



**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	Programación Lógica	DURACIÓN (HORA)
4	NOMBRE DE LA PRÁCTICA	Listas	2

1. INTRODUCCIÓN

Las listas en Prolog son una estructura de datos muy útil para almacenar datos. La manipulación apropiada de estas permite crear programas útiles en prolog.

2. OBJETIVO (COMPETENCIA)

Escribir programas que manipulen la información contenida en listas.

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



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

Formato para prácticas de laboratorio

3. FUNDAMENTO

SWI-Prolog cuenta con un buen conjunto de predicados para la manipulación de listas como se puede constatar en la página <http://gollem.science.uva.nl/SWI-Prolog/Manual/lists.html>, antes de emplearlas en sus programas, es interesante comprender exactamente como funcionan internamente. Por ello, en esta practica se harán implementaciones propias de estos predicados y se comparará su funcionamiento con el de los predicados que ya incluye SWI-Prolog.

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. Elaborar un predicado llamado **agregar(Lista1, Lista2, Lista3)** en el que *Lista3* contenga la concatenación de *Lista1* y *Lista2*.
2. Elaborar un predicado llamado **vecino(X, Y, Lista)** en el que tenga éxito el predicado cuando *Y* siga a *X* en *Lista*.
3. Elaborar un predicado llamado **borra(Lista1, X, Lista2)** en el que borre todos los miembros de *Lista1* que se unifiquen con *X* y coloca el resultado en *Lista2*.
4. Elaborar un predicado llamado **ultimo(Lista, X)** en el que sea verdadero si el último elemento de *Lista* es *X*.
5. Elaborar un predicado llamado **reversa(Lista1, Lista2)** en el que se invierta el orden de los elementos de *Lista1* y se unifique con los elementos de *Lista2*.
6. Elaborar un predicado llamado **permutacion(Lista1, Lista2)** en el que sea verdadero si *Lista1* es una permutación de *Lista2*.
7. Elabora un predicado **combina1(Lista1, Lista2, Lista2)** que combine las primeras dos listas y forme la tercera como se indica.
`combina1([1, 2, 3], [a, b, c], X).` $\Rightarrow X=[1, a, 2, b, 3, c]$
8. Elabora un predicado **combina2(Lista1, Lista2, Lista2)** que combine las primeras dos listas y forme la tercera como se indica.
`combina2([1, 2, 3], [a, b, c], X).` $\Rightarrow X=[[1, a], [2, b], [3, c]]$
9. Elabora un predicado **combina3(Lista1, Lista2, Lista2)** que combine las primeras dos listas y forme la tercera como se indica.
`combina3([1, 2, 3], [a, b, c], X).` $\Rightarrow X=[join(1, a), join(2, b), join(3, c)]$



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

Formato para prácticas de laboratorio

C) CÁLCULOS Y REPORTE

5. RESULTADOS Y CONCLUSIONES

6. ANEXOS

7. REFERENCIAS

Manual de Referencia de Prolog: Sección "Manipulación de Listas"

<http://gollem.science.uva.nl/SWI-Prolog/Manual/lists.html>