



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-2	5042	Programación Lógica

PRÁCTICA No.	LABORATORIO DE NOMBRE DE LA PRÁCTICA	Programación Lógica	DURACIÓN (HORA)
7	NOMBRE DE LA PRÁCTICA	Aspectos de Gramáticas de Clausula Definida	2

1. INTRODUCCIÓN

Las Gramáticas de Clausula Definida o DCG por sus siglas en inglés son una forma que proveen algunas versiones de Prolog para trabajar de una forma sencilla con gramáticas libres de contexto.

2. OBJETIVO (COMPETENCIA)

Utilizar DCGs para representar gramáticas de lenguajes formales.

Formuló	Revisó	Aprobó	Autorizó
Cecilia Curlango Rosas	MC Gloria E. Chavez Valenzuela		MC Miguel Angel Martínez Romero
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad

3. FUNDAMENTO



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO	MATERIAL DE APOYO
---------------------	-------------------

1. Computadoras con el sistema operativo Linux.
2. Acceso al interprete de Prolog SWI-Prolog.

B) DESARROLLO DE LA PRÁCTICA

1. Escriba el siguiente programa. Verifique la expansión que hace Prolog de la DCG, asegurese de comprender cómo se expande cada término.

```
s --> np_subject, vp.
np_subject --> det, n.
np_object --> det, n.
np_subject --> pro_subject.
np_object --> pro_object.
```

```
vp --> v, np_object.
vp --> v.
det --> [the].
det --> [a].
n --> [woman].
n --> [man].
v --> [shoots]
```

```
pro_subject --> [he].
pro_subject --> [she].
pro_object --> [him].
pro_object --> [her].
```

2. En modo de ejecución por pasos, verifique el proceso que se sigue para reconocer enunciado, a *man shoots her*.
3. Escriba el siguiente programa. Verifique la expansión que hace Prolog de la DCG, asegurese de comprender cómo se expande cada término.

```
s --> np(subject), vp.
np(_) --> det, n.
```



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

np(X) --> pro(X).
 vp --> v, np(object).
 vp --> v.
 det --> [the].
 det --> [a].
 n --> [woman].
 n --> [man].
 v --> [shoots].
 pro(subject) --> [he].
 pro(subject) --> [she].
 pro(object) --> [him].
 pro(object) --> [her].

4. En modo de ejecución por pasos, verifique el proceso que se sigue para reconocer los enunciados:
1. *a man shoots her*
 2. *she shoots the man*
 3. *he shoots*
5. Modifique la gramática anterior, agregando argumentos adicionales, para que también reconozca el sustantivo plural *men* y el verbo plural *shoot*. La gramática debe reconocer enunciados como:
1. *the men shoot her*
 2. *she shoots the men*
 3. *the men shoot the woman*
6. Escriba el siguiente programa. Verifique la expansión que hace Prolog de la DCG, asegurese de comprender cómo se expande cada término.

s(s(NP,VP)) --> np(NP), vp(VP).
 np(np(DET,N)) --> det(DET),n(N).
 vp(vp(V,NP)) --> v(V), np(NP).
 vp(vp(V)) --> v(V).
 det(det(the)) --> [the].
 det(det(a)) --> [a].
 n(n(woman)) --> [woman].
 n(n(man)) --> [man].
 v(v(shoots)) --> [shoots].



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

7. Modifique el programa anterior para que cree el árbol sintáctico de la gramática que desarrolló en el punto 5.
8. Verifique el funcionamiento de este programa:
`s(Count) --> ablock(Count),bblock(Count),cblock(Count).`
`ablock(0) --> [].`
`ablock(suc(Count)) --> [a],ablock(Count).`
`bblock(0) --> [].`
`bblock(suc(Count)) --> [b],bblock(Count).`
`cblock(0) --> [].`
`cblock(suc(Count)) --> [c],cblock(Count).`
9. Verifique el funcionamiento de este programa:
`s(Count) --> ablock(Count),bblock(Count),cblock(Count).`
`ablock(0) --> [].`
`ablock(NewCount) --> [a],ablock(Count), {NewCount is Count + 1}.`
`bblock(0) --> [].`
`bblock(NewCount) --> [b],bblock(Count), {NewCount is Count + 1}.`
`cblock(0) --> [].`
`cblock(NewCount) --> [c],cblock(Count), {NewCount is Count + 1}.`

C)	CÁLCULOS Y REPORTE
-----------	---------------------------

5. RESULTADOS Y CONCLUSIONES

6. ANEXOS

7. REFERENCIAS
