DETECCIÓN DE CONDICIONES FAVORABLES PARA LA ROYA EN EL CAFÉ BASADA EN SIMILITUD ENTRE GRAFOS.

Trabajo de Grado

Geraldin Valencia Valencia

Gersain de Jesus Lozada Minoli

Director: Mag. Ing. Emmanuel Gerardo Lasso Sambony

Codirector: PhD. Ing. Juan Carlos Corrales Muñoz

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería Electrónica y Telecomunicaciones

Popayán, Septiembre del 2017

Geraldin Valencia Valencia

Gersain de Jesus Lozada Minoli

DETECCIÓN DE CONDICIONES FAVORABLES PARA LA ROYA EN EL CAFÉ BASADA EN SIMILITUD ENTRE GRAFOS.

Trabajo de grado presentado en la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Ingeniero en:

Electrónica y Telecomunicaciones

Director:

Mag. Ing. Emmanuel Gerardo Lasso Sambony

Codirector:

PhD. Ing. Juan Carlos Corrales Muñoz

Popayán
2017

Agradecimientos

Principalmente queremos agradecerle a Dios por estar con nosotros en cada paso que damos, por fortalecer nuestro corazón e iluminar nuestra mente y por haber puesto en nuestro camino a aquellas personas que han sido nuestro soporte y compañía durante todo este proceso. A nuestros padres por su incondicional apoyo, por enseñarnos que con esfuerzo trabajo y constancia todo se consigue y por ser nuestro mejor ejemplo de lucha y constancia.

Agradecemos a nuestras familias por apoyarnos en nuestras decisiones e impulsarnos a ser cada día mejor. A nuestros amigos por crecer con nosotros y enseñarnos cada día cosas nuevas.

Agradecemos al ingeniero Emmanuel Lasso por la dirección y apoyo durante todo al proceso. A los evaluadores de este proyecto que con su visión incrementarán el aporte científico del mismo. Y por último agradecemos a la Universidad del Cauca por formarnos como profesionales.

Resumen estructurado

La proliferación de enfermedades en los diferentes cultivos ha llevado al sector agrícola a buscar diferentes iniciativas que permitan mejorar la calidad y la productividad de los mismos a través de diferentes técnicas basadas en las ciencias de la computación. Investigaciones recientes demuestran que las enfermedades que afectan al sector agrícola están comúnmente relacionadas a las condiciones climáticas y las propiedades agronómicas del cultivo. En este sentido, diferentes trabajos se han enfocado en identificar patrones que conducen a la prevención de la incidencia de estas enfermedades, haciendo uso de una representación basada en grafos de la información del cultivo.

Por lo anterior, este trabajo contribuye con un prototipo que busca encontrar la probabilidad de ocurrencia de la roya del café por medio de una técnica de emparejamiento de patrones de grafos tolerante a errores denominada *distancia de edición,* que permite encontrar un porcentaje de similitud entre las condiciones actuales de un cultivo y los patrones de roya previamente definidos que indican las tasas de infección de la roya del cafeto. Este prototipo implementa una adaptación para 4 algoritmos de emparejamiento de grafos que fueron clasificados teniendo en cuenta la precisión y el tiempo de ejecución.

La metodología definida para la construcción del prototipo comienza desde el reconocimiento de las necesidades del dominio de aplicación. Posteriormente, se realiza una selección de los algoritmos de emparejamiento de grafos tolerante a errores que cumplan con las características que se requieren para obtener la mejor precisión en los resultados. Después, se define e implementan los módulos que serán parte de la adaptación propuesta para los algoritmos seleccionados. Por último, se ejecutan una serie de pruebas con el fin de validar el prototipo.

Finalmente, se concluye que el cálculo de la similitud por medio de la distancia de edición obtenida a partir de la adaptación realizada a los algoritmos de emparejamiento de grafos tolerante a errores permite estimar la probabilidad de ocurrencia de la roya del café. Por otro lado, la adaptación puede ser utilizada en diversos dominios de aplicación donde el reconocimiento de patrones sea requerido, debido a la flexibilidad que aporta a través la parametrización de las funciones de costo y el tipo de normalización utilizada, la cual se puede adaptar a las características de diferentes dominios de aplicación.

Palabras Claves: Grafos, emparejamiento de grafos tolerante a errores, distancia de edición, roya del café, agricultura, reconocimiento de patrones.

Structured Abstract

The proliferation of diseases in different crops has led agricultural sector to look for different initiatives to improve the quality and productivity of those ones through different techniques based on computer science. Recent research shows that diseases affecting the agricultural sector are commonly related to climatic conditions and crop agronomic properties. In this sense, different studies have focused on identifying patterns that lead to prevent the incidence of these diseases, using a graph based representation of crop information.

Therefore, this work contributes with a prototype that seeks to find the occurrence probability of coffee rust by means of an error-tolerant graph pattern matching technique known as edit distance, which finds a percentage of similarity between the current crop conditions and previously defined patterns that indicate the coffee rust infection rates. This prototype implements an adaptation for 4 algorithms of graph matching that were classified considering precision and runtime.

The methodology defined for the construction of the prototype begins from the application domain needs recognition. Subsequently, a selection of error-tolerant graph matching algorithms is performed, searching for the algorithms that meets the characteristics required to obtain the best precision in the results. Later, the modules that will be part of the proposed adaptation for the selected algorithms are defined and implemented. Finally, several tests are made in order to validate the prototype.

Finally, it is concluded that the similarity calculation by means of the edit distance obtained from the adaptation made to the error-tolerant graph matching algorithms allows to estimate the coffee rust occurrence probability. On the other hand, the adaptation can be used in different application domains where the pattern recognition is required, due to the flexibility that it provides through the parameterization of the cost functions and the type of normalization used, which can be adapted to the characteristics of different application domains.

Keywords: Graphs, error-tolerant graph matching, editing distance, coffee rust, agriculture, pattern recognition.

Tabla de Contenido

Lista de figuras

Lista de tablas

Lista de ecuaciones

Lista de algoritmos

Capítulo 1

Introducción

Este capítulo tiene como propósito dar a conocer de manera general las principales consideraciones y motivaciones que impulsan el desarrollo de este proyecto. Seguidamente, se presentan los objetivos que indican las metas que se buscan alcanzar con el fin de aportar en el área de investigación. Finalmente, son expuestas las contribuciones y el esquema de la monografía.

1.1 Planteamiento del problema

Colombia es el cuarto productor cafetero y principal agricultor de café arábica en el mundo. El sector cafetero colombiano emplea directamente a 530 mil personas aproximadamente y unas 2,5 millones dependen de él [1]. Lo anterior hace que sea de suma importancia la optimización de la calidad del fruto con el fin de hacer frente al mercado global, siendo esencial la detección y el control de las enfermedades que se presentan en los cultivos.

Por otro lado, el cambio climático ha provocado el incremento de plagas y enfermedades ampliando el rango altitudinal en el que estas se pueden desarrollar [2], por consiguiente, en los últimos años se ha incrementado gradualmente el área afectada por enfermedades como la roya del cafeto. Esta enfermedad es causada por el hongo *Hemileia vastatrix* que afecta principalmente las hojas del árbol del café (cafeto), produciendo un deterioro en el follaje y por lo tanto una disminución significativa en la calidad y la cantidad del fruto [3]. Además, la ocurrencia de la enfermedad incrementa los costos de producción por el uso de fungicidas, generando un impacto socio-económico debido al gran número de familias colombianas que dependen de esta actividad [4]–[7].

Con el propósito de contrarrestar lo anterior, organizaciones gubernamentales, académicas y centros de investigación especializados, se han enfocado en proponer técnicas de control químico (pesticidas) y cambios en las propiedades

agronómicas del cultivo, de manera que la enfermedad pueda ser tratada antes de que ocasione efectos irreversibles en los árboles del café. Sin embargo, el constante uso de estos elementos genera altos riesgos de salud para los agricultores y además producen un deterioro en la capa de ozono lo cual representa un gran impacto ambiental [8]–[10]. Por este motivo, los investigadores se encuentran en búsqueda de nuevas técnicas que permitan detectar posibles brotes de la enfermedad de forma oportuna y de esta manera reducir el uso de pesticidas [11], [12].

En este sentido, existen investigaciones que a partir de la información proporcionada por plataformas tecnológicas de seguimiento de variables meteorológicas y ambientales [13], están enfocando sus esfuerzos en encontrar la relación entre las condiciones del cultivo (propiedades agronómicas[1] y condiciones climáticas) y la tasa de infección de la roya del cafeto [14], [15] para determinar qué variables están directamente relacionadas con la enfermedad [16]. Algunas de estas investigaciones proponen modelos predictivos [6] que hacen uso de técnicas basadas en grafos con el fin de representar la información de una mejor manera, debido a que estos brindan una mayor expresividad y dinamismo del fenómeno que se está estudiando [17]. Sin embargo, estas aproximaciones no determinan una probabilidad de aparición de la enfermedad con base en la cercanía entre las condiciones del cultivo y los modelos predictivos, siendo esto fundamental para conocer el nivel de riesgo del cultivo de café en una zona específica.

Considerando lo expuesto anteriormente y con pleno conocimiento de la necesidad que surge en el sector agrícola de prevenir las enfermedades en los cultivos, se plantea la siguiente pregunta de investigación.

¿Cómo estimar la probabilidad de ocurrencia de la roya en el café a partir de las condiciones del cultivo?

1.2 Escenario de motivación

Actualmente, el sector agrícola se encuentra en una búsqueda constante de tecnologías que permitan mejorar la calidad y la productividad de sus cultivos; por este motivo, se apoya en diferentes iniciativas, siendo las ciencias de la computación una de las áreas de investigación más influyentes que tienen

[1] Características del suelo y propiedades físicas del cultivo.

como objetivo buscar alternativas para la prevención y control de enfermedades y plagas en los cultivos.

Con base en las ciencias de la computación, han surgido diversas técnicas con las cuales es posible estudiar y prevenir los efectos (enfermedades y plagas) causados por las variaciones climáticas y agronómicas en los cultivos [13]. Por medio de estas técnicas, son obtenidos modelos que relacionan los parámetros agroclimáticos extraídos a partir de datos monitorizados en los cultivos, con el desarrollo de enfermedades y plagas. Sin embargo, a pesar de las numerosas aproximaciones en esta área, se requiere el desarrollo y la aplicación de técnicas que permitan obtener modelos de predicción más precisos. Por lo anterior, se utilizan algoritmos de emparejamiento junto con una estructura de información basada en grafos [14], debido a que éstos facilitan la comprensión de los datos y proporcionan una representación más versátil del dominio del problema.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema para el cálculo de similitud entre patrones de roya del café y las condiciones climáticas y agronómicas de un cultivo expresados como grafos.

1.3.2 Objetivos específicos

- Seleccionar y adaptar algoritmos para el cálculo de similitud entre grafos conforme a las necesidades del proyecto.
- Caracterizar la medida de similitud entre grafos con el dominio de aplicación del proyecto.
- Implementar un prototipo que determine la similitud entre grafos que representan las condiciones de un cultivo de café y patrones de roya.

1.4 Contribuciones

- Implementación y evaluación de la adaptación realizada a los algoritmos de emparejamiento de grafos tolerante a errores: A*, Beam, Hungarian y Volgenant-Jonker.
- Diseño de una herramienta software que permita encontrar la similitud entre dos grafos, por medio de la adaptación propuesta.
- Verificación del adecuado funcionamiento del prototipo implementado y análisis de los datos obtenidos por el mismo.
- Implementación de una solución que genera un aporte al sector agrícola colombiano en el ámbito de control de plagas y enfermedades.
- Fomento al estudio de las temáticas tratadas en la presente investigación, como: grafos y técnicas de emparejamiento de grafos.
- Apoyo al trabajo doctoral del MSc. Emmanuel Lasso, titulado: “Detección de condiciones favorables para la ocurrencia de enfermedades y plagas en el café basada en similitud de grafos”.

1.5 Esquema de la tesis

La monografía está organizada en 6 capítulos que contienen la información referente a este proyecto. Los cuales son presentados a continuación:

Capítulo 2: Investigación documental

En este capítulo se presenta el estado actual de la información y los trabajos relacionados con las temáticas tratadas en esta investigación.

Capítulo 3: Selección y adaptación de los algoritmos

Se presenta una introducción a las principales características del caso de estudio y una descripción de los factores que representan las condiciones agroclimáticas y su relación con las diferentes tasas de infección de la roya del café. Además, se exponen los criterios de selección de los algoritmos y una descripción general de los mismos. Finalmente, es descrito el proceso para la adaptación de los algoritmos seleccionados.

Capítulo 4: Normalización y Parametrización de resultados de acuerdo con el dominio de aplicación.

Se expone el procedimiento usado para caracterizar la medida de similitud por medio de la parametrización de cada una de las funciones de costo y normalización de los resultados.

Capítulo 5: Prototipo y experimentación

En este capítulo se presenta el desarrollo del prototipo que sigue las fases correspondientes a la metodología del “proceso unificado ágil” y que busca determinar la similitud entre grafos [18]. Además, se exponen los distintos artefactos que componen el prototipo y las principales características del mismo. Por otro lado, se describen las pruebas realizadas al prototipo y se efectúa un análisis de los resultados obtenidos con el fin de determinar la confiabilidad y la calidad de los mismos. Para realizar esta evaluación experimental, fue seguida la metodología propuesta por Wohlin [19].

Capítulo 6: Conclusiones y trabajos futuros

Se presentan las conclusiones obtenidas a partir del análisis de los resultados adquiridos con la realización del proyecto, así como los aportes y los trabajos futuros.

Capítulo 2

Investigación documental

En este capítulo se presenta el estado actual del conocimiento y los trabajos relacionados con la temática de este proyecto, los cuales fueron obtenidos siguiendo los pasos propuestos por el “modelo para la investigación documental” [20], teniendo como finalidad obtener una base y punto de partida para el desarrollo de la presente investigación.

2.1 Estado del arte

Con el propósito de contextualizar las diferentes técnicas utilizadas para comprender el ámbito de esta investigación, se presenta a continuación una aproximación a los conceptos relacionados con la temática vinculada al proyecto.

2.1.1 Grafos

Los grafos son una estructura de datos de naturaleza dinámica [21] para la representación de objetos y conceptos. En su estructura, los nodos típicamente representan objetos o partes de objetos, mientras que las aristas describen las relaciones entre estos [22].

La representación de la información como grafos permite una mayor expresividad a través de la especificación de atributos (etiquetas) de nodos y aristas. Además, la principal ventaja de una descripción de patrones basada en grafos es que permite una mejor representación por medio de las relaciones estructurales [23], manteniendo las propiedades esenciales de los objetos modelados, los cuales pueden ser representados gráficamente haciendo uso de los mismos grafos [14].

2.1.2 Emparejamiento de grafos

Una de las técnicas para el análisis de grafos es el emparejamiento de patrones de grafos; este concepto hace referencia al proceso de evaluar la similitud estructural [24], con el fin de encontrar la correspondencia entre los nodos y las aristas de dos grafos [25].

El emparejamiento de grafos puede dividirse en dos categorías: emparejamiento de grafos exacto, cuyo objetivo es detectar sub-estructuras idénticas de dos grafos y sus correspondientes atributos; y el emparejamiento de grafos tolerante a errores, también llamado emparejamiento de grafos inexacto, donde uno de los métodos más flexibles para este tipo de emparejamiento es la distancia de edición de grafos [17], [23], [26], la cual define la disimilitud de los grafos como la cantidad de distorsión (conjunto de operaciones de edición[2]) buscando el menor número de operaciones necesarias para transformar un grafo en otro [17], [27].

2.1.3 Similitud entre grafos

Consiste en determinar los grados de similitud entre un grafo modelo (Patrón) y cada grafo del conjunto de datos (Instancia). En este sentido, la similitud puede ser definida desde diferentes perspectivas como:

- Similitud de Etiquetas: Está basada en una comparación de las etiquetas de los nodos y/o las aristas, en el cual se hace uso de diferentes métricas dependiendo del dominio de aplicación [28].

- Similitud Estructural: Se basa en la topología de los datos representados como grafos [29], esta medida de similitud consiste en el cálculo de la distancia de edición del grafo, obtenida a partir del conteo de operaciones de edición necesarias (eliminación, inserción o sustitución de aristas y nodos) para transformar un grafo en otro [28].

[2] Operaciones de edición: Eliminación, inserción y sustitución de nodos y aristas.

2.2 Trabajos relacionados

A continuación, son presentados dos enfoques de los trabajos relacionados con esta propuesta de investigación. El primer enfoque expone algunos trabajos que están relacionados con el sector agrícola y que buscan plantear soluciones a los problemas de plagas y enfermedades en cultivos desde las ciencias de la computación. En el segundo enfoque se presentan investigaciones que buscan determinar cuál es el algoritmo de emparejamiento de grafos tolerante a errores más adecuado para su dominio de aplicación.

2.2.1 Enfoque agricultura

A continuación, son presentadas investigaciones relacionadas con el sector agrícola, que buscan plantear una solución que mitigue la aparición de la roya en los cultivos de café.

Recientes investigaciones se han enfocado en la predicción de plagas y enfermedades en los cultivos, haciendo uso de técnicas de aprendizaje supervisado (Redes Bayesianas, Árboles de Decisión, Máquinas de Vector de Soporte, Redes Neuronales Artificiales, entre otras.), las cuales aprenden por medio de ejemplos (datos de entrenamiento) relacionados con una variable objetivo, con el fin de predecir o detectar el valor de dicha variable dado un nuevo dato de entrada. Finalmente, después de un proceso de entrenamiento es creado un clasificador (hipótesis o modelo) [2].

En las investigaciones [30] y [31], se propone desarrollar y seleccionar modelos de alerta para predecir el aumento de la tasa de progreso de la roya del cafeto, teniendo en cuenta parámetros como las condiciones agroclimáticas. Además, en aproximaciones similares [16], [32]–[34], también es considerada la cantidad de fruto que el árbol de café posee. Los trabajos anteriores emplean técnicas de minería de datos como redes neuronales, árboles de decisión, máquinas de vectores de soporte y bosques aleatorios, con la intención de anticipar con éxito el desarrollo de la enfermedad.

Asimismo, en el trabajo desarrollado en [35], se plantea el uso de árboles de decisión para generar alertas de advertencia de aparición de la roya del café, estas alertas se activan cuando la enfermedad alcanza dos umbrales: el primer umbral induce a la aplicación de acciones preventivas mientras que el segundo requiere una acción curativa. Además, en investigaciones análogas son comparados los árboles de decisión con las redes bayesianas concluyendo

que los primeros tienen ventaja en los casos sensibles al contexto [36].

Por otra parte, en las aproximaciones [11] y [37], se propone implementar una adaptación para realizar predicciones del porcentaje de infección de la roya en las hojas del café, por medio de intervalos y no de valores puntuales, utilizando clasificadores no deterministas basados en una adaptación de máquinas de vector de soporte. Asimismo, en la investigación [38] es propuesto un sistema de alerta temprana que busca detectar cada una de las condiciones favorables que el hongo *Hemileia Vastatrix* requiere para infectar el cultivo y una vez identificadas estas condiciones, el sistema alerta a los agricultores sobre las condiciones de infección del cultivo.

Por su lado, la investigación desarrollada en [14], busca generar una representación basada en grafos por medio de la extracción de reglas para la detección de la roya del café. Estas reglas son expresadas como patrones de grafo que son modelados de acuerdo con las variables relacionadas con esta enfermedad y extraídas a partir de técnicas de inducción de árboles de decisión y conocimiento experto. Los patrones son usados con el fin de proporcionar una mayor expresividad e interpretabilidad de los fenómenos climáticos que favorecen la manifestación de la enfermedad. Con el propósito de darle continuidad a la investigación desarrollada anteriormente, en [39] proponen hacer uso de técnicas de minería de grafos como el emparejamiento de grafos exacto con el objetivo de validar las reglas y el conocimiento producido por expertos.

2.2.2 Enfoque algoritmos

En esta sección se presentan trabajos que realizan comparaciones de la precisión de algunos algoritmos de emparejamiento de grafos tolerante a errores más frecuentes en la literatura.

En la propuesta de investigación presentada en [29], se adopta un modelo en donde un proceso de negocio es definido por un grafo atribuido dirigido. Adicionalmente, es usado un repositorio de 100 procesos de negocio para realizar la comparación de 4 algoritmos de emparejamiento de grafos (Algoritmo Greedy, Algoritmo Exhaustivo, Algoritmo Heurístico, Algoritmo A*), con el fin de determinar cuál es el mejor algoritmo, teniendo en cuenta la precisión y el tiempo de ejecución. Para esto, se construye una lista ordenada con los resultados de la similitud que proporciona cada algoritmo y se compara con una lista de relevancia de los modelos de negocio.

En la investigación llevada a cabo en [40], se presenta un resumen de algunas de las herramientas disponibles para el emparejamiento de grafos y algunos formatos en los cuales los grafos pueden ser representados y almacenados. En este es mencionada una herramienta software denominada Graph Matching Toolkit (GMT) para el cálculo de la distancia de edición en grafos desarrollada por Riesen [24]. Esta herramienta permite encontrar una solución óptima al problema de emparejamiento de grafos y proporciona algoritmos para aproximaciones no óptimas, pero computacionalmente menos costosas, haciendo uso de una variante del algoritmo A* y diferentes implementaciones para el cálculo de distancia de edición en grafos bipartitos. Por otra parte, diferentes formatos de almacenamiento de grafos son presentados, entre ellos se encuentran dos formatos basados en XML (denominados Graph eXchange Language (GXL)[3] [41] y GraphML[4]).

En el trabajo desarrollado en [42], se expresa que los grafos proporcionan una herramienta eficiente y conveniente para la representación de objetos en varias aplicaciones de la visión artificial, y se realiza una comparación de tres algoritmos A*, Beam Search, Spectral Approach, Gradient Descent Method usando pares de grafos con estructuras similares.

Por su parte, en [23] proponen un novedoso algoritmo denominado BIPARTITE o BP, este permite calcular de manera subóptima la distancia de edición, pero de una manera sustancialmente más rápida; se basaba en el algoritmo munkres que permite resolver el problema de asignación, el cual consiste en encontrar una asignación de los elementos de dos conjuntos entre sí de tal manera que se minimice la función de costo. En esta investigación como punto de comparación utilizan dos sistemas de referencia para calcular la distancia de edición de grafos. El primer sistema de referencia está dado por el algoritmo de búsqueda de árbol óptimo (HEURISTIC-A*), mientras el segundo sistema de referencia es una modificación del HEURISTIC-A*, conocido como búsqueda Beam, el cual no explora el espacio de búsqueda completo, sino sólo se expanden aquellos nodos que pertenecen a los emparejamientos parciales más prometedores. Además, es verificado empíricamente por medio de datos semi-artificiales y reales de manuscritos, huellas digitales, moléculas y proteínas que la distancia de edición encontrada a partir de algoritmos de emparejamiento subóptimos es suficientemente precisa para varias aplicaciones de reconocimiento de patrones.

[3] http://www.gupro.de/GXL/

[4] http://graphml.graphdrawing.org/

En [43], realizan una comparación entre los algoritmos Volgenant-Jonker (VJ), Hungarian, Munkres y como punto de referencia se toma el algoritmo tradicional para el cálculo de la distancia de edición A*, llegando a la conclusión de que el algoritmo VJ se desempeña mejor que los otros algoritmos en cuanto al tiempo de ejecución, seguido por el algoritmo Hungarian. Adicionalmente, la precisión de la clasificación se ve poco afectada por la naturaleza subóptima de los algoritmos y en algunos casos la precisión del VJ y el Hungarian es la misma. Sin embargo, esto no se cumple para todos los conjuntos de datos usados para la comparación.

En la investigación desarrollada en [44] se propone el uso del algoritmo Munkres, algunas veces denominado algoritmo Hungarian, que es comparado con los algoritmos Beam Search, Pathlength y A*, este último es usado como punto de referencia debido a la precisión de sus resultados. Finalmente, se determina que el algoritmo Munkres ejecuta el cálculo de la distancia de edición más rápido que las demás aproximaciones; por otro lado, los resultados del algoritmo Beam Search son los más cercanos a los del algoritmo de referencia.

Por último, en [45] se realiza la comparación de 3 algoritmos que basan su funcionamiento en 3 técnicas de emparejamiento de grafos tolerante a errores diferentes (Fuerza Bruta, Greedy, Bipartito), con el fin de mejorar la búsqueda semántica en un servicio web. El algoritmo de fuerza bruta es tomado como punto de referencia para medir la precisión de los dos algoritmos restantes en donde se concluye que el algoritmo Hungarian que sigue la técnica de emparejamiento bipartito tiene los mismos resultados que el algoritmo de referencia pero con un tiempo de ejecución mucho menor.

2.2.2.1 Similitud entre grafos

A continuación, se presentan algunas investigaciones relacionadas con el cálculo de la similitud entre grafos. Los trabajos expuestos difieren uno del otro en la función de similitud utilizada dependiendo del dominio de aplicación.

En la investigación desarrollada en [29], son presentados y comparados cuatro algoritmos heurísticos para el cálculo de la similitud de los modelos de procesos de negocio basado en el emparejamiento de grafos. Este trabajo se centra en el problema de la clasificación de modelos de procesos en un repositorio de acuerdo con su similitud con respecto a un modelo de proceso dado. Por lo tanto, la necesidad de la búsqueda de similitud surge en múltiples escenarios, donde permite detectar la duplicación o superposición entre los modelos de procesos nuevos y existentes.

El trabajo presentado en [17], propone un nuevo método que redefine el concepto de similitud de dos nodos, el cual involucra el emparejamiento de sus vecinos intentado identificar propiedades deseables. Esta consideración no se encuentra presente en los métodos estudiados para la medida de similitud de nodos de grafos.

2.3 Brechas y aportes

A continuación, son presentados los aportes y brechas identificados en los trabajos relacionados, los cuales permiten establecer el alcance de las investigaciones desarrolladas hasta el momento y determinar aquellas técnicas que no han sido consideradas para el problema definido.

2.3.1 Enfoque agricultura

Sección - Trabajos	Aportes	Brechas
[16], [30]–[36]	Se estudian los efectos climáticos sobre el desarrollo de la roya del café haciendo uso de diversas técnicas computacionales como: redes bayesianas, árboles de decisión, entre otras, para obtener modelos predictivos más precisos de la incidencia de la roya del café. Estos enfoques permiten realizar una aproximación en la caracterización y predicción de la enfermedad. Proporcionan información basada en los resultados de	A pesar de la búsqueda en el mejoramiento de las predicciones de los modelos obtenidos por las técnicas de minería de datos, no se plantean nuevas variantes de investigación. Estos estudios realizan comparaciones en dominios de aplicación similar al de esta propuesta. Sin embargo, estas investigaciones no abordan técnicas como grafos para el reconocimiento de patrones en plagas y enfermedades en la agricultura.

	técnicas de minería de datos que sirve como soporte para la toma de decisiones preventivas por parte del productor.	
[11], [37], [38], [46]	Se propone el uso de intervalos para realizar la predicción de la incidencia de la roya del café, por medio de clasificadores. Se combinan múltiples clasificadores con el fin de mejorar la precisión en los modelos predictivos.	A pesar de que se consideran intervalos para realizar la predicción de la roya del café, al comparar los modelos predictivos con las condiciones actuales solo se tienen dos opciones: está dentro de un intervalo o no está. No es considerada una aproximación a los valores extremos de los intervalos, lo cual incrementa la cantidad de predicciones erróneas.
[7], [39]	Se plantea un nuevo mecanismo de extracción de reglas para la predicción de la roya del café. Se usa una representación basada en grafos para la extracción de las reglas de predicción. Se propone detectar la incidencia de la enfermedad por medio de emparejamiento de grafos.	El enfoque de estos trabajos está en la representación basada en grafos y técnicas de emparejamiento de grafos exactas. Sin embargo, no es considerado el emparejamiento de grafos tolerante a errores, el cual brinda una mayor flexibilidad en los resultados y permite encontrar el grado en que dos grafos se asemejan.

Tabla 1. Aportes y Brechas de los trabajos relacionados con la agricultura.
Fuente: Propia.

En los trabajos relacionados se exponen las diferentes investigaciones vinculadas con la temática, la cual da soporte a los aportes y brechas expuestos en la Tabla 1. Esto nos lleva a identificar que, a pesar de las numerosas aproximaciones en esta área, aún no se han reportado

investigaciones que lleven a cabo una exploración y aplicación de algunas de estas técnicas en la agricultura, específicamente, el emparejamiento de grafos tolerante a errores sobre modelos predictivos de la roya en el café.

De manera análoga, en otras investigaciones [3], [16], [36] se estudian los efectos climáticos sobre el desarrollo de la roya en el café haciendo uso de diversas técnicas computacionales como minería de datos, redes bayesianas, árboles de decisión, entre otras. Estos enfoques permiten realizar una aproximación en la caracterización y predicción de la enfermedad.

2.3.2 Enfoque algoritmos

Sección - Trabajos	Aportes	Brechas
[17], [23], [29], [40], [42], [44].	Se realizan evaluaciones comparativas de los diferentes algoritmos de emparejamiento de grafos para el cálculo de la distancia de edición; teniendo en cuenta la precisión y tiempo de ejecución como métricas de comparación. En las investigaciones hacen uso de las técnicas de emparejamiento de grafos aplicadas en de diversos conjuntos de datos como: manuscritos, moléculas, proteínas e imágenes entre otros, con el fin de determinar la similitud entre estas o identificar patrones.	A pesar de que los estudios realizan comparaciones en diferentes dominios de aplicación y diversos conjuntos de datos, no se tiene en cuenta el emparejamiento de grafos tolerante a errores, como parte de las técnicas para el reconocimiento de patrones en plagas y enfermedades en la agricultura.
[17], [29].	Diferentes técnicas para el cálculo de la similitud son expuestas, concluyendo que el método más flexible es el emparejamiento de grafos tolerante a errores basado en la distancia de edición de	Las funciones de similitud expuestas sólo consideran grafos etiquetados con valores numéricos o cadenas de texto. Sin embargo, no son consideradas etiquetas en nodos y aristas que expresen

	grafos. Diferentes funciones de similitud son adaptadas dependiendo del dominio de aplicación.	rangos numéricos.

Tabla 2. Aportes y Brechas de los trabajos relacionados con las técnicas.
Fuente: Propia

Múltiples propuestas se han realizado con el fin de encontrar los mecanismos más eficientes para el reconocimiento de patrones en diversas áreas, debido a su habilidad para gestionar grafos con estructuras aleatorias y diversos tipos de etiquetas tanto para nodos como aristas. Sin embargo, existen áreas de investigación [14] donde es crucial que la información que representan los grafos incluya rangos numéricos dentro de las etiquetas de los nodos. Debido a la versatilidad de los algoritmos de emparejamiento de grafos tolerante a errores, es posible plantear una función de costo particular, en donde se tenga en cuenta la aproximación entre los límites de los rangos y un valor evaluado.

Resumen

En este capítulo se presenta la investigación documental que está integrada por los conceptos de las técnicas más relevantes para el desarrollo de este proyecto, tales como: grafos, emparejamiento de grafos y similitud entre grafos. Posteriormente, se expone los trabajos relacionados desde dos enfoques: la agricultura y los algoritmos de emparejamiento de grafos tolerantes a errores. A partir de lo anterior, se obtienen las brechas y los aportes del presente trabajo, en donde el emparejamiento de grafos tolerante a errores, debido a las características que posee, puede surgir como una alternativa para apoyar al sector agrícola en la prevención y control de enfermedades.

Capítulo 3

Selección y adaptación de los algoritmos

En este capítulo se presenta una descripción que contiene los aspectos más relevantes de los factores relacionados con la ocurrencia de la roya del café, adicionalmente, se exponen las variables agroclimáticas que tienen mayor influencia en la aparición de la enfermedad y su representación como patrones de grafos. Por otra parte, se mencionan los aspectos más relevantes del proceso realizado para la selección de los algoritmos y una breve descripción de cada uno de ellos. Además, se presenta la adaptación realizada a los algoritmos para que estos puedan procesar los tipos de datos usados en el presente proyecto.

3.1 Introducción al caso de estudio

Se ha determinado en múltiples investigaciones [11], [13]–[15], que ciertas condiciones climáticas y agronómicas favorecen tanto la aparición como la disminución del hongo hemileia vastatrix. En una de ellas [47], se propone el desarrollo de un sistema experto que detecta estas condiciones haciendo uso de árboles de decisión para la construcción de un conjunto de reglas expresadas como patrones de grafos, las cuales fueron evaluadas a través de una técnica de emparejamiento de grafos exacta. Estas reglas fueron construidas a partir de los datos obtenidos en la granja experimental Los Naranjos, perteneciente a la empresa Supracafé, la cual está ubicada en el municipio de Cajibío (Cauca). El fin de esta investigación es poder alertar a los actores involucrados para que tomen acciones preventivas y correctivas, mejorando la productividad y competitividad de los cultivos cafeteros colombianos.

Es importante resaltar que el presente proyecto tiene como punto de partida el conjunto de reglas expresadas por medio de patrones de grafos para la detección de condiciones favorables para roya en el café [14], las cuales serán utilizadas con el fin de compararlas con un conjunto de variables predictivas para evaluar las tasas de infección de un cultivo de café, por medio de algunos algoritmos de emparejamiento de grafos tolerante a errores.

3.2 Factores relacionados con la ocurrencia de la enfermedad

El hongo Hemileia vastatrix necesita condiciones muy particulares para infectar las hojas de la planta de café [3], dentro de estas condiciones se encuentran factores climáticos como: la lluvia, la humedad, el viento, la temperatura entre otros, que bajo ciertas circunstancias permiten el incremento o la disminución de la tasa de infección del cafeto. Además, el manejo agronómico del cultivo puede influir fuertemente en las epidemias de roya del café, debido a las diferentes combinaciones de sombra, densidad del café, fertilización y poda [48]. Por lo tanto, dependiendo del manejo que el agricultor le da al cultivo, puede favorecer la aparición y desarrollo de la enfermedad [3].

3.3 Reglas expresadas como patrones de grafos

Para obtener el porcentaje de incidencia de la roya del café, el Centro Nacional de Investigación del Café(Cenicafé) ha desarrollado una metodología basada en la exploración de un lote en un área igual o menor a una hectárea. En donde inicialmente se seleccionan 60 árboles por lote y de cada árbol se elige la rama con mayor follaje. Posteriormente, se cuenta el número total de hojas y el número de estas afectadas por la roya. Finalmente, el porcentaje de incidencia es calculado dividiendo la suma total de hojas afectadas con roya en los 60 árboles entre la suma total de hojas de los mismos árboles por cien. A partir de lo anterior se puede calcular la tasa de infección evaluando el aumento o disminución del porcentaje de incidencia entre el mes analizado y el mes anterior, obteniendo tres clases o categorías.

- TI1 (<=0): Reducción o latencia, para tasas de infección negativas o nulas.
- TI2 (>0, <=2): Crecimiento moderado, para tasas de infección positivas, menores o iguales a 2 puntos porcentuales (pp).
- TI3 (>2): Crecimiento acelerado, para tasas de infección mayores a 2 pp.

Se ha establecido una relación entre los parámetros climáticos y las propiedades del cultivo, por un lado, y las tres categorías de tasas de infección [47]. De esta manera, a partir de la información que se dispone de las

condiciones del cultivo, es posible estimar en cuál categoría de tasa de infección se puede encontrar un cultivo.

Con base en el conocimiento de expertos, se definieron siete patrones de las condiciones climáticas y las propiedades agronómicas del cultivo que están asociadas a categorías específicas de tasas de infección. Estos patrones se representan mediante grafos como se explica a continuación [47].

Los atributos predictivos basados en información meteorológica fueron construidos a partir de los registros de la estación ubicada en la granja Los Naranjos. Los datos son usados para analizar la temperatura, precipitación, viento y humedad relativa, conforme la relación que los expertos en roya han definido entre estas variables climáticas y la enfermedad. Asimismo, también se tuvieron en cuenta la densidad, sombra y porcentaje de infección. Con base en lo anterior, se plantea un conjunto de variables, las cuales se pueden observar en la Tabla 3 [47].

Atributo	**Descripción**	**Tipo**	**Unidad**
ID	Nombre del patrón y tasa de infección de Roya. {0(TI1),1(TI2), 2(TI3)}	Nominal	-
DENSIDAD	Densidad del lote {3008, 4016, 5013, 6993}	Nominal	-
SOMBRA	Porcentaje de sombrío	Numérico	%
DFAV_ROYA	Cuantos días en el mes fueron favorables para la enfermedad	Numérico	Días
DLLUV	Número de días lluviosos (Precipitación >= 1 mm)	Numérico	Días
HORHR90	Media de número de horas diarias con humedad Relativa >= 90%	Numérico	Horas
HORHRN90	Media de número de horas nocturnas	Numérico	Horas

	diarias con humedad relativa >= 90%		
HR	Media de Humedad Relativa media diaria	Numérico	%
T_HR90	Media de temperatura media diaria durante horas con Humedad Relativa >= 90%	Numérico	°C
T_HRN90	Media de temperatura media diaria durante horas nocturnas con humedad relativa >= 90%	Numérico	°C
TMAX	Media de temperaturas máximas diarias	Numérico	°C
TMIN	Media de temperaturas mínimas diarias	Numérico	°C
DELTAT	Delta de temperatura máxima y mínima	Numérico	

Tabla 3. Variables del conjunto de datos Los Naranjos, Colombia. Tomado de [47].

Para tener una representación completa del conjunto de los elementos involucrados dentro del proceso de infección de la roya del café por medio de patrones expresados como grafos, es necesario considerar las siguientes entidades [47]:

• **Cultivo:** Entidad principal que representa un cultivo determinado, donde sus etiquetas corresponden a propiedades como: identificador único, nombre y ubicación.

• **Instancia:** Entidad relacionada con el registro de los atributos predictivos para una escala de tiempo determinado, que en este caso es un mes. Esta entidad está conectada con otras entidades que determinan los tipos de atributos

medidos y sus valores, como atributos de propiedades de cultivo, condiciones climáticas y control.

En [47] se propone una estructura para el grafo que se presenta en la Figura 1, donde las entidades son la base para su creación. Las entidades corresponden a nodos que contienen un conjunto de etiquetas representando rangos numéricos, los cuales expresan los valores de las variables asociadas con los factores que inciden sobre la enfermedad. Además, las relaciones entre nodos permiten dar significado a la interacción entre entidades por medio de las aristas dirigidas que los conectan y sus etiquetas.

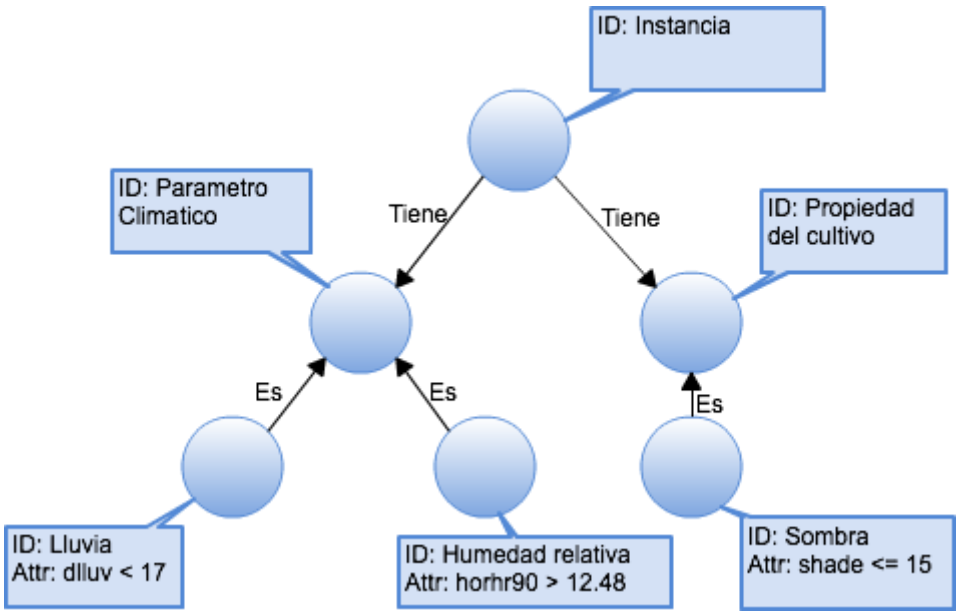


Figura 1. Ejemplo de un subgrafo del grafo de datos. Tomado de [47].

De la investigación [47] se obtienen 7 patrones de grafos que son divididos en tres grupos: 2 patrones para TI1 (Patrón P3 y P7), 3 patrones para TI2 (Patrón P1, P2 y P5) y 2 patrones para TI3 (Patrón P4 y P6). Además, se puede encontrar un resumen de las propiedades de cada uno de los patrones definidos (Ver anexo A).

3.4 Selección de algoritmos

Los algoritmos se seleccionaron con base en diferentes criterios de selección, inicialmente se tuvo en cuenta la recurrencia en que estos se presentan en las diferentes investigaciones. Asimismo, fue necesario considerar la precisión obtenida por cada uno de ellos como uno de los criterios más relevantes debido

a las características dadas del dominio de aplicación. Además, se tuvo en cuenta el enfoque de emparejamiento de grafos etiquetados tolerante a errores, el tiempo de ejecución y finalmente, la obtención del código fuente.

Con el fin de seleccionar los algoritmos que cumplieran con los criterios anteriormente expuestos, se realizó una investigación documental en 120 artículos y dos tesis, de los cuales fueron seleccionados 22 artículos y una tesis.

En la Tabla 4, se muestra una clasificación realizada de los algoritmos preseleccionados en cuanto a la precisión y el tiempo de ejecución. La clasificación se realiza calificando los criterios de selección establecidos en alto, medio y bajo. Con relación al código de fuente, se consideró si era posible obtenerlo o no.

Algoritmo	Artículo	Precisión	Tiempo de Ejecución	Código
A*	[24], [26], [27], [29].	Alto	Bajo	Si
Beam	[23], [24], [42], [44], [49].	Alto	Medio	Si
Greedy	[29], [50], [51]	Medio	Medio	No
Hungarian	[21], [23], [24], [26], [43], [45], [52].	Medio	Alto	Si
Neighbor Matching	[17]	Medio	Medio	Si
Umeyama	[53]–[56]	Bajo	Alto	Si
Volgenant - Jonker	[40], [43], [57], [58].	Medio	Alto	Si
Munkres	[24], [44], [52], [59], [60].	Medio	Alto	No

Tabla 4. Clasificación de las Investigaciones Seleccionadas. Fuente: Propia.

Después de realizar la clasificación de las investigaciones relacionadas con algoritmos de emparejamiento de grafos tolerante a errores (Tabla 4), los resultados conducen a que los algoritmos que cumplieron con los criterios de selección establecidos, y por lo tanto los más apropiados para el desarrollo de este proyecto, son A*, Beam, Hungarian y Volgenant-Jonker.

3.5 Algoritmos de emparejamiento de grafos

En la sección 3.4 fueron identificados los algoritmos de emparejamiento de grafos tolerante a errores más apropiados para el desarrollo de este proyecto; En consecuencia, se eligieron 4 algoritmos: dos de ellos siguen el proceso de evaluación clásico de búsqueda en árbol (A*, Beam) y los otros dos siguen el proceso de emparejamiento de grafos bipartito (Hungarian y Volgenant-Jonker). Las dos técnicas mencionadas anteriormente difieren en la forma en cómo procesan los grafos, dándole características particulares a cada algoritmo. A continuación, se presenta una descripción general de estas técnicas.

3.5.1 Distancia de edición por búsqueda en árbol

La idea básica de los algoritmos de emparejamiento basados en el método de búsqueda en árbol es organizar todos los posibles caminos de edición en un espacio de búsqueda ordenado como un árbol. El nodo raíz del árbol representa el punto de partida del procedimiento de búsqueda, los nodos internos del árbol de búsqueda corresponden a las soluciones parciales y las hojas[5] representan las soluciones completas, pero no necesariamente óptimas. El árbol de búsqueda es construido de manera dinámica por medio de un proceso iterativo, creando y enlazando con aristas los nodos sucesores que corresponden a alguna operación de edición [61].

En el Algoritmo 1 se puede observar el pseudocódigo de la distancia de edición de búsqueda en árbol.

Distancia de edición por búsqueda en árbol

Entrada: Grafos no vacíos $G_p = (V_p, E_p, f_{vp}, f_{ep})$ y $G_s = (V_s, E_s, f_{vs}, f_{es})$
Donde $V_p = \{u_1, \ldots, u_n\}$y $V_s = \{v_1, \ldots, v_n\}$
Salida: Camino de edición con el costo mínimo para transformar G_p en G_s
1: Inicializar $NODOS\ ABIERTOS$ como un conjunto vacío { }
2: Para cada nodo $w \epsilon\ V_s$, insertar operación de sustitución $\{u_1 \rightarrow w\}$ en $NODOS\ ABIERTOS$
3: Insertar operación de eliminación $\{u_1 \rightarrow \varepsilon\}$ en $NODOS\ ABIERTOS$
4: Ciclo
5: Eliminar $\lambda_{min} = arg\ min_{\lambda\ \epsilon\ ABIERTO}\{g(\lambda) + h(\lambda)\}$ de $NODOS\ ABIERTOS$
6: si λ_{min} es un camino de edición completo
7: Retornar λ_{min}
8: sino

---

[5] Hojas: Corresponde a los nodos externos del árbol.

9: Sea $\lambda_{min} = \{u_1 \rightarrow v_{\varphi 1}, \ldots, v_{\varphi k}\}$
10: si $k < n$ entonces
11: Para cada $w \epsilon V_2 \setminus \{v_{\varphi 1}, \ldots, v_{\varphi k}\}$, insertar $\lambda_{min} \cup \{u_{k+1} \rightarrow w\}$ en $NODOS\ ABIERTOS$
12: Insertar $\lambda_{min} \cup \{u_{k+1} \rightarrow \varepsilon\}$ en $NODOS\ ABIERTOS$
13: sino
14: Insertar $\lambda_{min} \cup \cap_{w \epsilon v2 \setminus \{v_{\varphi 1}, \ldots, v_{\varphi k}\}} \{e \rightarrow w\}$en $NODOS\ ABIERTOS$
15: fin si

16: fin si
17: fin ciclo

Algoritmo 1.Distancia de edición por búsqueda en árbol. Tomado de [61].

En el método de búsqueda en árbol para el cálculo óptimo de la distancia de edición, los nodos del grafo fuente G_p son procesados en un orden fijo pero arbitrario $u_1, u_2, \ldots, u_n$. La operación de sustitución (linea 11) y eliminación (línea 12) son consideradas simultáneamente, lo cual produce un número de nodos sucesores en el árbol de búsqueda. Si todos los nodos del primer grafo han sido procesados, los nodos restantes del segundo grafo son insertados en un solo paso (línea 14) [61].

El conjunto de $NODOS\ ABIERTOS$ contiene los nodos del árbol de búsqueda, estos representan los caminos de edición parciales o completos [62]. Con el fin de determinar el camino de edición parcial $\lambda \epsilon NODOS\ ABIERTOS$ más conveniente para la siguiente iteración del algoritmo en donde se agrega una operación de edición (nodo) al árbol, una función heurística $h(\lambda)$ es comúnmente utilizada (linea 5). Formalmente, para un camino de edición parcial λ en el árbol de búsqueda, se usa $g(\lambda)$ para denotar el costo acumulado de las operaciones de edición, y se usa $h(\lambda) > 0$ para denotar el costo estimado para completar el camino de edición λ. La suma de $g(\lambda) + h(\lambda)$ da como resultado el costo total asignado a un camino en el árbol de búsqueda [61].

Además, es importante resaltar que las operaciones de las aristas implicadas por las operaciones de los nodos, pueden ser derivadas de cada camino de edición parcial o completo durante el procedimiento de búsqueda realizado en el algoritmo. El costo de las operaciones de edición de las aristas implicadas es dinámicamente agregado al camino de edición $\lambda \epsilon NODOS\ ABIERTOS$ correspondiente. Para cada operación de edición de nodos se verifica si ya se han realizado operaciones sobre los nodos adyacentes y, si es así, el costo de las operaciones de edición de las aristas es agregado al costo general del camino de edición [62].

3.5.2 Distancia de edición Bipartita

El emparejamiento de grafos bipartito (Algoritmo 2) calcula la distancia de edición de dos grafos a partir de la generación de una matriz que contiene los costos de edición de los nodos; dentro de estos se incluyen los costos de edición de las aristas adyacentes para todas las combinaciones de nodos. En el segundo paso, el algoritmo de asignación calcula la asignación de nodo de costo mínimo; dado esta asignación se infieren las operaciones de edición implícitas de las aristas, y se puede calcular los costos acumulados de las operaciones de edición individuales tanto en los nodos como en las aristas. Por lo tanto, la distancia de edición exacta de la asignación dada se puede calcular en tiempo lineal. Es importante considerar que puede haber otras asignaciones con el mismo costo mínimo de asignación de nodos, pero posiblemente una distancia de edición exacta más pequeña, ya que la estructura de la arista explícita sólo se comprueba después de que se ha aplicado el algoritmo de asignación. Por lo tanto, las asignaciones de nodos y aristas implícitas encontradas por el algoritmo no tienen que corresponder a una solución óptima, lo que lleva a un costo de edición calculado mayor o igual que su distancia de edición del grafo real. Dado este mapeo, los costos sirven como una aproximación de la distancia de edición del grafo. Los valores aproximados de distancia de edición obtenidos por este procedimiento son iguales o mayores que los valores exactos de la distancia, ya que el enfoque subóptimo encuentra una solución óptima en un subespacio del espacio de búsqueda completo [43].

Distancia de edición Bipartita

1: Construir la matriz de costo $C^* = (c_{ij}^*)$ de acuerdo con los grafos de entrada $g1\ y\ g2$
2: Calcular la asignación óptima de nodos $\psi = \{u_1 \rightarrow v_{\psi 1} \ldots\ldots\ldots\ldots, u_{m+n} \rightarrow v_{\psi_{m+n}})\ en\ C^*\}$
3: Completar el camino de edición de acuerdo con ψ y retorna $d_\psi(g1, g2)$ y/o $d'_\psi(g1, g2)$.

Algoritmo 2. Distancia de edición bipartita. Tomado de [61].

En una primera instancia se construye la matriz de costos (Ecuación 1), en donde la parte superior izquierda de la matriz representa los costes de todas las posibles sustituciones de nodos, la diagonal de la parte superior derecha los

costos de todas las eliminaciones de nodos posibles, y la diagonal de la parte inferior izquierda los costos de todas las posibles inserciones de nodos. La parte inferior derecha de la matriz de coste se establece en cero puesto que las sustituciones de la forma $(\varepsilon \rightarrow \varepsilon)$ no deben causar ningún coste.

$$\mathbf{C} = \begin{array}{c|cccc|cccc} & v_1 & v_2 & \cdots & v_m & \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_n \\ \hline u_1 & c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ u_2 & c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ u_n & c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ \hline \varepsilon_1 & c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \varepsilon_2 & \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \varepsilon_m & \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{array}$$

Ecuación 1. Matriz de costos.

Por consiguiente, la entrada c_{ij} indica el coste $c(u_i \rightarrow v_j)$ de la sustitución de nodo $(u_i \rightarrow v_j)$, $c_{i\varepsilon}$ denota el costo $c(u_i \rightarrow \varepsilon)$ de la eliminación del nodo $(u_i \rightarrow \varepsilon)$, y $c_{\varepsilon j}$ denota el costo $c(\varepsilon \rightarrow v_j)$ de la inserción del nodo $(\varepsilon \rightarrow v_j)$.

Posteriormente, se calcula el camino de edición óptimo ψ que contiene todas las operaciones de edición para transformar $g_1$en $g_2$con el mínimo costo. Finalmente, el algoritmo retorna la distancia de edición $d'_{\psi}(g1, g2)$ asociada al camino de edición óptimo.

3.6 Descripción de los algoritmos

Con el propósito de identificar las características más relevantes de los algoritmos de búsqueda en árbol: A* y Beam y los algoritmos bipartitos: Hungarian y Volgenant-Jonker. A continuación, se hace una breve descripción de cada uno de ellos.

3.6.1 Algoritmo A*

Es un algoritmo de búsqueda en árbol ampliamente usado para calcular el valor exacto de la distancia de edición que explora todo el espacio de todos los posibles mapeos de nodos y aristas de ambos grafos atribuidos [59]. El algoritmo A* es uno de los algoritmos con mayor precisión según la literatura. Sin embargo, tiene un tiempo de ejecución exponencial con respecto a el tamaño del grafo y un alto consumo de recursos computacionales [26].

El algoritmo A* ejecuta una exploración de los posibles mapeos dentro de dos grafos, organizando el espacio de búsqueda subyacente como un árbol ordenado por medio de la creación de nodos sucesores enlazados por aristas a los nodos actualmente considerados en el árbol de búsqueda [63], [64]. Formalmente, para un nodo p en el árbol de búsqueda es usada una función $g(p)$ para denotar el costo del camino óptimo, desde la raíz del nodo al nodo actual p. Además, este algoritmo usa una función heurística $h(p)$ con el fin de determinar el nodo con las mejores condiciones en el árbol de búsqueda actual, y denotar el costo estimado desde el nodo p hasta un nodo externo. La suma de $g(p) + h(p)$ da el costo total asignado a un nodo abierto en el árbol de búsqueda. Finalmente, el costo mínimo total es obtenido por $pmin \leftarrow argmin\{g(p) + h(p)\}$ que es el camino más corto entre el nodo raíz y el nodo actual. La distancia entre dos grafos es medida aplicando una serie de operaciones de edición (Inserción, eliminación, y/o sustitución de nodos y aristas).

3.6.2 Algoritmo Beam

Este algoritmo está basado en el principio de funcionamiento del A*, sin embargo, proporciona la posibilidad de establecer el tamaño del espacio de búsqueda, y dependiendo de este varía proporcionalmente su precisión.

El método sigue una idea similar a la técnica descrita previamente, pero en lugar de expandir todos los nodos sucesores en la búsqueda del árbol, está limita el número s de nodos a ser procesados que son mantenidos en el conjunto de nodos abiertos en todo momento. Siempre que se agregue un nuevo camino de edición parcial, solo el camino de edición parcial p con el más bajo costo $g(p) + h(p)$ es mantenido y los caminos de edición parcial restantes son eliminados. Esto significa que no se explora el espacio de búsqueda completo, solo los nodos que pertenecen a los emparejamientos parciales más prometedores son expandidos [49], [63].

3.6.3 Algoritmo Hungarian

El algoritmo Hungarian fue propuesto por Kuhn [65] y es conocido por resolver el problema de asignación. En esta investigación se usa una adaptación del algoritmo original que permite calcular la distancia de edición de dos grafos, la cual puede ser aplicada a cualquier tipo de grafo [26]. Este método está basado en un procedimiento de optimización bipartito de mapeo de nodos o aristas, en donde los nodos o aristas de un grafo son mapeadas en otro [44].

Uno de los principales problemas en el emparejamiento de grafos es que los algoritmos estándar para el cálculo de similitud de grafos que basan su funcionamiento en una búsqueda en árbol, son exponenciales en el tiempo de ejecución dependiendo del número de nodos involucrados. Por lo tanto, estos algoritmos son aplicables a grafos pequeños solamente. Por este motivo, se hace uso del algoritmo Hungarian para calcular la distancia de edición, el cual calcula un emparejamiento completo de grafos, tal que la suma de los pesos de las aristas en el emparejamiento es minimizado [45], además su uso para el emparejamiento de grafos bipartitos es deseado debido a que proporciona un proceso mucho más rápido en donde la precisión es poco afectada [44].

3.6.4 Algoritmo Volgenant-Jonker

Este Algoritmo debido a su eficiencia ha sido objeto de atención en la literatura actual, consiste en tres pasos: Un método de preprocesamiento usado para encontrar la primera solución parcial, una etapa de dispersión para resolver una instancia con un número reducido de aristas seguidas por un procedimiento que iterativamente agrega aristas hasta obtener una solución óptima y finalmente, un procedimiento para encontrar el camino más corto [43].

El algoritmo Volgenant-Jonker es utilizado para resolver el problema de asignación, lo que conduce a un cálculo más rápido de la distancia de edición de grafos, ejecutándose en un tiempo polinomial; además, proporciona un cálculo de la distancia de edición subóptima, sin afectar considerablemente la precisión; la razón de su suboptimalidad es que la información de la arista es tomada en cuenta sólo de manera limitada durante el proceso de encontrar la asignación óptima del nodo entre dos grafos [43].

3.7 Adaptación de la función de similitud en los algoritmos seleccionados

Los grafos utilizados contienen información agroclimática en donde las etiquetas de los nodos expresan rangos numéricos y las aristas dan sentido a la interacción de las variables. Por este motivo, se hace necesario realizar la adaptación propuesta en esta sección, con el fin de buscar con mayor precisión la cercanía entre un patrón de roya y una instancia.

Dadas las características del dominio de aplicación es necesario considerar si un valor dado por una instancia está dentro de un intervalo específico de un patrón, debido a que, para determinar el incremento o disminución de la tasa de infección de un cultivo, es de gran importancia que los valores de los parámetros agroclimáticos estén entre los rangos establecidos por los patrones. Por ejemplo, un valor de la etiqueta de una instancia puede no encajar en un rango determinado por la etiqueta de un patrón, este hecho es penalizado con un costo de edición que disminuye la similitud entre las condiciones agroclimáticas y el patrón de roya evaluado. Por otro lado, también es tenida en cuenta la diferencia entre el valor y el límite del rango más cercano a este, lo que permite evaluar la cercanía entre las variables agroclimáticas presentes en un cultivo y los rangos establecidos por los patrones.

Por otra parte, las aristas de los grafos contienen la información semántica del dominio de aplicación y da una representación de la interacción de las variables agroclimáticas. Por consiguiente, es necesario realizar la comparación sólo de aquellas aristas que pertenecen al mismo tipo de nodo independientemente de la estructura de los grafos comparados, en donde se verifica si estas aristas tienen el mismo sentido y la misma etiqueta, de lo contrario, los resultados de la comparación no serían significativos puesto que se estarían comparando elementos de los grafos que no tienen ninguna relación.

Como se ha mencionado anteriormente, en esta investigación se toman las 3 operaciones básicas de la distancia de edición (Eliminación, Inserción y Sustitución). La eliminación e inserción tiene un costo de operación fijo, por ejemplo $Costo\ de\ eliminar\ un\ nodo\ o\ arista = 1$. La sustitución de nodos y aristas se compone de un conjunto de operaciones de edición efectuadas por las funciones de costo que fueron adaptadas a los algoritmos. La sustitución de una arista está compuesta por: el costo de la sustitución de la etiqueta y el costo de la inversión de la arista en el caso de existir. Asimismo, el costo de la sustitución de un nodo está dado por la diferencia entre los valores de las etiquetas y por un costo definido en el caso de que el valor evaluado no esté dentro del rango dado por una etiqueta de un patrón.

En este proyecto se hace uso de la definición de grafos dirigidos obtenida en [58], donde $G_p = (V_p, E_p, f_{vp}, f_{ep})$ es definido como un grafo; V_p es el conjunto de nodos y E_p es el conjunto de aristas, de manera que las aristas son dirigidas y conectan dos nodos V_i y V_j si $(V_i, V_j) \in E_p$. Adicionalmente, pueden ser encontrados nodos V_i con múltiples aristas dirigidas $E_k = (V_i, V_j)$ donde $k = 1 \ldots n$. De manera similar $f_{vp}(V_i)$ es definida como el conjunto de etiquetas de un nodo V_i, finalmente, $f_{ep}(E_i)$ es la etiqueta de la arista E_i.

Se asumen dos grafos $G_p = (V_p, E_p, f_{vp}, f_{ep})$ llamado grafo patrón y $G_s = (V_s, E_s, f_{vs}, f_{es})$ denominado grafo instancia, en donde el patrón contiene rangos de propiedades agronómicas y climáticas, y el grafo instancia mantiene los parámetros que se van a analizar. Para que los algoritmos puedan procesar estos dos tipos de grafos, se propone cuatro funciones de costo: función de costo de diferencia de etiquetas de nodos, función de costo de rango, función de costo de etiquetas de aristas y función de costo de inversión de aristas.

La función de etiquetas de nodos permite rangos numéricos como datos de entrada y considera si un valor evaluado está o no está dentro del rango. Además, la función debe considerar si este valor se aproxima más por la derecha o por la izquierda del rango.

Para mayor comprensión del principio operativo de la función de costo de etiquetas de nodos, se presenta una descripción de esta función por medio de un pseudocódigo mostrado en el Algoritmo 3.

Función de costo de diferencia de etiquetas de nodos

1: Procedimiento Función Propuesta (V_p, f_{vp},V_s, f_{vs})
2: Inicializar $costo$ en cero
3: Si existe al menos un $v_i \in V_p$ y un $v_j \in V_s$
4: Ciclo hasta mapear todos los nodos V_p en todos los nodos V_s
5: Si v_i y v_j son del mismo tipo
6: Ciclo hasta recorrer todas las etiquetas de v_i y v_j
7: Si el nombre del atributo $f_{vp} \in v_i$ y $f_{vs} \in v_j$ es igual
8: Evaluar valor f_{vs} en el rango f_{vp}
9: Determinar límite del rango f_{vp} más cercano a f_{vs}
10: $costo = costo + ValorAbs\ (\ valor\ f_{vs}$ – $límite\ más\ cercano)$
11: Fin si
12: Fin ciclo
13: Fin si
14: Fin ciclo
15: Fin si
16: Retornar $costo$

Algoritmo 3. Función de costo de diferencia de etiquetas de nodos

Los rangos están expresados como etiquetas en los nodos y son evaluadas usando la función propuesta. El emparejamiento de estas etiquetas busca encontrar la cercanía entre un valor dado por una instancia y el valor del rango más cercano a esta.

Función de costo de rango

1: Procedimiento Función Propuesta (V_p, f_{vp},V_s, f_{vs})
2: Inicializar $costo$ en cero
3: Si existe al menos un $v_i \epsilon V_p$ y un $v_j \epsilon V_s$
4: Ciclo hasta mapear todos los nodos V_p en todos los nodos V_s
5: Si v_i y v_j son del mismo tipo
6: Ciclo hasta recorrer todas las etiquetas de v_i y v_j
7: Si el nombre del atributo $f_{vp} \epsilon v_i$ y $f_{vs} \epsilon v_j$ es igual
8: Evaluar valor f_{vs} en el rango f_{vp}
9: Si f_{vs} no está en el rango f_{vp}
10: Incrementar $costo$ en 1
11: Fin si
12: Fin si
13: Fin ciclo
14: Fin si
15: Fin ciclo
16: Fin si
17: Retornar $costo$

Algoritmo 4. Función de costo de rango.

Adicionalmente, es definida la función de costo de rango (Algoritmo 4) que verifica si los nodos son del mismo tipo y las etiquetas tiene el mismo atributo en los dos grafos antes de realizar la comparación de sus valores, es necesario aclarar que no es tenida en cuenta la diferencia entre los valores, sino, si el valor evaluado está o no dentro del intervalo. En el caso de que el valor no esté dentro del rango se le asigna un peso.

Función de costo de etiquetas de aristas

1: Procedimiento Función Propuesta (V_p,E_p, f_{ep},V_s, E_s, f_{es})
2: Inicializar $costo$ en cero
3: Si $E_p$es una arista entre (V_i, V_i') y $E_s$es una arista entre (V_j, V_j')
4: Si V_i y V_j son del mismo tipo y V_i' y V_j' son del mismo tipo

5: Si $f_{ep}! = f_{es}$
6: Incrementar $costo$en 1
7: Fin si
8: Fin si
9: Fin si
10: Retornar $costo$

Algoritmo 5. Función de costo de etiquetas de aristas

La función de costo de etiquetas de aristas (Algoritmo 5), en donde sólo aquellas etiquetas del mismo tipo de nodos son comparadas y en el caso de que las etiquetas no sean las mismas, se asigna un peso.

V_i representa un nodo de un patrón y V_i' es otro nodo del patrón, de forma análoga para los nodos de las instancias V_jy V_j', se debe verificar que la arista pertenezca a nodos del mismo tipo en el patrón y en la instancia para poder que la etiqueta de la arista sea verificada.

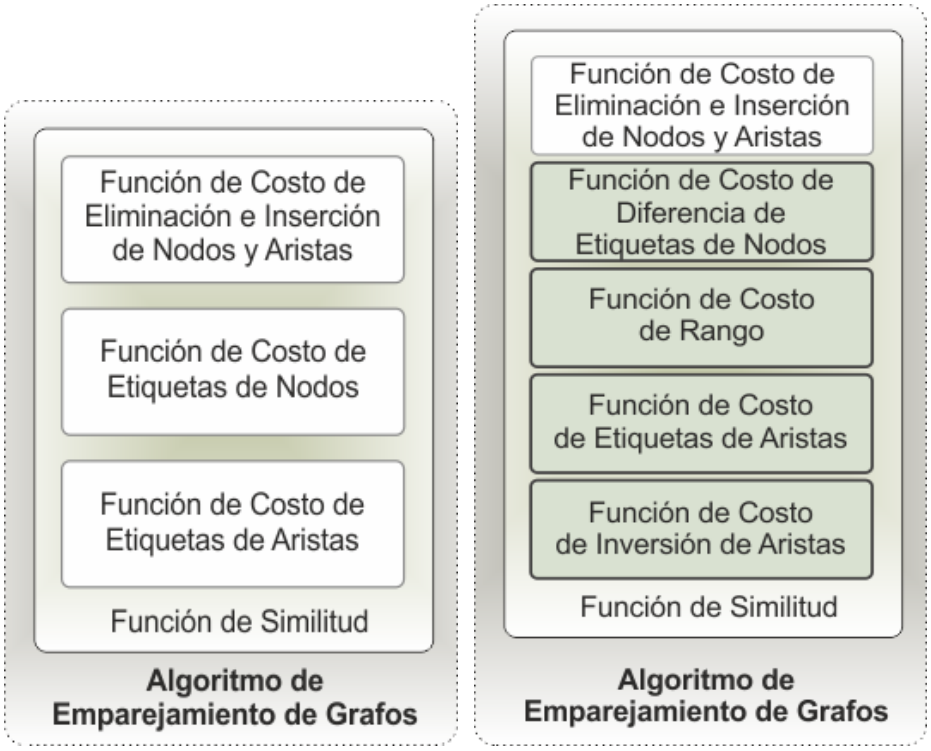
Asimismo, fue necesario considerar la inversión de una arista. Esta operación se presenta en el caso de que dos aristas (una de cada grafo) tengan direcciones contrarias en los dos grafos comparados. La función de inversión de arista es presentada en el Algoritmo 6.

Función de costo de inversión de aristas

1: Procedimiento Función Propuesta (V_p,E_p,V_s, E_s)
2: Inicializar $costo$ en cero
3: Si $E_p$es una arista entre (V_i, V_i') y $E_s$es una arista entre (V_j, V_j')
4: Si V_i y V_j' son del mismo tipo y V_i' y V_j son del mismo tipo
5: Incrementar $costo$ en 1
6: Fin si
7: Fin si
8: Retornar $costo$

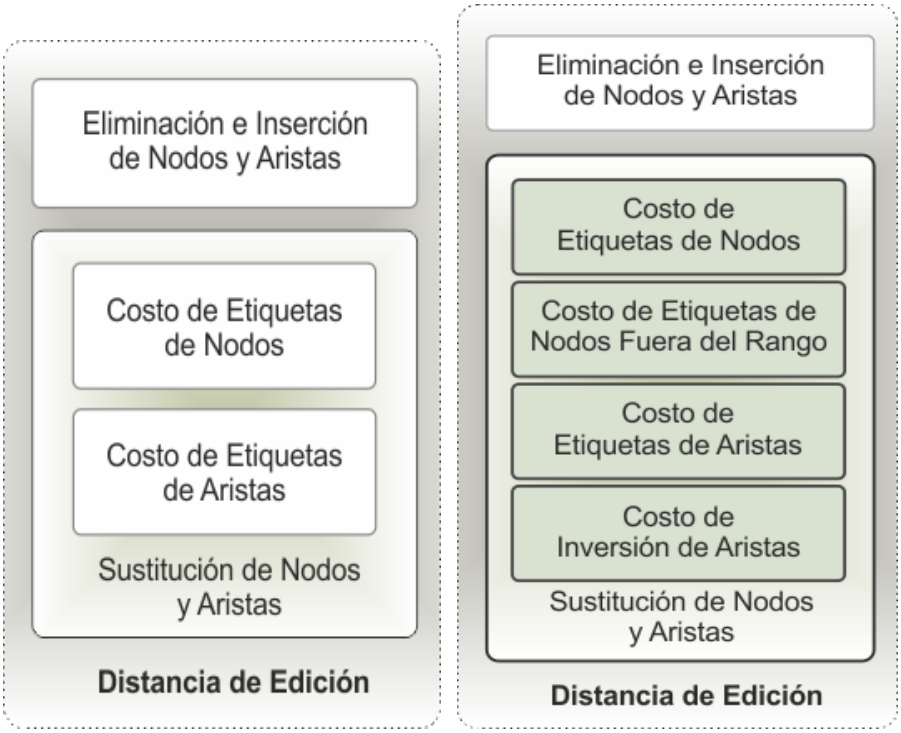
Algoritmo 6. Función de costo de inversión de aristas. Fuente: Propia.

Con el propósito de tener una mayor comprensión de la adaptación realizada, en la Figura 2 (a) se presenta un paralelo entre el diagrama de la composición inicial de los algoritmos y en (b) la adaptación propuesta para el dominio de aplicación.



(a) (b)

Figura 2. Paralelo entre (a) algoritmo original y (b) adaptación realizada. Fuente: Propia.



(a) (b)

Figura 3. Paralelo entre (a) operaciones de edición originales y (b) operaciones de edición adaptadas. Fuente: Propia.

Los componentes de las funciones de costo fueron modificados y en algunos casos sustituidas completamente por las funciones anteriormente propuestas. Con esta adaptación se busca una mayor flexibilidad en cuanto a las coincidencias encontradas correspondientes a los rangos definidos por los patrones utilizados.

Resumen

En este capítulo se realiza la introducción al caso de estudio en el que se describen brevemente las condiciones agroclimáticas que inciden en la ocurrencia de la roya del café y en su desarrollo. Además, se presenta un conjunto de reglas expresadas por medio de patrones de grafos para la detección de condiciones favorables para roya en el café, obtenidas de [47], que son usadas como punto de partida en esta investigación. Por otro lado, se muestra el proceso llevado a cabo para la selección de los algoritmos de emparejamiento de grafos tolerante a errores y se presenta una reseña concisa de los 4 algoritmos elegidos: dos de ellos siguen el proceso de evaluación clásico de búsqueda en árbol (A*, Beam) y los otros dos siguen el proceso de emparejamiento de grafos bipartito (Hungarian y Volgenant-Jonker). Finalmente, es expuesta la adaptación realizada a los algoritmos anteriormente mencionados, donde se especifica las funciones adaptadas y sus pseudocódigos con el fin de proporcionar un mayor entendimiento de esta adaptación.

Capítulo 4

Normalización y parametrización de los resultados de acuerdo con el dominio de aplicación

Este capítulo contiene el proceso efectuado para caracterizar la adaptación realizada a los algoritmos de emparejamiento de patrones en grafos con el dominio de aplicación. Por lo tanto, se presenta la normalización de los resultados obtenidos y la parametrización del peso de las funciones de costo. Estos aspectos permiten que la adaptación realizada proporcione resultados acordes a los buscados para el dominio de aplicación.

4.1 Normalización de los resultados

La normalización del conjunto de datos se encuentra entre los procesos de exploración de datos, en los que los datos de los atributos se escalan en un rango especificado. Además, la normalización es el paso central para estandarizar valores de características o atributos de diferentes rangos dinámicos en un rango específico [66].

En esta investigación se normalizan los resultados obtenidos por cada una de las funciones para que los valores estén en un intervalo de [0,1], donde 1 indica que los grafos son exactamente iguales y 0 que los grafos son totalmente distintos. Después de normalizar los resultados de las funciones de costo, se les asigna una ponderación (sección 4.4) y finalmente los valores obtenidos son sumados y acotados entre 0 y 1.

Es importante seleccionar un procedimiento de normalización específico de acuerdo con la naturaleza del conjunto de datos para el análisis. Por esta razón, la normalización utilizada en las funciones de *Costo de inserción y eliminación de aristas y nodos(Ce), Costo de inversión de aristas(Cia), Costo fuera de rango(Cfr) y Costo etiquetas de aristas(Cea)*, se obtiene dividiendo el costo de cada uno por la cantidad de comparaciones posibles. Para normalizar el $Costo\ etiquetas\ de\ nodos\ (Cen)$ se selecciona la normalización de máximos

propuesta en [66], la cual sigue el proceso de tomar los datos medidos en sus unidades y transformarlos en un valor entre 0 y 1; Esto proporciona una manera fácil de comparar valores que se miden usando diferentes escalas o diferentes unidades de tratamiento.

Para normalizar el costo de inserción y eliminación de aristas y nodos se hace uso de la ecuación 2, donde se tiene en cuenta los costos de operación de edición sobre cada nodo y arista, y la cantidad de nodos y aristas tanto del grafo patrón como del grafo instancia.

$$Ce' = \frac{Ce}{\lambda(V_p) + \lambda(V_s) + \lambda(E_p) + \lambda(E_s)} = \frac{\sum_{i=0}^{k} e_i}{\lambda(V_p) + \lambda(V_s) + \lambda(E_p) + \lambda(E_s)}$$

Ecuación 2. Costo de inserción y eliminación de aristas y nodos normalizado

Donde:

$e_i =$ Costo de inserción o eliminación de arista o nodo
$\lambda(V_p) =$ Cantidad de nodos del grafo patrón
$\lambda(V_s) =$ Cantidad de nodos del grafo instancia
$\lambda(E_p) =$ Cantidad de aristas del grafo patrón
$\lambda(E_s) =$ Cantidad de aristas del grafo instancia

La ecuación 3 es utilizada para calcular el costo de inversión de aristas normalizado, donde se tienen en cuenta parámetros como: cantidad total de aristas comparadas y el costo de operación sobre la inversión de las aristas.

$$Cia' = \frac{Cia}{\lambda(tec)} = \frac{\sum_{i=0}^{k} e_i}{\lambda(tec)}$$

Ecuación 3. Costo inversión de aristas normalizado

Donde:

$\lambda(tec) =$ Cantidad total de etiquetas de aristas comparadas.
$e_i =$ Costo de la operación de la inversión de arista.

La ecuación 4 es utilizada para calcular el costo de etiquetas de aristas normalizado, donde se tienen en cuenta parámetros como: cantidad total de etiquetas de aristas comparadas y el costo de operación sobre cada arista.

$$Cea' = \frac{Cea}{\lambda(tfec)} = \frac{\sum_{i=0}^{k} efe_i}{\lambda(tfec)}$$

Ecuación 4. Costo de etiquetas de aristas normalizado

Donde:

$\lambda(tfec) =$ Cantidad total de etiquetas de aristas comparadas.
$efe_i =$ Costo de la operación sobre cada etiqueta de las aristas comparada.

Como ya se ha mencionado anteriormente, para el costo de etiquetas de nodos se seleccionó la normalización de [66], debido a que esta es más apropiada para el tipo de datos que se desea analizar (ecuación 5).

$$Cen' = \frac{Cen}{\sum_{i=0}^{m} Maxefv_i} = \frac{\sum_{i=0}^{n} efv_i}{\sum_{i=0}^{m} Maxefv_i}$$

Ecuación 5. Costo de etiquetas de nodos normalizado

Donde:

$efv_i =$Costo de la operación sobre cada etiqueta de los nodos comparada.
$Maxefv_i =$Máxima diferencia entre las etiquetas de los nodos de las instancias y patrones.

Debido a las características del dominio de aplicación, es necesario normalizar la cantidad de comparaciones de las etiquetas de los nodos que están por fuera del rango (Ecuación 6).

$$Cfr' = \frac{Cfr}{\lambda(tfvc)} = \frac{\sum_{i=0}^{k} or_i}{\lambda(tfvc)}$$

Ecuación 6. Costo de valores fuera del rango normalizado

Donde:

$or_i =$ Cantidad de valores de instancias fuera del rango
$\lambda(tfvc) =$ Cantidad total de aristas de nodos comparados

Por último, se halla la similitud entre grafos (ecuación 7), en donde se tienen en cuenta los resultados de las ecuaciones de normalización descritas anteriormente. (Ecuaciones 3-6).

$$similitud = 1 - (Ce' * wCe + Cia' * wCia + Cea' * wCea + Cen' * wCen + Cfr' * wCfr)$$

Ecuación 7. Similitud entre grafos.

Para encontrar el valor total de la similitud, se asigna una ponderación a cada uno de los resultados de las funciones de costo (Ecuaciones 8-12). Esta

ponderación fue encontrada con base en las operaciones básicas de la distancia de edición (Eliminación, Inserción y Sustitución). La ponderación asociada a eliminación e inserción es la misma debido a que no existe una diferencia sustancial entre una y la otra, solo el orden en que se están comparando los grafos, es decir, si se desea transformar un grafo patrón g_p en un grafo instancia g_i, se deben insertar nodos y aristas, pero si se desea transformar un grafo instancia g_i en un grafo patrón g_p se deben eliminar. Por otra parte, como se dijo en la sección 3.7, las operaciones sobre las etiquetas de los nodos y aristas son las que tienen mayor influencia en la predicción de la tasa de infección. Por tal motivo, a todas las funciones de costo que componen la operación de sustitución se les asigna una ponderación individual, con el fin de tener un mayor control en los resultados y por lo tanto una mayor precisión en los mismos.

$$wCe = \frac{nec}{nec + eic + elc + orc + nlc}$$

Ecuación 8. Ponderación del costo de eliminación e inserción de aristas y nodos

$$wCia = \frac{eic}{nec + eic + elc + orc + nlc}$$

Ecuación 9. Ponderación del costo de inversión de arista

$$wCea = \frac{elc}{nec + eic + elc + orc + nlc}$$

Ecuación 10. Ponderación del costo de las etiquetas de las aristas

$$wCen = \frac{nlc}{nec + eic + elc + orc + nlc}$$

Ecuación 11. Ponderación del costo de las etiquetas de los nodos

$$wCfr = \frac{orc}{nec + eic + elc + orc + nlc}$$

Ecuación 12. Ponderación del costo de los valores fuera del rango

Donde:

nec =Peso de las operaciones sobre eliminación e inserción de aristas y nodos
eic =Peso de la inversión de una arista.
elc =Peso de las operaciones sobre las etiquetas de las aristas.
olc =Peso de los valores evaluados que se encuentra fuera del rango.
nlc =Peso de las etiquetas de los nodos.

4.2 Parametrización de los pesos de las funciones de costo

Para alcanzar una mayor precisión en los resultados obtenidos a través de los algoritmos, se asigna un peso a cada una de las funciones de costo, los cuales fueron variados de acuerdo con los resultados obtenidos entre la comparación de los datos proporcionados por el experto y los resultados de los algoritmos de emparejamientos de grafos, con el objetivo de encontrar la mayor similitud entre ambos.

A partir de la comparación de los datos proporcionados por el experto y los resultados de los algoritmos de emparejamientos de grafos, se les asignaron diferentes pesos a las funciones de costo siguiendo un proceso heurístico. En donde se realizaron un conjunto de pruebas con diferentes combinaciones de los pesos en cada una de ellas, buscando obtener los valores que proporcionen una mayor similitud entre los dos conjuntos de datos. Los pesos fueron validados a partir de la maximización de los verdaderos positivos adquiridos a través de la comparación de los resultados entre los dos conjuntos.

El rango establecido para los pesos de las funciones de costo está dado entre 0 y 10. Los valores que proporcionaron una mayor precisión son los siguientes:

Inserción y eliminación de nodos y aristas (nec)	Inversión de arista (eic)	Valores fuera del rango (elc)	Etiqueta de arista (olc)	Etiquetas de nodos (nlc)
8	1	7	1	6

Tabla 5. Pesos de las funciones de costo. Fuente: Propia

Con el fin de proporcionar una mayor comprensión del funcionamiento de la adaptación y del procedimiento que se realiza, se toma como ejemplo un grafo g_1 (Figura 4) correspondiente a un patrón de aparición de la roya y un grafo g_2 (Figura 5) que hace referencia a una instancia.

Es importante diferenciar entre los pesos asignados a las funciones de costo (tabla 5) y los pesos de las operaciones de edición. Como se explicó en la sección 3.7 existen 3 operaciones de edición básicas eliminación, inserción y sustitución de aristas y nodos. A las dos primeras se les asigna un peso de 1, el peso de la sustitución depende de la operación realizada. A las operaciones de inversión de arista, valor fuera del rango y sustitución de etiqueta de arista

se les da un valor de 1 y la operación de comparación de las etiquetas de los nodos se obtiene por medio de la diferencia mínima entre el valor evaluado y los límites del intervalo. A cada función de costo se le asigna un peso (Tabla 5) que posteriormente es transformado por el algoritmo en un coeficiente de ponderación para cada uno de los resultados de las funciones de costo.

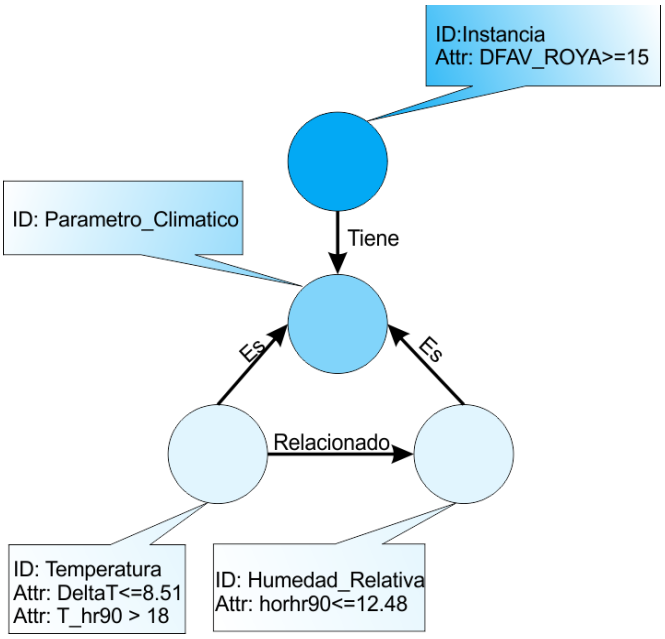


Figura 4. Grafo patrón g_1 . Tomado de [47]

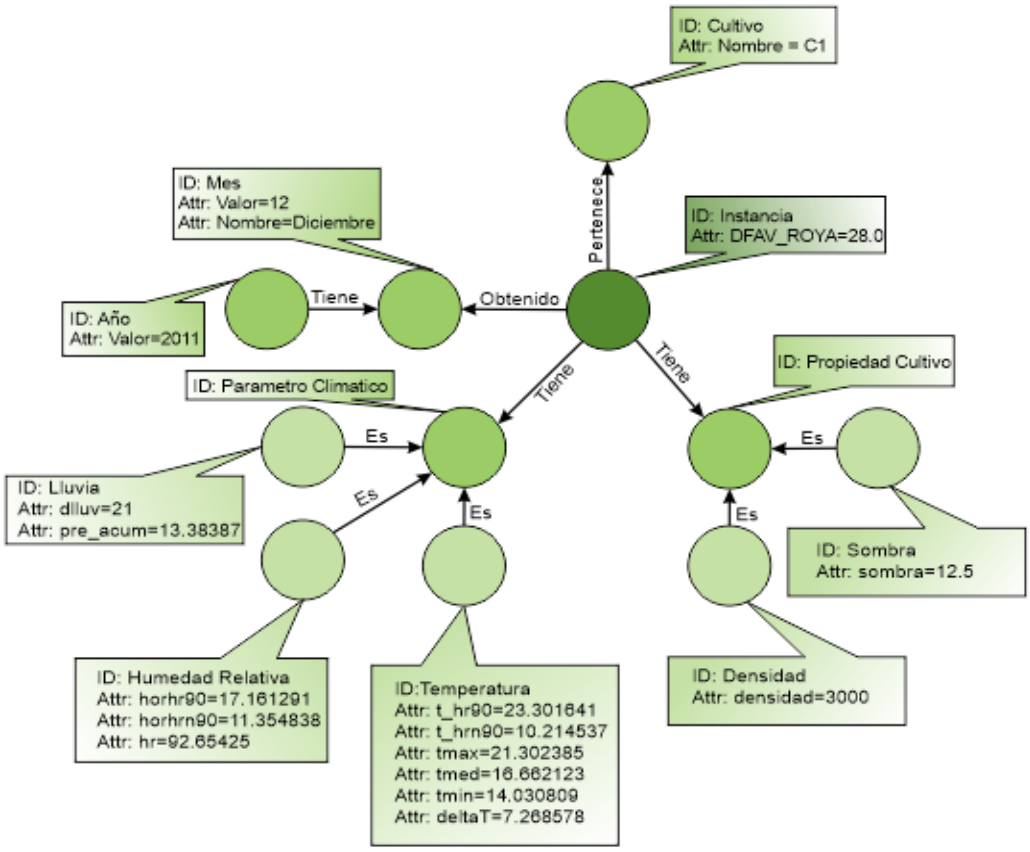


Figura 5. Grafo Instancia g_2. Tomado de [47]

En primer lugar, el algoritmo procesa los grafos de entrada (patrón e instancia) buscando encontrar la menor cantidad de operaciones (ecuación 13 y 14) para poder transformar estructuralmente un grafo en otro.

$$\lambda = \{(Cultivo \rightarrow \varepsilon), (Mes \rightarrow \varepsilon), (\text{Año} \rightarrow \varepsilon), (Propiedad\ Cultivo \rightarrow \varepsilon), (Sombra \rightarrow \varepsilon), (Densidad \rightarrow \varepsilon), (Lluvia \rightarrow \varepsilon)\}$$

Ecuación 13. Operaciones de edición de nodos.

Estas operaciones de edición de nodos implican las siguientes operaciones de edición sobre las aristas:

$$\{((Instancia, Cultivo) \rightarrow \varepsilon), ((Instancia, Mes) \rightarrow (\varepsilon)), ((Instancia, Propiedad\ Cultivo) \rightarrow (\varepsilon)), ((\text{Año}, Mes) \rightarrow (\varepsilon)), ((Sombra, Propiedad\ Cultivo) \rightarrow (\varepsilon)), ((Densidad, Propiedad\ Cultivo) \rightarrow (\varepsilon)), ((Lluvia, Parámetros\ Clima) \rightarrow (\varepsilon)), ((\varepsilon) \rightarrow (Temperatura, Humedad\ Relativa))\}$$

Ecuación 14. Operaciones de edición de aristas.

Cada operación de edición tiene un peso determinado, sin embargo, existen nodos que no son significativos en el momento de encontrar la similitud entre grafos, por lo tanto, las operaciones de edición que involucran los nodos con id: cultivo, mes y año, no tienen un peso sobre el costo de la distancia de edición, además, las operaciones sobre las aristas que los involucran tampoco tienen peso. A las operaciones de edición (inserción y eliminación de nodos y aristas) de los demás nodos y aristas se le asigna un peso.

En el ejemplo, se eliminan 7 nodos (ecuación 13) y 7 aristas, además, se realiza la inserción de una arista (ecuación 14), de los cuales no se tiene en cuenta las operaciones realizadas sobre 3 de los nodos y 3 de las aristas. Por lo tanto, la suma de las operaciones anteriores da un costo de edición parcial de 9 (ecuación 15).

$$Ce = \sum_{i=0}^{k} e_i = 9$$

Ecuación 15. Ejemplo - Costo de inserción y eliminación de aristas y nodos sin normalizar

Luego de efectuar el emparejamiento de los nodos y aristas de los grafos, se procede a comparar las etiquetas de los nodos y los valores de estas, donde solo son comparados los valores de las etiquetas que tiene el mismo nombre y pertenecen al mismo tipo de nodo, y los demás son descartados (Ecuación 16). En este caso es comparado la etiqueta $deltaT$ y T_hr90 del nodo Temperatura, la etiqueta $horhr90$ del nodo Humedad Relativa y la etiqueta $DFAV\_ROYA$ del nodo instancia, al finalizar el procedimiento se obtienen los siguientes resultados:

$$costo\ deltaT = abs(deltaT_{Patron} - deltaT_{Instacia}) = abs(8.51 - 7.26) = 1.25$$
$$costo\ T\_hr90 = abs(T\_hr90_{Patron} - T\_hr90_{Instancia}) = abs(18 - 23.301641) = 5.301641$$
$$costo\ horhr90 = abs(horhr90_{Patron} - horhr90_{Instacia}) = abs(12.48 - 17.16) = 4.68$$
$$costo\ DFAV\_ROYA = abs(DFAV\_ROYA_{Patron} - DFAV\_ROYA_{Instancia}) = abs(15 - 28) = 13$$

Ecuación 16. Ejemplo - Costo de edición de las etiquetas de los nodos

El costo de las etiquetas de los nodos (Ecuación 17), se obtiene por medio de la sumatoria de los resultados obtenidos anteriormente.

$$Cen = \sum_{i=0}^{n} efv_i = 1.25 + 5.301641 + 13 + 4.68 = 24.221641$$

Ecuación 17. Ejemplo - Costo etiquetas de los nodos sin normalizar

Los valores $deltaT, T\_hr90\ y\ DFAV\_ROYA$, están dentro del rango, de manera que no es asignado ningún costo. Por el contrario, el valor de $horhr90$ no se encuentra dentro del intervalo establecido en la etiqueta del nodo del patrón evaluado y, por lo tanto, se le asigna un costo de edición de 1 (ecuación 18).

$$Cfr = \sum_{i=0}^{k} or_i = 1$$

Ecuación 18. Ejemplo - Costo de los valores fuera del rango sin normalizar

Luego de realizar la comparación de las etiquetas de los nodos, se procede a comparar las etiquetas de las aristas, donde solo son comparadas aquellas etiquetas de las aristas que conectan los mismos tipos nodos, independientemente de la dirección (Ecuación 19). Por ejemplo, la arista $(Instancia, Parámetro\_Climático)$ del grafo patrón y la arista $(Instancia, Parámetro\_Climático)$ del grafo instancia, tienen la misma etiqueta, en el caso de no ser así, el costo asignado por la sustitución de una etiqueta es 1. Los resultados de la comparación de las etiquetas de las aristas son los siguientes:

$$Costo\ (Instancia, Parámetro\_Climático) = 0$$
$$Costo\ (Humedad\ Relativa, Parámetro\_Climático) = 0$$
$$Costo\ (Temperatura, Parámetro\_Climático) = 0$$

Ecuación 19. Ejemplo - Costo de edición de las etiquetas de las aristas

Posteriormente, se realiza la sumatoria del costo de las operaciones de edición sobre las etiquetas de las aristas (Ecuación 20).

$$Cea = \sum_{i=0}^{k} efe_i = 0$$

Ecuación 20. Ejemplo - Costo de etiquetas de aristas sin normalizar

Adicionalmente, se verifica la dirección de las aristas comparadas anteriormente, en donde se determina que el costo asignado por inversión de arista es (Ecuación 21).

$$Cia = \sum_{i=0}^{k} e_i = 0$$

Ecuación 21. Ejemplo - Costo inversión de aristas sin normalizar

Finalmente, se normalizan todos los valores obtenidos entre [0,1]. La normalización del $Costo\ de\ inserción\ y\ eliminación\ de\ aristas\ y\ nodos\ (Ce)$, está dada por la sumatoria del peso de las operaciones de edición dividida entre la suma de la cantidad de nodos y aristas de los dos grafos (Ecuación 22).

$$Ce' = \frac{Ce}{\lambda(V_p) + \lambda(V_s) + \lambda(E_p) + \lambda(E_s)} = \frac{\sum_{i=0}^{k-1} e_i}{\lambda(V_p) + \lambda(V_s) + \lambda(E_p) + \lambda(E_s)} = \frac{9}{4+8+4+7} = 0.39$$

Ecuación 22. Ejemplo - Costo de inserción y eliminación de aristas y nodos normalizado

El $Costo\ etiquetas\ de\ nodos\ (Cen)$ es normalizado, dividiendo el resultado entre la sumatoria de las diferencias máximas del valor de las etiquetas de los nodos de las instancias y el rango de las etiquetas de los nodos de los patrones (Ecuación 23).

$$Cen' = \frac{\sum_{i=0}^{n} efv_i}{\sum_{i=0}^{m} Maxefv_i} = \frac{24.221641}{54.2015483} = 0.45$$

Ecuación 23. Ejemplo - *Costo de etiquetas de nodos normalizado*

El valor del costo de las etiquetas de los nodos de las instancias que están por fuera del rango (Cfr), se normaliza dividiendo (Cfr) entre la cantidad de etiquetas de nodos comparadas (Ecuación 24).

$$Cfr' = \frac{Cfr}{\lambda(tfvc)} = \frac{\sum_{i=0}^{k} or_i}{\lambda(tfvc)} = \frac{1}{4} = 0.25$$

Ecuación 24. Ejemplo - Costo de valores fuera del rango normalizado

Luego, para normalizar el $Costo\ etiquetas\ de\ aristas\ (Cea)$ y $Costo\ de\ inversión\ de\ aristas\ (Cia)$, se divide su valor entre la cantidad de aristas comparadas (Ecuación 25 y 26).

$$Cea' = \frac{Cea}{\lambda(tfec)} = \frac{\sum_{i=0}^{k} efe_i}{\lambda(tfec)} = \frac{0}{3} = 0$$

Ecuación 25. Ejemplo - Costo de etiquetas de aristas normalizado

$$Cia' = \frac{Cia}{\lambda(tec)} = \frac{\sum_{i=0}^{k} e_i}{\lambda(tec)} = \frac{0}{3} = 0$$

Ecuación 26. Ejemplo - Costo de inversión de aristas normalizado

La similitud es encontrada a través del complemento de la disimilitud, en donde este último se halla multiplicando el valor de cada función de costo por su ponderación y sumando los resultados (Ecuación 27).

$$similitud\ = 1 - (\ Ce' * wCe\ + Cia' * wCia\ + Cea' * wCea\ + Cen' * wCen\ + Cfr' * wCfr)$$

Ecuación 27. Ejemplo - Similitud entre grafos

La ponderación de las funciones de costo es dada a través de los pesos de las mismas, como se indica en las Ecuaciones 28-32.

$$wCe = \frac{nec}{nec + eic + elc + orc + nlc} = \frac{8}{8+1+7+1+6} = 0{,}34$$

Ecuación 28. Ejemplo - Ponderación del costo de eliminación e inserción de aristas y nodos.

$$wCia = \frac{eic}{nec + eic + elc + orc + nlc} = \frac{1}{8+1+7+1+6} = 0{,}043$$

Ecuación 29. Ejemplo - Ponderación del costo de inversión de arista

$$wCea = \frac{elc}{nec\ + eic + elc + orc + nlc} = \frac{1}{8+1+7+1+6} = 0{,}043$$

Ecuación 30. Ponderación del costo de las etiquetas de las aristas

$$wCen = \frac{nlc}{nec\ + eic + elc + orc + nlc} = \frac{6}{8+1+7+1+6} = 0{,}26$$

Ecuación 31. Ponderación del costo de las etiquetas de los nodos

$$wCfr\ = \frac{orc}{nec\ + eic + elc + orc + nlc} = \frac{7}{8+1+7+1+6} = 0{,}30$$

Ecuación 32. Ponderación del costo de los valores fuera del rango

Para finalizar, la similitud de los dos grafos está dada por la siguiente expresión (Ecuación 33).

$$similitud\ = 1 - (\ 0.39 * 0.34\ + 0 * 0{,}043\ + 0{,}45 * 0.26\ + 0.25 * 0.26\ + 0 * 0.032) =\ 0.355$$

Ecuación 33. Similitud Total

De esta manera, la parametrización y normalización sobre las funciones de costo dan como resultado la similitud anterior entre el patrón de roya y la instancia evaluada del ejemplo expuesto anteriormente. Este valor permite conocer la tasa de infección de la enfermedad a la que un cultivo está más propenso en un determinado momento.

Resumen

En este capítulo inicialmente se expone el desarrollo efectuado para normalizar los datos de cada una de las funciones de costo usadas (Eliminación, Inserción, Sustitución de aristas y nodos, inversión de arista, valores fuera del rango y Sustitución de etiquetas de aristas). Después de normalizar los resultados de estas funciones, estas son multiplicadas por una ponderación obtenida a través de los pesos asignados a las funciones de costo. La parametrización de estas funciones permite tener en cuenta cuales son las que presentan una mayor relevancia caracterizando de mejor manera el dominio de aplicación**.** Por último, se presenta un ejemplo con el proceso total realizado por los algoritmos de emparejamiento de grafos tolerante a errores, la parametrización de las funciones de costo y la normalización de los resultados, buscando encontrar la similitud entre un grafo patrón y un grafo instancia.

Capítulo 5

Prototipo y experimentación

En este capítulo se presentan las principales características relacionadas con el desarrollo del prototipo, tales como: la arquitectura, la interfaz y los módulos por los que está compuesto el sistema. Asimismo, se exponen las pruebas realizadas junto con el análisis de los resultados obtenidos.

5.1 Características del prototipo

El prototipo se compone de diversos módulos que en su conjunto permiten obtener la similitud entre dos grafos (patrón e instancia), con el fin de detectar si el cultivo que se está analizando presenta condiciones favorables para la aparición de la roya del café.

5.1.1 Casos de uso

Se realizan los casos de uso del sistema con el fin de describir la respuesta del sistema a un evento que inicia el experto en agronomía

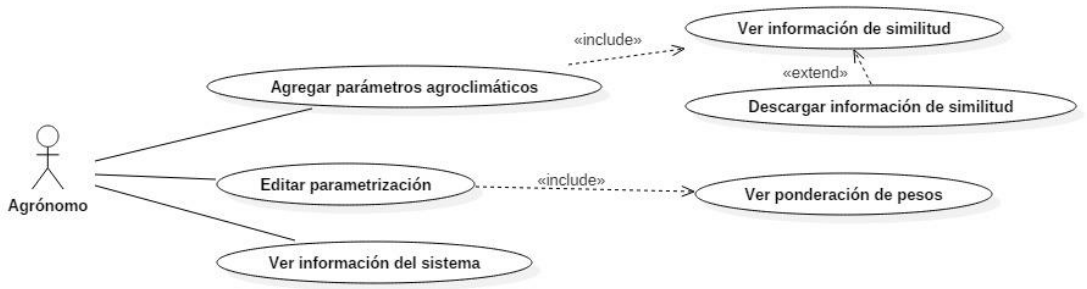


Figura 6. Diagramas de casos de uso del sistema. Fuente: Propia

La interacción de algunos de los componentes del sistema es presentada en 2 diagramas de secuencia (Figura 7 y 8) que ilustran el comportamiento de los dos servicios implementados en el sistema, asimismo se muestra el catálogo de los servicios (Tabla 6 y 7).

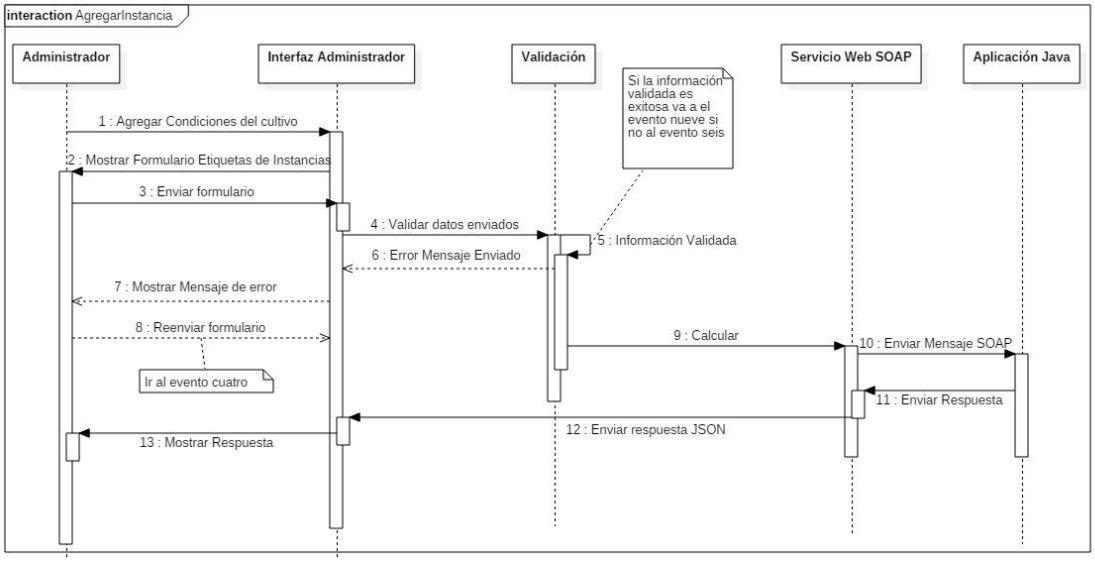


Figura 7. Diagrama de secuencia de Agregar Instancia. Fuente: Propia.

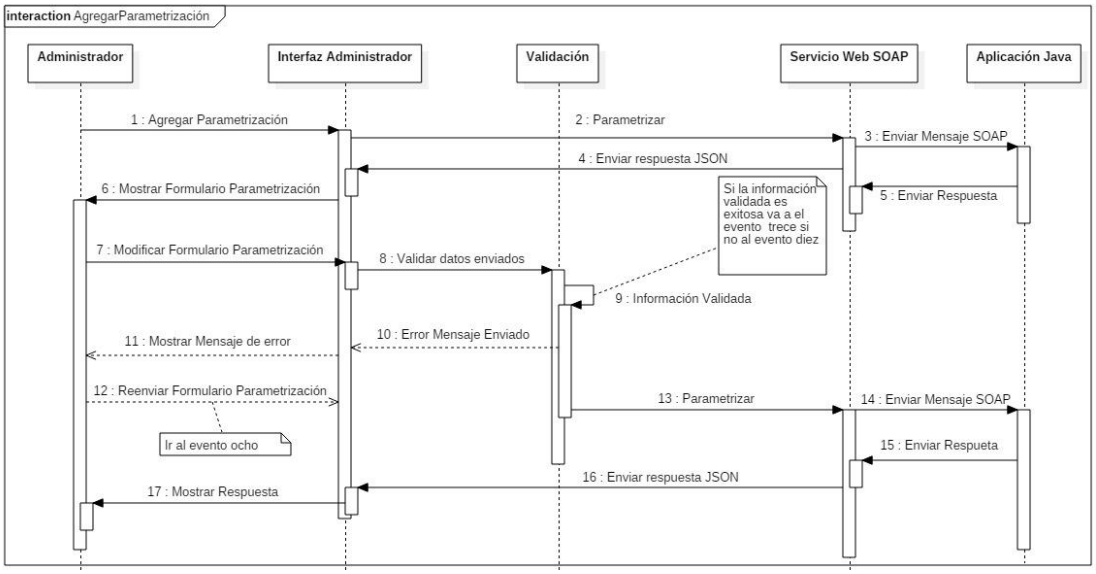


Figura 8. Diagrama de secuencia de Agregar Parametrización. Fuente: Propia.

Servicio Web Calcular: Este servicio permite intercambiar información relacionada con los parámetros agroclimáticos y los datos obtenidos del emparejamiento de grafos.			
	Descripción	**Nombre**	**Tipo de Dato**
	Datos de Identificación	Nombre Instancia	String

Parámetros de Entrada		Mes	String
		Año	String
	Instancia	DFAV_ROYA	String
	Lluvia	Dlluv	String
		Pre_acum	String
	Humedad Relativa	Horhr90	String
		Horhrn90	String
		Hr	String
	Temperatura	T_hr90	String
		T_hrn90	String
		Tmax	String
		Tmed	String
		Tmin	String
		DeltaT	String
	Sombra	Shade	String
	Densidad	Density	String
	Nombre de la instancia.	Instancia	String
	Nombre del patrón.	Patrón	String
	Tasa de infección de la roya asociada a cada patrón.	Tasa de infección	String
	Cantidad de nodos del patrón	Tamaño Patrón	String

Parámetros de Salida	evaluado.		
	Cantidad de nodos de la instancia evaluada.	Tamaño Instancia	String
	Representa cuántas etiquetas nodos de la instancia encajan dentro del rango establecido por cada patrón.	Comparaciones dentro del Rango	String
	Muestra la cantidad de etiquetas de nodos comparadas.	Cantidad de Etiquetas de nodos comparadas	String
	Tiempo de ejecución del algoritmo en procesar la comparación entre la instancia y cada patrón. Este está dado en milisegundos.	Tiempo de ejecución	String
	Porcentaje de similitud obtenido entre la instancia y el patrón.	Similitud(%)	String
	Tasa de Infección de la roya del café	Tasa de infección	String

Tabla 6. Parámetros de entrada y salida del servicio web Calcular. Fuente: Propia

Servicio Web Parametrizar: Servicio Web que permite el intercambio de información de los pesos asociados a cada función de similitud.			
	Descripción	**Nombre**	**Tipo de Dato**
	Parámetro de peso de la eliminación/inserción o sustitución de un nodo	node	String
	Parámetro de peso de la eliminación/inserción o sustitución de una arista	edge	String
	Parámetro de peso de las operaciones sobre las etiquetas de los nodos.	nodelabel	String

Parámetros de Entrada y Salida	Parámetro de peso de las operaciones sobre las etiquetas de las aristas.	edgelabel	String
	Parámetro de peso de la inversión de una arista	edgeinversion	String
	Parámetro de peso de la inversión las etiquetas de los nodos de las instancias que están por fuera del rango de la etiqueta de un nodo de un patrón.	outofrange	String

Tabla 7. Parámetros de entrada y salida del servicio web parametrizar. Fuente: Propia

5.1.2 Arquitectura del sistema

En la Figura 9 es presentada la arquitectura del sistema desde una vista de capas, donde se encuentran los componentes con sus respectivas relaciones.

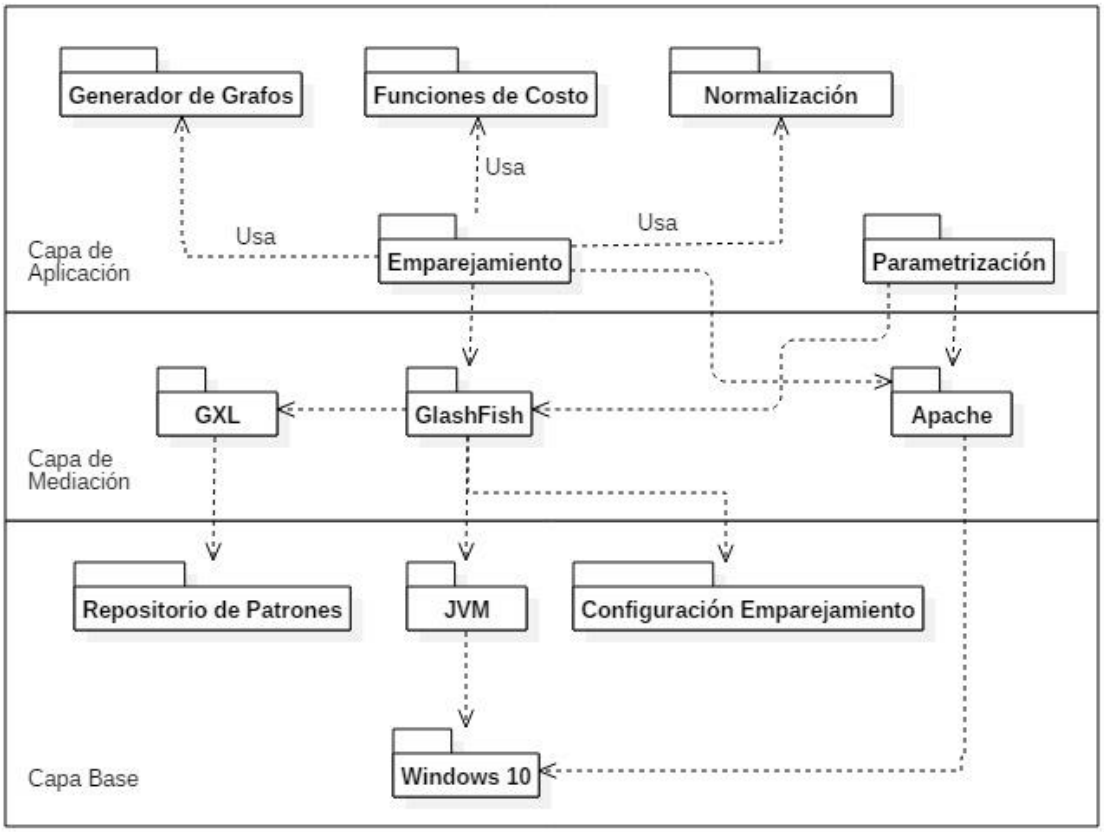


Figura 9. Arquitectura del sistema. Fuente: Propia.

A continuación, se presenta una descripción de los componentes del sistema:

Generador de Grafos: En este componente se generan los grafos instancia a partir de los datos ingresados en el formulario web.

Normalización: Este componente obtiene los resultados de las funciones de costo y los transforma en valores que están en un intervalo de [0,1], con el fin de estandarizar los resultados y poder realizar operaciones entre ellos.

Funciones de Costo: Este módulo contiene las 5 funciones de costo utilizadas para encontrar la similitud entre dos grafos.

- **Función de eliminación e inserción de aristas y nodos:** Contiene las operaciones de edición: eliminación e inserción de aristas y nodos. A estas operaciones se les asigna un peso con el propósito de encontrar el costo de la similitud estructural entre dos grafos, a partir de la cantidad de operaciones de edición efectuadas para transformar un grafo en otro.

- **Función de costo de diferencia de etiquetas de nodos:** Este módulo está diseñado para procesar grafos patrones con rangos numéricos en las etiquetas de los nodos, donde se tiene en cuenta la aproximación entre los límites de los rangos.

- **Función de costo de rango:** Determina si un valor de una instancia evaluado está dentro del intervalo establecido por un patrón.

- **Función de costo de etiquetas de aristas:** En este componente se realiza una comparación entre las etiquetas de las aristas de los patrones e instancias, con el objetivo de encontrar si existe correspondencia entre una y otra. En el caso de no existir, es asignado un costo a la operación.

- **Función de inversión de aristas:** Realiza la inversión de una arista en el caso de que dos aristas, una de–cada grafo (patrón e instancias) conecten el mismo par de nodos, pero tengan direcciones opuestas.

Emparejamiento: En este módulo se encuentran los 4 algoritmos de emparejamiento de grafos adaptados que evalúan la correspondencia entre los nodos y aristas de los grafos patrones e instancias, con el propósito de obtener la similitud.

Parametrización: Permite al usuario modificar los pesos de las funciones de costo.

Repositorio de Patrones: Este repositorio contiene los grafos patrones que representan las tasas de infección de la roya del cafeto.

GXL: Módulo que permite la lectura de los grafos almacenados en formato GXL, la cual es una extensión de XML y se define como un formato de intercambio estándar para grafos.

Glassfish: Servidor de aplicaciones que permite intercambiar datos entre aplicaciones de diferente naturaleza por medio de un servicio web, en este caso, el algoritmo de emparejamiento de grafos desarrollado en java y la interfaz de usuario.

Apache: Servidor web que permite la ejecución y visualización de la interfaz de usuario.

Configuración de emparejamiento: Archivo que almacena los pesos asignados a cada función de costo.

JVM (Java Virtual Machine): Permite la ejecución de un código compilado en java independientemente del sistema operativo que se esté utilizando.

Windows 10: Es el sistema operativo que soporta el prototipo.

5.1.3 Interfaz de usuario

En esta sección, se presenta la interfaz de usuario del prototipo por medio de las Figuras 10 - 12. En este componente el usuario puede ingresar los datos relacionados con las condiciones agroclimáticas del cultivo (ver Sección 3.3) en un determinado instante de tiempo. A partir de los datos ingresados, el sistema genera un grafo (Instancia) que es comparado con los patrones de roya cargados previamente. Además, en la interfaz web para efectos de estudio se brinda la opción de seleccionar un algoritmo de emparejamiento de grafos entre los que se encuentran: A*, Beam, Hungarian y Volgenant-Jonker (Figura 10); finalmente, es mostrado un ranking de los patrones más similares a las condiciones agroclimáticas ingresadas (Figura 11).

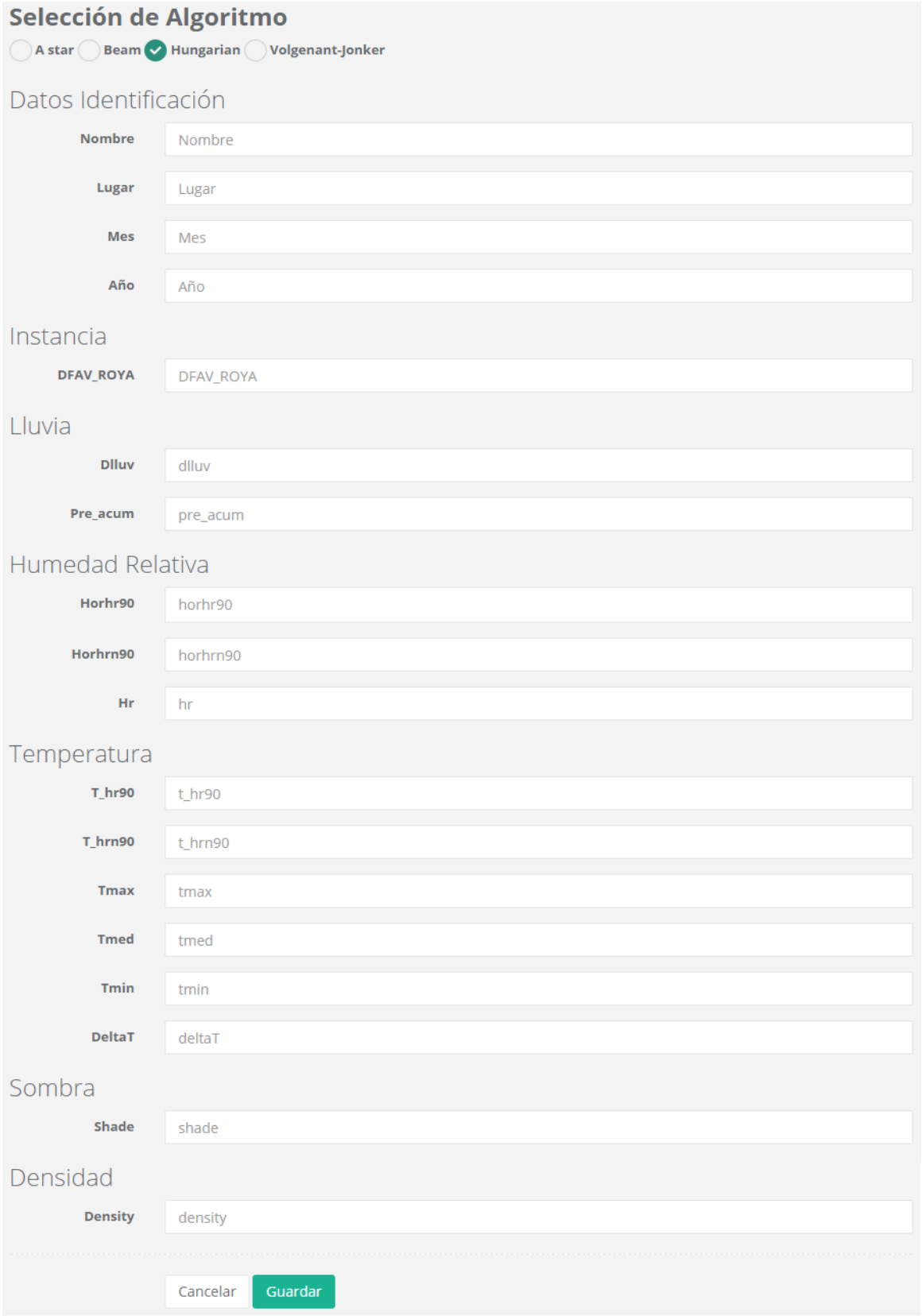
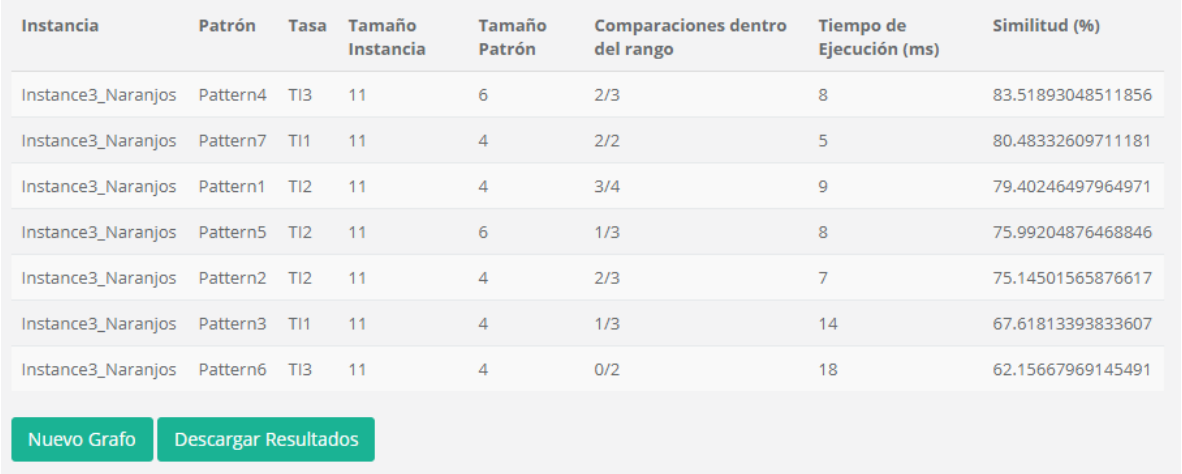


Figura 10.Formulario datos agroclimáticos. Fuente: Propia.

En la Figura 11 se muestra los resultados obtenidos por el algoritmo seleccionado, donde se puede evidenciar los patrones ordenados desde el más similar hasta el menos similar a la instancia evaluada, el porcentaje de similitud y otros parámetros, los cuales permite realizar un mejor análisis de las condiciones del cultivo que se está evaluando.



Instancia	Patrón	Tasa	Tamaño Instancia	Tamaño Patrón	Comparaciones dentro del rango	Tiempo de Ejecución (ms)	Similitud (%)
Instance3_Naranjos	Pattern4	TI3	11	6	2/3	8	83.51893048511856
Instance3_Naranjos	Pattern7	TI1	11	4	2/2	5	80.48332609711181
Instance3_Naranjos	Pattern1	TI2	11	4	3/4	9	79.40246497964971
Instance3_Naranjos	Pattern5	TI2	11	6	1/3	8	75.99204876468846
Instance3_Naranjos	Pattern2	TI2	11	4	2/3	7	75.14501565876617
Instance3_Naranjos	Pattern3	TI1	11	4	1/3	14	67.61813393833607
Instance3_Naranjos	Pattern6	TI3	11	4	0/2	18	62.15667969145491

Figura 11. Resultados del emparejamiento de grafos. Fuente: Propia.

Como ya se ha mencionado en el capítulo anterior, la parametrización permite caracterizar de forma apropiada el prototipo implementado con respecto al dominio de aplicación que se está trabajando. Variar los pesos de las funciones de costo brinda la posibilidad de acercarse a los resultados deseados. Por otro lado, los datos de parametrización de las funciones podrán ser modificados. Cabe resaltar que estas modificaciones deberán ser realizadas por un experto en el tema. A continuación, se presenta el formulario de parametrización (Figura 12).

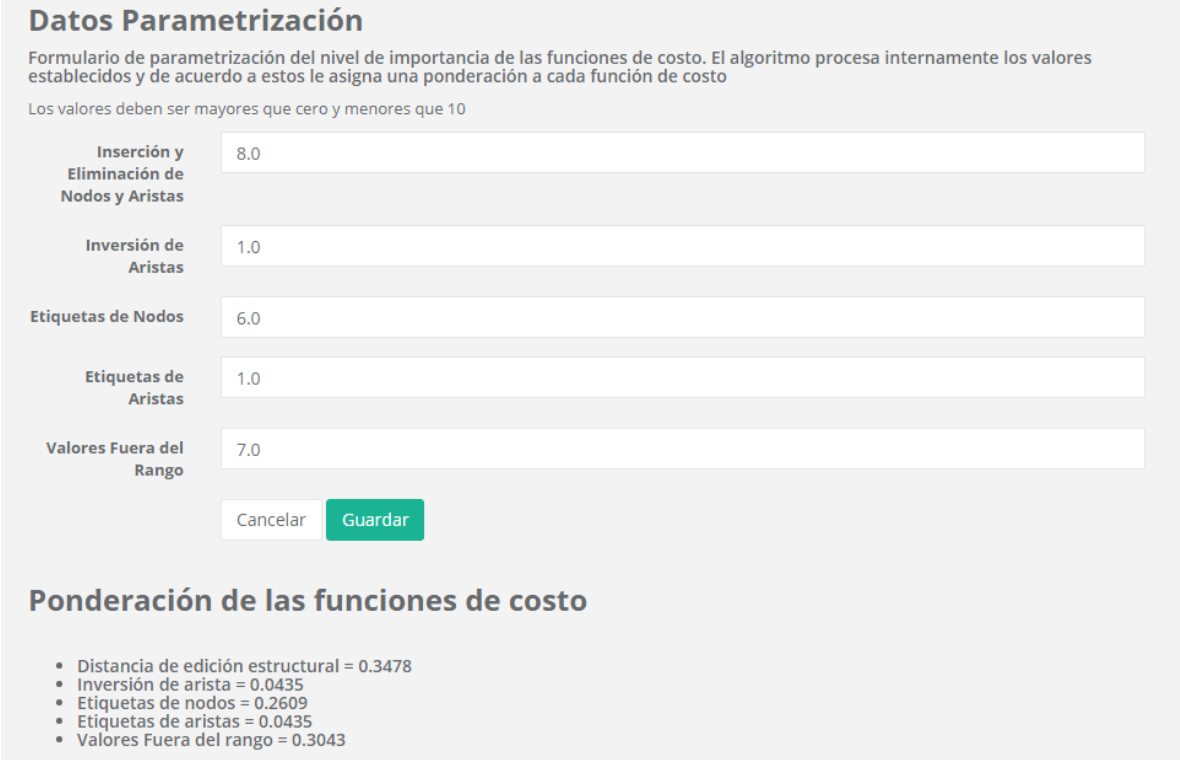


Figura 12. Formulario de parametrización. Fuente: Propia.

5.2 Evaluación experimental

En esta sección se presenta una evaluación experimental de los algoritmos discutidos, con la que se busca encontrar una clasificación de los patrones más similares con una instancia evaluada y así encontrar la tasa de infección de la roya del café en un cultivo. Los resultados obtenidos por el prototipo son comparados con los datos proporcionados por un experto con el propósito de determinar el nivel de precisión de los algoritmos y clasificarlos con base en este último parámetro y el tiempo de ejecución. Es importante mencionar que los patrones e instancias que se utilizaron para evaluar el sistema fueron las indicadas en el capítulo 3 y la parametrización de los pesos de las operaciones de edición se indicaron en el capítulo 4.

5.2.1 Criterios de evaluación

En esta sección, es evaluado empíricamente el prototipo propuesto con base en los siguientes criterios:

5.2.1.1 Parámetros de precisión de los resultados obtenidos

- **Falsos Positivos (FP)**: Patrones clasificados en la clase a la cual no pertenece.

- **Falsos Negativos (FN):** Patrones incorrectamente clasificados en las clases diferentes a la evaluada.

- **Verdaderos Positivos (VP):** Patrones correctamente clasificados.

- **Verdaderos Negativos (VN):** Patrones clasificados correctamente en clases diferentes a la evaluada.

- **Tasa de Verdaderos Positivos (TVP) (Sensibilidad):** Probabilidad de clasificar correctamente un patrón.

$$TVP = \frac{VP}{VP + FN}$$

Ecuación 34. Cálculo de tasa de verdaderos positivos

- **Tasa de Falsos Positivos (TFP) (Especificidad)**: Probabilidad de clasificar incorrectamente un patrón.

$$TFP = \frac{FP}{VN + FP}$$

Ecuación 35. Cálculo de tasa de Falsos positivos

- **Precisión (Valor Predictivo Positivo):** Probabilidad de ocurrencia de la tasa de infección si el patrón fue correctamente clasificado [67].

$$VPP = \frac{VP}{VP+FP}$$

Ecuación 36. Valor Predictivo Positivo

- **Exactitud (Índice Rand):** Es una medida relacionada con la precisión que mide la similitud entre el conjunto de resultados proporcionados por el sistema experto y el conjunto de resultados dados por los algoritmos de emparejamiento.

$$IR = \frac{VP + VN}{VP + FP + VN + FN}$$

Ecuación 37. Cálculo del índice Rand

- **Exhaustividad (Tasa de Verdaderos Positivos):** esta medida, conocida en inglés como “recall”, es calculada de la misma forma que la tasa de verdaderos positivos (ecuación 34) [67].

- **Medida F:** Es la relación que existe entre la precisión y la exhaustividad. El factor β define el nivel de importancia que se le da estas dos medidas, si $\beta = 1$ los dos tienen la misma importancia, si $\beta > 1$ es más importante la exhaustividad y viceversa [67].

$$F_\beta = (1+\beta^2) * \frac{VPP*TVP}{(\beta^2*VPP)+TVP} = \frac{(1+\beta^2)*VP}{((1+\beta^2)*VP)+(\beta^2*FN)+(FP)}$$

$$F_1 = 2\frac{VPP*TVP}{VPP+TVP} = \frac{2VP}{2VP+FP+FN}$$

Ecuación 38. Cálculo de la medida F

- **Coeficiente de Correlación de Matthews:** Mide la calidad de la clasificación de los patrones según la instancia evaluada, este retorna un valor entre -1 y 1. en donde 1 representa una predicción perfecta, 0 indica que el resultado es similar a una predicción aleatoria, y -1 que existe una completa discrepancia entre la predicción y el valor real [68].

$$CCM = \frac{(VP*VN)-(FP*FN)}{\sqrt{(VP+FP)(VP+FN)(VN+FP)(VN+FN)}}$$

Ecuación 39. Coeficiente de Correlación de Matthews.

5.2.1.2 Tiempo de ejecución

Para evaluar el tiempo de ejecución de los algoritmos se generaron dos conjuntos de grafos análogos a los patrones e instancias, con grafos de diferentes tamaños y etiquetas de valores aleatorios en un rango establecido para determinar la relación entre la cantidad de nodos y aristas, con el tiempo de ejecución.

5.2.1.3 Consumo de recursos computacionales

Las pruebas se efectuaron en un equipo con 8192 MB de RAM, un procesador Intel Core i5-4210U @ 2.4Ghz (4CPUs) y un sistema operativo Windows 10 de 64 bits, usando los grafos sintéticos generados se determinó el consumo promedio de CPU y Memoria RAM.

5.3 Planificación

En la Tabla 8, se presenta el plan de pruebas llevado a cabo para determinar la precisión y rendimiento del prototipo. Cabe aclarar que estas pruebas son realizadas sobre el sistema completo.

Ref.	Módulo	Prueba
P01	Algoritmos de emparejamiento de Grafos	Precisión, exactitud, exhaustividad y calidad de los resultados obtenidos por los algoritmos.
P02	Algoritmos de emparejamiento de Grafos	Tiempo de ejecución con instancias variables y grafos conexos.
P03	Algoritmos de emparejamiento de Grafos	Tiempo de ejecución con instancias variables y grafos doblemente conexos.
P04	Algoritmos de emparejamiento de Grafos	Consumo de recursos computacionales entre instancias variables y grafos conexos.
P05	Algoritmos de emparejamiento de Grafos	Consumo de recursos computacionales entre instancias variables y grafos doblemente conexos.
P06	Servicio Web e Interfaz Web	Envío y recepción de datos

Tabla 8. Plan de pruebas. Fuente: Propia

5.4 Resultados

La evaluación de los parámetros relacionados con la precisión se efectúa por medio del resultado obtenido de los algoritmos a partir de un conjunto de grafos que representan datos agroclimáticos reales, y los resultados proporcionados por un experto, los cuales se componen de la clasificación de los 3 patrones más similares a una instancia evaluada. Esta clasificación fue realizada para 98 instancias y 7 patrones que representan datos agroclimáticos reales. Estos

resultados son tomados como punto de referencia para la evaluación experimental del prototipo.

Para determinar el consumo de recursos computacionales y el tiempo de ejecución promedio, se creó un conjunto de grafos sintéticos dirigidos doblemente conexos y grafos dirigidos conexos análogos a los grafos reales. Se realizaron las pruebas en primer lugar con patrones variables e instancias constantes y en segundo lugar con instancias variables y patrones constantes. El objetivo fue analizar el comportamiento de los algoritmos frente a los grafos establecidos.

Por otra parte, no es considerado el algoritmo A* dentro de las gráficas destinadas para el análisis de consumo de recursos computacionales y tiempo de ejecución, debido a que este no presenta un amplio rango en la cantidad de nodos que pueda emparejar antes de que se desborde la memoria, lo que no lo hace comparable con los demás algoritmos; según los resultados, el tamaño máximo de los grafos que puede emparejar es de 9 nodos en las instancias y 9 nodos en los patrones.

Para las pruebas realizadas se establece un valor para la profundidad del árbol del algoritmo Beam de n=443192. Este parámetro se obtuvo buscando que la diferencia entre los resultados de similitud del algoritmo Beam con el Algoritmo A* sea mínima, debido a que este último según la literatura presenta una mayor precisión [23], [43], [58], por ser un algoritmo de emparejamiento de grafos óptimo.

De acuerdo con los criterios de evaluación seleccionados anteriormente, a continuación, se presentan los resultados obtenidos.

5.4.1 P01 - Precisión de los resultados obtenidos

El experto genera una clasificación de los tres primeros patrones más similares a la instancia evaluada, que son ubicados en tres clases dependiendo de su similitud, siendo la clase 1 la que contiene los patrones más similares. A partir de la comparación de los resultados obtenidos por el experto y los algoritmos, se establece la precisión y exactitud mediante los siguientes parámetros: Tasa de verdaderos positivos (TVP), Tasa de falsos positivos (TFP), Valor predictivo positivo (VPP), Índice Rand (IR), Medida F (F1) y Coeficiente de Correlación de Matthew (CC) (Tabla 7).

Algoritmo	Clase	FP	FN	VP	VN	TVP	TFP
A*	1	4	2	93	78	0,9789	0.0487
	2	57	21	39	131	0,65	0,3031
	3	62	30	31	135	0,5081	0,3147
Beam	1	4	2	92	87	0,9787	0,0439
	2	45	29	47	133	0,6184	0,2528
	3	58	27	34	143	0,5573	0,2885
Hungarian	1	4	2	93	69	0,9789	0,0547
	2	67	21	30	132	0,5882	0,3366
	3	61	40	33	124	0,4520	0,3297
VJ	1	4	2	93	67	0,9789	0,0563
	2	65	25	34	126	0,5762	0,3403
	3	67	29	30	128	0,5084	0,3435

Tabla 9. Medidas de evaluación. Fuente: Propia

La Tabla 10 muestra los parámetros evaluados para obtener la precisión, exactitud y la calidad de los resultados obtenidos por cada uno de los algoritmos.

Algoritmo	Clase	Precisión	Exactitud	Exhaustividad	Medida F	Coeficiente de Correlación:
A*	1	0,9587	0,9661	0,9789	0,9687	0,9319
	2	0,4062	0,6854	0,65	0,5	0,3049
	3	0,3333	0,6434	0,5082	0,4025	0,1712
Beam	1	0,9583	0,9675	0,9787	0,9684	0,9353
	2	0,5108	0,7086	0,6184	0,5595	0,3483
	3	0,3695	0,6755	0,5574	0,4444	0,2380
Hungarian	1	0,9587	0,9642	0,9789	0,9687	0,9273
	2	0,3092	0,648	0,5882	0,4054	0,2080
	3	0,3510	0,6085	0,4520	0,3952	0,1144
VJ	1	0,9588	0,9638	0,9789	0,9687	0,9262
	2	0,3434	0,64	0,5763	0,4304	0,2049
	3	0,3093	0,6221	0,5084	0,3846	0,1433

Tabla 10. Criterios de evaluación de la precisión. Fuente: Propia

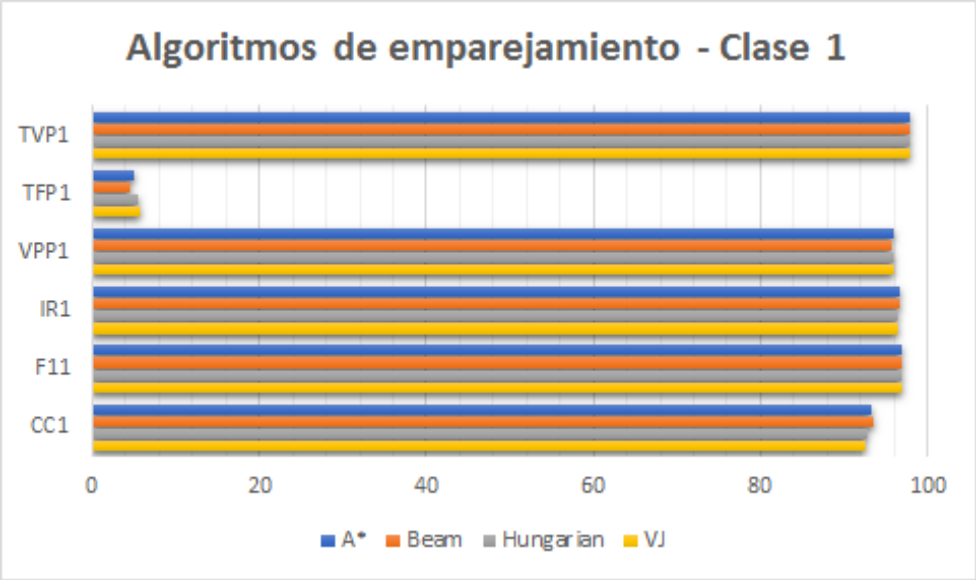


Figura 13. Algoritmos de emparejamiento de grafos - precisión y exactitud clase 1. Fuente: Propia.

En la Figura 13 se presenta la clase 1, la cual es la más representativa en la predicción de la tasa de infección de la roya del cafeto. La diferencia de los resultados de los 4 algoritmos es mínima, sin embargo, en la tasa de falsos positivos y en la calidad de los resultados se destaca el algoritmo Beam.

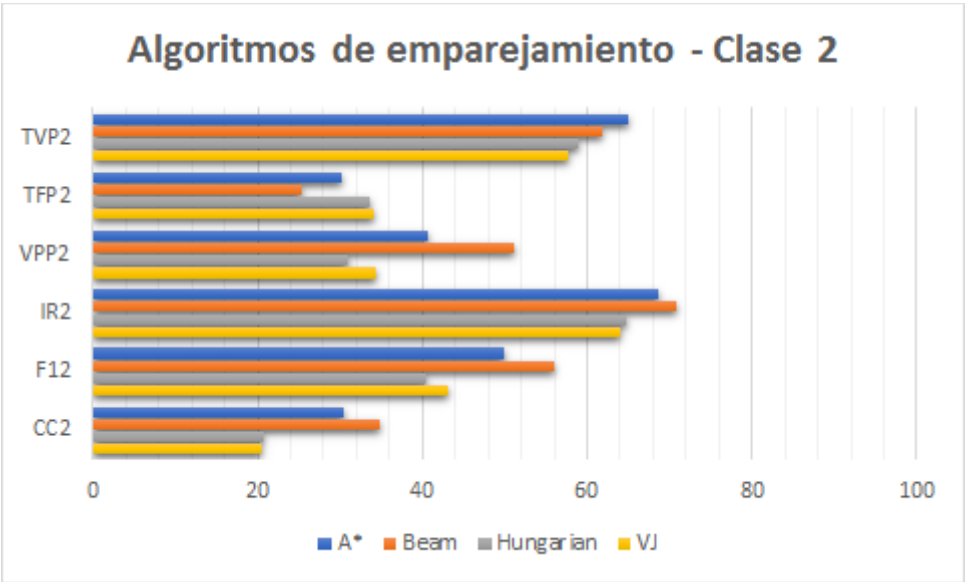


Figura 14. Algoritmos de Emparejamiento de grafos - precisión y exactitud clase 2. Fuente: Propia.

En Clase 2 (Figura 14) existe una mayor variación en los resultados, aun así el Algoritmo Beam es considerablemente más preciso, exacto y presenta una mejor calidad en los resultados, seguido por el Algoritmo A* y los dos algoritmos de emparejamiento bipartito.

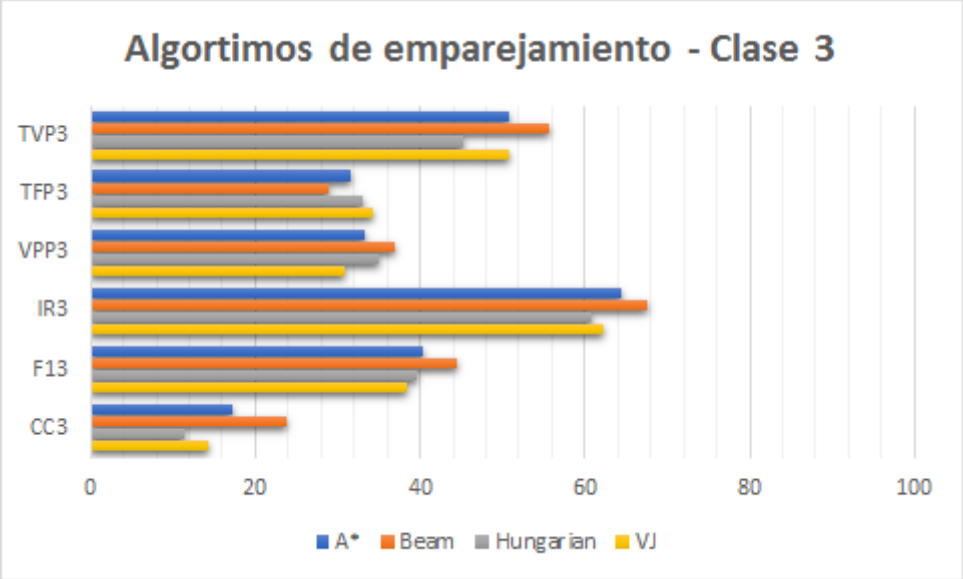


Figura 15. Algoritmos de Emparejamiento de grafos - precisión y exactitud clase 3. Fuente: Propia.

La clase 3 es la menos representativa en cuanto a la predicción de la tasa de infección, aun así, es considerada como un parámetro para establecer la precisión de los algoritmos. En la Figura 15 se puede observar que los resultados obtenidos fueron similares a los de la clasificación de la clase 2.

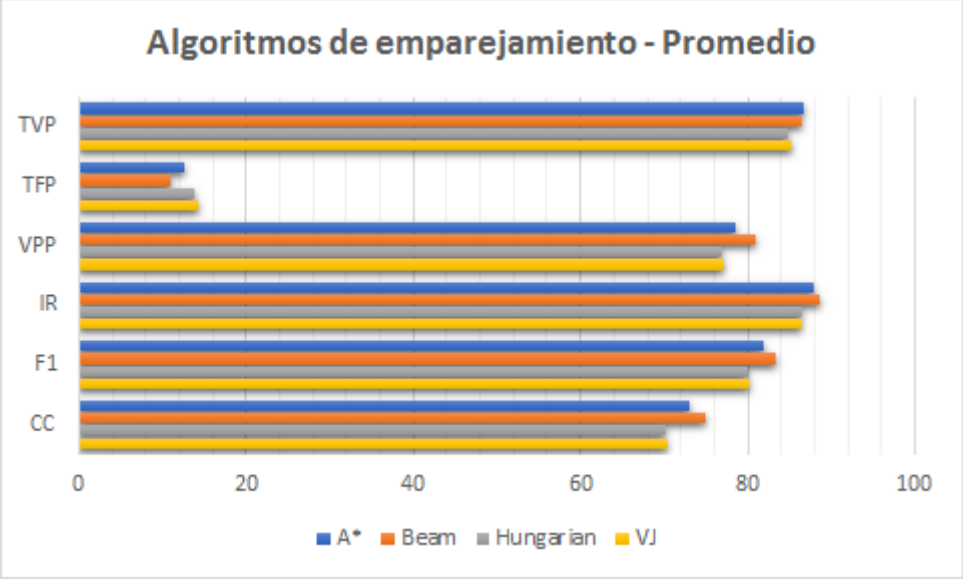


Figura 16. Algoritmos de Emparejamiento de grafos - promedio de precisión y exactitud. Fuente: Propia.

Para obtener el promedio de los parámetros anteriores, se le asignó una ponderación de 70%, 20% y 10% a la clase 1, 2 y 3 respectivamente. Esta ponderación fue asignada con el fin de establecer el nivel de importancia de cada una de las clases, en donde la clase 1 permite identificar la tasa de infección con mayor probabilidad de aparición en el mes analizado y las clases 2 y 3 permiten determinar la tendencia que tiene el cultivo a otra tasa de infección. En la Figura 16 se observa que en la tasa de verdaderos positivos (TVP) que representa la probabilidad de clasificar correctamente un patrón, el Algoritmo A* presenta ligeramente un mejor resultado que los demás algoritmos. En la tasa de falsos positivos (TFP), que indica la probabilidad de clasificar incorrectamente un patrón y el valor predictivo positivo (VPP) que es uno de los parámetros más importantes ya que este da la probabilidad de ocurrencia de la tasa de infección, el Algoritmo Beam es considerablemente mejor que los demás. Por otro lado, en el índice rand (IR) se puede observar que el Algoritmo Beam proporciona una similitud más alta entre los resultados dados por el experto y los resultados obtenidos por los algoritmos de emparejamiento de grafos, seguido por los algoritmos A*, Volgenant-Jonker y Hungarian. Finalmente, los últimos parámetros Medida F (F1) y Coeficiente de Correlación de Matthew (CC), indican que el Algoritmo Beam tiene una mejor relación entre la precisión y exhaustividad, por lo tanto, brinda una mejor calidad en sus resultados.

5.4.2 P02, P03 - Tiempo de ejecución del emparejamiento

Inicialmente se realiza la prueba de tiempo de ejecución variando los nodos de las instancias en grafos dirigidos doblemente conexos (Figura 17) y grafos dirigidos conexos (Figura 18). Estos dos grupos de grafos fueron usados con el fin de determinar qué tan fuerte es la relación entre la cantidad de aristas y el tiempo de ejecución de cada algoritmo.

Figura 17. Grafos dirigidos doblemente conexos-Instancias Variables. Fuente: Propia.

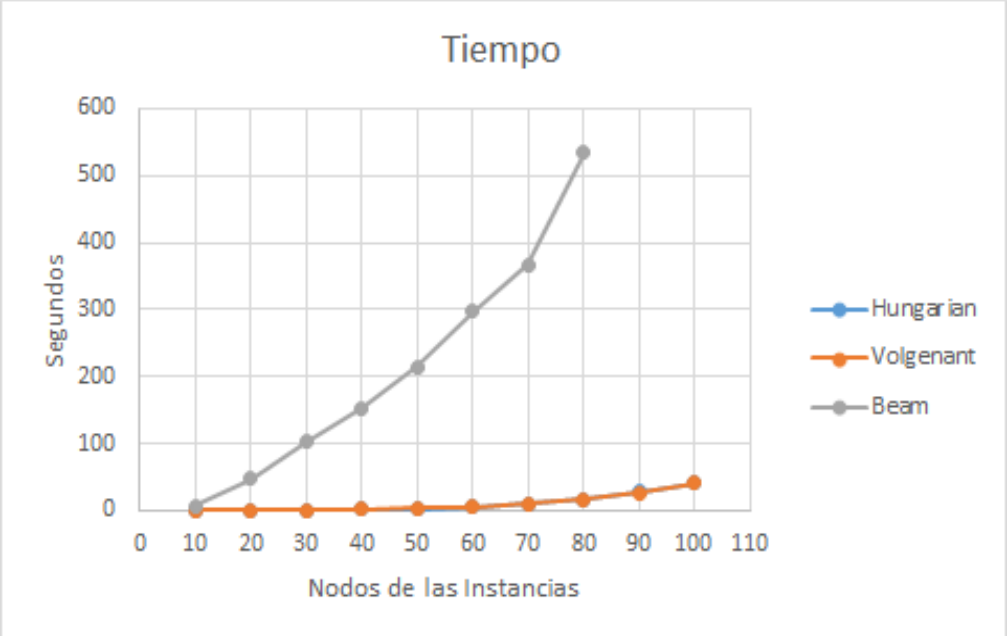


Figura 18. Grafos dirigidos conexos-Instancias Variables. Fuente: Propia.

Finalmente, en la Figura 17 y 18 se puede observar el comportamiento de los algoritmos frente a la variación de los nodos y las aristas. Estas gráficas indican que el algoritmo Beam tiene un tiempo de ejecución mayor comparado con los algoritmos Hungarian y Volgenant-Jonker, donde estos últimos presentan un comportamiento similar. Por otro lado, se infiere que el tiempo de ejecución de todos los algoritmos en los grafos dirigidos conexos respecto a los grafos dirigidos doblemente conexos disminuye debido a la reducción de las aristas.

Asimismo, la cantidad de aristas afectan significativamente los resultados, especialmente en los algoritmos bipartitos.

5.4.3 P04, P05 - Consumo de Recursos Computacionales

Para evaluar el consumo de recursos computaciones de cada uno de los algoritmos elegidos (Beam, Hungarian y Volgenant-Jonker), se realiza una comparación del consumo promedio de RAM y porcentaje del uso de CPU, con las variaciones de los grafos dirigidos doblemente conexos (Figura 19 y 20) y grafos dirigidos conexos (Figura 21 y 22).

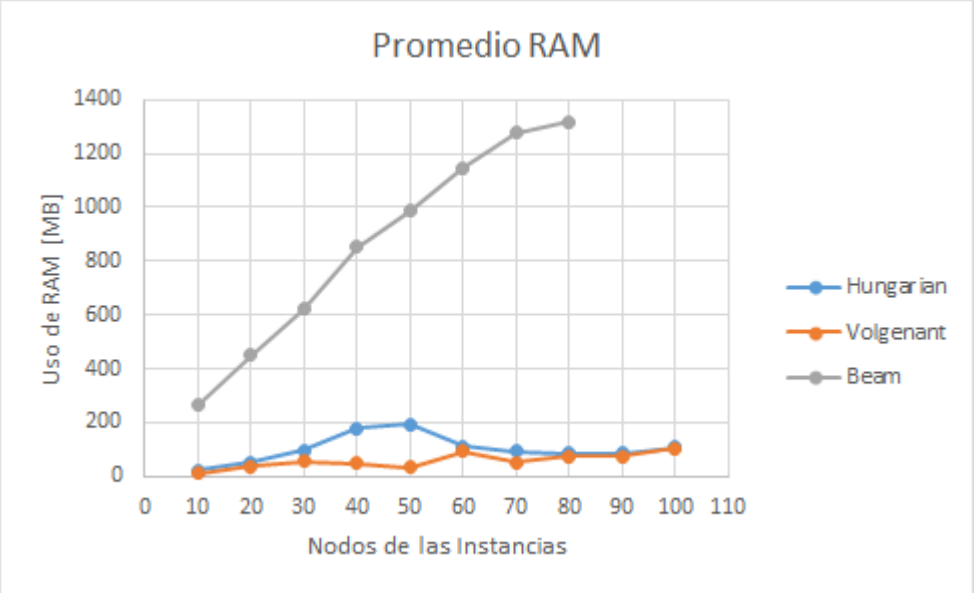


Figura 19. Grafos dirigidos doblemente conexos-Instancias Variables. Fuente: Propia.

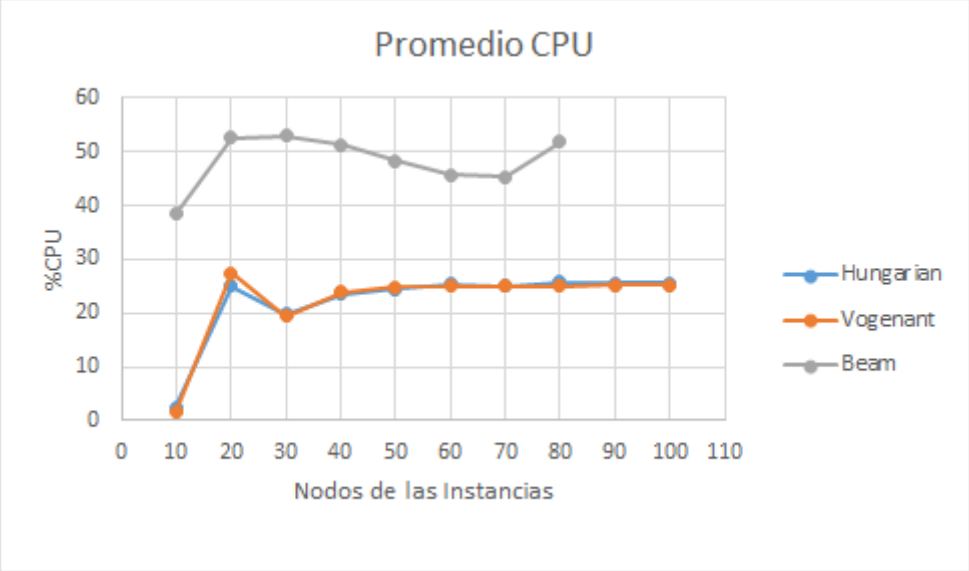


Figura 20. Grafos dirigidos doblemente conexos-Instancias Variables. Fuente: Propia.

Según los resultados obtenidos, el promedio de uso de CPU del algoritmo Beam se encuentra entre el 40% y 50%**.** Es importante resaltar que la memoria es incremental dependiendo del número de nodos. Por otro lado, en los algoritmos Hungarian y Volgenant-Jonker el uso de CPU y de memoria se mantiene estable a pesar de que se incremente el número de nodos, estando alrededor del 25% y 80MB respectivamente. De lo anterior se infiere que el algoritmo Beam es computacionalmente más exigente respecto a los otros algoritmos estudiados.

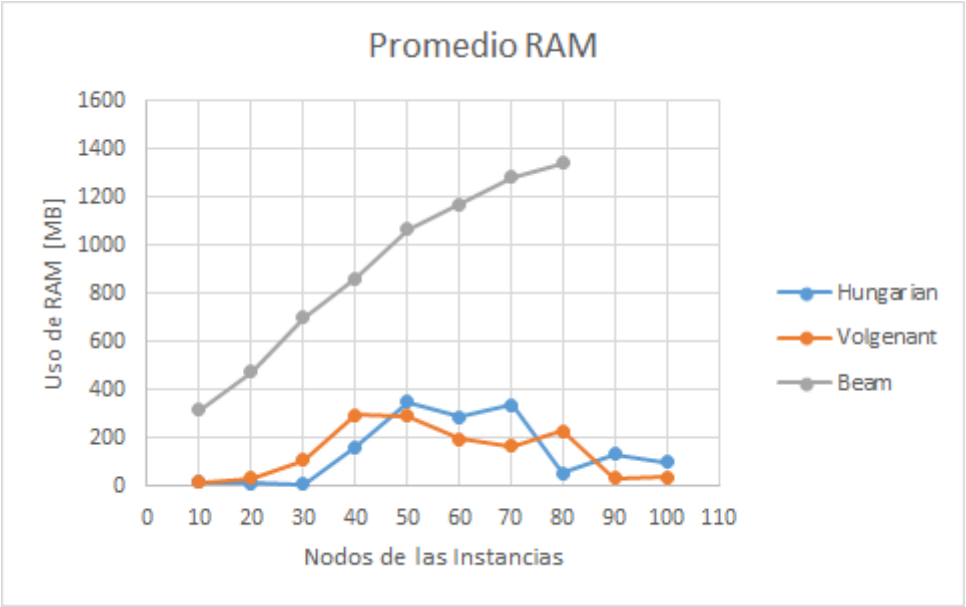


Figura 21. Grafos dirigidos conexos-Instancias Variables. Fuente: Propia.

Los resultados obtenidos en la Figura 21 muestran que memoria RAM presentan un comportamiento similar respecto a las pruebas ejecutadas con los grafos dirigidos doblemente conexos.

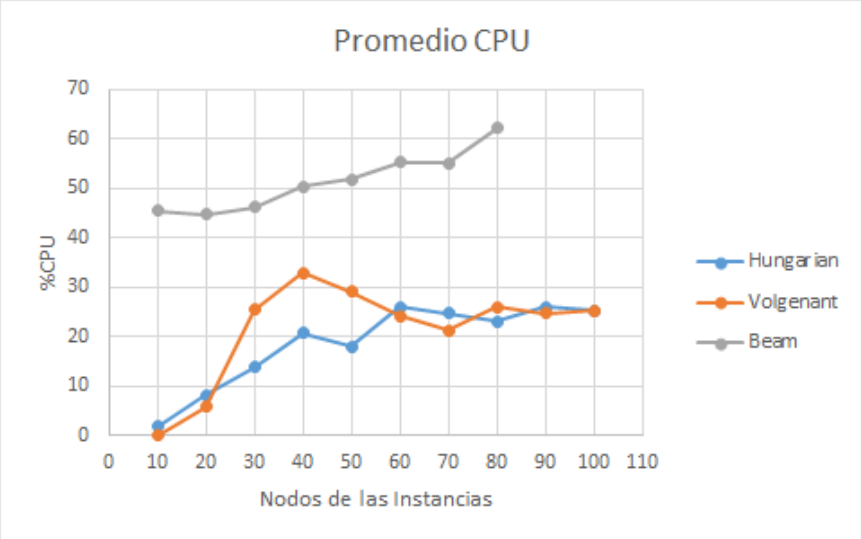


Figura 22. Grafos dirigidos conexos-Instancias Variables. Fuente: Propia.

En los grafos dirigidos conexos con instancias variables el promedio de CPU en el algoritmo Beam está entre el 45 % y 60 %, mientras que el comportamiento de los algoritmos Hugarian y Volgenat-Jonker se mantiene similar a pesar de la variación de los grafos (Figura 22).

Los algoritmos Hungarian y Volgenant-Joker presentan una mayor estabilidad respecto al consumo de recursos computacionales, debido a que se observa que a pesar de que se incrementen las aristas y los nodos, el uso de memoria RAM y CPU se mantiene en un rango definido. Por otro lado, en el algoritmo Beam, el promedio de memoria RAM se ve bastante afectado por la cantidad de nodos de los grafos. Sin embargo, el consumo de CPU se mantiene estable. Además, se observa que el consumo de recursos computacionales no es afectado si se comparan grafos atribuidos conexos o grafos atribuidos doblemente conexos.

Con las pruebas efectuadas a los algoritmos adaptados, se obtiene la siguiente clasificación de los algoritmos con base en los criterios de evaluación (Tabla 11).

Parámetro	Precisión	Tiempo de Ejecución	Recursos computacionales
Ranking Algoritmos	Beam	Volgenat-Jonker	Volgenat-Jonker
	A*	Hungarian	Hungarian
	Volgenant-Jonker	Beam	Beam
	Hungarian	A*	A*

Tabla 11. Clasificación de los algoritmos. Fuente: Propia

De acuerdo con la naturaleza del dominio de aplicación, el algoritmo más adecuado para realizar la predicción de la tasa de infección de la roya del cafeto es el Algoritmo Beam, debido a que es el algoritmo con mayor precisión y presenta un tiempo de ejecución y consumo de recursos computacionales aceptables para la aplicación.

5.4.4 P06 - Envio y recepción de datos

Las pruebas sobre el funcionamiento del Servicio Web y la Interfaz de Usuario se realizó de manera manual mientras se ejecutaba la prueba P01. De igual manera se realizaron pruebas con varias instancias de la interfaz de usuario y solicitudes simultáneas al servicio web, las cuales dieron como resultado un correcto funcionamiento del mismo.

Resumen

En este capítulo se expone cada uno de los módulos por los que está integrado el prototipo implementado. Asimismo, se presentan los factores más relevantes de su funcionamiento y la interacción de cada uno de ellos. Además, se presentan los criterios de evaluación de los resultados de los algoritmos, donde se busca determinar la precisión, exactitud, calidad de los resultados, tiempo de ejecución y consumo de recursos computacionales. Estas pruebas fueron realizadas haciendo uso de grafos reales y grafos sintéticos análogos a los reales.

Capítulo 6

Conclusiones y trabajos futuros

En este capítulo se exponen cada una de las conclusiones obtenidas durante el desarrollo del proyecto. Asimismo, se presentan los trabajos futuros para enriquecer aún más la propuesta planteada en esta investigación.

6.1 Conclusiones

- La pregunta de investigación: ¿Cómo estimar la probabilidad de ocurrencia de la roya en el café a partir de las condiciones del cultivo?, es resuelta calculando la similitud entre los grafos patrones y una instancia determinada, por medio de los algoritmos de emparejamiento de grafos adaptados en esta investigación. Los grafos patrones representan las tasas de infección de la roya del café por medio de variables agroclimáticas y sus relaciones, y las instancias contiene la información agroclimática de un cultivo en un intervalo de tiempo. La similitud permite encontrar la probabilidad de ocurrencia de una tasa de infección con base en una instancia y su cercanía a un patrón.

- La adaptación propuesta permite acercar el emparejamiento de grafos tolerante a errores a los dominios de aplicación, que requieran el reconocimiento de patrones y que proporcionen grafos patrones dirigidos con rangos numéricos en las etiquetas de los nodos y grafos instancia dirigidos con etiquetas numéricas en los nodos, debido a que el emparejamiento de grafos tolerante a errores considera un grado de proximidad en la tarea de búsqueda de patrones de grafos. Asimismo, proporciona una mayor flexibilidad en las coincidencias correspondientes a los intervalos definidos por los patrones utilizados.

- Los resultados de esta adaptación muestran que: el Algoritmo Beam presenta un mejor desempeño en relación con el problema de la roya del café, ya que mantiene una buena relación entre la precisión, el tiempo de ejecución y el consumo de recursos computacionales; haciéndolo

adecuado para dominios de aplicación donde se requiera una alta precisión.

- El Algoritmo A* presenta un crecimiento exponencial en el tiempo de ejecución y en el consumo de recursos computacionales, debido al aumento del número de nodos. Como se observó en los resultados, la precisión de los 4 algoritmos es muy similar, por lo que se concluye que el algoritmo A* no es adecuado para el emparejamiento de patrones, en donde el producto de la cardinalidad de los nodos de los grafos emparejados sea mayor que 81.

- El número de aristas del grafo tiene una influencia mayor en el tiempo de ejecución de los algoritmos que están basados en la distancia de edición bipartita, respecto a los algoritmos de búsqueda en árbol.

- La normalización propuesta en este trabajo permite caracterizar mejor la medida de similitud entre dos grafos, ya que considera el dinamismo que se presenta en los valores de las etiquetas de los nodos.

- Los pesos asignados a las funciones de costo y la ponderación permiten caracterizar de mejor manera los resultados y adaptar los algoritmos a diferentes dominios de aplicación.

- La comparación realizada entre los resultados obtenidos por los algoritmos de emparejamiento de grafos y los resultados proporcionados por el experto, permite comprobar el comportamiento de los algoritmos directamente en el dominio de aplicación y ajustar los parámetros para el óptimo funcionamiento del prototipo.

6.2 Trabajos futuros

- Explorar el comportamiento de la adaptación propuesta en otros tipos de cultivos, para verificar si el emparejamiento de grafos puede ser utilizado como una posible solución general en la predicción de las tasas de infección de plagas y enfermedades en el sector agrícola.

- Realizar pruebas sobre diferentes normalizaciones, con el fin de buscar mecanismos que brinden mejores resultados en la tarea de búsqueda de patrones.

- Incluir en el prototipo la persistencia de los resultados, obtenidos por los algoritmos en una base de datos, con la intención de mejorar el análisis de los resultados históricos.

- Implementar un módulo de normalización, que permita seleccionar el lugar desde donde provienen los datos, con el propósito de generar una normalización específica para cada área de cultivo y así, permitir que los resultados se adapten mejor a las características agroclimáticas de cada lugar.

- Agregar un módulo que permita crear y/o modificar los grafos patrones desde una interfaz para ampliar la adaptación propuesta en esta investigación a otro tipo de plagas, enfermedades y cultivos de una manera más simple.

- Verificar si los algoritmos de emparejamiento de grafos tolerante a errores adaptados pueden ser utilizados como una herramienta para identificar la cercanía entre una instancia determinada y patrones de diversas enfermedades con el fin estimar la probabilidad de ocurrencia de una enfermedad u otra.

7. Bibliografía

[1] J. A. P. Toro, «“Apreciaciones sobre el ciclo cafetero en Colombia durante el siglo xx», *Econ. Cafe. Desarro. Económico En Colomb.*, Universidad de Bogotá Jorge Tadeo Lozano, Facultad de Ciencias Sociales, Programa de Relaciones Internacionales, pp. 31-33, 2013.

[2] D. C. Corrales, J. C. Corrales, y A. Figueroa-Casas, «Towards detecting crop diseases and pest by supervised learning», *Ing. Univ.*, vol. 19, nº 1, pp. 207–228, 2015.

[3] C. A. Rivillas, C. A. Serna, M. A. Cristancho, y A. L. Gaitan, «La roya del cafeto en Colombia: Impacto manejo y costos del control», *Cenicafé*, vol. 1, 2011.

[4] C. Morales, «Impacto Socioeconómico y productivo de la Roya del Café en los países de la Región. Situación actual y perspectivas», presentado en Memorias Del Seminario Científico Internacional Manejo Agroecológico De La Roya Del Café, Ciudad de Panamá, 2015, p. 6.

[5] J. A. Hruska, «Manejo Agroecológico de la roya del café», presentado en Memorias Del Seminario Científico Internacional Manejo Agroecológico De La Roya Del Café, Ciudad de Panamá, 2015, p. 1.

[6] N. Arrieta, «Resistencia genética en café: Estrategia de manejo no químico de la Roya del Cafeto», presentado en Memorias Del Seminario Científico Internacional Manejo Agroecológico De La Roya Del Café, Ciudad de Panamá, 2015, pp. 68-69.

[7] C. Aristizabal y H. Duque, «Análisis económico del efecto de la roya en la variedad caturra y progenies con resistencia incompleta», *Cenicafé*, vol. 58, nº 3, pp. 167-184, 2007.

[8] A. Temis, A. López, y M. Sosa, «Producción de café (coffea arabica L.): cultivo, beneficios, plagas y enfermedades», *Temas Sel. Ing. Aliment.*, vol. 5, nº 2, pp. 54–74, 2011.

[9] npic, «Pesticidas y salud humana,» [online] Universidad Estatal de Oregón y la Agencia de Protección Ambiental de los Estados Unidos, 2015 Disponible en: http://npic.orst.edu/health/humhealth.es.html

[10] A. C. Rivillas, «Acciones emprendidas por Colombia en el manejo de la Roya del Cafeto», presentado en Memorias Del Seminario Científico Internacional Manejo Agroecológico De La Roya Del Café, Ciudad de Panamá, 2015, pp. 11-16.

[11] O. Luaces, L. H. A. Rodrigues, C. A. A. Meira, y A. Bahamonde, «Using nondeterministic learners to alert on coffee rust disease», *Expert Syst. Appl.*, vol. 38, nº 11, pp. 14276–14283, 2011.

[12] A. Hruska, «Apoyo de FAO en el manejo de enfermedades en cultivos de Centroamérica», presentado en Memorias Del Seminario Científico Internacional Manejo Agroecológico De La Roya Del Café, Ciudad de Panamá, 2015, p. pp 3-5.
[13] D. C. Corrales *et al.*, «Plataforma para el seguimiento de variables meteorológicas y ambientales para el sector agropecuario», en *VII Congreso Ibérico de AgroIngeniería y Ciencias Hortícolas, Madrid*, 2013.
[14] E. Lasso, T. T. Thamada, C. A. A. Meira, y J. C. Corrales, «Graph Patterns as Representation of Rules Extracted from Decision Trees for Coffee Rust Detection», en *Research Conference on Metadata and Semantics Research*, 2015, pp. 405–414.
[15] S. Georgiou, P. Imbach, F. Anzueto, G. del Carmen Calderón, y J. Avelino, «Weather and climate indicators for coffee rust disease», en *AGU Fall Meeting Abstracts*, 2014.
[16] D. C. Corrales, A. Ledezma, A. J. Peña, J. Hoyos, A. Figueroa, y J. C. Corrales, «A new dataset for coffee rust detection in Colombian crops base on classifiers», *Sist. Telemática*, vol. 12, nº 29, pp. 9–23, 2014.
[17] M. Nikolic, «Measuring Similarity of Graphs and their Nodes by Neighbor Matching», *ArXiv Prepr. ArXiv10095290*, 2010.
[18] M. Sliger y S. Broderick, *The software project manager's bridge to agility*. Addison-Wesley Professional, 2008.
[19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, y A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
[20] K. Petersen, S. Vakkalanka, y L. Kuzniarz, «Guidelines for conducting systematic mapping studies in software engineering: An update», *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.
[21] A. Armiti, «Geometric Graphs: Matching, Similarity and Indexing», 2015.
[22] D. Grigori, J. C. Corrales, y M. Bouzeghoub, «Behavioral matchmaking for service retrieval: Application to conversation protocols», *Inf. Syst.*, vol. 33, nº 7, pp. 681–698, 2008.
[23] K. Riesen y H. Bunke, «Approximate graph edit distance computation by means of bipartite graph matching», *Image Vis. Comput.*, vol. 27, nº 7, pp. 950–959, 2009.
[24] K. Riesen, S. Emmenegger, y H. Bunke, «A novel software toolkit for graph edit distance computation», en *International Workshop on Graph-Based Representations in Pattern Recognition*, 2013, pp. 142–151.
[25] D. Conte, P. Foggia, C. Sansone, y M. Vento, «Thirty years of graph matching in pattern recognition», *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, nº 03, pp. 265–298, 2004.
[26] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, y H. Bunke, «A fast matching algorithm for graph-based handwriting recognition», en *International*

Workshop on Graph-Based Representations in Pattern Recognition, 2013, pp. 194–203.
[27] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, y P. Martineau, «An exact graph edit distance algorithm for solving pattern recognition problems», en *4th International Conference on Pattern Recognition Applications and Methods 2015*, 2015.
[28] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärik, y J. Mendling, «Similarity of business process models: Metrics and evaluation», *Inf. Syst.*, vol. 36, nº 2, pp. 498–516, 2011.
[29] R. M. Dijkman, M. Dumas, y L. García-Bañuelos, «Graph Matching Algorithms for Business Process Model Similarity Search.», en *BPM*, 2009, vol. 5701, pp. 48–63.
[30] T. THAMATA, C. DI GIROLAMO NETO, y C. A. Meira, «Sistema de alerta da ferrugem do cafeeiro: resultado de um processo de mineração de dados.», en *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*, 2013.
[31] C. A. A. MEIRA y L. H. A. RODRIGUES, «Mineração de dados no desenvolvimento de sistemas de alerta contra doenças de culturas agrícolas», en *CONGRESSO BRASILEIRO DE AGROINFORMÁTICA*, 2005, vol. 5.
[32] C. G. Neto, L. H. A. Rodrigues, y C. A. A. Meira, «Modelos de predição da ferrugem do cafeeiro (Hemileia vastatrix Berkeley & Broome) por técnicas de mineração de dados», *Coffee Sci.*, vol. 9, nº 3, pp. 408–418, 2014.
[33] C. A. A. Meira, L. H. A. Rodrigues, S. A. de Moraes, y others, «Modelos de alerta para o controle da ferrugem-do-cafeeiro em lavouras com alta carga pendente», *Pesqui. Agropecuária Bras.*, 2009.
[34] C. DI GIROLAMO NETO, L. H. A. Rodrigues, T. T. Thamada, y C. A. A. Meira, «Desenvolvimento e seleção de modelos de alerta para a ferrugem do cafeeiro em anos de alta carga pendente de frutos.», en *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*, 2013.
[35] M. E. Cintra, C. A. Meira, M. C. Monard, H. A. Camargo, y L. H. Rodrigues, «The use of fuzzy decision trees for coffee rust warning in Brazilian crops», en *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference*, 2011, pp. 1347–1352.
[36] C. B. Pérez-Ariza, A. E. Nicholson, y M. J. Flores, «Prediction of coffee rust disease using bayesian networks», en *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models*, 2012, pp. 259–266.
[37] O. Luaces, L. H. A. Rodrigues, C. A. A. Meira, J. R. Quevedo, y A. Bahamonde, «Viability of an alarm predictor for coffee rust disease using interval regression», en *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2010, pp. 337–346.

[38] D. C. Corrales, A. Figueroa, A. Ledezma, y J. C. Corrales, «An empirical multi-classifier for coffee rust detection in colombian crops», en *International Conference on Computational Science and Its Applications*, 2015, pp. 60–74.
[39] E. Lasso y J. C. Corrales, «Expert system for crop disease based on graph pattern matching: a proposal», *Rev. Ing. Univ. Medellín*, vol. 15, nº 29, pp. 81–98, 2016.
[40] K. Lasinger, «Overview of Existing Software Tools for Graph Matching», Vienna University of Technology, Vienna Austria, Tech. Rep, PRIP-TR-130, 21, sep, 2013.
[41] R. C. Holt, A. Winter, y A. Schurr, «GXL: Toward a standard exchange format», en *Reverse Engineering, 2000. Proceedings. Seventh Working Conference on*, 2000, pp. 162–171.
[42] M. Zaslavskiy, F. Bach, y J.-P. Vert, «Many-to-many graph matching: a continuous relaxation approach», *Mach. Learn. Knowl. Discov. Databases*, pp. 515–530, 2010.
[43] S. Fankhauser, K. Riesen, y H. Bunke, «Speeding up graph edit distance computation through fast bipartite matching», *Graph-Based Represent. Pattern Recognit.*, pp. 102–111, 2011.
[44] K. Riesen, M. Neuhaus, y H. Bunke, «Bipartite graph matching for computing the edit distance of graphs», *GbRPR*, vol. 4538, pp. 1–12, 2007.
[45] U. Bellur y R. Kulkarni, «Improved matchmaking algorithm for semantic web services based on bipartite graph matching», en *Web Services, 2007. ICWS 2007. IEEE International Conference*, 2007, pp. 86–93.
[46] D. C. Corrales, Q. Peña, J. Andrés, C. León, A. Figueroa, y J. C. Corrales, «Early warning system for coffee rust disease based on error correcting output codes: a proposal», *Rev. Ing. Univ. Medellín*, vol. 13, nº 25, pp. 57–64, 2014.
[47] E. Lasso, «Sistema experto basado en emparejamiento de patrones», Tesis de Maestría, Universidad Del Cauca, Popayán, 2016.
[48] J. Avelino, L. Willocquet, y S. Savary, «Effects of crop management patterns on coffee rust epidemics», *Plant Pathol.*, vol. 53, nº 5, pp. 541–547, 2004.
[49] M. Neuhaus, K. Riesen, y H. Bunke, «Fast suboptimal algorithms for the computation of graph edit distance», en *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2006, pp. 163–172.
[50] P.-A. Champin y C. Solnon, «Measuring the similarity of labeled graphs», en *International Conference on Case-based Reasoning, ICCBR*, 2003.
[51] E. Bengoetxea, P. Larrañaga, I. Bloch, y A. Perchant, «Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems», en *EMMCVPR*, 2001, vol. 1, pp. 454–468.

[52] F. Serratosa y X. Cortés, «Edit distance computed by fast bipartite graph matching», en *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2014, pp. 253–262.
[53] W.-J. Lee y R. P. Duin, «An inexact graph comparison approach in joint eigenspace», en *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2008, pp. 35–44.
[54] Y. Lu, K. Huang, y C.-L. Liu, «A fast projected fixed-point algorithm for large graph matching», *Pattern Recognit.*, vol. 60, pp. 971–982, 2016.
[55] M. Zaslavskiy, F. Bach, y J.-P. Vert, «A path following algorithm for the graph matching problem», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, nº 12, pp. 2227–2242, 2009.
[56] B. Luo y E. R. Hancock, «Structural graph matching using the EM algorithm and singular value decomposition», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, nº 10, pp. 1120–1136, 2001.
[57] W. Jones, A. Chawdhary, y A. King, «Revisiting Volgenant-Jonker for approximating graph edit distance», en *International Workshop on Graph-Based Representations in Pattern Recognition*, 2015, pp. 98–107.
[58] F. Serratosa, «Computation of graph edit distance: reasoning about optimality and speed-up», *Image Vis. Comput.*, vol. 40, pp. 38–48, 2015.
[59] F. Serratosa, «Fast computation of bipartite graph matching», *Pattern Recognit. Lett.*, vol. 45, pp. 244–250, 2014.
[60] H. Zhu y M. Zhou, «Efficient role transfer based on Kuhn–Munkres algorithm», *IEEE Trans. Syst. Man Cybern.-Part Syst. Hum.*, vol. 42, nº 2, pp. 491–496, 2012.
[61] K. Riesen, «Structural pattern recognition with graph edit distance: Approximation Algorithms and Applications», Springer Publishing Company. Switzerland.2016.
[62] K. Riesen, «Graph edit distance», en *Structural Pattern Recognition with Graph Edit Distance*, Springer, 2015, pp. 29–44.
[63] P. E. Hart, N. J. Nilsson, y B. Raphael, «A formal basis for the heuristic determination of minimum cost paths», *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, n.º 2, pp. 100–107, 1968.
[64] B. T. Messmer y H. Bunke, «A decision tree approach to graph and subgraph isomorphism detection», *Pattern Recognit.*, vol. 32, nº 12, pp. 1979–1998, 1999.
[65] H. W. Kuhn, «The Hungarian method for the assignment problem», *Nav. Res. Logist. NRL*, vol. 2, nº 1-2, pp. 83–97, 1955.
[66] N. X. Vinh, J. Epps, y J. Bailey, «Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance», *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, 2010.

[67] D. M. Powers, «Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation», 2011.
[68] J. Chen, H. Liu, J. Yang, y K.-C. Chou, «Prediction of linear B-cell epitopes using amino acid pair antigenicity scale», *Amino Acids*, vol. 33, nº 3, pp. 423–428, 2007.

Anexos

Anexo A

Patrones P1-P7

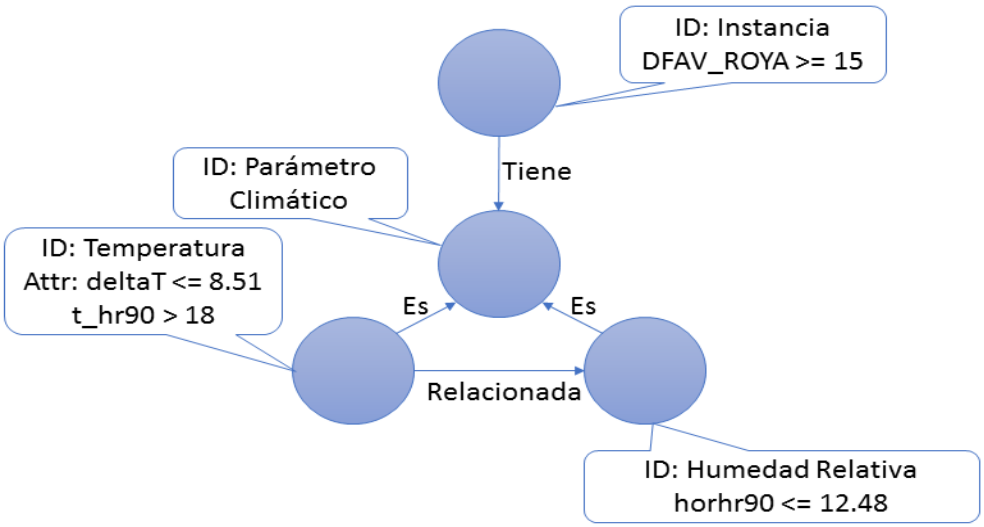


Figura 23. Patrón 1.

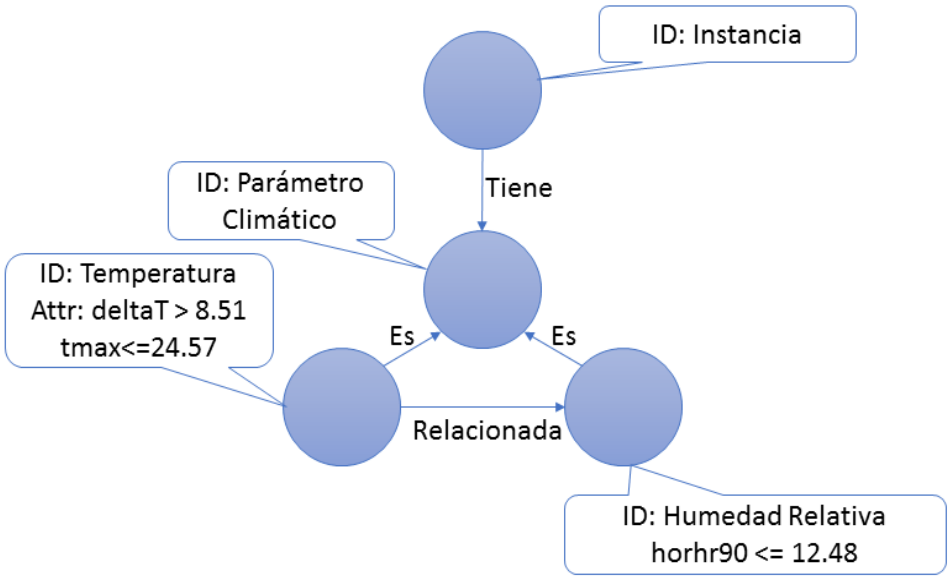


Figura 24. Patrón 2

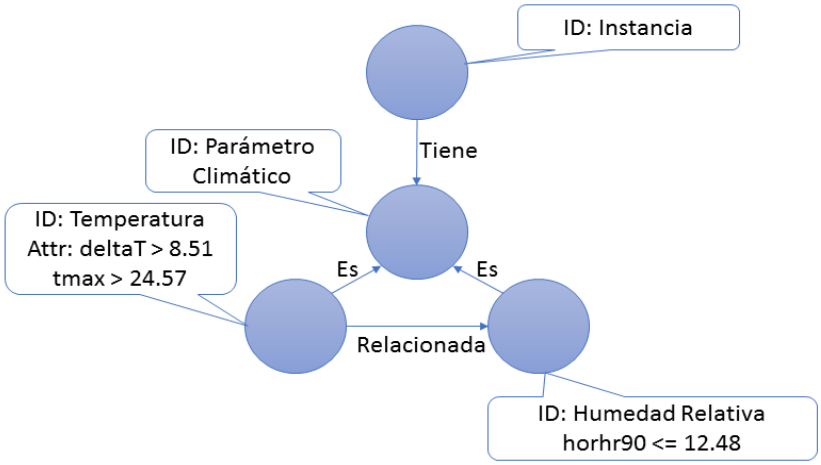


Figura 25. Patrón 3

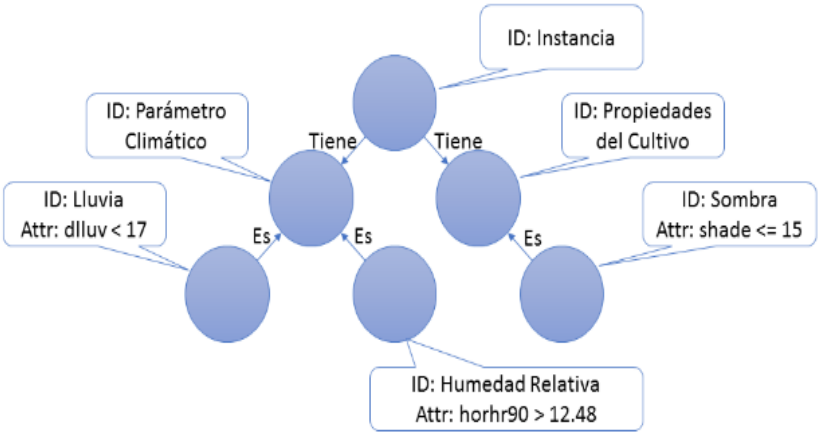


Figura 26. Patrón 4

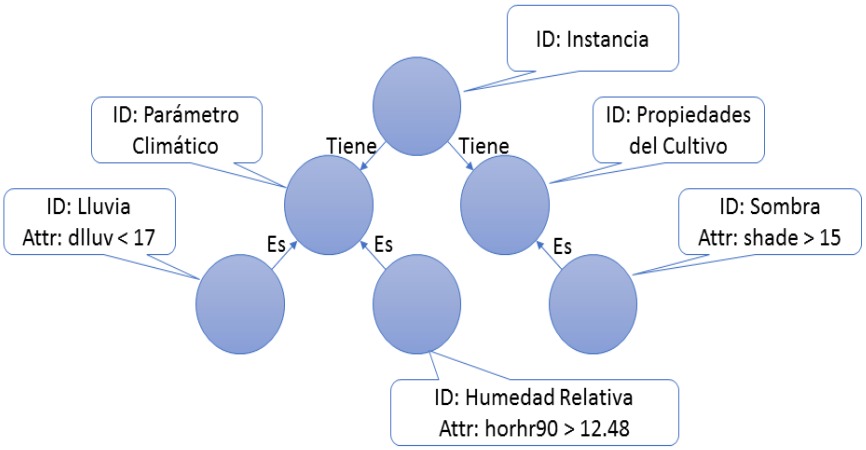


Figura 27. Patrón 5

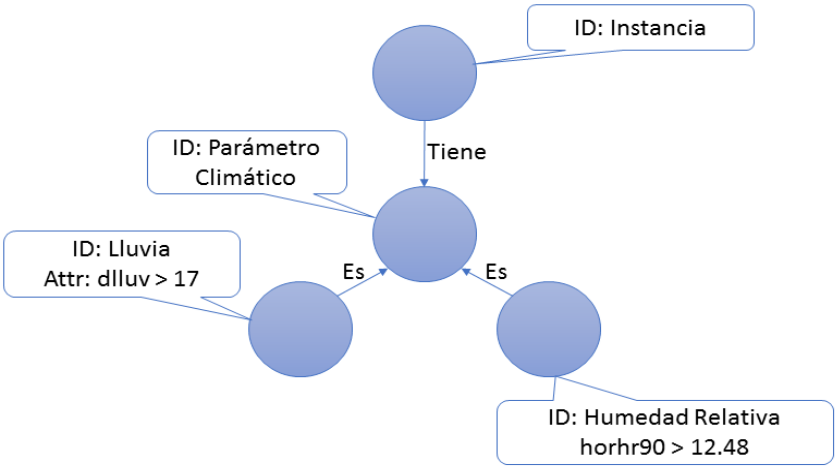


Figura 28. Patrón 6

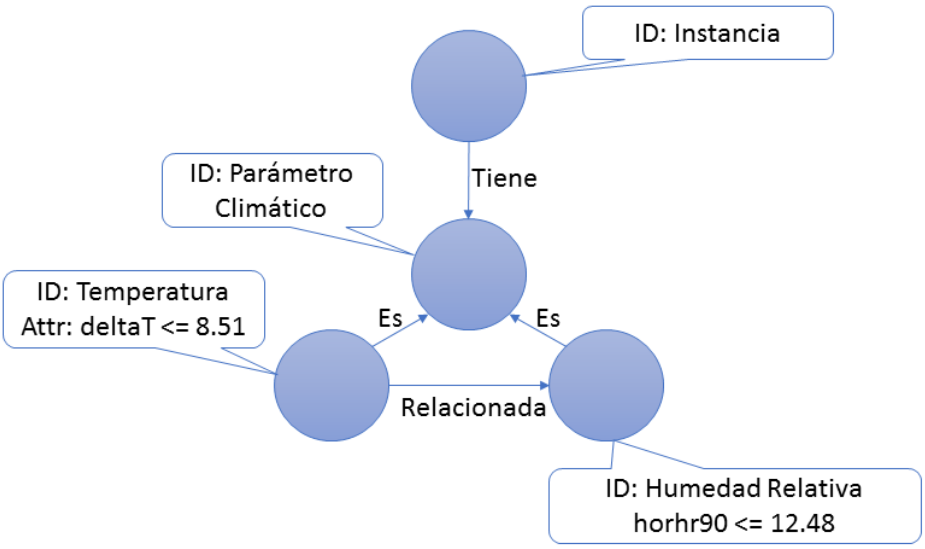


Figura 29. Patrón 7