

Universidad Michoacana de San Nicolás de
Hidalgo

Facultad de Ingeniería Eléctrica

Ingeniería en Computación

Proyecto Final
Modelos Probabilísticos

Implementación de Algoritmos para
Modelos Gráficos Probabilísticos

Integrantes:

Abraham Flores Avila
2100679B

Profesor:

Dr. Mauricio Reyes

Morelia, Michoacán
Noviembre–Diciembre 2025

Resumen

Este proyecto presenta la implementación completa de un sistema web para el análisis y visualización de Modelos Gráficos Probabilísticos, desarrollado íntegramente en PHP puro sin dependencias de frameworks externos. El sistema implementa **6 algoritmos fundamentales** distribuidos en tres módulos principales: Redes Bayesianas, Cadenas de Markov y Modelos Ocultos de Markov (HMM).

La arquitectura del sistema utiliza el servidor integrado de PHP, eliminando la necesidad de configurar Apache, Nginx o servidores web externos, lo que garantiza portabilidad total y facilidad de despliegue. La interfaz web, desarrollada con HTML5, CSS3 y JavaScript puro, proporciona visualización interactiva de grafos mediante vis.js y permite la manipulación intuitiva de modelos probabilísticos.

Entre las características principales se incluyen: inferencia exacta en Redes Bayesianas mediante enumeración y eliminación de variables, cálculo de distribuciones estacionarias en Cadenas de Markov, y decodificación de secuencias en HMM utilizando el algoritmo de Viterbi. El sistema implementa técnicas avanzadas de estabilidad numérica mediante aritmética logarítmica para evitar problemas de underflow en probabilidades pequeñas.

El proyecto incluye cuatro ejemplos demostrativos completamente funcionales, documentación técnica exhaustiva, y un conjunto de pruebas que validan la correctitud de los algoritmos implementados. Los resultados obtenidos muestran que es posible implementar algoritmos probabilísticos complejos en PHP manteniendo precisión numérica y rendimiento adecuados para fines educativos y de investigación.

Palabras clave: Redes Bayesianas, Cadenas de Markov, Modelos Ocultos de Markov, Inferencia Probabilística, PHP, Algoritmos de Grafos, Modelos Gráficos.

Índice general

Resumen	1
Lista de Figuras	4
Lista de Tablas	5
1. Manual de Usuario	6
1.1. Introducción	6
1.1.1. Alcance del Proyecto	6
1.2. Requisitos del Sistema	6
1.2.1. Software Necesario	7
1.2.2. Verificación del Entorno	7
1.3. Instalación	7
1.3.1. Descarga del Proyecto	7
1.3.2. Estructura del Proyecto	8
1.4. Ejecución	10
1.4.1. Servidor Integrado de PHP (Recomendado)	10
1.5. Guía de Uso	10
1.5.1. Módulo: Redes Bayesianas	11
1.5.2. Módulo: Cadenas de Markov	11
1.5.3. Módulo: HMM	12
1.6. Ejemplos Pre-cargados	12
1.6.1. Ejemplos de Redes Bayesianas	12
1.6.2. Ejemplos de Cadenas de Markov	12
1.6.3. Ejemplos de HMM	13
2. Explicación de los Algoritmos	14
2.1. Redes Bayesianas	14
2.1.1. Algoritmo de Enumeración	14
2.1.2. Eliminación de Variables	14
2.2. Algoritmos de Cadenas de Markov	14
2.2.1. First Order Markov Chain	14
2.2.2. Transition Matrix Calculator	14
2.2.3. Stationary Distribution	15
2.3. Modelos Ocultos de Markov	15
2.3.1. Algoritmo Viterbi	15

3. Decisiones de Diseño	16
3.1. Arquitectura del Sistema	16
3.1.1. Backend: PHP Puro	16
3.2. Representación de Datos	16
3.2.1. Redes Bayesianas	16
4. Conclusiones y Trabajo Futuro	17
4.1. Limitaciones Actuales	17
4.2. Reflexión Final	17
A. Instalación de PHP por Sistema Operativo	18
A.1. Windows	18
B. Generación del Árbol de Archivos	19
C. Referencias Bibliográficas	20

Índice de figuras

Índice de tablas

1.1. Requisitos de software del sistema	7
1.2. Algoritmos disponibles en el módulo HMM	12

Manual de Usuario

1.1 Introducción

Este manual proporciona las instrucciones necesarias para instalar, configurar y utilizar el sistema de Modelos Probabilísticos desarrollado como proyecto final de la materia.

Nota

Este proyecto implementa **6 algoritmos fundamentales** distribuidos en tres módulos principales: Redes Bayesianas (RB), Cadenas de Markov (CM) y Modelos Ocultos de Markov (HMM). El sistema está desarrollado completamente en **PHP puro**, sin frameworks externos, utilizando únicamente el servidor integrado de PHP.

1.1.1 Alcance del Proyecto

Este sistema permite:

- **Redes Bayesianas:** Inferencia exacta mediante enumeración y eliminación de variables
- **Cadenas de Markov:** Análisis de transiciones y cálculo de distribuciones estacionarias
- **Modelos Ocultos de Markov:** Decodificación de secuencias (Viterbi)
- **Visualización interactiva:** Grafos dinámicos con vis.js
- **Ejemplos predefinidos:** Casos de uso reales para cada módulo

1.2 Requisitos del Sistema

1.2.1 Software Necesario

Componente	Versión Mínima	Recomendada
PHP	7.4	8.0 o superior
Navegador Web	Chrome 90+ / Firefox 88+	Última versión
Git (opcional)	2.0+	Última versión
Conexión Internet	Requerida (CDN)	Banda ancha
Sistema Operativo	Windows 10 / Linux / macOS	Cualquiera

Tabla 1.1: Requisitos de software del sistema

Importante

No se requiere: Apache, Nginx, XAMPP, MAMP ni ningún otro servidor web externo. El sistema utiliza exclusivamente el servidor integrado de PHP (`php -S`).

1.2.2 Verificación del Entorno

Abra una terminal y ejecute:

```
alumno@DESKTOP-J4QHEU8:~
```

```
# Verificar PHP
alumno@DESKTOP-J4QHEU8:~$ php -v
PHP 8.0.30 (cli) (built: Aug 23 2023 12:00:00) ( NTS )
Copyright (c) The PHP Group
Zend Engine v4.0.30, Copyright (c) Zend Technologies

# Verificar Git (opcional)
alumno@DESKTOP-J4QHEU8:~$ git --version
git version 2.25.1
```

Advertencia

Si PHP no está instalado, consulte el capítulo [A](#) para instrucciones de instalación según su sistema operativo (Windows, Linux, macOS).

1.3 Instalación

1.3.1 Descarga del Proyecto

Opción 1: Clonar desde Git

```
alumno@DESKTOP-J4QHEU8:~  
  
# Clonar el proyecto  
alumno@DESKTOP-J4QHEU8:~$ git clone https://github.com/  
    usuario/ProyectoFinal_ModelosProbabilistas2526.git  
Cloning into 'ProyectoFinal_ModelosProbabilistas2526'...  
remote: Enumerating objects: 120, done.  
remote: Counting objects: 100% (120/120), done.  
remote: Compressing objects: 100% (95/95), done.  
Receiving objects: 100% (120/120), 1.2 MiB | 2.5 MiB/s,  
    done.  
  
# Entrar al directorio  
alumno@DESKTOP-J4QHEU8:~$ cd  
    ProyectoFinal_ModelosProbabilistas2526  
alumno@DESKTOP-J4QHEU8:~/  
    ProyectoFinal_ModelosProbabilistas2526$
```

Opción 2: Descarga Directa

1. Visite el repositorio en GitHub
2. Haga clic en el botón **Code** → **Download ZIP**
3. Extraiga el archivo en la ubicación deseada
4. Navegue al directorio extraído

Opción 3: Script Automático (Windows)

Si está en Windows, puede usar el script de configuración incluido:

```
alumno@DESKTOP-J4QHEU8:~/Proyecto  
  
# Ejecutar script de configuracion  
alumno@DESKTOP-J4QHEU8:~/Proyecto$ crear_estructura.bat  
# El script creara directorios faltantes y verificara  
    archivos
```

Nota

Este script es útil si descargó el proyecto como ZIP y algunos directorios no se crearon correctamente.

1.3.2 Estructura del Proyecto

```
ProyectoFinal_ModelosProbabilistas2526/
|-- index.php                # Pagina principal
|-- config.php              # Configuracion global
|-- README.md               # Documentacion
|-- assets/                 # Recursos estaticos
|   |-- css/                # Hojas de estilo
|       |-- style.css       # Estilos globales
|       |-- bayesian.css
|       |-- markov.css
|       |-- 'hmm.css
|   |-- js/                 # Scripts JavaScript
|       |-- main.js
|       |-- visualization.js
|       |-- bayesian.js
|       |-- markov.js
|       |-- 'hmm.js
|-- modules/                # Modulos de algoritmos
|   |-- bayesian/           # Redes Bayesianas
|       |-- index.php
|       |-- network.php
|       |-- inference.php
|       |-- save.php
|       |-- visualization.php
|       |-- algorithms/     # Algoritmos especificos
|           |-- enumeration.php
|           |-- 'variable_elimination.php
|       |-- 'examples/      # Ejemplos JSON
|           |-- alarm.json
|           |-- medical.json
|           |-- diagnostic.json
|           |-- 'weather.json
|   |-- markov/             # Cadenas de Markov
|       |-- index.php
|       |-- chain.php
|       |-- stationary.php
|       |-- visualization.php
|       |-- algorithms/
|           |-- first_order.php
|           |-- stationary.php
|           |-- 'transition_matrix.php
|       |-- 'examples/
|           |-- weather.json
|           |-- customer_behavior.json
|           |-- stock_market.json
|           |-- 'text_generation.json
|-- 'hmm/                   # Modelos HMM
|   |-- index.php
|   |-- model.php
|   |-- visualization.php
```

```
|      |-- algorithms/
|      |      '-- viterbi.php
|      '-- examples/
|          |-- weather_prediction.json
|          |-- speech_recognition.json
|          |-- bioinformatics.json
|          '-- stock_trading.json
|-- includes/                # Archivos PHP comunes
|   |-- header.php
|   |-- footer.php
|   |-- navbar.php
|   '-- functions.php
|-- docs/                    # Documentacion
|   '-- proyecto_final_corregido.tex
'-- crear_estructura.bat      # Script de instalacion (Windows)
```

Listing 1.1: Estructura de directorios

1.4 Ejecución

1.4.1 Servidor Integrado de PHP (Recomendado)

La forma más sencilla y portable es usar el servidor web integrado de PHP:

alumno@DESKTOP-J4QHEU8:~/Proyecto

```
# Navegar al directorio del proyecto
alumno@DESKTOP-J4QHEU8:~$ cd
    ProyectoFinal_ModelosProbabilistas2526

# Iniciar servidor en el puerto 8000
alumno@DESKTOP-J4QHEU8:~/
    ProyectoFinal_ModelosProbabilistas2526$ php -S localhost
:8000
[Mon Dec 08 19:30:00 2025] PHP 8.0.30 Development Server (
    http://localhost:8000) started
```

Importante

Importante: NO cierre esta ventana de terminal mientras utilice el sistema. El servidor se detendrá si cierra la terminal.

Luego abra su navegador en: <http://localhost:8000>

1.5 Guía de Uso

1.5.1 Módulo: Redes Bayesianas

Crear una Nueva Red

1. Seleccione número de nodos (recomendado: 5–15)
2. Defina relaciones padre-hijo haciendo clic en nodos
3. Ingrese probabilidades condicionales (CPT) para cada nodo
4. Valide que las probabilidades sumen 1.0
5. Guarde la red

Realizar Consultas

1. Seleccione variable(s) de consulta
2. Defina evidencia observada (opcional)
3. Elija algoritmo:
 - **Enumeración:** Simple, más lento
 - **Eliminación de Variables:** Más eficiente
4. Presione “Calcular”
5. Revise resultados en distribución de probabilidad

1.5.2 Módulo: Cadenas de Markov

Configuración

1. Especifique número de estados (ej: 3)
2. Ingrese matriz de transición (debe ser estocástica)
3. (Opcional) Defina distribución inicial

Operaciones Disponibles

- **Simular:** Genere secuencias de estados aleatorias
- **Calcular Estacionaria:** Distribución límite π
- **Visualizar Grafo:** Diagrama interactivo de transiciones

1.5.3 Módulo: HMM

Definir Modelo

Configure los siguientes parámetros:

1. **Estados ocultos:** Ej: {Feliz, Triste}
2. **Observaciones posibles:** Ej: {Caminar, Comprar, Limpiar}
3. **Probabilidades iniciales** (π): Distribución inicial de estados
4. **Matriz de transición** (A): Transiciones entre estados ocultos
5. **Matriz de emisión** (B): Probabilidades de observaciones dado estado

Algoritmos Disponibles

Algoritmo	Propósito
Viterbi	Encuentra secuencia de estados más probable: $\arg \max_Q P(Q O, \lambda)$

Tabla 1.2: Algoritmos disponibles en el módulo HMM

1.6 Ejemplos Pre-cargados

El sistema incluye ejemplos demostrativos listos para usar.

1.6.1 Ejemplos de Redes Bayesianas

El sistema incluye cuatro ejemplos predefinidos:

- **Red Alarma-Terremoto-Ladrón** (*alarm.json*): Ejemplo clásico de inferencia causal
- **Red Médica** (*medical.json*): Diagnóstico basado en síntomas
- **Red de Diagnóstico** (*diagnostic.json*): Sistema de control de calidad
- **Red Climática** (*weather.json*): Predicción meteorológica

1.6.2 Ejemplos de Cadenas de Markov

- **Clima** (*weather.json*): Estados Soleado/Nublado/Lluvioso
- **Comportamiento del Cliente** (*customer_behavior.json*): Análisis de compra
- **Mercado de Valores** (*stock_market.json*): Predicción financiera
- **Generación de Texto** (*text_generation.json*): Modelo de lenguaje simple

1.6.3 Ejemplos de HMM

- **Predicción del Clima** (*weather_prediction.json*): Robot infiere clima desde actividades
- **Reconocimiento de Voz** (*speech_recognition.json*): Fonemas ocultos desde audio
- **Bioinformática** (*bioinformatics.json*): Predicción de genes
- **Trading Bursátil** (*stock_trading.json*): Estados del mercado

Explicación de los Algoritmos

2.1 Redes Bayesianas

2.1.1 Algoritmo de Enumeración

Algoritmo

```
1 ENUMERACION-PREGUNTAR(X, e, bn):  
2   Q(X) = distribucion vacia sobre X  
3   para cada valor xi de X:  
4       Q(xi) = ENUMERAR-TODO(bn.VARS, extend(e, X, xi))  
5   retornar NORMALIZAR(Q(X))
```

Listing 2.1: Pseudocódigo: Enumeración en Redes Bayesianas

2.1.2 Eliminación de Variables

Idea Principal

Mejora la eficiencia factorizando y sumando variables una por una en orden estratégico, reutilizando cálculos intermedios.

2.2 Algoritmos de Cadenas de Markov

2.2.1 First Order Markov Chain

Implementación básica de cadenas de Markov de primer orden donde el estado futuro depende únicamente del estado presente.

2.2.2 Transition Matrix Calculator

Cálculo y validación de matrices de transición estocásticas.

2.2.3 Stationary Distribution

Cálculo del vector π tal que $\pi = \pi P$.

2.3 Modelos Ocultos de Markov

2.3.1 Algoritmo Viterbi

Algoritmo Viterbi

Recursión ($2 \leq t \leq T$):

$$\begin{aligned}\delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(o_t) \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}]\end{aligned}\tag{2.1}$$

Decisiones de Diseño

3.1 Arquitectura del Sistema

3.1.1 Backend: PHP Puro

- **Universalidad:** Amplia disponibilidad en servidores
- **Sin dependencias:** No requiere instalación de frameworks
- **Servidor integrado:** Desarrollo rápido con `php -S`

3.2 Representación de Datos

3.2.1 Redes Bayesianas

Clase Nodo

```
1 class Nodo {  
2     public $nombre;  
3     public $padres = [];  
4     public $cpt = []; // Tabla probabilidad condicional  
5 }
```

Listing 3.1: Estructura de Nodo en Red Bayesiana

Conclusiones y Trabajo Futuro

4.1 Limitaciones Actuales

Algoritmos HMM Faltantes

Actualmente solo está implementado **Viterbi**. Los siguientes algoritmos están pendientes:

- **Forward Algorithm** (*algorithms/forward.php*)
- **Backward Algorithm** (*algorithms/backward.php*)
- **Forward-Backward (Baum-Welch)** (*algorithms/baum_welch.php*)

Prioridad: Alta - Estos son esenciales para un módulo HMM completo.

4.2 Reflexión Final

Conclusión Principal

Es completamente viable implementar algoritmos probabilísticos complejos en PHP puro, proporcionando una herramienta educativa robusta y práctica.

Instalación de PHP por Sistema Operativo

A.1 Windows

Visite <https://windows.php.net/download/> y configure el PATH a C:\php.

Generación del Árbol de Archivos

Para generar el árbol de estructura actual del proyecto:

alumno@DESKTOP-J4QHEU8:~/Proyecto

```
# Linux/Mac: Instalar tree
```

```
alumno@DESKTOP-J4QHEU8:~/Proyecto$ sudo apt install tree #  
Ubuntu/Debian
```

```
alumno@DESKTOP-J4QHEU8:~/Proyecto$ brew install tree #  
macOS
```

```
# Generar arbol ignorando node_modules, vendor, etc.
```

```
alumno@DESKTOP-J4QHEU8:~/Proyecto$ tree -I 'vendor|  
node_modules|.git' -L 4 > arbol.txt
```

```
# Windows: Usar PowerShell
```

```
PS C:\Proyecto> tree /F > arbol.txt
```

Referencias Bibliográficas

1. Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson Education.
2. Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.