



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



**TECNOLÓGICO NACIONAL DE MÉXICO**

**TECNOLÓGICO NACIONAL DE MÉXICO**

**Instituto Tecnológico de Minatitlán**

Ingeniería En Sistemas Computacionales

**“MANUAL DE PRÁCTICAS DE LA MATERIA DE  
PROGRAMACION LOGICA Y FUNCIONAL”**

MINATITLÁN, VER. OCTUBRE 2023



# ÍNDICE

<b>UNIDAD 1 – Conceptos Fundamentales</b>	<b>4</b>
Competencias Especificas	4
Practica 1.1	4
Objetivo de la práctica	4
Actividades previas	4
Introducción	4
Desarrollo de la práctica	5
Observaciones y conclusiones	5
EVALUACION	5
<b>UNIDAD 2 – Modelo de Programación Funcional</b>	<b>8</b>
Competencias Especificas	8
Practica 1.2	8
Objetivo de la práctica	8
Actividades previas	8
Introducción	8
Desarrollo de la práctica	9
Observaciones y conclusiones	9
EVALUACIÓN	10
Practica 2.2	12
Objetivo de la práctica	12
Actividades previas	12
Introducción	12
Desarrollo de la práctica	12
Observaciones y conclusiones	13
EVALUACIÓN	13
<b>UNIDAD 3 – Programación Lógica</b>	<b>16</b>
Competencias Especificas	16
Practica 1.3	16
Objetivo de la práctica	16
Actividades previas	16
Introducción	16
Desarrollo de la práctica	17

Observaciones y conclusiones .....	17
EVALUACIÓN.....	17
<b>UNIDAD 4 – Modelo de Programación Lógica. ....</b>	<b>20</b>
Competencias Especificas .....	20
Practica 1.4.....	20
Objetivo de la práctica .....	20
Actividades previas.....	20
Introducción.....	20
Desarrollo de la práctica.....	21
Observaciones y conclusiones.....	21
EVALUACIÓN.....	22
Practica 2.4.....	25
Objetivo de la práctica .....	25
Actividades previas.....	25
Introducción.....	25
Desarrollo de la práctica.....	25
Observaciones y conclusiones.....	26
EVALUACIÓN.....	26
<b>REFERENCIAS.....</b>	<b>29</b>

# **UNIDAD 1 – Conceptos Fundamentales.**

## **Competencias Especificas**

- Identificar los paradigmas de los lenguajes de programación.

## **Practica 1.1 - Elaboración de mapas conceptuales y/o mentales de la programación lógica y funcional.**

### **Objetivo de la práctica:**

El objetivo de esta práctica es desarrollar mapas conceptuales y/o mentales que representen visualmente los conceptos fundamentales y las estructuras de programación en los paradigmas de programación lógica y funcional. Estos mapas ayudarán a comprender y organizar la información clave relacionada con la programación en estos paradigmas.

### **Actividades previas:**

- Estudiar los conceptos básicos de la programación lógica, como la lógica de predicados y Prolog.
- Comprender los fundamentos de la programación funcional, incluyendo conceptos como funciones de orden superior y recursión.
- Revisar ejemplos de código en lenguajes relevantes, como Prolog y Haskell.

### **Introducción:**

Los mapas conceptuales y mentales son herramientas visuales efectivas para organizar y representar información compleja. En este contexto, se utilizan para capturar los principios y conceptos clave de la programación lógica y funcional, lo que facilita su comprensión y asimilación.

**Desarrollo de la práctica:**

1. Elegir un software o herramienta de generación de mapas conceptuales o mentales (por ejemplo, CmapTools, XMind, MindMeister, etc.).
2. Identificar los conceptos clave de la programación lógica y funcional, como la unificación en Prolog, la recursión en Haskell y la inmutabilidad de datos en programación funcional.
3. Crear un mapa conceptual o mental que muestre relaciones y jerarquías entre estos conceptos.
4. Etiquetar y conectar los nodos del mapa con explicaciones claras y ejemplos.
5. Asegurarse de que el mapa sea fácil de entender y siga una estructura lógica.

**Observaciones y conclusiones:**

En observaciones, se pueden notar los desafíos enfrentados al crear los mapas, como la selección de conceptos clave o la elección de la herramienta adecuada. Además, se pueden mencionar las ventajas de usar mapas conceptuales en el proceso de aprendizaje.

En conclusión, se destaca la importancia de los mapas conceptuales como recursos de estudio efectivos para comprender y retener conceptos en programación lógica y funcional. También se resalta cómo esta práctica contribuye a la consolidación del conocimiento en estos paradigmas de programación.

**EVALUACION**

1. ¿Cuál es la diferencia principal entre la programación lógica y la programación funcional?
  - A) La programación lógica se basa en funciones matemáticas, mientras que la programación funcional se basa en reglas lógicas.
  - B) La programación lógica se enfoca en la manipulación de datos, mientras que la programación funcional se enfoca en la lógica de predicados.

- C) La programación lógica se basa en reglas lógicas y la resolución de consultas, mientras que la programación funcional se basa en funciones matemáticas y la evaluación de expresiones.
- D) La programación lógica y funcional son términos intercambiables y se refieren a lo mismo.

**Respuesta Correcta: C) La programación lógica se basa en reglas lógicas y la resolución de consultas, mientras que la programación funcional se basa en funciones matemáticas y la evaluación de expresiones.**

2. ¿Cuál es un concepto clave en programación lógica que se utiliza para describir reglas lógicas y relaciones entre elementos?
- A) Función de orden superior.
  - B) Unificación.
  - C) Recursión.
  - D) Evaluación perezosa.

**Respuesta Correcta: B) Unificación.**

3. En la programación funcional, ¿qué se entiende por "evaluación perezosa"?
- A) Evaluar expresiones lo más rápido posible.
  - B) Evaluar solo las funciones de orden superior.
  - C) Evaluar las expresiones solo cuando sea necesario para el resultado.
  - D) Evaluar todas las expresiones en paralelo.

**Respuesta Correcta: C) Evaluar las expresiones solo cuando sea necesario para el resultado.**

4. ¿Cuál de las siguientes no es una característica común de la programación funcional?

- A) Inmutabilidad de datos.
- B) Recursión.
- C) Evaluación perezosa.
- D) Cambio de estado global.

**Respuesta Correcta: D) Cambio de estado global.**

5. Cuando creas un mapa conceptual de programación lógica, ¿cuál es un concepto importante que debes incluir en el mapa?

- A) Funciones de orden superior.
- B) Recursión.
- C) Unificación.
- D) Evaluación perezosa.

**Respuesta Correcta: C) Unificación.**

## **UNIDAD 2 – Modelo de Programación Funcional.**

### **Competencias Específicas**

- Realiza una aplicación dando solución a un problema del entorno usando el paradigma de la programación funcional.

### **Practica 1.2 - Desarrollo de programas funcionales con un grado creciente de complejidad, utilizando herramientas de programación funcional, que den solución a problemas reales.**

#### **Objetivo de la práctica:**

El objetivo de esta práctica es desarrollar programas funcionales que aborden problemas reales y complejos. Se busca aplicar los conceptos y técnicas de programación funcional estudiados en la materia para resolver desafíos prácticos.

#### **Actividades previas:**

- Estudiar los conceptos fundamentales de la programación funcional, como funciones de orden superior, recursión y evaluación perezosa.
- Conocer y practicar con un lenguaje de programación funcional, como Haskell o Lisp, que se utilizará para esta práctica.
- Explorar ejemplos de problemas reales que se pueden abordar de manera eficiente con programación funcional.

#### **Introducción:**

La programación funcional se basa en el uso de funciones y datos inmutables para resolver problemas. Esta práctica se enfoca en aplicar estos principios a problemas de la vida real con un grado creciente de complejidad.



**Desarrollo de la práctica:**

1. Seleccionar un problema real que pueda resolverse mediante programación funcional. Ejemplos incluyen análisis de datos, procesamiento de texto, o simulación.
2. Definir los requisitos y objetivos claros para la solución del problema.
3. Utilizar un lenguaje de programación funcional para desarrollar una solución. Comenzar con problemas más simples y avanzar gradualmente hacia desafíos más complejos.
4. Aplicar conceptos como funciones de orden superior para abordar la modularidad y la reutilización de código.
5. Utilizar estructuras de datos inmutables para garantizar la integridad de los datos y facilitar la depuración.
6. Documentar el código de manera adecuada y aplicar prácticas de programación funcional.

**Observaciones y conclusiones:**

En observaciones, podemos notar los desafíos específicos que se enfrentaron durante el desarrollo de programas funcionales. Esto incluye la elección de las estructuras de datos adecuadas, la optimización de rendimiento y la depuración de código funcional.

En conclusión, destacamos los beneficios de la programación funcional para abordar problemas reales. Se resalta cómo la modularidad, la inmutabilidad y el uso de funciones de orden superior permiten crear soluciones más limpias y mantenibles. Además de mencionar las lecciones aprendidas y cómo estas experiencias contribuyen al desarrollo de habilidades en programación funcional.

## EVALUACIÓN

1. ¿Qué es la programación funcional y cuáles son sus conceptos fundamentales?
  - A) La programación funcional se basa en el uso de variables mutables y bucles.
  - B) La programación funcional se centra en el uso de funciones de orden superior y datos inmutables.
  - C) La programación funcional no utiliza funciones.
  - D) La programación funcional solo se aplica a problemas matemáticos.

**Respuesta Correcta: B) La programación funcional se centra en el uso de funciones de orden superior y datos inmutables.**

2. ¿Cuál es una ventaja clave de la programación funcional al abordar problemas reales?
  - A) Mayor complejidad del código.
  - B) Mayor énfasis en la mutabilidad de datos.
  - C) Mayor facilidad para depurar código.
  - D) Mayor reutilización de funciones y modularidad.

**Respuesta Correcta: D) Mayor reutilización de funciones y modularidad.**

3. ¿Qué significa la "evaluación perezosa" en el contexto de la programación funcional?
  - A) Evaluar expresiones lo más rápido posible.
  - B) Evaluar todas las expresiones de inmediato.
  - C) Evaluar las expresiones solo cuando sea necesario para el resultado.
  - D) No evaluar nunca las expresiones.

**Respuesta Correcta: C) Evaluar las expresiones solo cuando sea necesario para el resultado.**

4. ¿Qué se entiende por "funciones de orden superior" en programación funcional?

- A) Funciones que siempre tienen un alto rendimiento.
- B) Funciones que son difíciles de comprender.
- C) Funciones que toman otras funciones como argumentos o devuelven funciones como resultado.
- D) Funciones que solo se pueden utilizar una vez.

**Respuesta Correcta: C) Funciones que toman otras funciones como argumentos o devuelven funciones como resultado.**

5. En la programación funcional, ¿qué significa "datos inmutables"?

- A) Datos que no se pueden utilizar en funciones.
- B) Datos que no se pueden modificar una vez creados.
- C) Datos que solo se pueden modificar de manera asincrónica.
- D) Datos que cambian constantemente.

**Respuesta Correcta: B) Datos que no se pueden modificar una vez creados.**

## **Practica 2.2 - Diseñar y construir una base de conocimiento a través de programación funcional.**

### **Objetivo de la práctica:**

El objetivo de esta práctica es diseñar y construir una base de conocimiento utilizando programación funcional. Se busca aplicar los principios y técnicas de la programación funcional para crear una base de conocimiento que pueda utilizarse para representar datos y reglas en un dominio específico.

### **Actividades previas:**

Estudiar los conceptos de programación funcional, incluyendo la definición de funciones, evaluación perezosa y manipulación de datos inmutables.

Familiarizarse con un lenguaje de programación funcional que se utilizará para esta práctica, como Haskell o Lisp.

Comprender los fundamentos de la representación del conocimiento y la lógica de dominio.

### **Introducción:**

La práctica se enfoca en la creación de una base de conocimiento mediante programación funcional. Una base de conocimiento es esencial para representar información y reglas en un dominio específico, lo que resulta valioso en aplicaciones como sistemas expertos y procesamiento de lenguaje natural.

### **Desarrollo de la práctica:**

1. Seleccionar un dominio o área de conocimiento para el cual se construirá la base de conocimiento. Ejemplos podrían incluir información médica, datos de productos, reglas de un juego, etc.
2. Definir las estructuras de datos necesarias para representar la información en el dominio elegido. Esto puede incluir listas, árboles, grafos u otras estructuras específicas.

3. Implementar funciones y reglas lógicas que permitan la manipulación y consulta de datos en la base de conocimiento.
4. Utilizar características de programación funcional, como recursión, funciones de orden superior y evaluación perezosa, según sea necesario para crear una base de conocimiento eficiente y mantenible.
5. Documentar la base de conocimiento, incluyendo las estructuras de datos y las reglas utilizadas.

### **Observaciones y conclusiones:**

En las observaciones, se pueden destacar los desafíos específicos enfrentados al diseñar y construir la base de conocimiento, como la elección de las estructuras de datos más adecuadas o la implementación de reglas lógicas complejas.

En las conclusiones, se resalta la importancia de la programación funcional en la representación del conocimiento y se discuten los beneficios de esta aproximación en términos de modularidad y reutilización de código. También se pueden mencionar las aplicaciones potenciales de la base de conocimiento en el dominio seleccionado.

### **EVALUACIÓN**

1. ¿Qué es una base de conocimiento en el contexto de la programación funcional?
  - A) Un archivo de texto que contiene información sobre programación funcional.
  - B) Un conjunto de funciones matemáticas en programación funcional.
  - C) Una estructura de datos que almacena información y reglas en un dominio específico.
  - D) Un lenguaje de programación funcional.

**Respuesta Correcta: C) Una estructura de datos que almacena información y reglas en un dominio específico.**

2. ¿Por qué es importante utilizar programación funcional en la construcción de una base de conocimiento?
- A) La programación funcional no es importante para la construcción de bases de conocimiento.
  - B) La programación funcional permite la manipulación eficiente de datos mutables.
  - C) La programación funcional promueve la modularidad y la reutilización de código, lo que es beneficioso en la construcción de bases de conocimiento.
  - D) La programación funcional solo se aplica a problemas matemáticos.

**Respuesta Correcta: C) La programación funcional promueve la modularidad y la reutilización de código, lo que es beneficioso en la construcción de bases de conocimiento.**

3. ¿Qué estructuras de datos se pueden utilizar para representar información en una base de conocimiento en programación funcional?
- A) Variables mutables.
  - B) Listas, árboles, grafos y otras estructuras específicas.
  - C) Números enteros.
  - D) Funciones recursivas.

**Respuesta Correcta: B) Listas, árboles, grafos y otras estructuras específicas.**

4. ¿Qué técnica de programación funcional se utiliza comúnmente en la construcción de bases de conocimiento para aplicar reglas lógicas?

- A) Bucles.
- B) Expresiones condicionales.
- C) Evaluación perezosa.
- D) Funciones de orden superior.

**Respuesta Correcta: D) Funciones de orden superior.**

5. ¿Por qué es importante documentar una base de conocimiento?

- A) La documentación es opcional y no es importante.
- B) La documentación ayuda a garantizar que la base de conocimiento sea secreta.
- C) La documentación facilita la comprensión de la base de conocimiento y su uso futuro.
- D) La documentación solo se aplica a lenguajes de programación tradicionales.

**Respuesta Correcta: C) La documentación facilita la comprensión de la base de conocimiento y su uso futuro.**

## **UNIDAD 3 – Programación Lógica.**

### **Competencias Específicas**

- Conoce las ventajas y desventajas del paradigma de programación lógica.
- Identifica los elementos de la programación lógica.

### **Practica 1.3 - Desarrollo de programas lógicos con un grado creciente de complejidad, utilizando herramientas de programación lógica, que den solución a problemas reales.**

#### **Objetivo de la práctica:**

El objetivo de esta práctica es desarrollar programas lógicos utilizando herramientas de programación lógica, como Prolog. Los programas deben abordar problemas reales y aumentar gradualmente en complejidad para que los estudiantes adquieran experiencia en la resolución de problemas utilizando la lógica y la programación declarativa.

#### **Actividades previas:**

Estudiar los fundamentos de la programación lógica, incluyendo la representación de hechos y reglas mediante predicados en Prolog.

Familiarizarse con el entorno de desarrollo y las herramientas específicas utilizadas para programar en Prolog.

Comprender la resolución de consultas y la inferencia lógica en Prolog.

#### **Introducción:**

La práctica se centra en el desarrollo progresivo de programas lógicos que resuelvan problemas del mundo real. La programación lógica, como la que ofrece Prolog, es especialmente adecuada para la resolución de problemas basada en reglas y relaciones lógicas.



**Desarrollo de la práctica:**

1. Seleccionar un conjunto de problemas del mundo real que puedan abordarse mediante programación lógica. Estos problemas pueden variar en complejidad y alcance.
2. Comenzar con un problema relativamente sencillo y desarrollar un programa lógico en Prolog que proporcione una solución. Esto puede implicar la definición de hechos, reglas y consultas.
3. Gradualmente, aumentar la complejidad de los problemas y los programas desarrollados. Introducir conceptos avanzados de Prolog, como listas, recursión y manipulación de estructuras de datos.
4. Documentar claramente el proceso de desarrollo, incluyendo la especificación de cada problema, la implementación de programas y ejemplos de consultas exitosas.

**Observaciones y conclusiones:**

En las observaciones, se pueden destacar los desafíos específicos enfrentados al desarrollar programas lógicos más complejos, como la gestión de reglas, la depuración y la optimización.

En las conclusiones, se resalta la utilidad de la programación lógica en la resolución de problemas del mundo real y se discuten las lecciones aprendidas a lo largo de la práctica. También se pueden mencionar las aplicaciones potenciales de Prolog en diferentes dominios.

**EVALUACIÓN**

1. ¿Qué es la programación lógica y en qué se diferencia de otros paradigmas de programación?
  - A) La programación lógica es un enfoque basado en la resolución de problemas mediante la ejecución de instrucciones en un orden específico. Se diferencia

de otros paradigmas en que se basa en la lógica formal y en la representación de conocimiento mediante reglas y relaciones.

- B) La programación lógica se centra en la programación de dispositivos lógicos. Se diferencia de otros paradigmas en que no se utiliza un lenguaje de programación.
- C) La programación lógica utiliza algoritmos matemáticos para resolver problemas complejos. Se diferencia de otros paradigmas en que no permite la representación de hechos y reglas.
- D) La programación lógica es un enfoque orientado a objetos. Se diferencia de otros paradigmas en que se centra en la abstracción de datos.

**Respuesta Correcta: A) La programación lógica es un enfoque basado en la resolución de problemas mediante la ejecución de instrucciones en un orden específico. Se diferencia de otros paradigmas en que se basa en la lógica formal y en la representación de conocimiento mediante reglas y relaciones.**

2. ¿Qué herramienta de programación lógica se menciona en la práctica como una opción común para desarrollar programas lógicos?

- A) Python.
- B) Prolog.
- C) Java.
- D) C++.

**Respuesta Correcta: B) Prolog.**

3. ¿Por qué es importante aumentar gradualmente la complejidad de los problemas y programas en la práctica?

- A) Para hacer que la práctica sea más desafiante para los estudiantes.

- B) Para demostrar la versatilidad de la programación lógica en la resolución de problemas de diferentes niveles de complejidad.
- C) Para evitar que los estudiantes se sientan abrumados por problemas complejos.
- D) Para limitar el alcance de la práctica y centrarse en problemas sencillos.

**Respuesta Correcta: B) Para demostrar la versatilidad de la programación lógica en la resolución de problemas de diferentes niveles de complejidad.**

4. ¿Qué elementos se utilizan comúnmente en la programación lógica para representar información y reglas?

- A) Variables y estructuras de control.
- B) Funciones y clases.
- C) Hechos y reglas en forma de predicados.
- D) Matrices y listas.

**Respuesta Correcta: C) Hechos y reglas en forma de predicados.**

5. ¿Cuál es uno de los conceptos avanzados de Prolog mencionados en la práctica?

- A) Variables mutables.
- B) Evaluación perezosa.
- C) Programación orientada a objetos.
- D) Bucles.

**Respuesta Correcta: B) Evaluación perezosa.**

## **UNIDAD 4 – Modelo de Programación Lógica.**

### **Competencias Específicas**

- Realiza una aplicación dando solución a un problema del entorno usando el paradigma de la programación lógica.

### **Practica 1.4 - A partir de una situación real, diseñar y construir una base de conocimiento a través de herramientas de sistemas expertos basado en programación lógica.**

#### **Objetivo de la práctica:**

El objetivo de esta práctica es diseñar y construir una base de conocimiento utilizando herramientas de sistemas expertos basados en programación lógica. Se abordará una situación real y se aplicarán conceptos de programación lógica para desarrollar un sistema experto que pueda tomar decisiones o proporcionar recomendaciones dentro del dominio seleccionado.

#### **Actividades previas:**

- Comprender los principios de los sistemas expertos y su aplicación en la resolución de problemas específicos.
- Estudiar las herramientas y lenguajes de programación utilizados para desarrollar sistemas expertos basados en lógica, como Prolog.
- Analizar la situación o problema real que se abordará con el sistema experto y definir los requisitos y reglas necesarios.

#### **Introducción:**

La práctica se enfoca en el desarrollo de sistemas expertos utilizando programación lógica como base. Los sistemas expertos son aplicaciones que imitan la toma de decisiones humanas expertas en un dominio específico.

**Desarrollo de la práctica:**

1. Seleccionar una situación o dominio real para el desarrollo del sistema experto. Esto podría incluir diagnóstico médico, soporte técnico, recomendaciones de productos, entre otros.
2. Definir las reglas y conocimientos necesarios para abordar la situación. Establecer hechos, reglas y relaciones lógicas en Prolog u otra herramienta de programación lógica.
3. Implementar el sistema experto utilizando las herramientas disponibles. Programar las reglas y consultas lógicas que permitan al sistema realizar diagnósticos o recomendaciones.
4. Realizar pruebas exhaustivas del sistema experto para garantizar su funcionamiento correcto y su capacidad para proporcionar respuestas precisas en función de la entrada proporcionada.

**Observaciones y conclusiones:**

En las observaciones, se pueden destacar los desafíos específicos enfrentados al desarrollar el sistema experto, como la definición de reglas precisas, la depuración del sistema y la eficiencia en la toma de decisiones.

En las conclusiones, se resalta la utilidad de la programación lógica en la construcción de sistemas expertos y se discuten las lecciones aprendidas a lo largo de la práctica. También se pueden mencionar las aplicaciones potenciales de los sistemas expertos en diversos campos.

## EVALUACIÓN

1. ¿Qué es un sistema experto y cuál es su propósito en la programación lógica?

A) Un sistema experto es una aplicación de inteligencia artificial que imita la toma de decisiones humanas expertas en un dominio específico. Se utiliza para proporcionar recomendaciones y tomar decisiones basadas en reglas lógicas.

B) Un sistema experto es un software de diagnóstico que utiliza algoritmos de aprendizaje automático para tomar decisiones. Se utiliza principalmente en medicina.

C) Un sistema experto es un lenguaje de programación utilizado para desarrollar aplicaciones lógicas. No tiene un propósito específico.

D) Un sistema experto es un programa que se ejecuta en hardware especializado para realizar cálculos complejos.

**Respuesta Correcta: A) Un sistema experto es una aplicación de inteligencia artificial que imita la toma de decisiones humanas expertas en un dominio específico. Se utiliza para proporcionar recomendaciones y tomar decisiones basadas en reglas lógicas.**

2. ¿Qué lenguaje de programación es comúnmente utilizado para desarrollar sistemas expertos basados en programación lógica?

A) Java.

B) C++.

C) Python.

D) Prolog.

**Respuesta Correcta: D) Prolog**

3. ¿Por qué es importante definir reglas y conocimientos específicos en un sistema experto?

- A) Para hacer que el sistema experto sea más complejo.
- B) Para mejorar la eficiencia de procesamiento del sistema experto.
- C) Para permitir que el sistema experto tome decisiones y proporcione recomendaciones en función de la información proporcionada.
- D) Para aumentar el tamaño de la base de conocimiento del sistema experto.

**Respuesta Correcta: C) Para permitir que el sistema experto tome decisiones y proporcione recomendaciones en función de la información proporcionada.**

4. ¿Qué tipo de pruebas se deben realizar en un sistema experto antes de considerarlo listo para su implementación?

- A) Pruebas de hardware.
- B) Pruebas de rendimiento.
- C) Pruebas exhaustivas para garantizar que el sistema proporciona respuestas precisas y adecuadas en función de la entrada proporcionada.
- D) Pruebas de interfaz de usuario.

**Respuesta Correcta: C) Pruebas exhaustivas para garantizar que el sistema proporciona respuestas precisas y adecuadas en función de la entrada proporcionada.**

5. ¿Cuál es uno de los principales objetivos al desarrollar un sistema experto basado en programación lógica?

- A) Imitar la toma de decisiones humanas expertas en un dominio específico.
- B) Aumentar la velocidad de procesamiento de datos.
- C) Crear una interfaz de usuario atractiva.
- D) Automatizar la entrada de datos.

**Respuesta Correcta: A) Imitar la toma de decisiones humanas expertas en un dominio específico.**



## **Practica 2.4 - Construir un sistema experto a partir de la base de conocimiento creada en programación lógica.**

### **Objetivo de la práctica:**

El objetivo de esta práctica es construir un sistema experto utilizando la base de conocimiento previamente creada en programación lógica. Se busca aplicar los principios de la programación lógica para desarrollar un sistema que pueda tomar decisiones o proporcionar recomendaciones dentro de un dominio específico.

### **Actividades previas:**

- Crear una base de conocimiento en programación lógica, como se realizó en una práctica anterior.
- Estudiar los conceptos y técnicas necesarios para el desarrollo de sistemas expertos basados en la base de conocimiento existente.
- Definir el dominio o situación en la que se aplicará el sistema experto y comprender las reglas y conocimientos necesarios.

### **Introducción:**

La práctica se enfoca en la construcción de un sistema experto utilizando la base de conocimiento previamente creada en programación lógica. Los sistemas expertos son aplicaciones que pueden tomar decisiones lógicas o proporcionar recomendaciones basadas en la información contenida en la base de conocimiento.

### **Desarrollo de la práctica:**

1. Utilizar la base de conocimiento existente, desarrollada en prácticas anteriores, como punto de partida para la construcción del sistema experto.
2. Definir las reglas y consultas necesarias en el sistema experto para abordar el dominio seleccionado.
3. Implementar el sistema experto utilizando lenguajes y herramientas de programación lógica, como Prolog.

4. Realizar pruebas exhaustivas para verificar que el sistema experto funcione correctamente y sea capaz de proporcionar respuestas precisas y lógicas.

### **Observaciones y conclusiones:**

En las observaciones, se pueden destacar los desafíos específicos enfrentados durante la construcción del sistema experto a partir de la base de conocimiento existente. Se pueden mencionar aspectos como la integración de reglas, la eficiencia en la toma de decisiones y la depuración del sistema.

## **EVALUACIÓN**

1. ¿Qué es un sistema experto y cuál es su función principal?
  - A) Un sistema experto es un programa de diseño gráfico.
  - B) Un sistema experto es un sistema de seguridad informática.
  - C) Un sistema experto es una aplicación que puede tomar decisiones lógicas o proporcionar recomendaciones basadas en una base de conocimiento.
  - D) Un sistema experto es un sistema operativo.

**Respuesta Correcta: C) Un sistema experto es una aplicación que puede tomar decisiones lógicas o proporcionar recomendaciones basadas en una base de conocimiento.**

2. ¿Cuál es el propósito de utilizar una base de conocimiento en un sistema experto?
  - A) Las bases de conocimiento se utilizan para almacenar archivos.
  - B) Las bases de conocimiento son útiles para respaldar los datos del sistema.
  - C) Las bases de conocimiento permiten al sistema experto tomar decisiones y proporcionar recomendaciones en función de la información contenida.

- D) Las bases de conocimiento son simplemente un elemento decorativo en un sistema experto.

**Respuesta Correcta: C) Las bases de conocimiento permiten al sistema experto tomar decisiones y proporcionar recomendaciones en función de la información contenida.**

3. ¿Qué lenguaje de programación es comúnmente utilizado para desarrollar sistemas expertos basados en programación lógica?

- A) Java.
- B) C++.
- C) Python.
- D) Prolog.

**Respuesta Correcta: D) Prolog.**

4. ¿Por qué es importante realizar pruebas exhaustivas en un sistema experto antes de su implementación?

- A) Para comprobar si el sistema experto se ejecuta en hardware de alta gama.
- B) Para evaluar la eficiencia del sistema experto.
- C) Para verificar que el sistema experto funcione correctamente y sea capaz de proporcionar respuestas precisas y lógicas.
- D) Para asegurarse de que el sistema experto sea compatible con varios sistemas operativos.

**Respuesta Correcta: C) Para verificar que el sistema experto funcione correctamente y sea capaz de proporcionar respuestas precisas y lógicas.**

5. ¿Cuáles son algunos ejemplos de dominios o situaciones en los que los sistemas expertos basados en programación lógica pueden ser aplicados con éxito?

- A) Diagnóstico médico, recomendaciones de productos, soporte técnico, análisis de datos, entre otros.
- B) Redes sociales, juegos de video, gestión de proyectos, edición de video.
- C) Programación web, diseño gráfico, comercio electrónico, música.
- D) Navegadores web, sistemas operativos, hojas de cálculo, edición de imágenes.

**Respuesta Correcta: A) Diagnóstico médico, recomendaciones de productos, soporte técnico, análisis de datos, entre otros.**

## REFERENCIAS

### Impresas:

- Nilsson, n. J. (2001). *Inteligencia artificial. Una nueva síntesis*. Mc graw hill.
- Poole, d., mackworth, a. Y goebel, r. (1998). *Computational intelligence (a logical Approach)*. Oxford university press.
- Bratko, i. (1990). *Prolog programming for artificial intelligence (2nd ed.)*. Addison Wesley.
- Mitchell, t. M. (1997). *Machine learning*. Mc graw hill.
- Flach, p. (1994). *Simply logical (intelligent reasoning by example)*. John wiley.
- Bird, richard. (2000). *Introducción a la programación funcional con haskell*. Segunda Ed. Prentice hall..
- Fokker, jeroen. (1995). *Programación funcional*. Universidad de utrecht, Departamento de informática.
- Julian, p., alpuente, m. (2007). *Programación lógica. Teoría y práctica*. Pearson Prentice hall.
- Hogger, c. (1990). *Essentials of logic programming*. Clarendon press, oxford.
- Bratko. (1991). *Prolog programming for artificial intelligence*. Segunda edición). Addison wesley.
- Sterling & shapiro. (1994). *The art de prolog*. Mit.
- Lucas, p. Y gaag, l.v.d. (1991). *Principles of expert systems*. Addison–wesley.

### Electrónicas:

- 13. Garbusi Pablo. *Diseño de compiladores*. Obtenido de [http://www.fing.edu.uy/inco/cursos/compil/teoricos/01\\_Introduccion.pdf](http://www.fing.edu.uy/inco/cursos/compil/teoricos/01_Introduccion.pdf)
- 14. Ortiz Triviño, Jorge Eduardo. *Lenguajes Regulares*. Obtenido de <http://www.youtube.com/watch?v=2caZNHXsj88>
- 15. Cubur, Alex. *Expresion Regular a DFA en JFlap*. Obtenido de [http://www.youtube.com/watch?v=S6y0Wu\\_qp6I](http://www.youtube.com/watch?v=S6y0Wu_qp6I)
- 16. Bonelli, Eduardo. *Resolución SLD y PROLOG*. 2006. Obtenido de <http://www-2.dc.uba.ar/materias/plp/20062C/download/clase10.pdf>

- 17. Gallardo, José, Gutiérrez, Paco, Ruíz, Blas. Tomado de: Hudak, Paul, Peterson, John, Fasel, Joseph. *A gentle introduction to Haskell Version 98*. 2001. Obtenido de <http://www.lcc.uma.es/~blas/pfHaskell/gentle/>
- 18. Gallardo, José, Gutiérrez, Paco, Ruíz, Blas. Tomado de: Hudak, Paul, Peterson, John, Fasel, Joseph. *A gentle introduction to Haskell Version 98*. 2001. Obtenido de <http://www.lcc.uma.es/~blas/pfHaskell/gentle/>
- 19. Juganaru, Mihaela. *Lenguaje PROLOG*. 2010. Obtenido de <http://ce.azc.uam.mx/profesores/mjm/pub/prolog.pdf>
- 20. HolaCódigo. *Analizador Lexico con JFlex en Java (NetBeans)*. 2012. Obtenido de <http://www.youtube.com/watch?v=w-KfjJdRas8>