



UAEM

Universidad Autónoma
del Estado de México



Simulación de Inteligencia Artificial Unidad 1. Nociones Básicas de Lógica

Centro Universitario UAEM Valle de México



Fecha de elaboración: Julio 2016
Semestre: Agosto – Diciembre 2016

Licenciatura en Informática Administrativa

Ph. D. Victor Manuel Landassuri Moreno
vmlandassurim@uaemex.mx
landassuri@gmail.com

Índice de Contenidos

- Visión global de la Unidad de Aprendizaje (UDA)
- Descripción de la unidad de aprendizaje
- Unidad 1. Nociones Básicas de Lógica
- Resumen
- Guión Explicativo
- Bibliografía



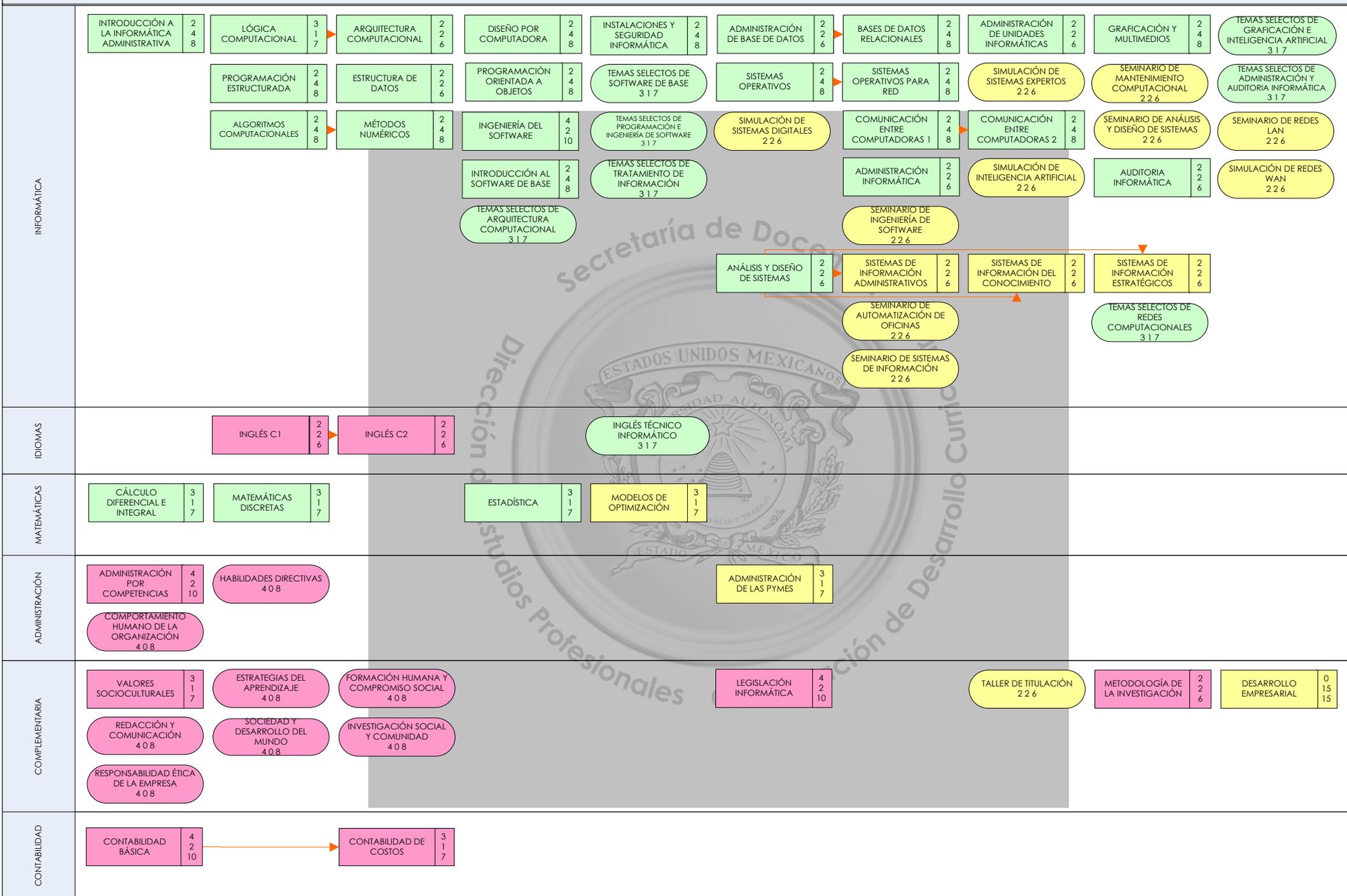
**Programa de Estudio por Competencias
 SIMULACIÓN DE INTELIGENCIA ARTIFICIAL**

1. IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

ORGANISMO ACADÉMICO: Facultad de Contaduría y Administración.								
Programa Educativo: Licenciatura en Informática Administrativa.					Área de docencia: Informática			
Aprobación por los H.H. Consejos Académico y de Gobierno			Fecha: 21 y 27 Marzo 2007		Programa elaborado por: M en Sc. Román Ramírez Nájera MA Ed. Ma. de la Luz Sánchez Paz		Fecha de elaboración 15 febrero 2007	
Clave	Horas de teoría	Horas de práctica	Total de horas	Créditos	Tipo de Unidad de Aprendizaje	Carácter de la Unidad de Aprendizaje	Núcleo de formación	Modalidad
L30060	2	2	4	6	T	optativa	integral	Presencial
Prerrequisitos					Unidad de Aprendizaje		Unidad de Aprendizaje Consecuente:	
Programas educativos en los que se imparte: Licenciatura en Informática Administrativa								

MAPA CURRICULAR DE LA LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA 2003

CRÉDITOS TOTALES: 400



FINANZAS

ANÁLISIS Y PLANEACIÓN FINANCIERA	3 1 7
----------------------------------	-------------

PROYECTOS DE INVERSIÓN	3 1 7
------------------------	-------------

ECONOMÍA

MICROECONOMÍA	3 1 7	MACROECONOMÍA	4 2 10
---------------	-------------	---------------	--------------

SIMBOLOGÍA

8 ÁREAS CURRICULARES

HT HORAS TEÓRICAS
HP HORAS PRÁCTICAS
CR CRÉDITOS

11 LÍNEAS DE SERIACIÓN

NÚCLEO BÁSICO OBLIGATORIAS CURSAR Y ACREDITAR 11 UA	34 HT 18 HP 86 CR
---	-------------------------

NÚCLEO BÁSICO OPTATIVAS ACREDITAR 3 UA PARA CUBRIR 24 CRÉDITOS
--

TOTAL DEL NÚCLEO BÁSICO 14 UA 110 CRÉDITOS
--

NÚCLEO SUSTANTIVO OBLIGATORIAS CURSAR Y ACREDITAR 26 UA	58 HT 76 HP 192 CR
---	--------------------------

NÚCLEO SUSTANTIVO OPTATIVAS ACREDITAR 2 UA PARA CUBRIR 14 CRÉDITOS
--

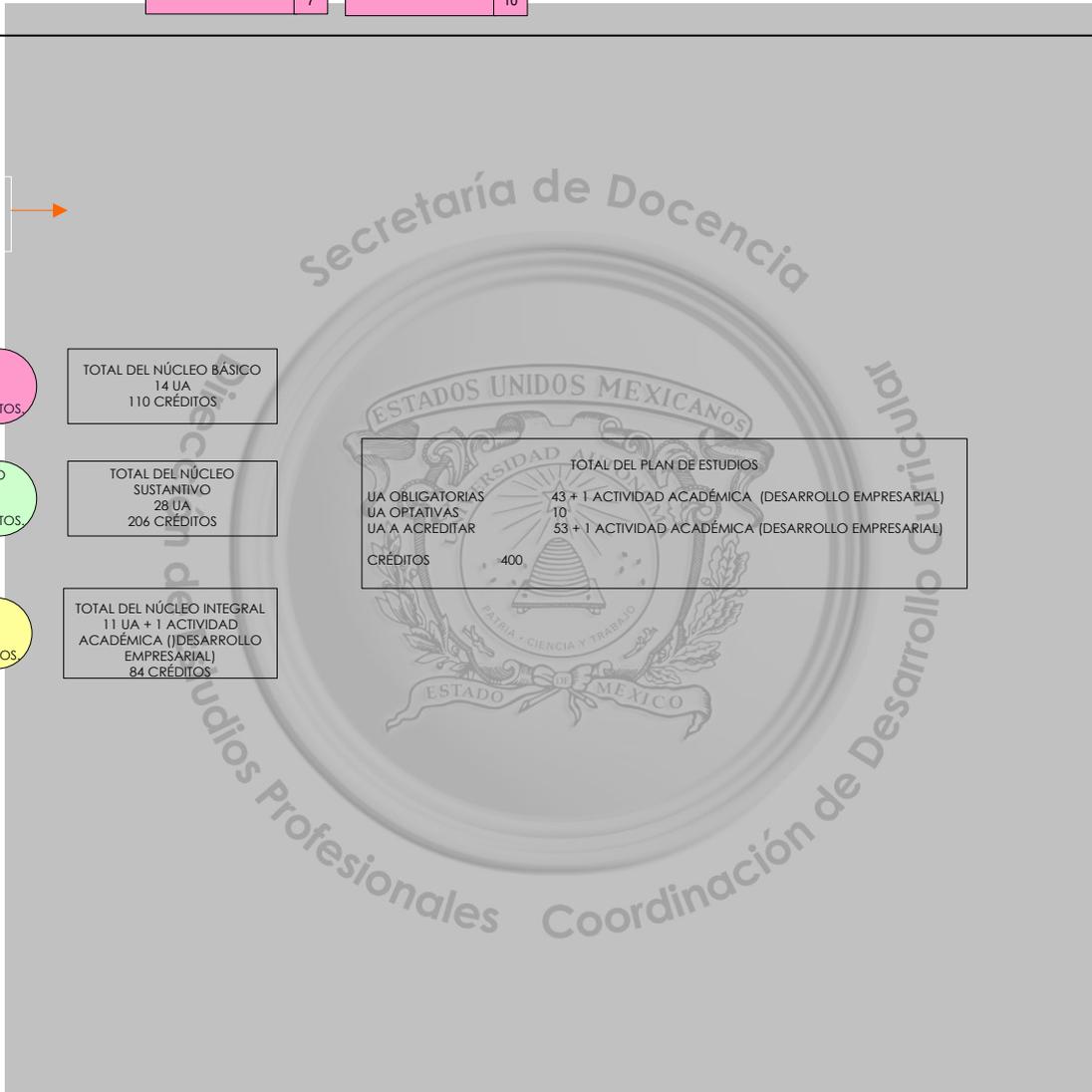
TOTAL DEL NÚCLEO SUSTANTIVO 28 UA 206 CRÉDITOS
--

NÚCLEO INTEGRAL OBLIGATORIAS CURSAR Y ACREDITAR 6 UA + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)	15 HT 24 HP 54 CR
---	-------------------------

NÚCLEO INTEGRAL OPTATIVAS ACREDITAR 5 UA PARA CUBRIR 30 CRÉDITOS
--

TOTAL DEL NÚCLEO INTEGRAL 11 UA + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL) 84 CRÉDITOS
--

TOTAL DEL PLAN DE ESTUDIOS	
UA OBLIGATORIAS	43 + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)
UA OPTATIVAS	10
UA A ACREDITAR	53 + 1 ACTIVIDAD ACADÉMICA (DESARROLLO EMPRESARIAL)
CRÉDITOS	400



Descripción de la unidad de aprendizaje

Identificación del Curso

Licenciatura en Informática Administrativa

Horas de Teoría: 2 hrs.

Horas de Práctica: 2 hrs.

Créditos: 6

Unidad de Aprendizaje Antecedente:

Ninguna

Unidad de Aprendizaje Consecuente:

Ninguna

Presentación

Debido a los requerimientos en el manejo de la información, se hace imprescindible el desarrollo de estrategias y algoritmos que permitan el manejo eficiente de la misma en todas las áreas del saber humano. La inteligencia artificial desarrolla métodos para el manejo y administración de ésta.

Lineamientos

Del Profesor

- **Cubrir el programa en su totalidad**
- **Puntualidad**
- **Asistencia del 100% de las sesiones**
- **Aplicación del marco teórico y práctico de lo visto en una sesión de clase proporcionalmente a un 33% de peso en el tiempo de la sesión, de tal forma que se pueda asegurar el aprendizaje y puesta en práctica de cada alumno.**
- **Coordinar asistencia a una exposición nacional de equipo de cómputo**
- **Elaborar y publicar material didáctico en el portal académico de la universidad**

Del Alumno

- **Asistencia mínima de un 80% del curso.**
- **Puntualidad con un máximo de tolerancia de 15 minutos**
- **Elaborar y cumplir los trabajos teóricos y prácticos que se soliciten**
- **Propuesta de talleres externos sobre el armado de computadoras**
- **Asistir una exposición nacional de equipo de cómputo**
- **Consultas en el portal de Internet de la Universidad**
- **Comunicarse por lo menos en dos foros de consulta en el portal académico de la Universidad**
- **Contar con una cuenta de correo electrónico para intercambio de material didáctico de la materia**

Propósito

El propósito será la simulación de los sistemas de inteligencia artificial para proponer soluciones prácticas a problemas reales.

Competencias genéricas

Identificar los componentes y funcionamiento de un sistema de simulación de inteligencia artificial.

Ámbito de desempeño

El proceso de enseñanza-aprendizaje se realizará mediante sesiones de explicación en clase, ejercicios grupales y realización de trabajos extraclase que complementen lo enseñado en el aula. Foros y consulta de materiales en internet.

Estructura

Unidad 1.

- Nociones Básicas de Lógica

Unidad 2.

- Representación del Conocimiento

Unidad 3.

- Introducción al lenguaje Prolog

Unidad 4.

- Introducción al lenguaje Clips

Unidad 5.

- Aplicaciones de Inteligencia Artificial

Estructura por unidad

- **Unidad 1.** Nociones Básicas de Lógica
 - *Tablas de verdad*
 - *Semántica del lenguaje*
 - *Formulación de Oraciones Lógicas*
- **Unidad 2.** Representación del Conocimiento
 - *Algoritmos de búsqueda de datos*
 - *Algoritmos de ubicación de datos*
 - *Máquina de inferencia*

Estructura por unidad

- **Unidad 3.** Introducción al lenguaje Prolog
 - *Conocimiento del sistema operativo de la PC*
 - *Operación básica de la PC*
 - *Instalación del software*
 - *Comandos básicos del lenguaje*
- **Unidad 4.** Introducción al lenguaje Clips
 - *Instalación del software*
 - *Conceptos básicos del lenguaje*

Estructura por unidad

- **Unidad 5.** Aplicaciones de la Inteligencia Artificial
 - *Desarrollo de Modelos de Simulación de Inteligencia Artificial*

Procedimientos de Evaluación

2 Exámenes escritos	40%
Proyecto	40%
Trabajo, investigaciones y tareas	20%

Unidad 1.

Nociones Básicas de Lógica

Contenido

1. Nociones Básicas de Lógica

1.1 Tablas de verdad

1.2 Semántica del lenguaje

1.3 Formulación de Oraciones Lógicas

1.1 Tablas de Verdad

La **LÓGICA** es una **CIENCIA FORMAL** así como una rama de la filosofía que **ESTUDIA** los **PRINCIPIOS** de la **DEMOSTRACIÓN** e **INFERENCIA** válida.

Así mismo trata los métodos de razonamiento útiles al momento de analizar y programar problemas de la vida real en la computadora.

Ésta proporciona las **REGLAS** para **DETERMINAR SI** un **ARGUMENTO ES VÁLIDO** o se cumple en determinada situación. Por ejemplo en las matemáticas sirve para la demostración de teoremas y en las ciencias de la computación (donde se aplica la Inteligencia Artificial) sirve para determinar si un programa es correcto o no.

1.1 Tablas de Verdad

Una oración que tenga la propiedad de ser falsa o verdadera y sólo una de estas posibilidades se llamará **proposición lógica o proposición**.

La *verdad* o *falsedad* de una proposición es llamada su **valor de verdad**. Si una proposición lógica es *verdadera* diremos que su valor de verdad es V o 1 ; si es *falsa*, su valor de verdad es F o 0 .

1.1 Tablas de verdad

Ejemplos:

- a) *Washington, D.C., es la capital de los Estados Unidos*, es una proposición porque se comprueba que es una declaración Verdadera.
- b) $x > 3$, no es una proposición, porque no se puede determinar si es Verdadera o Falsa. A estos enunciados se les denomina **expresiones abiertas**.
- c) $2 + 2 = 5$, es una proposición y es falsa.
- d) *Esta proposición es falsa*, no es una proposición. Es una **paradoja** ya que si suponemos que es Verdadera, entonces es Falsa. Y si suponemos que es Falsa, entonces es Verdadera.

1.1 Tablas de Verdad

Una proposición puede considerarse como un caso particular de una expresión abierta.

Dentro de las matemáticas, no todas las proposiciones juegan un papel similar. En algunos casos, la veracidad de las proposiciones es postulada y aceptada, y se llama **axioma**; en otras, su veracidad debe ser demostrada mediante un procedimiento aceptado dentro de dicha teoría, y se le llama **teorema**; otras proposiciones establecen propiedades de la propia teoría o sistema, o relaciones entre teorías o sistemas y se les llama **metateorema**.

1.1 Tablas de Verdad

Para denotar proposiciones usaremos letras, y las denominaremos **PROPOSICIONES ATÓMICAS**.

Con las proposiciones podemos construir proposiciones compuestas usando **CONNECTIVOS LÓGICOS**, que son:

\wedge (*and*)

\vee (*o*)

\Rightarrow (*si...entonces*)

\neg (*no*)

\Leftrightarrow (*si y solo sí*)

1.1 Tablas de Verdad

LA CONJUNCIÓN se representa por “ \wedge ”, está compuesta por dos o más oraciones. Aquí, ambas condiciones tienen que ser verdaderas para que el resultado de la conjunción sea verdadero.

Si p y q son proposiciones, la conjunción de p y q se denota $p \wedge q$ y se lee p y q .

1.1 Tablas de Verdad

LA DISYUNCIÓN se representa por “ \vee ”, está compuesta por dos o más oraciones. Aquí, si al menos una de las partes es verdadera, el resultado será verdadero

Si p y q son proposiciones, la conjunción de p y q se denota $p \vee q$ y se lee p ó q .

1.1 Tablas de Verdad

En lógica distinguiremos dos tipos de disyunciones: la forma **inclusiva** y la forma **exclusiva**.

En el caso de la forma exclusiva es no pueden suceder ambas partes de proposición. Por ejemplo:

El día de mañana será frío o caluroso

Se sugiere que habrá frío o calor, pero no ambas.

1.1 Tablas de Verdad

En el caso de la forma inclusiva ambas pueden suceder.
Por ejemplo:

7 es primo o impar

Se sugiere que el número 7 podría satisfacer ambos criterios y por lo tanto la disyunción es inclusiva.

1.1 Tablas de Verdad

Si $p \vee q$ son proposiciones:

La disyunción inclusiva de p y q se denota $p \vee q$ y se lee p o q .

La disyunción exclusiva de p y q se denota $p \underline{\vee} q$ y se lee p o q .

1.1 Tablas de Verdad

LA NEGACIÓN se representa por “ \neg ”, y se antepone a la proposición. Si p es una proposición, su negación se denota $\neg p$ y se lee *no p*.

LA CONDICIONAL es una proposición compuesta por dos o más oraciones y se usa el símbolo “ \Rightarrow ”, donde el resultado es falso si únicamente la primera proposición es verdadera y la segunda falsa.

Si p y q son proposiciones, la condicional de p y q se denota $p \Rightarrow q$ y se lee *si p entonces q*.

1.1 Tablas de Verdad

Existen diversos tipos de condicional:

Contraria: $\neg p \Rightarrow \neg q$

Recíproca: $q \Rightarrow p$

Contrarecíproca: $\neg q \Rightarrow \neg p$

1.1 Tablas de Verdad

LA BICONDITIONAL es una proposición compuesta por dos o más proposiciones y se usa el símbolo “ \Leftrightarrow ”, donde el resultado es verdadero si ambas partes tienen el mismo valor.

Si p y q son proposiciones, la bicondicional de p y q se denota $p \Leftrightarrow q$ y se lee *si p si y sólo si q* .

1.1 Tablas de Verdad

Una **TABLA DE VERDAD** es una tabla donde se comparan las proposiciones (todas las combinaciones de valores de cada una de ellas) para determinar si es verdadera o falsa.

Si todos los valores de las proposiciones son verdaderas se denomina **TAUTOLOGÍA**, si son falsos **CONTRADICCIÓN**, de lo contrario se denomina como **INDETERMINADA O CONTINGENCIA**.

1.1 Tablas de Verdad

Si la proposición consta de n proposiciones simples, entonces se tendrán 2^n combinaciones posibles, dado que sólo pueden tener verdadero o falso. Esto es:

Diagram illustrating the construction of truth tables for two and three variables.

For two variables (p and q), the number of possible combinations is $2^2 = 4$.

	p	q
	V	V
	V	F
	F	V
	F	F

For three variables (p , q , and r), the number of possible combinations is $2^3 = 8$.

	p	q	r
	V	V	V
	V	V	F
	V	F	V
	V	F	F
	F	V	V
	F	V	F
	F	F	V
	F	F	F

1.1 Tablas de Verdad

Tabla 1. Tabla de Verdad de la **DISYUNCIÓN** entre dos proposiciones

p	q	$p \vee q$
V	V	V
V	F	F
F	V	F
F	F	F

Tabla 2. Tabla de Verdad de la **CONJUNCIÓN** entre dos proposiciones

p	q	$p \wedge q$
V	V	V
V	F	V
F	V	V
F	F	F

1.1 Tablas de Verdad

Tabla 3. Tabla de Verdad
Condicional

p	q	$p \Rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Tabla 4. Tabla de Verdad
Bicondicional

p	q	$p \Leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

2.1 Tablas de Verdad

Tabla 5. Tabla de Verdad de la **CONJUNCIÓN EXCLUSIVA** entre dos proposiciones

p	q	$p \underline{\vee} q$
V	V	F
V	F	V
F	V	V
F	F	F

ó

Tabla 6. Tabla de Verdad de la **CONJUNCIÓN EXCLUSIVA** entre dos proposiciones

p	q	$p \oplus q$
V	V	F
V	F	V
F	V	V
F	F	F

1.1 Tablas de Verdad

Tabla 7. Tabla de Verdad de la
NEGACIÓN

p	$\neg p$
V	F
F	V

1.1 Tablas de Verdad

Sean p y q son proposiciones, diremos que p y q son lógicamente *equivalentes*, si $p \Leftrightarrow q$ es una **TAUTOLOGÍA**.

La equivalencia lógica de p y q se denota $p \equiv q$ ó $p \vdash q$

p	q	$\neg p$	$\neg q$	$\neg(p \wedge q)$	$(\neg p \vee \neg q)$
V	V	F	F	F	F
V	F	F	V	V	V
F	V	V	F	V	V
V	F	V	V	V	V

1.1 Tablas de Verdad

Equivalencia	Nombre
$p \wedge V \equiv p$ $p \wedge F \equiv p$ $p \vee V \equiv V$ $p \wedge F \equiv F$	Leyes de identidad
$\neg(\neg p) \equiv p$	Leyes de involución
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$ $p \oplus q \equiv q \oplus p$ $p \leftrightarrow q \equiv q \leftrightarrow p$	Leyes conmutativas
$[(p \vee q) \vee r] \equiv [p \vee (q \vee r)]$ $[(p \oplus q) \oplus r] \equiv [p \oplus (q \oplus r)]$ $[(p \wedge q) \wedge r] \equiv [p \wedge (q \wedge r)]$ $[(p \leftrightarrow q) \leftrightarrow r] \equiv [p \leftrightarrow (q \leftrightarrow r)]$	Leyes asociativas

1.1 Tablas de Verdad

Equivalencia	Nombre
$p \wedge p \equiv p$ $p \vee p \equiv p$	Leyes de idempotencia
$[p \wedge (q \vee r)] \equiv [(p \wedge q) \vee (p \wedge r)]$ $[p \vee (q \wedge r)] \equiv [(p \vee q) \wedge (p \vee r)]$	Leyes distributivas
$\neg(p \vee q) \equiv (\neg p \wedge \neg q)$ $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$	Leyes De Morgan
$[p \wedge (p \vee q)] \equiv p$ $[p \vee (p \wedge q)] \equiv p$	Leyes de absorción
$p \vee \neg p \equiv V$ $p \wedge \neg p \equiv F$ $\neg F \equiv V$ $\neg V \equiv F$	Leyes de complemento

1.1 Tablas de Verdad

Tabla 8. Equivalencias lógicas para proposiciones condicionales

$$p \Rightarrow q \equiv \neg p \vee q$$

$$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$$

$$p \Rightarrow q \equiv \neg p \Rightarrow q$$

$$p \Rightarrow q \equiv \neg(p \Rightarrow q)$$

$$\neg(p \Rightarrow q) \equiv p \wedge \neg q$$

$$(p \Rightarrow q) \wedge (p \Rightarrow r) \equiv p \Rightarrow (q \wedge r)$$

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \equiv (p \vee q) \Rightarrow r$$

$$(p \Rightarrow q) \vee (p \Rightarrow r) \equiv p \Rightarrow (q \vee r)$$

$$(p \Rightarrow r) \vee (q \Rightarrow r) \equiv (p \wedge q) \Rightarrow r$$

1.1 Tablas de Verdad

Tabla 9. Equivalencias lógicas para proposiciones bicondicionales

$$\begin{aligned}p \Leftrightarrow q &\equiv (p \Rightarrow q) \wedge (q \Rightarrow p) \\p \Leftrightarrow q &\equiv \neg p \Leftrightarrow \neg q \\p \Leftrightarrow q &\equiv (p \wedge q) \vee (\neg p \wedge \neg q) \\ \neg(p \Leftrightarrow q) &\equiv p \Leftrightarrow \neg q\end{aligned}$$

1.2 Semántica del lenguaje

Un Lenguaje de programación (LP) debe ser definido como una notación formal para describir algoritmos ejecutables por una computadora. Donde este posee una **Sintaxis** la cual es un conjunto de reglas formales que especifican la composición de programas a partir de ciertas unidades de construcción, y una **Semántica** la cual especifica el significado de un programa sintácticamente válido cuando es escrito en un lenguaje de programación determinado.

1.2 Semántica del lenguaje

```
int main ()
{
    int i, j;
    Get (i, j);
    while (i != j )
        if (i > j)
            i -= j;
        else
            j -= i;
    printf ("%d\n", i);
}
```

1.2 Semántica del lenguaje

La semántica del LP nos indica que se reserva espacio para dos elementos del tipo entero (i y j).

Por otro lado, se tiene que el ciclo del **while** se estará evaluando mientras la expresión ($i \neq j$) se cumpla.

Así es como se pueden hacer las preguntas de:

- ¿Es válido este programa?
- ¿Qué significa?
- ¿Cómo hacer un compilador para saber cómo traducir el programa?

1.2 Semántica del lenguaje

De esta forma, se debe de escribir un LP de forma que esté definido con el suficiente nivel para poder determinar:

**Si un programa es válido (Sintaxis) entonces,
¿Cuál es su significado o efecto? (Semántica).**

Así como podemos determinar cuál es la semántica de cada sección del programa, también lo podemos hacer para la semántica completa de este. Es decir, podemos preguntarnos qué es lo que hace cada sección del programa y para qué sirve todo el programa por completo.

1.2 Semántica del lenguaje

El significado puede expresarse mediante una asignación de cada constructor del lenguaje en un dominio cuya semántica sea conocida. Así se tiene diversas clasificaciones de semánticas:

- **Semántica denotacional:** Las denotaciones son considerados objetos matemáticos por las cuales cada frase de un lenguaje es interpretado. Sin embargo, no es un requerimiento que sean objetos matemáticos.
- **Semántica operacional:** aquí no se hace una traducción como en el anterior, en cambio, la ejecución del programa se describe directamente. Estas semánticas pueden definir una maquina abstracta.
- **Semántica Axiomática:** Los axiomas lógicos en ellas son descritos al darle significado a las frases que las describe.

1.3 Formulación de Oraciones Lógicas

Las proposiciones simples comentadas anteriormente, deben ser atómicas o simples, lo que significa que no se pueden descomponer en más oraciones simples. En su contraparte, se tiene las moleculares, o complejas que si se pueden descomponer. Como se comentó anteriormente, una proposición compuesta es el resultado de juntar dos o más proposiciones simples con un operador (AND, OR, u otro).

Algunos de los elementos de la lógica de enunciados (oraciones) son: variables, constantes y auxiliares.

1.3 Formulación de Oraciones Lógicas

- **Variables:** las variables son símbolos que se sustituyen proposiciones o enunciados, y como su nombre lo indica pueden ir cambiando. Usualmente se usan las literales: p, q, r, s, t . Pero si se llegase a dar el caso de requerir más de estas, se pueden utilizar ellas mismas con subíndices $p_1, q_1, r_1, s_1, t_1, \dots, p_n, q_n, r_n, s_n, t_n$. Por ejemplo la letra p podría simbolizar: la tierra es plana ($p =$ la tierra es plana). El uso de mayúsculas en los símbolos, suele implicar, metalenguajes, donde cada variable puede expresar cualquier proposición, atómica o molecular.

1.3 Formulación de Oraciones Lógicas

Constantes: son los conectores proposicionales que tienen la función de alterar, relacionar así como conectar enunciados atómicos haciéndolos compuestos. Estos fueron comentados anteriormente y son la negación, conjunción, el condicional, el bicondicional y la disyunción.

Símbolos auxiliares: Estos son los paréntesis y corchetes, y por si solos no tienen ningún significado lógico, pero sirven para clarificar la comprensión de las proposiciones compuestas.

1.3 Formulación de Oraciones Lógicas

Ahora, para formar oraciones lógicas, se tienen que seguir ciertas reglas, es decir, que no toda expresión es admitida como fórmula bien realizada, donde una fórmula es una secuencia ordenada de símbolos.

Una fórmula es una formula (expresión) bien formada (FBF o EBF), si cumple con alguna de las siguientes características:

1.3 Formulación de Oraciones Lógicas

- Una proposición sola, es una FBF
- Una FBF precedida por una negación es una FBF
- Una FBF seguida por cualquiera de las constantes, seguida de una FBF, haciendo buen uso de los paréntesis es una FBF

1.3 Formulación de Oraciones Lógicas

Veamos un ejemplo de cómo se forma una proposición y como distinguir si es correcta: Si p es la proposición “hace frío” y q es la proposición “llueve” la proposición simbólica $\neg p \wedge q$ puede traducirse como:

- a) No hace frío pero llueve.
- b) Hace frío y no llueve.
- c) No llueve y no hace frío.

1.3 Formulación de Oraciones Lógicas

Solución: Si p es la proposición “hace frío”, la proposición $\neg p$ es “no hace frío”. La proposición “no hace frío pero llueve” es equivalente a decir “no hace frío y llueve”, es decir la conjunción de $\neg p$ y q que se expresa así: $\neg p \wedge q$. Por lo tanto la opción correcta es la a)

1.3 Formulación de Oraciones Lógicas

Un sistema formal de razonamiento, también llamado una *Teoría Formal*, está compuesto de tres elementos:

- Un conjunto de FBFs
- Un conjunto de axiomas
- Un conjunto de reglas de inferencia

Las FBFs ya son conocidas. Los axiomas, son FBFs que se sabe previamente que son tautologías, o bien, a las que asignaremos arbitrariamente el valor de verdad V para establecer condiciones de inicio del problema.

1.3 Formulación de Oraciones Lógicas

Por último, una regla de inferencia, es el mecanismo que permite llegar a conclusiones en un razonamiento. Formalmente, una regla de inferencia es un criterio que asocia una o más FBFs, llamadas **antecedentes** o **hipótesis**, con una única FBF llamada **consecuente** o **conclusión**.

En general, si R es una regla de inferencia que asocia las hipótesis h_1, h_2, \dots, h_N con la conclusión C , la cual denotaremos como $R(h_1, h_2, \dots, h_N) = C$, entonces se dice que “a partir de h_1, h_2, \dots, h_N se infiere C ”

1.3 Formulación de Oraciones Lógicas

Reglas de inferencia

Conjunción

$$\frac{a, b}{\therefore a \wedge b}$$

Simplificación

$$\frac{a \wedge b}{\therefore a}$$

Agregación

$$\frac{a}{\therefore a \vee b}$$

Modus Ponens

$$\frac{a \Rightarrow b, a}{\therefore b}$$

Modus Tollens

$$\frac{a \Rightarrow b, \neg b}{\therefore \neg a}$$

1.3 Formulación de Oraciones Lógicas

Reglas de inferencia

Silogismo Hipotético

$$\frac{a \Rightarrow b, b \Rightarrow c}{\therefore a \Rightarrow c}$$

Silogismo Disyuntivo

$$\frac{a \vee b, \neg a}{\therefore b}$$

1.3 Formulación de Oraciones Lógicas

Una **demostración** es una secuencia finita de FBFs en la cual cada elemento de la secuencia, o bien es un axioma, o bien, puede ser inferida a partir de algunas FBFs previas en la misma secuencia.

La primera expresión de una demostración es siempre un axioma, y la última expresión se denomina **teorema**.

1.3 Formulación de Oraciones Lógicas

Por ejemplo, un razonamiento *modus ponens* sería:

Si está soleado, entonces es de día

Esta soleado.

Por lo tanto es de día.

Expresado más formalmente el *modus ponens*, queda:

Si A, entonces B

A

Por lo tanto, B


$$a \Rightarrow b$$
$$\frac{a}{\therefore b}$$

Resumen

- Aquí se pudo observar la lógica proposicional, la cual nos permite aplicarlo a tablas de verdad
 - Lógica requerida en lenguajes de programación
 - Sobre todo para evaluar expresiones booleanas (*es decir, expresiones que regresan falso o verdadero*)
- También se comentó sobre la Semántica del lenguaje y la formulación de oraciones lógicas, mostrando su importancia.
- Aspectos considerados como el inicio para posteriormente ahondar en conceptos de Inteligencia Artificial

Guión Explicativo

- Las diapositivas deben leerse en el orden que aparecen
- Aquí fue presentada una introducción a la Unidad I de Aprendizaje
- El siguiente juego de diapositivas, debe iniciar con la Unidad II. Representación del Conocimiento. Algoritmos y Búsqueda de datos.

Bibliografía Básica

- Allen Collin, Hand Michael, Logic Primer, ed. MIT Press, 2001
- Giarratano Joseph C., Riley Gary, Expert Systems: Principles and Programming, ed. Thompson, 2005
- Rosen Kenneth H., Discrete Mathematics and its Applications, ed. McGraw Hill, 2007
- Johnsonbaugh Richard, Matematicas Discretas, ed. Prentice Hall, 2005