



Universidad
Zaragoza

TRABAJO FIN DE GRADO

Adaptación del software Elvira a la nube

Autor:

Sergio Langarita Benítez

Directores:

María Piedad Garrido Picazo

Francisco José Martínez Domínguez

Escuela Universitaria Politécnica Teruel

2021

Repositorio de la Universidad de Zaragoza – Zaguán
<http://zaguán.unizar.es>

Tabla de contenido

1. Agradecimientos	4
2. Resumen y Palabras Clave	5
3. Introducción y Objetivos	7
4. Estado del Arte	10
5. Análisis, Diseño e Implementación	13
5.1. Análisis	13
5.2. Diseño	19
5.3. Implementación	20
6. Evaluación	23
6.1. Test de accesibilidad	26
6.2. Navegadores	29
7. Licencia Software y Documental	32
8. Conclusiones y Trabajo Futuro	34
9. Referencias Bibliográficas	35
10. Anexos	37
10.1. Manual de instalación	37
10.2. Manual de usuario	38

Índice de figuras

5.1. Gestión de la información principal de un nodo en Elvira	15
5.2. Gestión de los valores de un nodo en Elvira	15
5.3. Gestión de los padres de un nodo en Elvira	16
5.4. Gestión de la relación entre nodos en Elvira	17
5.5. Mock up previo	20
5.6. Funcionamiento de la aplicación	21
6.1. Visualización del resultado en W3c	29
6.2. Visualización de la web en Mozilla Firefox	30
6.3. Visualización de la web en Google Chrome	30
6.4. Visualización de la web en Microsoft Edge	31
7.1. Logotipo de la licencia BSD	32
7.2. Logotipo de la licencia GNU	33
7.3. Licencia de ZAGUAN	33
10.1. Visualización del ejemplo	39
10.2. Información del nodo Robo representado como B	39
10.3. Información del nodo Terremoto representado como E	40
10.4. Información del nodo Alarma representado como A	41
10.5. Información del nodo John llama representado como J	42
10.6. Información del nodo Mary Llama representado como M	43
10.7. Resultado del ejercicio	44

Índice de tablas

4.1. Softwares alternativos a Elvira	10
4.2. Resultado del análisis de softwares alternativos	12
6.1. Resumen de la evaluación de la norma ISO/IEC 25010	25
6.2. Problemas encontrados en el primer análisis	27
6.3. Problemas restantes	28
10.1. Tabla de resultados del ejercicio	38

Capítulo 1

Agradecimientos

En este apartado quisiera dar las gracias en primer lugar a Piedad y a Paco por ayudarme en este proyecto. En segundo lugar a mis abuelos Dionisio Langarita, Pilar Alhambra, Manuel Benítez y Benilde Sebastián, también a mis padres Victor Langarita y Benilde Benítez, porque gracias sus esfuerzos ha sido posible realizar este proyecto. Y por último mencionar a mi hermano Rubén, por su apoyo constante, incansable y tenaz.

Capítulo 2

Resumen y Palabras Clave

Palabras clave - Redes Bayesianas, Grafos, Elvira, Payara, Javascript.

En este trabajo se expone la adaptación de Elvira a la nube, que pretende abordar los problemas de obsolescencia e incompatibilidad que suelen generar este tipo de aplicaciones con bastantes años de recorrido, además de facilitar el uso del software a todo aquel que tenga acceso a internet.

Para la realización del front-end se ha utilizado Javascript con la biblioteca React. El back-end se utilizará en código obtenido de la aplicación de escritorio de Elvira. Y para la conexión entre ambos códigos se ha implementado una API REST con la ayuda de un servidor de Payara.

This work presents the adaptation of the software Elvira to the cloud, which pretends solve problems like obsolescence and incompatibility that use to make this kind of applications when they are some years old, and also make an easier access of software to everyone who have internet.

For the develop of front-end it been used Javascript with React. At the back-end it had used source code from the desktop app of Elvira. And the connection between both sides it has been implemented an API REST under by Payara Server.

Capítulo 3

Introducción y Objetivos

La directora del proyecto, entre otras propuestas de Trabajo Fin de Grado (TFG), propuso la idea de adaptar Elvira a la nube. Elvira es un software que implementa modelos probabilísticos basados en grafos. Se trata de una aplicación de escritorio con una interfaz gráfica de usuario. La decisión por realizar este proyecto y no otro fue debido a su posible utilidad en un futuro. Para obtener el código fuente se envió un mensaje a los autores. Los mismos autores redireccionaron a otra página web para poder descargar el código.

El programa Elvira [1, 2] está implementado en el lenguaje de programación Java por la Universidad de Granada (UGR), Universidad de Almería (UAL), Universidad del País Vasco campus en San Sebastián (EHU), Universidad Nacional de Educación a Distancia (UNED), Universidad de Castilla La Mancha campus en Albacete y Ciudad Real (UCML), Universidad de Murcia (UM), Universidad Politécnica de Madrid (UPM) y Universidad E.A Nebrija. Financiado por la CICYT y el Ministerio de Ciencia y Tecnología. Al estar escrito en Java, permite ser utilizado por todos los sistemas operativos que usen una máquina virtual de Java (MS-DOS/Windows, Linux, Solaris, etc.)

La utilidad de Elvira, principalmente, se encuentran en el campo de la medicina, aunque en la actualidad también en la agricultura, y la ganadería. Por ejemplo en medicina, se consigue obtener un resultado sobre el diagnóstico de una enfermedad analizando la inferencia de los síntomas.

El objetivo principal de este TFG es adaptar el programa de escritorio Elvira a la nube, ya como evidencian proyectos europeos como "School in the cloud" (Acción 3 del programa de for digital citizenship, Aprendizaje permanente de la Unión Europea(UE), avalan este tipo de adaptaciones [3])

Los objetivos específicos, que permiten conseguir el objetivo principal son:

- Rediseñar y programar el código de la aplicación para adaptarlo a la nube
- Corregir los bugs que se vayan detectando durante el proceso de adaptación
- Conseguir una versión más estable y adaptable, puesto que se trata de un software muy útil, gracias al que se pueden llevar a cabo tareas de edición y evaluación de modelos gráficos probabilísticos, concretamente redes bayesianas y diagramas de influencia, de manera visual.

Además se busca cumplir con algunos Objetivos de Desarrollo Sostenible (ODS) impulsados por la Organización de Naciones Unidas (ONU) [4]:



Objetivo 4: Una educación de calidad con acceso igualitario. Cumpliendo los objetivos 4.3 y 4.4



Objetivo 9: Industria, innovación e infraestructura inclusivas y sostenibles, donde promueban la nuevas tecnologías, faciliten el comercio internacional y permitan el uso eficiente de los recursos. Realizando los objetivos 9.b y 9.c

Objetivo 4.3: Asegurar el acceso igualitario de todos los hombres y las mujeres a una formación técnica, profesional y superior de calidad, incluida la enseñanza universitaria

Objetivo 4.4: Aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento.

Objetivo 9.b: Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas.




Objetivo 9.c: Aumentar significativamente el acceso a la tecnología de la información y las comunicaciones y esforzarse por proporcionar acceso universal y asequible a Internet en los países menos adelantados.

Este documento se divide en las siguientes secciones: un estudio de softwares similares, un análisis sobre el código y documentación, así como un diseño e implementación propios de una solución para adaptar Elvira a la nube. Una validación de lo propuesto siguiendo algunas características de la norma ISO/IEC 25010, la aplicación de licencias tanto en el software como en la documentación, unas conclusiones y trabajo futuro, y las referencias bibliográficas para finalizar.

Capítulo 4

Estado del Arte

En la búsqueda de algún software sobre redes bayesianas con el que poder comparar Elvira, se han encontrado varios.

Logotipo	URL
OpenBugs	https://www.openbugs.net/w/FrontPage
	https://www.bayesfusion.com/genie/
JAGS	https://mcmc-jags.sourceforge.io/
	https://mc-stan.org/
 NIMBLE	https://r-nimble.org/

Cuadro 4.1: Softwares alternativos a Elvira

El proyecto **OpenBUGS** [5] es un proyecto open source que nace del proyecto WinBUGS. Desarrollado por Medical Research Council (MRC) Biostatistics Unit (BSU) de la Universidad de Cambridge. Escrito en C y R funciona en los sistemas operativos de Windows, Unix/Linux y Mac. Permite copiar

y distribuir pero no cambiar el código. Puede ser una instalación difícil si se presentan incompatibilidades.

Otro software desarrollado por la universidad de Pittsburgh, aunque el copyright lo tiene la empresa BayesFusion es **Genie** [6]. Hay dos versiones disponibles, la versión comercial y la versión académica. Solo está escrito para el sistema operativo Windows, pero se puede ejecutar en otros sistemas operativos con Wine. Se permite usar la versión académica con fines de investigación y docencia para instituciones académicas. Genie utiliza las librerías de SMILE Engine, desarrolladas también por la Universidad de Pittsburgh, escritas en C++. Las relaciones en los nodos no se pueden personalizar, son únicas respecto al nodo. Deficiencia que se suple con la creación de más tipos de nodos.

JAGS [7] es un software que está bajo el copyright de Martyn Plummer. La última versión (4.3) salió en 2017, escrito en C y usando librerías de R con una licencia GNU GENERAL PUBLIC LICENSE (GNU/GPL), versión 2, donde está permitido copiar y distribuir el documento pero no está permitido el cambio. Está disponible en sistemas operativos Debian, Ubuntu, Mac y algunas distribuciones Linux. La instalación es compleja por incompatibilidad de versiones.

Stan [8] proporciona interfaces con algoritmos estadísticos en diferentes lenguajes. Está creado por profesores de varias universidades, destacando Columbia University, asociados con NumFOCUS. Utiliza los lenguajes Python, shell, MATLAB, Julia y Stata. Se puede usar en Linux, Mac y Windows, código bajo la licencia GPLv3 que permite usar, compartir y modificar el software. La complejidad de la instalación dependerá del lenguaje que se utilice.

Por último citar a **Nimble** [9], hecho por UC Berkeley con la colaboración de University of California. Usando C++ con R. Antes de instalar es necesario tener R. Se puede instalar en los sistemas operativos Linux, Mac y Windows. El código está bajo la licencia GPL con permiso para usar, compartir y modificar el software.

Tal como se puede deducir del resumen presentado en el cuadro 4.2, los software de simulación de redes bayesianas requieren altos conocimientos de matemáticas y programación. Estas necesidades hacen que los simuladores sean escasos. Todos ellos están escritos sobre lenguajes como C o R, por su rapidez en el cálculo, y buscan la adaptación a varios sistemas operativos, ya sea por la implementación nativa o por el uso de programas de reimplementación en otros sistemas operativos como Wine. Entre todos los software

Software	Lenguaje	SO	Usar	Compartir	Modificar
OpenBUGS	C y R	Todos	Sí	Sí	No
Genie	C++	Todos	No	No	No
JAGS	C y R	Linux, Mac	Sí	Sí	No
Stan	Depende	Todos	Sí	Sí	Sí
Nimble	C y R	Todos	Sí	Sí	Sí

Cuadro 4.2: Resultado del análisis de softwares alternativos

expuestos, Genie es el más fácil de usar e instalar pero también es el software con más restricciones en sus licencias, además, es el único software con interfaz gráfica de usuario, el resto realiza la simulación de las redes mediante lenguaje de programación o comandos.

Capítulo 5

Análisis, Diseño e Implementación

5.1. Análisis

Un grafo [17] es un conjunto de nodos enlazados por aristas que no tienen la por qué estar dirigidos. En los grafos dirigidos [18] pueden tener aristas que parten de un mismo nodo dirigidas a un segundo nodo en la misma dirección si las aristas representan cosas diferentes. También se pueden unir dos nodos, mediante múltiples aristas, lo cual se considera un multigrafo [19].

Los grafos acíclicos tienen la característica de que al realizar el recorrido desde cualquier nodo no se creará un bucle. Los grafos cíclicos pueden realizar un bucle, al recorrer los nodos, porque su estado actual contiene información de estados pasados.

El grafo mixto [20] es un grafo con aristas tanto dirigidas como no dirigidas.

Elvira implementa varios modelos probabilísticos basados en grafos: Las redes bayesianas [22] son grafos dirigidos acíclicos, las redes de Márkov [21] en cambio son sistemas cíclicos, las redes de influencia es una generalización de las redes bayesianas donde participan la inferencia bayesiana y análisis de decisión para resolver el problema y Chain Graph es grafo mixto acíclico utilizado en modelos gráficos por inferencias bayesianas.

El software Elvira no está implementado completamente. Sólo están disponibles las redes Bayesianas, las redes de influencia, y las redes de análisis de decisiones. El tipo de variables son discretas, continuas y ambas, aunque

sólo se pueden seleccionar variables discretas.

En las redes bayesianas, los elementos base son los nodos y los enlaces entre sí.

Respecto a los nodos en las redes bayesianas, se pueden añadir nodos aleatorios de variables discretas y nodos continuos. Los nodos aleatorios con variables discretas, en la sección de relación, solo se puede seleccionar 'General' y 'Árbol de probabilidad'. La opción árbol de probabilidad da problemas, si se quisiera volver a cambiar a otra opción, lo mejor es borrar el nodo para crear otro. Si se reabren las propiedades del nodo, las opciones de las relaciones aumentan. Los nodos continuos es un nodo con la relación de árbol de probabilidad.

Elvira cuenta con dos modos de actuación sobre el proyecto: edición e inferencia. En el modo inferencia se muestra la propagación de la red. Donde se visualizan las probabilidades a priori, se pueden crear y visualizar los casos, simular que el estado de un nodo se da por cierto y ver su propagación e incidencia entre el resto de la red.

En las redes de influencia además de nodos y enlaces se encuentran los nodos observadores, de decisión y de utilidad. Se implementan las acciones de añadir restricciones en el diagrama, expandir árbol de decisión y realizar un análisis de sensibilidad. Cuando se cambia del modo edición al modo inferencia, el cuadro de visualización se queda totalmente en blanco sin la posibilidad de ver el resultado.

Las redes de análisis de decisiones no funcionan correctamente al cambiar de modo edición a inferencia, es cuando se cierra el programa.

Para obtener la información que maneja y estructura, es necesario mirar los datos de los nodos y el archivo guardado.

La información de los nodos se separa en cuatro secciones: información principal, valores, padres y relación. El archivo con extensión .elv guarda las diferentes redes, en otras cuatro secciones, que se van a comentar a continuación:

La primera sección de los archivos .elv son las propiedades de la red (tipo de grafo, precisión visual, versión, estados de los nodos por defecto).

El segundo apartado de los archivos .elv lo califican como variables, guardan la información principal de los nodos y el apartado de valores, sumando la posición x e y para dibujar los nodos. La ejecución del programa Elvira se organiza en dos apartados, Nodo y Valores. Ver figuras 5.1 y 5.2.

La información principal de todos los nodos son: nombre, título, relevancia (que es una enumeración 0 a 10), comentario, y función (Tratamiento,

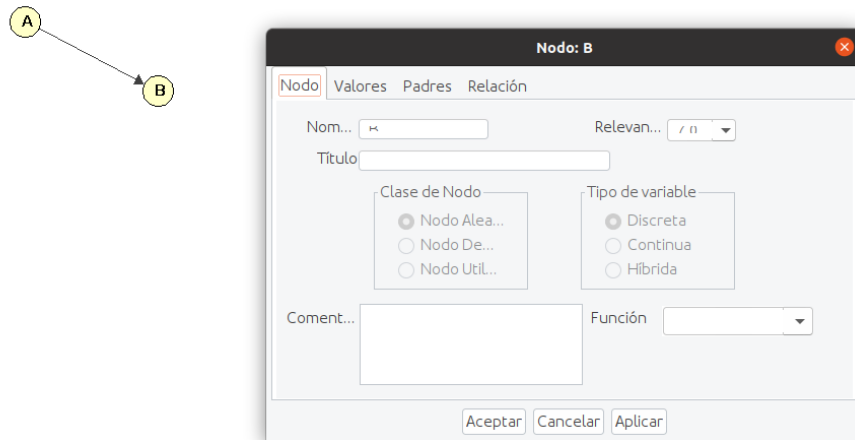


Figura 5.1: Gestión de la información principal de un nodo en Elvira

Factor de riesgo, Síntomas, Signo, Prueba, Enfermedad, Auxiliar, Otro, Definir nuevo). Además se puede observar la clase de nodo y el tipo de variables pero no se puede cambiar su valor. En el apartado valores, el nodo de utilidad no tiene este apartado, pero se gestionan los posibles estados del nodo.

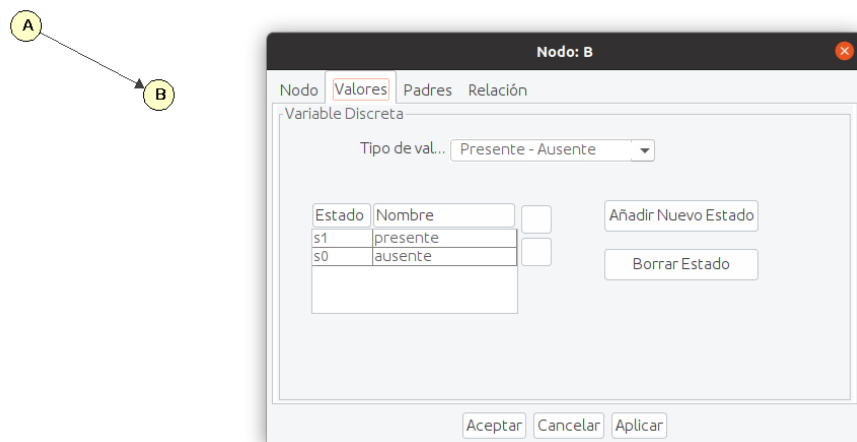


Figura 5.2: Gestión de los valores de un nodo en Elvira

El tercer apartado son los enlaces del grafo que utiliza la sección padres, en el nodo, donde se visualizan todos los enlaces que apuntan al nodo(ver figura 5.3).

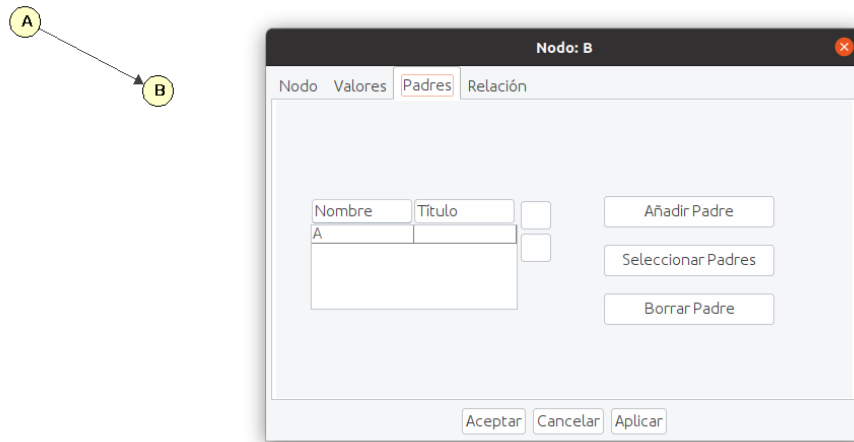


Figura 5.3: Gestión de los padres de un nodo en Elvira

La última sección, llamada relaciones en la red, guarda la relación de los nodos en el grafo y cómo se comportan entre sí las variables con los algoritmos: General, Or, Max Casual, Max generalizada, And, Min y Arbol Prob (ver figura 5.4).

Respecto a la organización que existe en el código que se ha proporcionado, se han de mencionar las clases mas simples: FiniteStates, Link, Relation y la agrupación de estas clases en listas NodeList, LinkList y RelationList. Las clases que se utilizan para representar una red bayesiana: Bnet, Network, Graph y, como clase auxiliar: Potential.

- FiniteStates es la clase que representa los nodos, y tiene herencia de Node, donde lo principal es que son todos nodos de probabilidad y estados finitos. Y los argumentos interesantes son dos LinkList, uno que refleja los nodos padres y otro los hijos y, la relevancia.
- Link contiene dos objetos Node: tail como padre y head como hijo.
- Potential es una clase abstracta y en sus herencias es donde se almacenan los datos.
- Relation representa cómo se relacionan los nodos entre sí y contiene un objeto de tipo Potencial, un entero que indica el tipo de relación, y un objeto NodeList donde están todos los nodos de la relación, tanto los nodos padres como el nodo hijo.

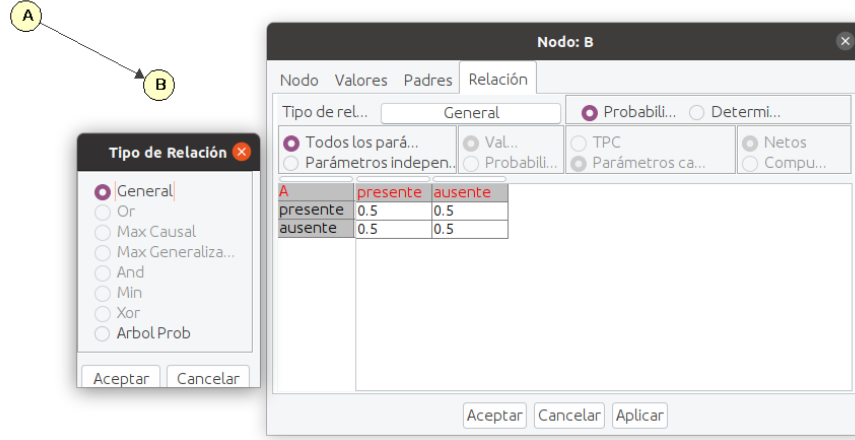


Figura 5.4: Gestión de la relación entre nodos en Elvira

- Graph es la clase más simple de la representación de la red bayesiana donde se almacenan un NodeList y LinkList. Después NetWork extiende el Graph y se incorpora como objeto RelationList. Y por último, Bnet, es la clase final, que se usará para ejecutar el cálculo.

Las filas representan cuantos estado tiene el nodo al que se aplicará el cálculo y las columnas las diferentes inferencias que producen, los nodos padres. Tanto las columnas de forma individual antes del cómputo como la suma final de las posibilidades debe ser 1.

El cálculo a realizar cuando un nodo es huérfano y no tiene padres presenta un vector de 1 columna y n filas siendo n el número de estados posibles. El mismo vector es el resultado de posibilidades de los estados.

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \quad (5.1)$$

La operación a realizar con el nodo al cual tiene la inferencia de un padre es la siguiente: se tiene una matriz de m columnas siendo los estados del nodo padre y o filas siendo los estados del nodo actual. Esta matriz se debe multiplicar con el resultado del nodo padre. La suma de las filas es un vector que refleja el resultado.

$$\begin{bmatrix} b_{00} & b_{01} & \cdots & b_{0m} \\ b_{10} & b_{11} & \cdots & b_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{o0} & b_{o1} & \cdots & b_{om} \end{bmatrix} * \begin{bmatrix} a_0 & a_1 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} b_{00} * a_0 & b_{01} * a_1 & \cdots & b_{0m} * a_n \\ b_{10} * a_0 & b_{11} * a_1 & \cdots & b_{1m} * a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_{o0} * a_0 & b_{o1} * a_1 & \cdots & b_{om} * a_n \end{bmatrix} \rightarrow \quad (5.2)$$

$$\begin{bmatrix} b_{00} * a_0 + b_{01} * a_1 + \cdots + b_{0m} * a_n \\ b_{10} * a_0 + b_{11} * a_1 + \cdots + b_{1m} * a_n \\ \vdots \\ b_{o0} * a_0 + b_{o1} * a_1 + \cdots + b_{om} * a_n \end{bmatrix}$$

Si la inferencia al nodo se produjera desde varios nodos la operación sería la misma, no así la creación de la matriz ni del vector. Para obtener el número de columnas de la matriz y la longitud del vector es necesario calcular el productorio de los estados de los nodos que inferen.

$$\prod_{i=0}^p x_i = x_0 \cdot x_1 \cdots x_p \quad (5.3)$$

Si se tienen tres nodos con el número de estados q, r, s . Se obtiene p de la siguiente forma $p = q * r * s$. El contenido del vector es la multiplicación de las posibilidades entre los estados de los nodos padres representado así:

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_q \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_r \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_s \end{bmatrix} \rightarrow \begin{bmatrix} a_0 * b_0 * c_0 \\ a_0 * b_0 * c_1 \\ \vdots \\ a_0 * b_0 * c_s \\ a_0 * b_1 * c_0 \\ \vdots \\ a_0 * b_r * c_s \\ a_1 * b_0 * c_0 \\ \vdots \\ a_q * b_r * c_s \end{bmatrix} \quad (5.4)$$

Tal y como se puede observar en el análisis realizado y a pesar del potencial de la herramienta, está sin terminar. El autor de este TFG, sólo ha

cursado 30 créditos relacionados con la materia, por lo que no está capacitado para diseñar ni desarrollar lo que falta. Es más, lo deseado sería que hubiera por detrás un equipo multidisciplinar para hacerlo.

5.2. Diseño

Solo se posee como recurso disponible previo el código original escrito en Java. Por lo tanto, se debe crear el cliente desde cero con la libertad de uso de cualquier software. Utilizar el código Java para crear el servidor limita las herramientas a usar. La web obliga a la división y el reparto de las funciones entre los clientes y en el servidor. La creación de una red y ejecutar un cálculo de propagación sobre ella será la acción principal. Realizar las dos acciones en el cliente plantea el problema de rehacer todo el código por completo. Si las dos acciones se producen en el servidor se crearía mucho tráfico hacia el servidor, al crear y editar una red. La división de crear la red en el cliente y consultar al servidor para hacer los cálculos permite repartir la carga, reusar el código Java y en un futuro, el escalado de la web como indica la norma ISO/IEC 25010 [13], dependiendo del número de clientes activos.

Como evidencia de diseño se aporta un prototipo de baja fidelidad de la pantalla principal de la nueva versión de Elvira.

Dicho prototipo ha sido realizado con la herramienta Blasamiq, usada en la asignatura Interacción Persona Ordenador del plan de estudios de la titulación de Grado de Ingeniería Informática (GII) en la Escuela Universitaria Politécnica de Teruel (EUP-T).

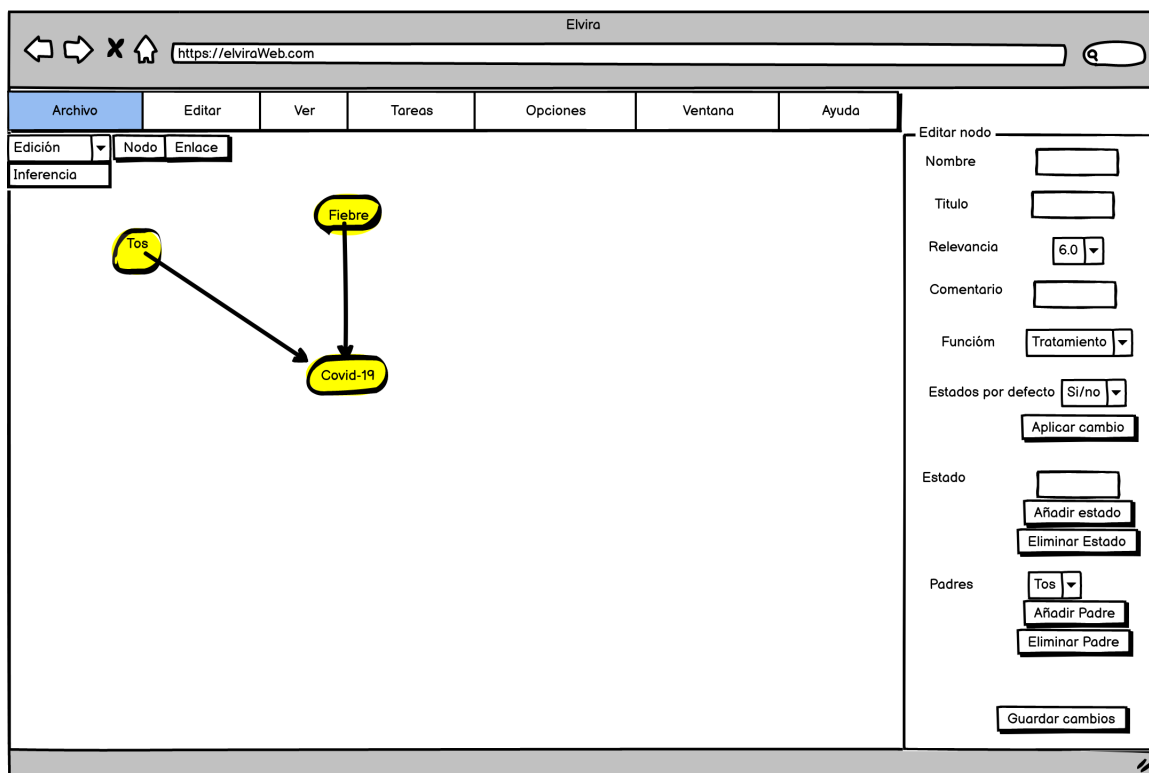


Figura 5.5: Mock up previo

5.3. Implementación

Para la implementación del programa se parte de la aplicación de escritorio de Elvira, que se compila y se inicia mediante Java, versión java-14-openjdk-amd64. En el back-end se ha usado NetBeans 11.3 y Payara Server 5.194, utilizando las versiones de java: java-11-openjdk-amd64 y java-8-openjdk-amd64. En el front-end se ha usado la biblioteca React versión 17.

La adaptación de Elvira a un entorno web, y por tanto a la nube, facilita su uso debido a no tener que instalar ningún tipo de software en el ordenador del usuario. Si se habla del Elvira prístino, se debe lanzar con Java, y podría darse la posibilidad de que la versión del compilador no llegase a ser compatible con el código ya que éste se puede quedar anticuado. Además, se añade que no importa el sistema operativo que se use. Lo importante y necesario en la web es el navegador. En el front-end acoge la parte de creación de la red. En el

back-end contiene el código java de Elvira como soporte para la organización y creación de la red, las funciones que realizan el cómputo de la red y las llamadas de REST API para invocar todo el proceso. La actualización de las partes es totalmente independiente.

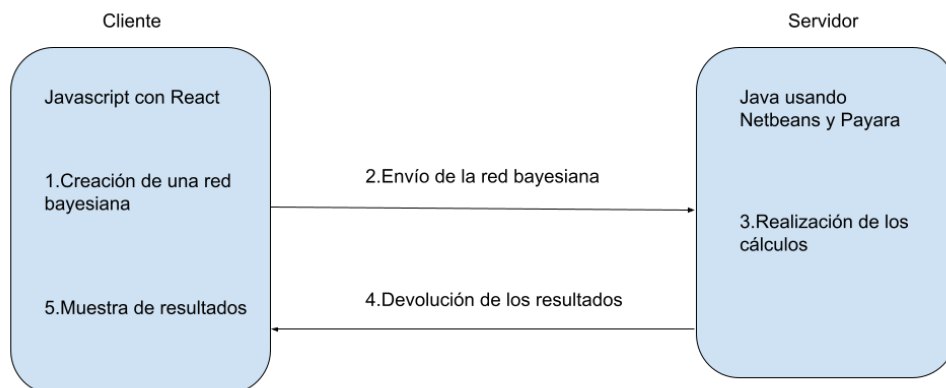


Figura 5.6: Funcionamiento de la aplicación

Las dificultades encontradas durante la realización han sido varias y de diversa índole. La instalación de Netbeans y la conexión con Payara. El posicionamiento de los distintos elementos que conforman la web y la interacción entre ellos, el estudio del código proporcionado de Elvira, entender la propagación y los cálculos en los distintos nodos, junto con la realización de éstos, el canvas donde se dibujan los nodos y los enlaces y por dónde se pueden mover, además que presentaba problemas con los inputs en los cuales no se podía escribir. La comunicación entre cliente y servidor no se ha podido realizar mediante el envío de objetos, por tanto se ha sustituido con el envío de una cadena de texto con un formato JSON(JavaScript Object Notation) y en la parte que recibe el mensaje lo trata.

En la implementación de los cálculos en el servidor hay mayor dificultad en crear el vector, que en la operación. Para la creación el vector es necesario el resultado de los nodos padres, representando cada uno en forma de vector double y todos ellos agrupados en un vector de objetos, al se llamará índice y el número total de columnas, que es el productorio del número de estado de los padres. Se creará un vector con una longitud igual al número de columnas inicializado a 1, al que se denominará vector final. Para realizar todas las

combinaciones posibles de manera ordenada se realizará un bucle a los índices desde el último al primero. En la primera iteración se obtiene en el vector final los resultados del último nodo padre de manera repetida. Para las siguientes iteraciones, se ha de repetir la multiplicación por el mismo número en las siguientes posiciones del vector, tantas como estados tenía el nodo anterior multiplicado por las veces que se había repetido la iteración anterior.

Si se tienen 3 nodo, 2 de ellos con 2 estados y un tercero con 3, el vector final será de una logitud de 12. Se puede observar en la siguiente

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_0 \\ c_1 \\ c_2 \\ c_0 \\ c_1 \\ c_2 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} \rightarrow \begin{bmatrix} c_0 * b_0 \\ c_1 * b_0 \\ c_2 * b_0 \\ c_0 * b_1 \\ c_1 * b_1 \\ c_2 * b_1 \\ c_0 * b_0 \\ c_1 * b_0 \\ c_2 * b_0 \\ c_0 * b_1 \\ c_1 * b_1 \\ c_2 * b_1 \end{bmatrix} \rightarrow \begin{bmatrix} c_0 * b_0 * a_0 \\ c_1 * b_0 * a_0 \\ c_2 * b_0 * a_0 \\ c_0 * b_1 * a_0 \\ c_1 * b_1 * a_0 \\ c_2 * b_1 * a_0 \\ c_0 * b_0 * a_1 \\ c_1 * b_0 * a_1 \\ c_2 * b_0 * a_1 \\ c_0 * b_1 * a_1 \\ c_1 * b_1 * a_1 \\ c_2 * b_1 * a_1 \end{bmatrix} \quad (5.5)$$

Se puede consultar el código del back-end [10] y el código [11] y la visualización [12] front-end en las referencias [10, 11, 12].

Capítulo 6

Evaluación

La norma ISO/IEC 25010 es una estandarización de un modelo de calidad del producto software, que satisface una serie de requisitos de los usuarios aportando valor añadido al software.

Adecuación funcional [13]: Cubre todas las tareas y los objetivos del usuario en condiciones específicas y provee de resultados correctos con el nivel de precisión requerido. Además de proporcionar un conjunto apropiado de funciones con completitud, corrección y pertinencia.

Eficiencia [13]: Los tiempos de respuesta, procesamiento del sistema y las cantidades y tipos de recursos utilizados cuando el software lleva a cabo sus funciones bajo condiciones determinadas.

Compatibilidad [13]: Capacidad del producto para coexistir con otro software independiente, sistemas o componentes en un entorno común, para intercambiar información y compartir recursos comunes sin detrimento.

Usabilidad [13]: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades, características y discapacidades, aprender su aplicación, operarlo y controlarlo con facilidad mientras satisface la interacción del usuario y los protege de cometer errores.

Fiabilidad [13]: Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales, estar operativo y accesible para su uso cuando se requiere también para operar según lo previsto en presencia de fallos hardware o software y en caso de error recuperar los datos directamente afectados y reestablecer el estado deseado del sistema.

Seguridad [13]: Capacidad de protección contra el acceso o modificación de datos e información no autorizados, ya sea accidental o deliberadamente. Demostrar que las acciones o eventos que han tenido lugar, de manera que

no puedan ser repudiados posteriormente. Rastrear y demostrar las acciones e identidad de un sujeto o recurso.

Mantenibilidad [13]: Capacidad de un sistema que permite que un cambio en un componente tenga un impacto mínimo en los demás. La capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos. La facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño. La facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Portabilidad[13]: Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso. La facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno. La capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Expuestos ya las propiedades que debe de tener un software para cumplir la norma ISO/IEC 25010 se expone si la adaptación a la nube cumple o no las características de calidad.

Se observa que el software cumple la adecuación funcional porque cubre la funcionalidad deseada.

En la eficacia está influido por el hardware usado.

La compatibilidad con otros módulos se puede llevar a cabo por la implementación independiente del back-end y front-end.

Respecto usabilidad es difícil de entender en un primer momento, fácil de aprender y de recordar. En el caso de este TFG no se puede asegurar que se cumpla, puesto que no se han realizado aún pruebas reales.

La fiabilidad del sistema es alta debido a la capacidad de crear y borrar la red por varias vías y reestablecer el estado deseado.

La seguridad del sistema no es un problema, porque no se guardan datos importantes, por lo tanto no hay accesos indeseados a los datos, o posibles modificaciones. Así que las acciones o eventos que se produzcan no tienen una importancia mayor, ya que no pueden cambiar o acceder a datos importantes.

La mantenibilidad es sencilla porque tanto el front-end como el back-end

están implementados de manera independiente y se pueden modificar de la misma forma. El cambio de una de las partes por otra no cambiaría nada, siempre que se mantenga la comunicación.

La portabilidad está implementada de forma que cualquier persona con acceso a internet tenga acceso al contenido. El cliente no necesita instalar ningún software.

Adecuación funcional	Eficiencia	Compatibilidad	Usabilidad	Fiabilidad
Sí	Dependen	Sí	No evaluado	Sí
		Seguridad	Mantenibilidad	Portabilidad
		Sí	Sí	Sí

Cuadro 6.1: Resumen de la evaluación de la norma ISO/IEC 25010

Llegados a este punto y antes de tratar una subcaracterística más en profundidad, la accesibilidad, se va a proceder a comentar la norma vigente actual, la norma UNE-EN 301549:2020, donde se relantan varios requisitos a cumplir para que una aplicación se accesible. Dichos requisitos son:

- "especifica, de forma adecuada para su uso en la contratación pública dentro de Europa, los requisitos funcionales de accesibilidad aplicables a los productos y servicios que incorporan TIC, junto con una descripción de los procedimientos de prueba y la metodología de evaluación para cada requisito de accesibilidad. Este documento está diseñado para su utilización con tecnologías basadas en la web, tecnologías no web y tecnologías híbridas que utilizan ambas. Abarca tanto el software como el hardware, así como los servicios. Está dirigido principalmente a su empleo por proveedores y contratantes, aunque se espera que también sea útil para muchos otros usuarios." [14]
- "la relación entre este documento y los requisitos esenciales de la Directiva 2016/2102 sobre la accesibilidad de los sitios web y aplicaciones para dispositivos móviles de los organismos del sector público." [14]
- "Este documento contiene los requisitos funcionales necesarios y constituye una fuente de consulta, que hará posible que los resultados de las pruebas sean similares y la interpretación de esos resultados sea clara, aun cuando los procedimientos sean seguidos por distintos actores. Las descripciones de las pruebas y la metodología de evaluación contenidas

en este documento se elaboran a un nivel de detalle que se ajusta a la Norma ISO/IEC17007:2009 [i.14] para que la prueba de conformidad pueda proporcionar resultados conclusivos.” [14]

La norma referencia a los siguientes documentos necesarios para su correcta aplicación:

- ETSI ETS 300 381 (edición 1) (diciembre de 1994), Telefonía para personas de audición defectuosa. Acoplamiento inductivo de auriculares telefónicos a audífonos.
- ETSI ES 200 381-1 (V1.2.1) (octubre de 2012), Telephony for hearing impaired people; Inductive coupling of telephone earphones to hearing aids; Part 1: Fixed-line speech terminals.
- ETSI ES 200 381-2 (V1.1.1) (octubre de 2012), Telephony for hearing impaired people; Inductive coupling of telephone earphones to hearing aids; Part 2: Cellular speech terminals.
- WC3 Recommendation (diciembre 2008)/ISO/IEC40500:2012 Web Content Accessibility Guidelines (WCAG) 2.0. NOTADisponible en WCAG 2.0.
- WC3 Recommendation (junio2015): Web Content Accessibility Guidelines (WCAG) 2.1.

6.1. Test de accesibilidad

Siempre se aspira a que una aplicación sea accesible y usable para la mayor gente posible. Para evaluar cómo de accesible es la implementación se escaneará mediante el test de accesibilidad web TAW [15]. La elección de TAW es porque éste analiza tanto el código html como el javascript, también las páginas publicadas mediante github. Los fallos encontrados en el análisis son los siguientes:

Tipología	Comprobación	Incidencias
Perceptible		
	1.1.1-Contenido no textual	
Formularios	Controles de formulario sin etiquetar	1
Imágenes	Imágenes sin atributo alt	58
Navegación	Enlaces consecutivos al mismo recurso	4
	1.3.1-Información y relaciones	
Formularios	Controles de formulario sin etiquetar	1
Estructura y semántica	Inexistencia de elemento h1	1
Operable		
	2.4.4-Propósito de los enlaces	
Navegación	Enlaces sin contenido	1
	2.4.10-Encabezados de sección	
Estructura y semántica	Inexistencia de elemento h1	1
Comprensible		
	3.1.1-Idioma de la página	
Idioma	declarado e idioma	1
	3.3.2-Etiquetas o instrucciones	
Formularios	Etiquetado de los controles de formulario	1
Robusto		
	4.1.1-Procesamiento	
Página web	Página 'bien formada'	1
	4.1.2-Nombre, función, valor	
Formularios	Controles de formulario sin etiquetar	1

Cuadro 6.2: Problemas encontrados en el primer análisis

Los cambios introducidos para mejorar la accesibilidad y la usabilidad, para in nivel AA, son los siguientes. En las imágenes se ha introducido el argumento alt, si la interfaz no llegase a cargar las imágenes éstas se sustituirían por el texto. Se ha modificado el menú desplegable para que si la imagen introducida es una cadena de texto vacía no proyecte ninguna imagen. Se ha cambiado el idioma del documento principal html.

Algunos de los errores vigentes se encuentran en el menú desplegable porque está creado siendo lo mas fiable al software original, pensado para trabajos futuros, pero no contiene funcionalidad y los enlaces están vacíos. Para saber qué errores contiene solo el desplegable se han realizado dos análi-

sis. Uno con el desplegable (ver cuadro 6.2) y otro sin él (ver cuadro 6.3). El resultado final sin estos fallos quedaría reflejado en el siguiente cuadro:

Tipología	Comprobación	Incidencias
Perceptible		
	1.1.1-Contenido no textual	
Formularios	Controles de formulario sin etiquetar	1
	1.3.1-Información y relaciones	
Formularios	Controles de formulario sin etiquetar	1
Estructura y semántica	Inexistencia de elemento h1	1
Operable		
	2.4.10-Encabezados de sección	
Estructura y semántica	Inexistencia de elemento h1	1
Comprensible		
	3.3.2-Etiquetas o instrucciones	
Formularios	Etiquetado de los controles de formulario	1
Robusto		
	4.1.1-Procesamiento	
Página web	Página 'bien formada'	1
	4.1.2-Nombre, función, valor	
Formularios	Controles de formulario sin etiquetar	1

Cuadro 6.3: Problemas restantes

Tal y como se puede observar, las diferencias radican en unos resultados de 4 errores perceptibles localizados en el punto 1.1.1-Contenido no textual con una tipología de navegación por 'Enlaces consecutivos al mismo recurso', y 1 error operable localizado en el punto 2.4.4-Propósito de los enlaces también de navegación por 'Enlaces sin contenido'.

También se ha evaluado el código HTML mediante los estándares de W3c [16], sin encontrarse ningún error o aviso alguno.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://sergiolangaritabenitez.github.io/ElviraWeb/>

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by address

<https://sergiolangaritabenitez.github.io/ElviraWeb/>

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was utf-8.
Total execution time 75 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 21.6.16

Figura 6.1: Visualización del resultado en W3c

6.2. Navegadores

Se ha testado la visualización del front-end con varios navegadores: Microsoft Edge, Google Chrome y Mozilla Firefox. Las principales diferencias que se han encontrado, entre ellos, son las configuraciones por defecto de los mismos. Si no hay una especificación sobre la visualización en los archivos .css el navegador escoge la configuración por defecto.

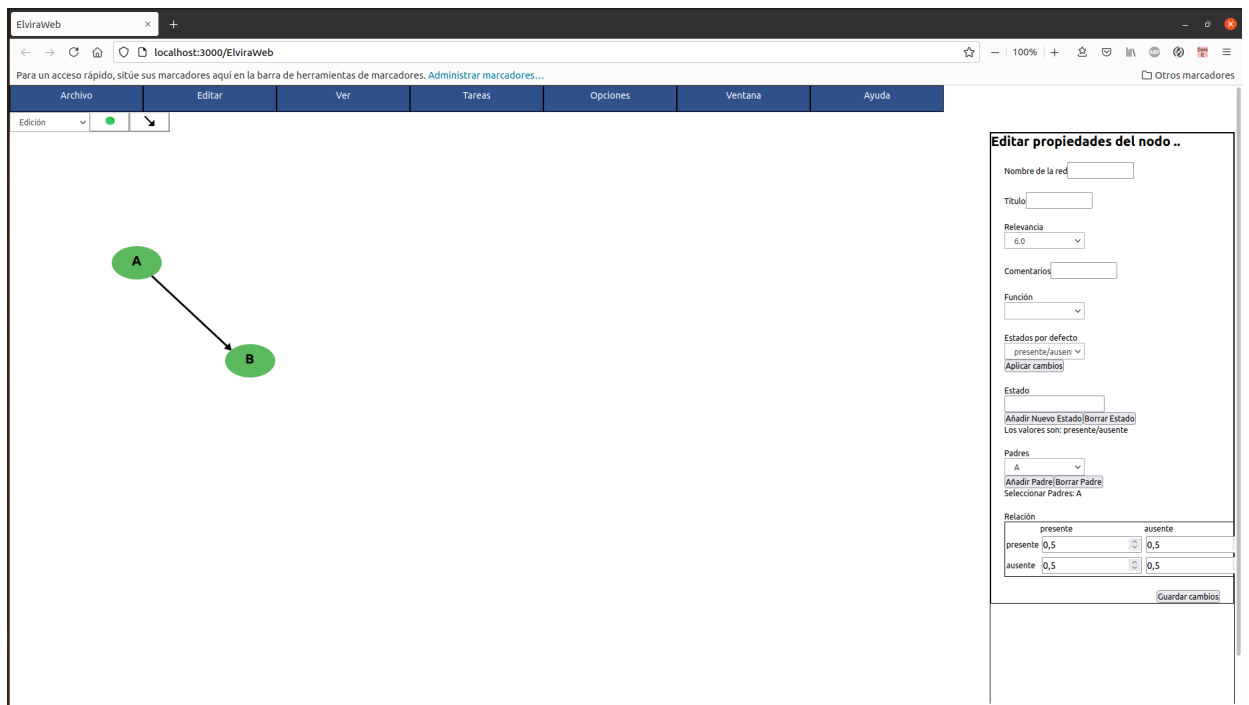


Figura 6.2: Visualización de la web en Mozilla Firefox

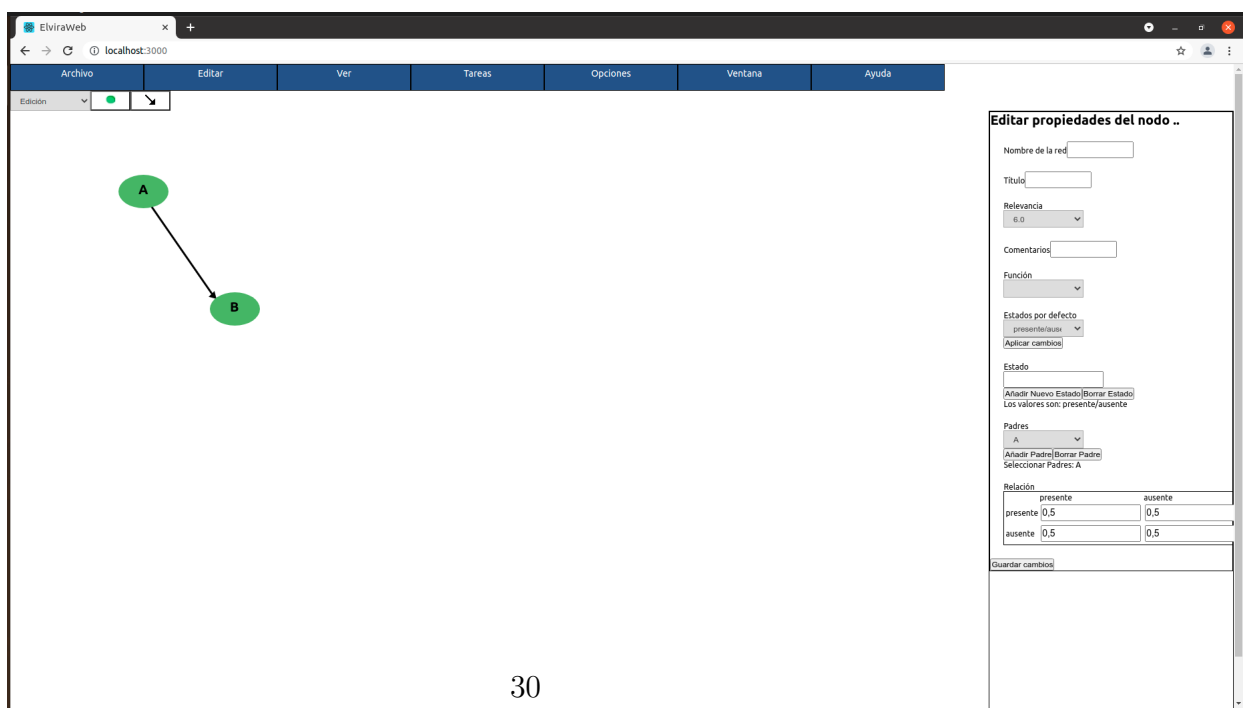


Figura 6.3: Visualización de la web en Google Chrome

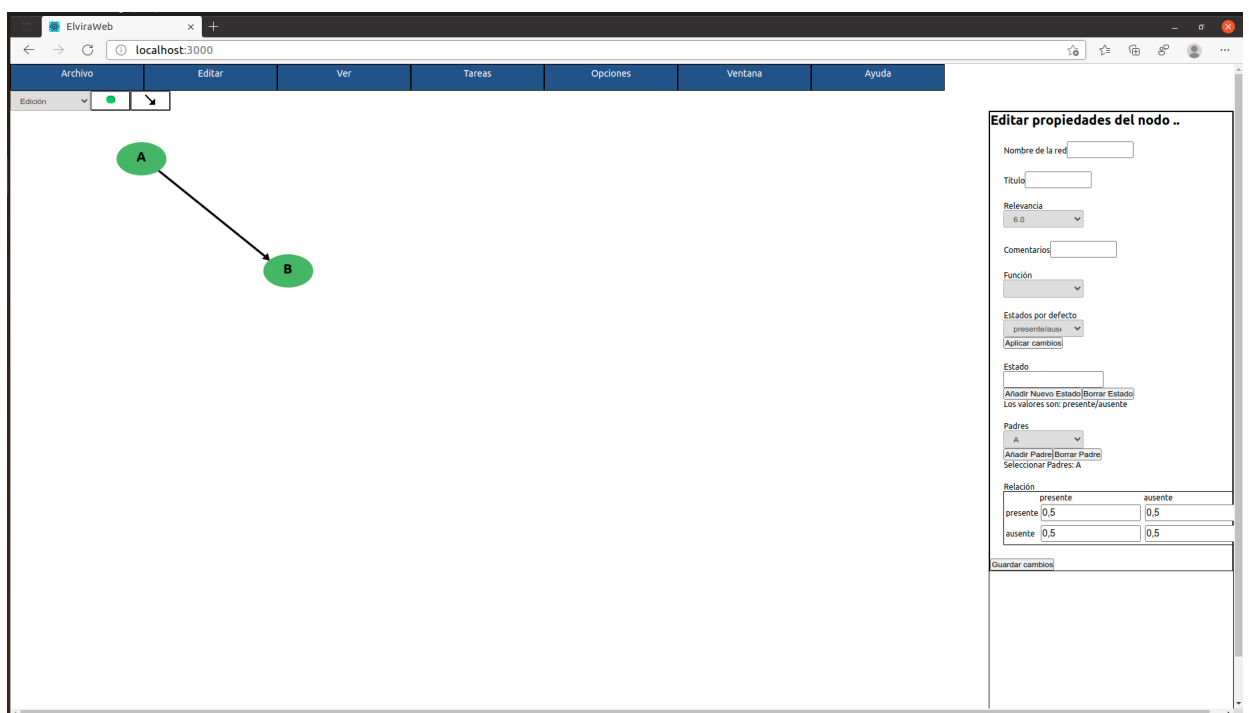


Figura 6.4: Visualización de la web en Microsoft Edge

Capítulo 7

Licencia Software y Documental

Llegados a este apartado, se va a proceder a comentar tanto la licencia de software como la licencia documental.

En cuanto a la licencia de software se va a emplear Berkeley Software Distribution (BSD). Se trata de una licencia de software libre permisiva como puede ser OpenSSL o la MIT License. Existen diferentes tipos de licencias, en el caso de este TFG se ha utilizado la licencia “BSD modificada”, “BSD revisada”, “BSD-3” o “BSD de 3 cláusulas” [23]



Figura 7.1: Logotipo de la licencia BSD

Al igual que sucede en el mundo del software, se tienen que buscar formas de garantizar las libertades asociadas al trabajo elaborado y su inviolabilidad futura. Para garantizar que la libertad esté asociada al documento se buscan métodos, uno de ellos es la licencia GNU Free Documentation License (GFDL).



Figura 7.2: Logotipo de la licencia GNU

El propósito de esta Licencia es hacer que en el caso de este TFG sea 'gratuito' en el sentido de libertad: para asegurar a todos la libertad efectiva de copiarlo y redistribuirlo, con o sin modificarlo, ya sea comercial o no comercialmente. En segundo lugar, esta licencia preserva para el autor y el editor una forma de obtener crédito por su trabajo, sin ser considerado responsable de las modificaciones realizadas por otros. Es una especie de 'copyleft', lo que significa que las obras derivadas del documento deben ser libres en el mismo sentido. Si por algún motivo se emplea este documento y se modifica, debe realizar una serie de acciones indicadas en el sitio web oficial de GNU [24].

Si por algún motivo se emplea este documento y se modifica, se debe realizar una serie de acciones indicadas en el sitio web oficial de GNU [24]. Tampoco hay que olvidar que este documento, por defecto, está al amparo de la licencia 7.3, por su inclusión en el Repositorio Institucional de Documentos de la Universidad de Zaragoza: ZAGUAN

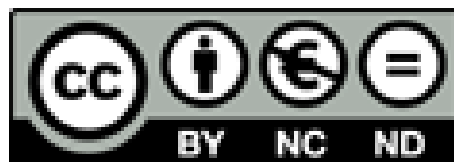


Figura 7.3: Licencia de ZAGUAN

Capítulo 8

Conclusiones y Trabajo Futuro

Se ha podido adaptar el software Elvira a la nube, de manera parcial, y los trabajos que se pueden llevar a cabo para futuros proyectos son funcionalidades tales como: importar o exportar redes de un archivo, operar con casos, trabajar con varias redes al mismo tiempo, implementar más tipos de redes, guardar la red en un archivo, y cargar de un archivo una red.

Los conocimientos adquiridos en el transcurso de este TFG han sido la comprensión en profundidad las redes bayesianas complementando al conocimiento adquirido en la asignatura de Inteligencia Artificial. La implementación de una página web mediante el uso de tecnologías web usando los conocimientos adquiridos tanto en Ingeniería Web como Sistemas y Tecnologías Web, así como una evaluación en usabilidad de la propia página utilizando lo aprendido en la asignatura de Interacción Persona Ordenador.

Capítulo 9

Referencias Bibliográficas

- [1] Elvira, 2021. Ia.uned.es [online],
- [2] Home Page Elvira System, 2021. Leo.ugr.es [online],
- [3] 2021. Schoolonthecloud.net [online],
- [4] GAMEZ, MARIA, 2021, Objetivos y metas de desarrollo sostenible. Desarrollo Sostenible [online]. 2021. [Accessed 16 June 2021]. Available from: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [5] OpenBUGS. Mrc-bsu.cam.ac.uk [online]. 2021. [Accessed 16 June 2021]. Available from: <https://www.mrc-bsu.cam.ac.uk/software/bugs/openbugs/>
- [6] GeNIe Modeler – BayesFusion, 2021. Bayesfusion.com [online],
- [7] JAGS - Just Another Gibbs Sampler, 2021. Mcmc-jags.sourceforge.io [online],
- [8] Stan, 2021. stan-dev.github.io [online],
- [9] NIMBLE – An R package for programming with BUGS models and compiling parts of R., 2021. R-nimble.org [online],
- [10] SergioLangeritaBenitez/ElviraWebBackEnd, 2021. GitHub [online],

- [11] SergioLangaritaBenitez/ElviraWeb, 2021. GitHub [online],
- [12] ElviraWeb, 2021. Sergiolangaritabenitez.github.io [online], <https://sergiolangaritabenitez.github.io/ElviraWeb/>
- [13] DISCERN, CGM, 25000, LOS, PA, EL, CHAPP, SICAMAN and PROMETEUS DELFOS 1.0.0, NUEVO CERTIFICADO ISO/IEC 25000, 2021, ISO 25010. Iso25000.com [online]. 2021. [Accessed 16 June 2021]. Available from: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- [14] UNE-EN 301549:2020 Requisitos de accesibilidad para productos ..., 2021. Une.org [online]
- [15] TAW Servicios de accesibilidad y movilidad web, 2021. tawis.net [online],
- [16] The W3C Markup Validation Service, 2021. Validator.w3.org [online],
- [17] Grafo - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online],
- [18] Grafo dirigido - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online],
- [19] Multigrafo - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online],
- [20] Mixed graph - Wikipedia, 2021. En.wikipedia.org [online],
- [21] Cadena de Márkov - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online],
- [22] Red bayesiana - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online],
- [23] Licencia BSD - Wikipedia, la enciclopedia libre, 2021. Es.wikipedia.org [online]
- [24] The GNU Operating System and the Free Software Movement, 2021. Gnu.org [online]
- [25] Russell, S. y Norvig, P. Inteligencia artificial (Un enfoque moderno), segunda edición (Prentice–Hall Hispanoamericana, 2004)

Capítulo 10

Anexos

10.1. Manual de instalación

Para el despliegue del back-end es necesario:

- Descargar e instalar de Payara Server 5.194.
- Desplegar el archivo: "(Carpeta principal del proyecto) /target/ElviraWebs-1.war", para comprobar que se ha desplegado correctamente realizar una petición "GET (Dirección):(Puerto)/(Nombre de despliegue del paquete del back-end)/rest/elvira/activo".

Para el despliegue del front-end en una dirección pública los pasos a seguir dependerá de donde se publique y cuales son las herramientas que se utilizan pero los siguientes pasos son útiles para usarlo de manera local.

- Instalar NodeJS 6.14.4 o superior.
- Abrir el documento variables.js y buscar la función llamar, dentro de la función, cambiar la variable url por lo siguiente:"(Dirección):(Puerto)/(Nombre de despliegue del paquete del back-end)/rest/elvira/graph".
- Comprobar el funcionamiento lanzando el comando "npm start".

10.2. Manual de usuario

Para usar Elvira el usuario debe desenvolverse en el entorno con facilidad, he aquí un manual para mejorar su aprendizaje. Las acciones importantes que un usuario debe realizar son: gestionar los elementos principales, nodos y enlaces, e introducir la información sobre la red bayesiana.

Para crear un nodo pulsa el botón correspondiente en la pala de herramientas, si se pulsa de nuevo se desactivará, y realizando un click izquierdo en la parte inferior de la paleta se visualizará el nuevo nodo. Para mover el nodo por el lienzo mantén el click izquierdo sobre el nodo y muévelo a la posición deseada. Para eliminar un nodo mantén pulsando ctrl y realiza un click izquierdo cerca del nodo a eliminar.

Para crear enlace pulsa el botón adecuado en la pala de herramientas, realizar un click izquierdo cerca del nodo padre y repetir la acción cerca del nodo hijo. Para eliminar enlace mantén pulsando ctrl y realiza un click izquierdo sobre el lienzo. También se pueden gestionar los padres de un nodo editando las propiedades del nodo. Para ver los resultados cambia del modo edición al modo inferencia.

Para modificar la información de un nodo haz click izquierdo sobre el nodo y aparecerá un tablón a la derecha de la pantalla.

Ahora se expondrá el correcto funcionamiento de Elvira, ejecutando un ejemplo de una red Bayesiana (ver la figura 10.1), extraído de una diapositiva de la asignatura Inteligencia Artificial. El ejemplo expone que en caso de robo o de producirse terremoto suena una alarma y al sonar la alarma, las posibilidades que tienen John y Mary de llamar a urgencias. Resultado en la tabla 10.1):

	Sí	No
Robo	0.001	0.999
Terremoto	0.002	0.998
Alarma	0.002516442	0.997483558
John	0.052138975700000006	0.9478610243
Mary	0.01173634498	0.98826365502

Cuadro 10.1: Tabla de resultados del ejercicio

Ejemplo: Alarma Antirrobo

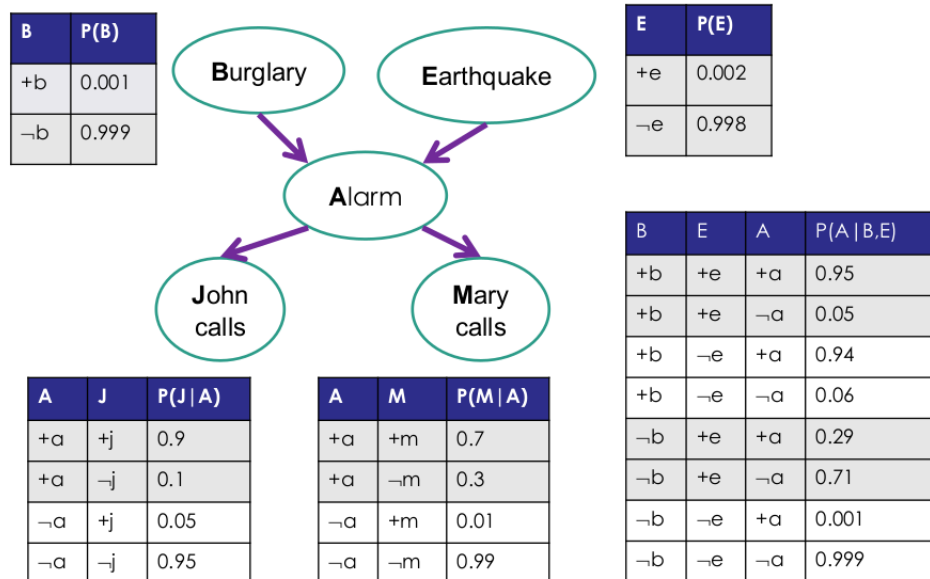


Figura 10.1: Visualización del ejemplo

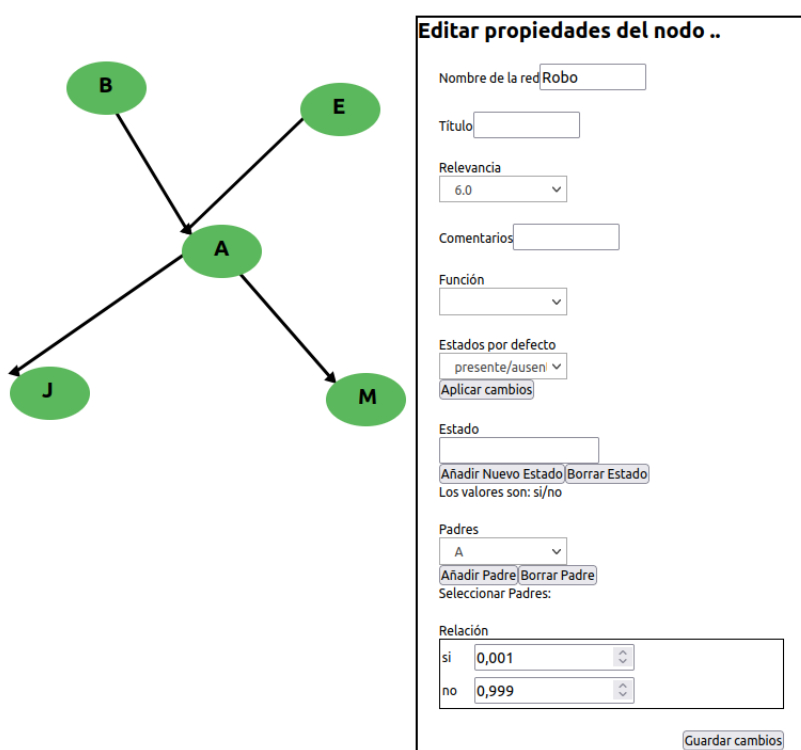


Figura 10.2: Información del nodo Robo representado como B

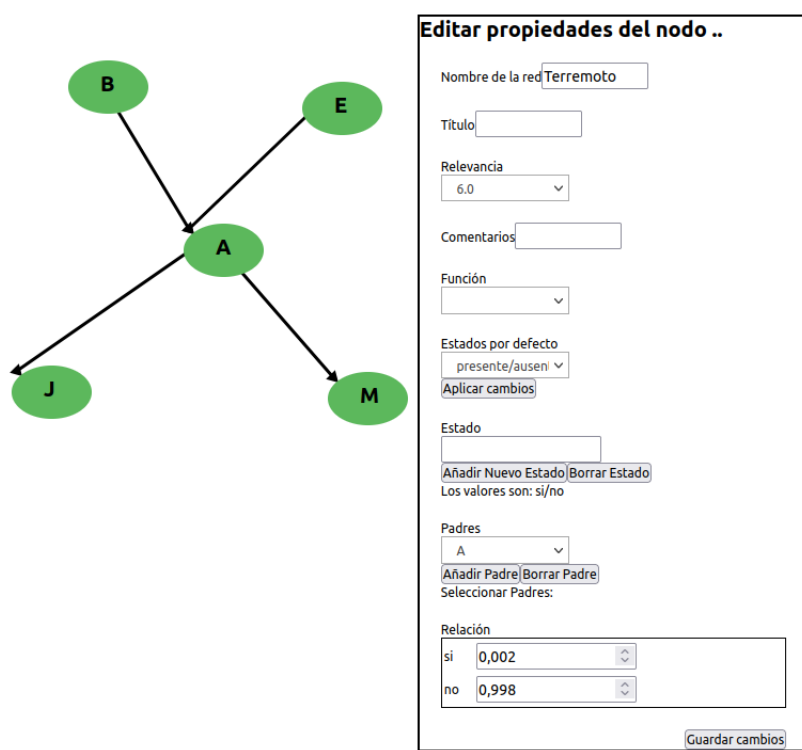
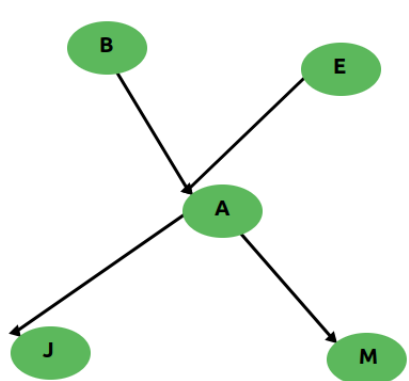


Figura 10.3: Información del nodo Terremoto representado como E



Editar propiedades del nodo ..

Nombre de la red:

Título:

Relevancia:

Comentarios:

Función:

Estados por defecto:

Estado:

Los valores son: si/no

Padres:

Seleccionar Padres: B/E

Relación

	si	si	no	no
	si	no	si	no
si	<input type="text" value="0,95"/>	<input type="text" value="0,94"/>	<input type="text" value="0,29"/>	<input type="text" value="0,001"/>
no	<input type="text" value="0,05"/>	<input type="text" value="0,06"/>	<input type="text" value="0,71"/>	<input type="text" value="0,999"/>

Figura 10.4: Información del nodo Alarma representado como A

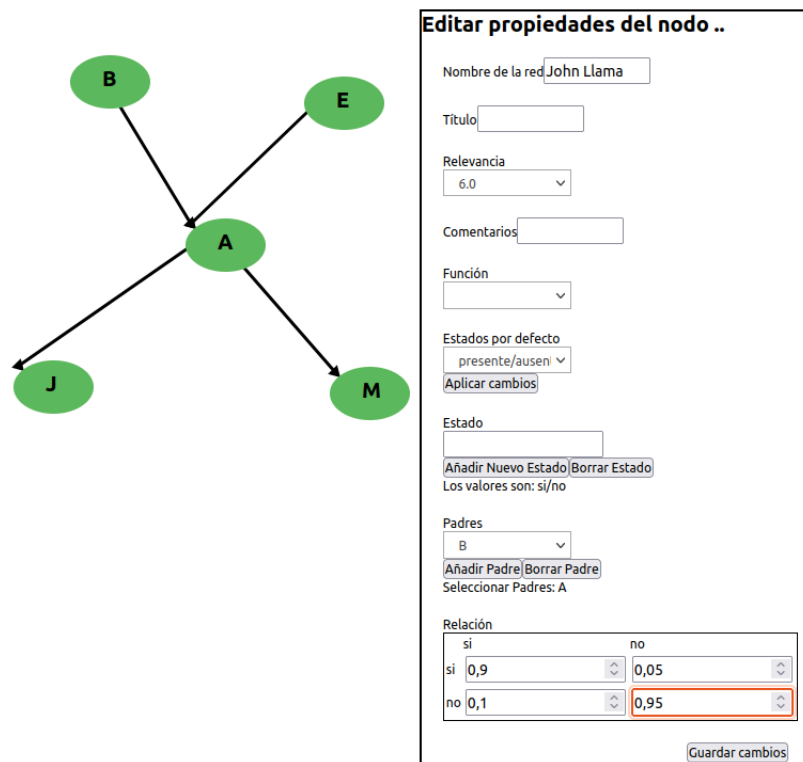


Figura 10.5: Información del nodo John llama representado como J

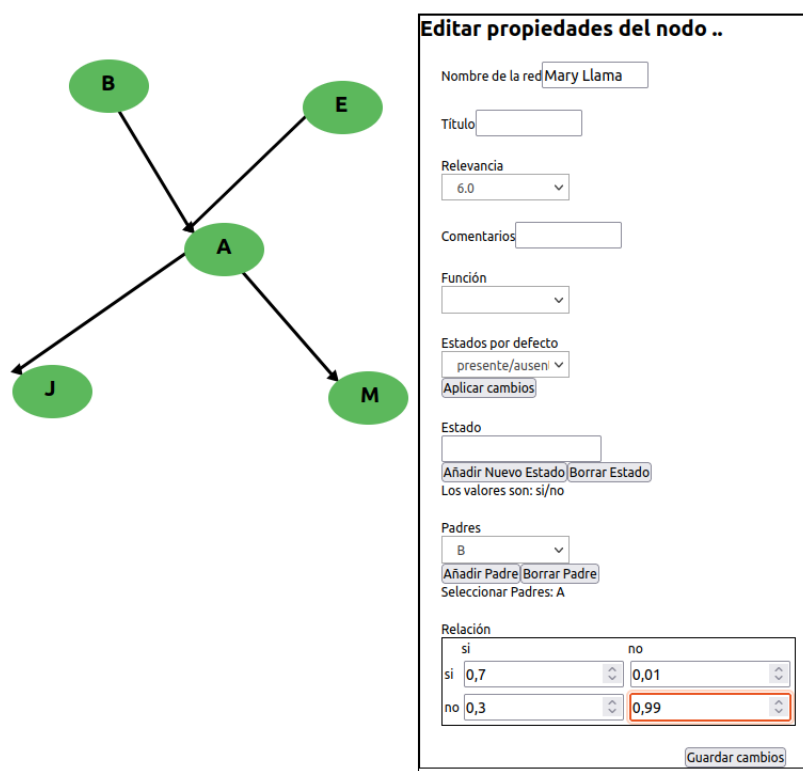


Figura 10.6: Información del nodo Mary Llama representado como M

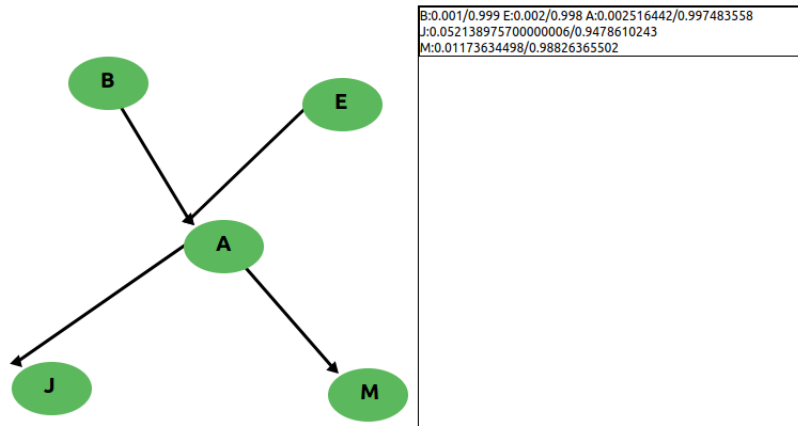


Figura 10.7: Resultado del ejercicio

Copyright (C) SERGIO LANGARITA BENÍTEZ. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being ADAPTACIÓN DEL SOFTWARE ELVIRA A LA NUBE, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".