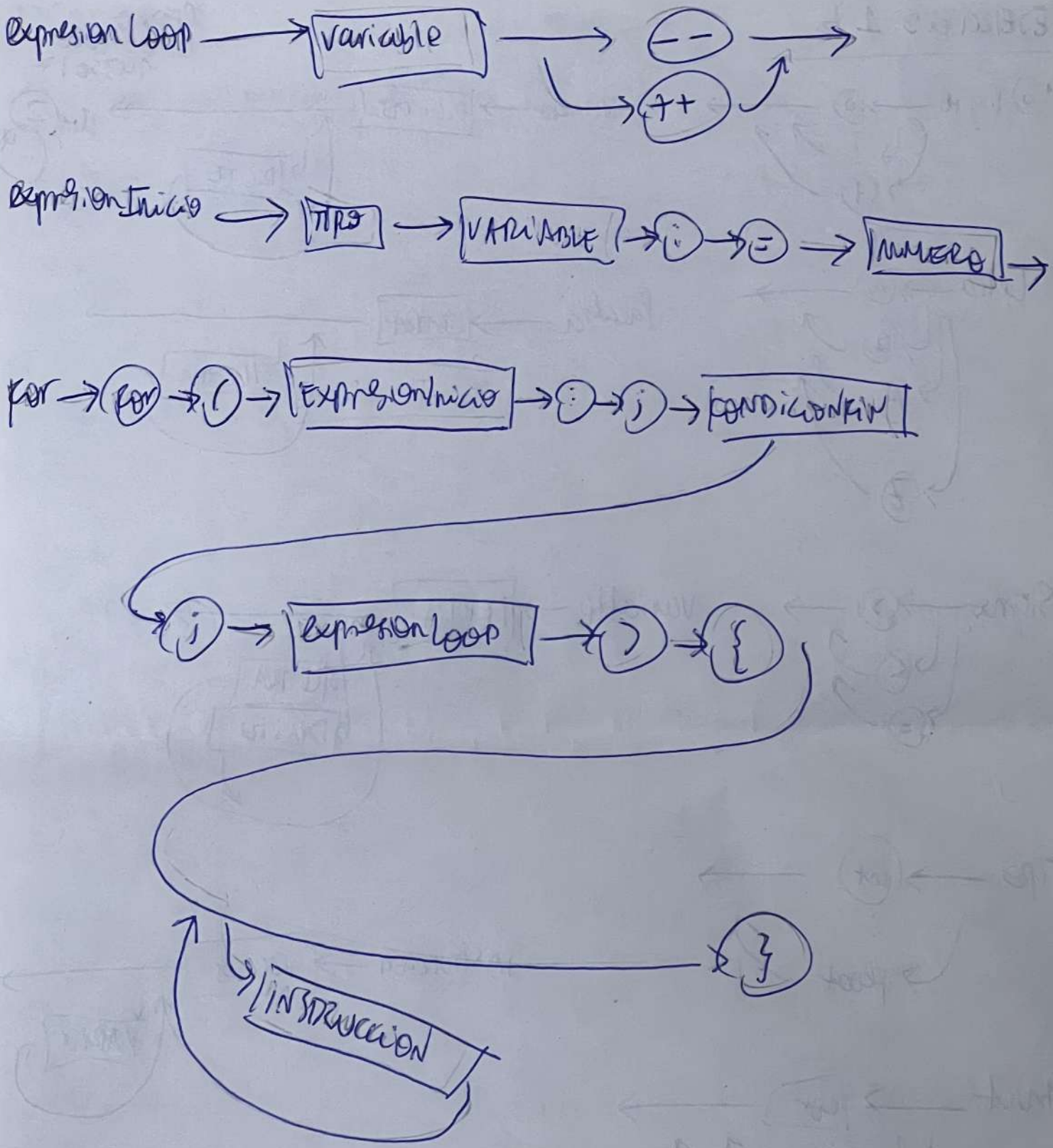











Expresion Loop
variable
--
++
ExpresionInicio
TIPO
VARIABLE
i
=
NUMERO
for
for
(
EXPRESIONINICIO
;
j
CONDICIONFIN
j
EXPRESIONLOOP
)
{
}
INSTRUCCION

EJERCICIO 3

PEDRO CARBALLO
14350/7

Miprograma
Tres | Dos | Uno
Fun

a) CADENA DINÁMICA:

*1	Reg. Act. MiPrograma PR EE=- ED=- i=~~1~~..6 x=~~4~~ a[1]=2 a[2]=4 a[3]=6 a[4]=8 a[5]=10 a[6]=~~12~~8 Procedure Tres Procedure ~~Dos~~ Procedure Uno VR	i=3 x=5 imprime 3 4 i=1..6 imprime 2 4 6 8 10 8
*2	Reg. Act. Uno PR EE=*1 ED=*1 i=1~~..6~~~~8~~7 VR	imprime 7

*3	Reg Act ~~Dos~~
	PR
	EE=*1
	ED=*2
	x=7
	Funcion Fun
	VR=8
		x=7
		a[1]=Fun();
		a[6]=Fun();
		a[6]=8

*4	Reg Act Fun PR EE=*3 ED=*3 VR	i=i-1 return x+1 return 8

*5	Reg Act Tres PR EE=*1 ED=*2 VR	i=i+2 i=5+2=7 x=x-1 x=5-1=4

b) Campos que componen un registro de activación:

Punto de Retorno (PR) = sirve para indicar el lugar en memoria donde sigue el programa luego de volver de otro procedimiento/función que fue llamado.

Enlace o Link estático (EE) = sirve para indicar la unidad que lo contiene

Enlace o Link dinámico (ED) = sirve para indicar la unidad que lo llamó

Variables = Son las variables de esa unidad

Procedimientos/funciones = son los procedimientos/funciones que contiene esa unidad.

Valor de retorno (VR) = sirve para indicar el valor que devuelve la función que llamo. En caso de devolver más de un valor, el VR se actualiza.

EJERCICIO 4

PEDRO CARBALLO
1435017

a) Variable local se denomina a aquellas variables que se encuentran declaradas dentro de una unidad local tal como un procedimiento o una función. Su alcance es (desde la línea siguiente a su declaración) dentro de esa unidad únicamente, y su tiempo de vida es desde la declaración de esa unidad hasta el final.

Variable global se denomina a aquellas variables que se encuentran declaradas para de todos los procedimientos/funciones locales, generalmente arriba de ellas.
Su alcance (desde la línea siguiente a su declaración) es todo el programa, y su tiempo de vida es todo el programa.

b) Un error sintáctico es un error de cómo está escrito ese código, y no lo que significa. Es decir, es un error sintáctico cuando no se siguen las reglas léxicas del programa. Ej: (en ADA)

```
C = 2;
```

} error sintáctico ya que faltan los dos puntos en la asignación: Debería ser C:=2;

Depende de cada lenguaje.

Estos errores se detectan en compilación.

Un error semántico es aquel error de código que está bien escrito sintácticamente pero carece de sentido para el lenguaje: [illegible] Depende de cada lenguaje.

Ejemplo en Pascal

```
Procedure Uno
var i: integer;
var S: string;
begin
   i:=6;
   S:="a";
   i:= i + S;
end
```

sintácticamente está bien escrito pero no tiene sentido ya que no se puede sumar una variable entera con un string.

Los errores semánticos (semántica dinámica) se detectan en ejecución.

c) No alcanza con definir una gramática EBNF para la construcción de números romanos, ya que, la gramática EBNF es una gramática libre de contexto.