ESCUELA DE INGENIERÍA EN COMPUTACIÓN

MAESTRÍA EN COMPUTACIÓN CON ÉNFASIS EN CIENCIAS DE LA COMPUTACIÓN

Identificación de documentos PICSs relevantes usando tráfico de red y la extracción automática de objetos y propiedades BACnet

Tesis para optar por el grado de Magister Scientiae en Computación

San José, mayo del 2022

Autor

Randall Jiménez Morales

Profesor asesor

Herson Esquivel Vargas

Instituto Tecnológico de Costa Rica
Escuela de Computación
Unidad de Posgrado

ACTA DE APROBACION DE TESIS

Identificación de documentos PICSs relevantes usando tráfico de red y la extracción automática de objetos y propiedades BACnet

Por: Randall Mauricio Jiménez Morales

TRIBUNAL EXAMINADOR

HERSON TOBIAS ESQUIVEL VARGAS (FIRMA)
Firmado digitalmente por HERSON TOBIAS ESQUIVEL VARGAS (FIRMA)
Fecha: 2022.06.01 13:56:45 -06'00'

MSc. Herson Esquivel Vargas
Profesor Asesor

KEVIN MORAGA GARCIA (FIRMA)
PERSONA FISICA, CPF-02-0617-0362.
Fecha declarada: 30/05/2022 05:39:43 PM
Esta representación visual no es fuente de confianza. Valide siempre la firma.

MSc. Kevin Moraga García
Profesor Lector

JOSE ENRIQUE ARAYA MONGE (FIRMA)
Firmado digitalmente por JOSE ENRIQUE ARAYA MONGE (FIRMA)
Fecha: 2022.06.01 17:16:04 -06'00'

Dr. José Enrique Araya Monge
Lector Externo

LILIANA SANCHO CHAVARRIA (FIRMA)
Firmado digitalmente por LILIANA SANCHO CHAVARRIA (FIRMA)
Fecha: 2022.06.16 11:28:37 -06'00'

Dra.-Ing. Lilliana Sancho Chavarría
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

23 de mayo, 2022

Tabla de contenido

Resumen

Con el aumento en la popularidad de los edificios inteligentes, la administración de los dispositivos especializados que ayudan a la automatización de las tareas se ha vuelto cada vez más necesaria. Con el fin de contar con un sistema interconectado de estos dispositivos se recurre al uso de algún protocolo de comunicación de datos, entre los existentes se cuenta con el ISO 16484-5 conocido como BACnet, por sus siglas en inglés (Building Automation and Control Networks).

Las capacidades documentadas de los dispositivos bajo el protocolo BACnet se encuentran en archivos en formato PDF llamados PICS, por sus siglas en inglés (Protocol Implementation Conformance Statement). Existen diversas razones por las cuales puede resultar muy útil conocer las capacidades de cada dispositivo como por ejemplo comunicarse con los dispositivos, desarrollar aplicaciones con ellos entre otros.

Independientemente la razón por la que se quiera conocer la información contenida en los PICSs, la extracción manual de su información resulta poco escalable y muy tediosa si se realiza de manera manual.

Aunque existen trabajos anteriores donde se ha conseguido la extracción automática de información contenida en los PICS, su alcance sea ha sido limitado a un pequeño grupo (10 PICSs). Este trabajo parte de esa idea y tiene como intención realizar el procesamiento total de objetos y propiedades de los dispositivos (PICSs) contenidos en el repositorio oficial del sitio de BACnet.

Aparte de la creación de un algoritmo capaz de realizar la extracción de los objetos y propiedades BACnet escritos como texto no estructurado en todos los documentos PICS existentes en el repositorio oficial, este trabajo busca mejorar la usabilidad existente en el estado del arte.

Como trabajo en general, se tiene como objetivo la creación de un sistema donde de manera automática un usuario suba a una página web un archivo de extensión PCAP, donde su contenido sea tráfico de red de un edificio inteligente, y el sistema extraiga la información necesaria para determinar los dispositivos que han generado tráfico, luego seleccione los PICSs que se encuentren relacionados a dichos dispositivos, y devuelva al usuario dos archivos por dispositivo encontrado: su respectivo PICS y un archivo que contenga los objetos y propiedades contenidos en ese PICS.

Para poder realizar dicho trabajo, además del algoritmo mencionado se realiza la creación de un algoritmo de recuperación de PICS según la información obtenida en el tráfico de red suministrado, basado en recuperación textual de la información.

Palabras clave: BACnet, edificios inteligentes, PICS.

Abstract

With the rise in popularity of smart buildings, the management of specialized devices that help automate tasks has become increasingly necessary. To have an interconnected system of these devices, some data communication protocol is used, among the existing ones is ISO 16484-5 known as BACnet (Building Automation and Control Networks).

The documented capabilities of the devices under the BACnet protocol are found in PDF files called PICS, (Protocol Implementation Declaration of Conformity). There are several reasons why it can be very useful to know the capabilities of each device, such as communicating with the devices, developing applications with them, for example.

Regardless of the reason why you want to know the information contained in the PICSs, the manual extraction of its information is not very scalable and very tedious if it is done manually.

Although there are previous works where the automatic extraction of information contained in the PICS has been achieved, its scope has been limited to a small group (10 PICSs). This work is based on that idea and aims to perform the total processing of objects and device properties (PICSs) contained in the official repository of the BACnet site.

In addition to the creation of an algorithm capable of extracting BACnet objects and properties written as unstructured text in all existing PICS documents in the official repository, this work seeks to improve the existing usability in the state of the art.

As a general work, the objective is to create a system where a user automatically uploads a file with a PCAP extension to a web page, where its content is network traffic from an intelligent building, and the system extracts the necessary information. to determine the devices that have generated traffic, then select the PICS that are related to those devices, and return to the user two files per device found: its respective PICS and a file containing the objects and properties contained in that PICS.

To carry out this work, in addition to the algorithm mentioned above, the creation of a PICS recovery algorithm is carried out according to the information obtained in the supplied network traffic, based on the recovery of textual information.

Keywords: BACnet, intelligent buildings, PICS.

Glosario

BACnet: Abreviación de “Building Automation and Control Networks”. Es un protocolo de comunicación de datos diseñado para comunicar entre sí a los diferentes dispositivos electrónicos presentes en los edificios, por ejemplo: alarmas, sensores de paso, aire acondicionado, luces, etc.

PICS: Siglas en inglés de "Protocol Implementation Conformance Statement", en español declaración de conformidad de implementación de protocolo, describe las capacidades de BACnet de una implementación de BACnet en particular.

PDF: Siglas en inglés de Portable Document Format, en español formato de documento portátil. Es un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

Ranking: Es un anglicismo que hace referencia a una relación entre un conjunto de elementos de tal manera que para cualquiera de los dos elementos el primero está "clasificado más alto que", "clasificado más bajo que" o "clasificado igual a" el segundo. También se usan en español los términos lista, tabla clasificatoria o escalafón para referirse a esto.

RS-485: También conocido como EIA-485, que lleva el nombre del comité que lo convirtió en estándar en 1983. Es un estándar de comunicaciones en bus de la capa física del Modelo OSI.

SAE: Sistema de automatización de edificios es un término general, que también se conoce como sistema de gestión de edificios, BMS que se utiliza para referirse a una amplia gama de sistemas computarizados de control de edificios, como por ejemplo controladores especiales y estaciones remotas independientes.

Stop words: el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto).

1.Introducción

Con el auge de los edificios inteligentes, su correcta administración ha cobrado una relevancia cada vez mayor. El mantenimiento de los dispositivos que ayudan a la automatización de las tareas de dichos edificios es tanto imprescindible como un reto. Entre las posibles técnicas de supervisión y mantenimiento de los dispositivos está el monitoreo de su actividad, mediante el tráfico de red generado por ellos mismos.

En esta dirección, la identificación de los dispositivos mediante el tráfico de red, y conocer la manera esperada de su comportamiento, resulta vital para su mantenimiento. Entre las posibles formas de conocer su comportamiento esperado se cuenta con acceder a los archivos de especificaciones técnicas dadas por cada fabricante.

Con el constante aumento de estos dispositivos especializados, consultar sus especificaciones técnicas de manera manual, resulta poco práctico y escalable. Así que encontrar una manera de automatizar la búsqueda y recuperación de la información deseada es importante.

Esta tesis se basa en el primer esfuerzo realizado en el 2017 por varios investigadores [1], los cuales crearon un algoritmo para la recuperación de información contenida como texto no estructurado en documentos en formato PDF, con la ayuda del tráfico de red.

Los archivos a procesar corresponden al protocolo de comunicación de datos ISO 16484-5 conocido como BACnet. Los archivos en formato PDF contienen las especificaciones BACnet de dispositivos bajo una implementación BACnet en particular. Dichos archivos escritos como texto no estructurado se conocen como PICS por sus en inglés (Protocol Implementation Conformance Statement).

A nivel operativo, el trabajo se divide en tres fases. La primera en la creación de un algoritmo de identificación de los archivos PICSs basada en análisis de tráfico de red. La segunda en la creación de un algoritmo que extraiga los objetos y propiedades de cada PICSs representado en archivos en formato PDF, y por último una interfaz web donde el usuario suba un archivo de tráfico de red en formato PCAP y el sistema de manera automática busque el PICS correcto, y devuelva además del PICS un archivo con los objetos y propiedades contenidos en dicho archivo.

1.1 Definición del problema

Desde hace unos años es cada vez más normal contar con edificios con algún grado de automatización en asuntos referentes a control de acceso, comodidad y en aspectos de ahorro de consumo de energía y recursos en general. Con el avance en la tecnología se ha

pasado de edificios con cierto grado de automatización hasta los que cuentan con sistemas de información en toda su edificación, permitiendo que estén automatizados y autorregulados. Estos últimos son los llamados edificios inteligentes.

Los sistemas de automatización de edificios (SAE) supervisan y controlan los procesos físicos en los edificios modernos. Las implementaciones típicas no solo automatizan los servicios antes mencionados, también alarmas, ascensores, control de acceso físico, entre otros. A diferencia de los sistemas de TI estándar, los SAEs pueden influir en el entorno físico.

Para poder realizar todas las funciones inherentes en la automatización de los edificios, es necesario contar con dispositivos electrónicos especializados en las actividades que se desean automatizar; y con el fin de contar con un sistema interconectado compuesto de los dispositivos antes mencionados se debe utilizar algún protocolo de comunicación de datos, entre ellos existe el protocolo abierto ISO 16484-5:2017 (BACnet).

Las capacidades documentadas de los dispositivos bajo el protocolo BACnet se encuentran en archivos en formato PDF llamados PICS. PICS son siglas en inglés que en español significan declaración de conformidad de implementación de protocolo, y describen las capacidades BACnet de una implementación de BACnet en particular.

Independientemente de si se es desarrollador o si se desea comunicarse con los dispositivos, es necesario conocer el protocolo, y muy específicamente los objetos y propiedades de cada dispositivo.

Cualquiera que sea de la razón por la cual se desee conocer la información contenida en los PICSs, la extracción de la información resulta muy tediosa y poco escalable si se realiza de manera manual.

1.2 Justificación

Existen muchas situaciones por las cuales acceder a la información escrita en los PICSs es necesaria. Entre esas están, tener un inventario de las capacidades de cada dispositivo, conocer cuando un dispositivo, en alguna potencial actualización, va a quedar rezagado u obsoleto, para la implementación de seguridad basada en comparar el tráfico de red con las capacidades documentadas de cada dispositivo para buscar inconsistencias, entre otras.

La información referente a los dispositivos BACnet está organizada en objetos, lo cual le convierte en un protocolo orientado a objetos. Cada objeto representa un componente del propio dispositivo o un conjunto de información que puede ser solicitada por otro dispositivo a través de medios como RS-485, Ethernet, etc. De esta manera, si se desea conocer la información de cada dispositivo de manera manual es necesario acceder al archivo PICS, el cual es provisto por cada fabricante de los dispositivos.

Actualmente aunque existen maneras 100% automáticas de extraer información de los PICSs, se ha realizado con un éxito limitado a 10 PICSs, y no de una forma muy amigable con el usuario [1].

La finalidad de este trabajo es crear un algoritmo que sea capaz de extraer los objetos y propiedades BACnet contenidas en todos los archivos PICS disponibles en el repositorio oficial [2], con el fin de contar con toda la información indexada para poder realizar cualquier tarea como las mencionadas anteriormente.

Aparte de la creación de un algoritmo con el cual analizando el tráfico de red generado en un SAE se puedan recuperar de manera automática los PICS que hacen referencia a la información contenida en el tráfico, se pretende crear un algoritmo innovador para la extracción automática de objetos y propiedades BACnet de documentos técnicos no estructurados (PICS).

1.3 Contribuciones

Como se ha mencionado en apartados anteriores, ya existen esfuerzos donde se han podido extraer de manera automática los objetos y propiedades de los archivos PICSs, la contribución de este trabajo con respecto al estado del arte va orientado sobre dos vertientes diferentes: usabilidad y alcance de efectividad de la propuesta.

Con respecto a la usabilidad, se propone la creación de una interfaz gráfica web con la cual el usuario final pueda interactuar directamente. Dejando de lado todos los procesos necesarios para la obtención y manipulación realizada para conseguir el PICS y sus respectivos objetos y propiedades; todo con solo subir un archivo (extensión pcap) que contenga el tráfico de red.

La herramienta presenta una usabilidad mejorada con respecto al estado del arte, ya que con solo ejecutar un archivo el sistema procesará el conjunto de PICSs que se encuentren en una ubicación especifica (configurable fácilmente), y sin ningún otro trabajo, se generará la información y estructuras necesarias para obtener los objetos y propiedades de cada PICS, así como para realizar la búsqueda de cada PICS mediante recuperación textual de la información basada en el tráfico de red.

Con respecto a la efectividad, se ha fijado la meta de procesar el 100% de los PICSs existentes en el repositorio oficial (2504 para la fecha que se realizó el trabajo). Mejorando así el estado del arte, cuya evaluación considera únicamente un subconjunto de 10 PICS.

Además, entre las contribuciones de este proyecto, está la precisión en la recuperación de la información. En un enfoque tradicional de recuperación de la información, luego de una consulta, habitualmente los sistemas regresan un conjunto de documentos ranqueados

según un criterio de similitud. En este caso, el sistema debe devolver solo un documento, el correcto, ya que si se escoge el documento incorrecto la extracción de las propiedades de los dispositivos también será errada. En pocas palabras se debe minimizar la posibilidad de error lo máximo posible.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar un sistema de recuperación automática de documentos PICSs y de recuperación de sus respectivos objetos y propiedades basada en el análisis de texto no estructurado y de tráfico de red; con un nivel de precisión, recuperación y usabilidad equivalente o superior al estado del arte.

1.4.2 Objetivo específicos

1. Crear un algoritmo de recuperación automática de PICS según la información obtenida en el tráfico de red suministrado.

2. Crear un algoritmo para la recuperación de objetos y propiedades BACnet escritos como texto no estructurado en documentos PICS.

3. Comparar el nivel de precisión y recuperación del algoritmo de la extracción de la información de los dispositivos BACnet con el estado del arte.

4. Implementar la interfaz de usuario, que facilite a las personas usuarias la ejecución del algoritmo en una plataforma centralizada.

1.5 Preguntas de investigación

- ¿Cómo seleccionar automáticamente los PICS de aquellos dispositivos identificados en usando tráfico de red?

- ¿De qué manera se pueden recuperar los objetos y propiedades de los documentos PICs automáticamente?

- ¿Cómo facilitar la usabilidad de un sistema que retorne objetos y propiedades BACnet de los dispositivos identificados en el tráfico de red suministrado como entrada?

2.Marco teórico

2.1 Recuperación de la información

La recuperación de información es el conjunto de actividades orientadas a facilitar la localización de determinados datos. Al contrario de lo que se puede pensar, no es un área nueva, sino que se viene desarrollando desde finales de la década de 1950. Sin embargo, hoy en día adquiere un rol más importante debido al valor que tiene la información.

De manera general el problema de la recuperación de la información puede ser estudiado desde dos puntos de vista: el computacional y el humano [3]. El primer caso tiene que ver con la construcción de estructuras de datos y algoritmos eficientes que mejoren la calidad de las respuestas. El segundo caso corresponde al estudio del comportamiento y de las necesidades de los usuarios.

En esencia un sistema de recuperación de la información parte de un conjunto de documentos de texto, los cuales están compuestos por sucesiones de palabras que forman estructuras gramaticales. Dichos documentos están escritos en lenguaje natural y expresan ideas sobre un determinado tema. El conjunto de todos los documentos con los que se trata y sobre los que se deben realizar operaciones de recuperación de la información se denomina colección (corpus, o base de datos textual o documental). Para poder realizar operaciones sobre una colección, es necesario obtener primero una representación lógica de todos sus documentos, la cual puede consistir en un conjunto de términos, frases u otras unidades sintácticas o semánticas que permitan de alguna manera caracterizarlos.

La representación lógica de los documentos crea un proceso de indexación que genera la construcción de estructuras de datos (normalmente denominadas índices), cada tipo de representaciones dependen en gran medida al modelo especifico de recuperación que se use [4].

Definición:

Un modelo de recuperación de la información es un cuádruple [5] < D, Q, F, $R(q_i,d_j)$ > donde:

1. D: Es el conjunto de vistas lógicas (o representaciones) de los documentos en la colección.
2. Q: Es el conjunto de vistas lógicas (o representaciones) de las necesidades de información de los usuarios; esto son las consultas.

3. F: Es un marco conceptual para modelar las representaciones de los documentos, las consultas y sus relaciones.

4. $R(q_i,d_j)$: Función de escalafón (ranking) que asocia un número real con un $q_i \in Q$ y un documento $d_j \in D$. Esto define un orden para los documentos con relación a la consulta q_i.

Un modelo se construye elaborando representaciones para documentos y necesidades de información, para luego establecer un marco que permita definir una función de escalafón.

En la Ilustración 1 se muestra la arquitectura básica de un sistema de recuperación de la información.

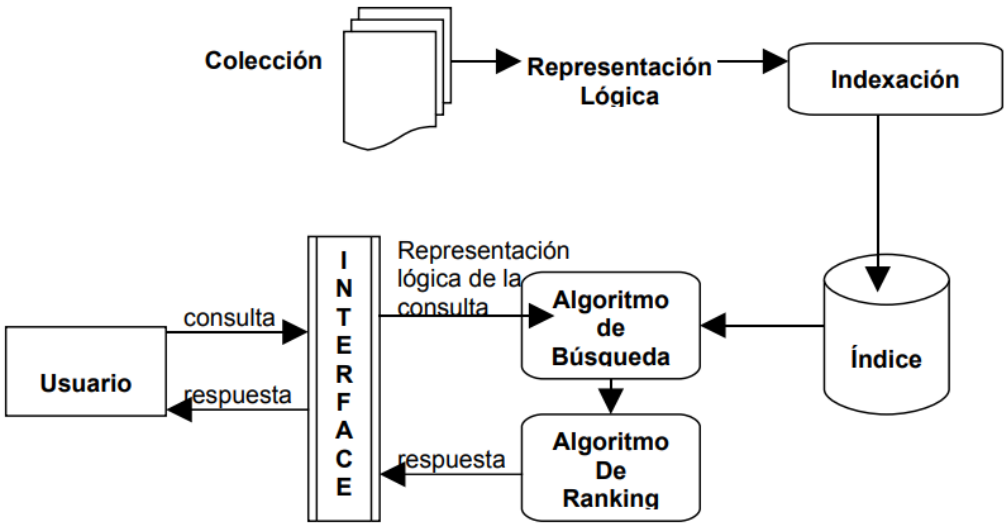

Ilustración 1. Arquitectura básica de un sistema de recuperación de la información.[4]

A continuación, y luego de una investigación previa sobre los modelos existentes se describen los dos modelos seleccionados para su implementación, los cuales están agrupados según la base matemática que utilizan para su funcionamiento.

2.1.1 Modelos basados en teoría de conjuntos

Se caracterizan en que los documentos se representan como un conjunto de palabras o frases, entre los más comunes se encuentran:

Modelo Booleano.

Las consultas son expresadas como expresiones booleanas que tienen un significado muy preciso.

Ventajas

- Es de los primeros y más extendidos modelos existentes.
- Es fácil de implementar gracias a su simpleza.

Desventajas

- La estrategia de búsqueda se basa en un criterio binario sin que permita grados o escalas en las que un documento pueda clasificarse; es más una estrategia de extracción de datos que de búsqueda textual.
- Si bien una expresión booleana tiene un significado preciso, con frecuencia no es fácil traducir las necesidades de información en una expresión booleana.

Definiciones [5]:

- Para el modelo booleano los pesos son binarios {*true*, *false*} frecuentemente representados como {0, 1}.
- Una consulta es una expresión booleana compuesta de términos ligados por conectivas and, or y not.
- La similitud entre un documento y una consulta se obtiene evaluando la expresión de la consulta para los valores presentes en el documento:

$$\mathrm{sim}(d_j, q) = \begin{cases} 1 & \text{si q evalúa true} \\ 0 & \text{de cualquier otra manera} \end{cases}$$

- Si $\mathrm{sim}(d_j, q) = 1$, entonces el modelo predice que el documento es relevante, de lo contrario se considera no relevante.
- Para el modelo booleano un documento es relevante o no es relevante.
- No usa la noción de resultado parcial:
  - Para una consulta con n términos ligados con *and* da lo mismo fallar todos los términos que fallar solo uno.
  - Para una consulta con n términos ligados con *or* da lo mismo acertar todos los términos que acertar solo uno.

Su simplicidad es su ventaja, los resultados del modelo booleano pueden ser mejorados usando pesos no binarios en los términos.

En la
Ilustración *2* se muestra un ejemplo del modelo [5]:

La colección consta de 8 documentos y hay 11 términos distintos. La distribución de los términos y los valores de k_i se muestran a continuación:

	1	2	3	4	5	6	7	8	9	10	11
k_i	a	b	c	d	e	f	g	h	i	j	k
1	1	1		1		1			1		
2		1	1	1			1				1
3	1		1	1				1	1	1	
4	1			1		1	1				
5		1				1		1		1	
6			1				1		1		
7				1						1	
8		1			1	1			1		1
n_i	3	4	3	5	1	4	3	2	4	3	2

Ilustración 2. Ejemplo de la ejecución del algoritmo del Modelo Booleano.

La consulta involucra a los términos a, b y c.
Se plantean dos variantes de la consulta:

- a **and** b **and** c
- a **or** b **or** c

doc	and	or
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	0
8	0	1

Ilustración 3. Ejemplo de una consulta en el Modelo Booleano.

En este ejemplo se nota la crudeza del modelo booleano, por un lado, se obtiene muy poca información por ser muy estricto, o se obtienen demasiados documentos sin ningún criterio de clasificación entre ellos.

2.1.2 Modelos Algebraicos

En estos modelos los documentos y las consultas se representas como vectores, matrices o tuplas.

Modelo Vectorial

En este modelo se intenta recoger la relación de cada documento D_i, de una colección de N documentos, con el conjunto de las w características de la colección. Formalmente un documento puede considerarse como un vector que expresa la relación del documento con cada una de esas características [6].

El vector para un documento d_j es representado por:

$$\overrightarrow{d_j} = (w_{1j}, w_{2j}, ..., w_{tj})$$

Es decir, ese vector identifica en qué grado el documento D_i satisface cada una de las w características. En ese vector, c_{ik} es un valor numérico que expresa en qué grado el documento D_i posee la característica k. El concepto "característica" suele concretarse en la ocurrencia de determinadas palabras o términos en el documento, aunque nada impide tomar en consideración otros aspectos.

El peso de un término para un documento debe aumentar con la frecuencia con que dicho término aparece en el documento y debe disminuir en proporción al porcentaje de documentos de la colección en que aparece el término.

Los pesos pueden ser calculados de muchas maneras, a manera general se pueden calcular de la siguiente manera:

$$peso = log(N/n_i)$$

Siendo:

N: número total de documentos en el sistema

n_i: número de documentos en los cuales parece el término k_i

Este enfoque solo toma en cuenta el porcentaje de documentos de la colección en que aparece el término, omite la frecuencia del término en el documento, ya que por el tipo de información que se desea trabajar, no importa si un termino a buscar aparece más de una vez.

A continuación, se muestra un ejemplo del modelo [5]:

Colección de documentos:

d_1	La mayoría de los dinosaurios fueron comidos por otros dinosaurios.
d_2	La mayoría es la mitad más uno.
d_3	La mitad de dos tercios no es mayoría.

	mayoria	dinosaurio	fueron	comido	otro	mitad	mas	uno	dos	tercios	no	MAX freq
d_1	1	2	1	1	1							2
d_2	1					1	1	1				1
d_3	1					1			1	1	1	1
n_i	3	1	1	1	1	2	1	1	1	1	1	
f	0.0	0.48	0.48	0.48	0.48	0.18	0.48	0.48	0.48	0.48	0.48	
d_1	0.0	0.48	0.24	0.24	0.24							
d_2	0.0					0.18	0.48	0.48				
d_3	0.0					0.18			0.48	0.48	0.48	

Ilustración 4. Ejemplo de la ejecución del algoritmo del Modelo Vectorial.

Donde: t=11 N=3 $\log_{10}(3/1) = 0.477 = 0.48$ $\log_{10}(3/2) = 0.176 = 0.18$

Ventajas

- El uso de pesos mejora los resultados de las búsquedas.
- Permite una estrategia de correspondencia parcial que permite extraer documentos que solo aproximan las condiciones de una consulta.

Desventajas

- El costo computacional del cómputo del escalafón puede ser enorme en colecciones de documentos suficientemente grandes.
- Al construir los vectores términos a partir de los documentos, al modificar la colección de documentos habrá que recalcular nuevamente todos los valores almacenados en el sistema, tanto de los vectores términos como de los vectores documentos.

Entre sus principales características están [5]:

- Usa pesos no binarios para los términos que aparecen en documentos y en consultas.

- Estos términos permiten calcular el grado de similitud entre cada documento y la consulta.
- Permite ordenar los documentos en forma decreciente por ese grado. Esto permite que los documentos que se estiman más similares a la consulta se le presentan primero al usuario.
- También permite considerar los documentos con similitud parcial.
- El escalafón de documentos del modelo vectorial es más preciso que el conjunto de documentos del modelo booleano.
- A pesar de su simplicidad, la estrategia de rankeo del modelo vectorial es muy conveniente.
- Permite dar respuestas que son muy difíciles de mejorar sin usar refinamientos de las consultas.
- Una gran variedad de métodos alternativos de rankeo han sido comparados con el modelo vectorial, pero el modelo vectorial se desempeña tan bien o mejor que las alternativas en documentos no muy extensos, en caso contrario es recomendable el metodo probabilistico BM25.
- Es simple y rápido.
- Es muy usado en la práctica.

2.1.3 Métodos de evaluación

Existen un conjunto de medidas de evaluación que habitualmente se aplican a los sistemas de recuperación de la información con el fin de medir la efectividad de este. A continuación, se describen las métricas [7].

La Exhaustividad

Es la proporción de los documentos relevantes que han sido recuperados. Permite evaluar la habilidad del sistema para encontrar todos los documentos relevantes de la colección [4].

Su fórmula es la siguiente:

$$\text{Exhaustividad (E)} = \frac{Cantidad\ de\ documentos\ relevantes\ \ recuperados}{Cantidad\ de\ documentos\ relevantes}$$

La Precisión

Es la proporción de los documentos recuperados que son relevantes y permite evaluar la habilidad del sistema para posicionar de primero la mayoría de los documentos relevantes [4].

Su fórmula es la siguiente:

Precisión (P) = $\frac{Cantidad\ de\ documentos\ relevantes\ \ recuperados}{Cantidad\ de\ documentos\ recuperados}$

La exhaustividad y la precisión se encuentran altamente relacionadas. De manera informal se ha comprobado que una alta exhaustividad se acompaña de una muy baja precisión y viceversa, es decir, existe una relación inversa entre ambas. Esto se explica en el hecho de que la salida de un sistema de recuperación de la información es un conjunto aproximado (no exacto) y por lo tanto entre ésta se encontrarán documentos no relevantes. Por el contrario, si recuperamos unos pocos documentos y todos son relevantes se tendrá una precisión máxima, pero seguramente se están perdiendo documentos útiles por no ser recuperados. El sistema ideal es aquel que siempre recupera todos los documentos relevantes y solo esos, situación que hasta el momento no existe.

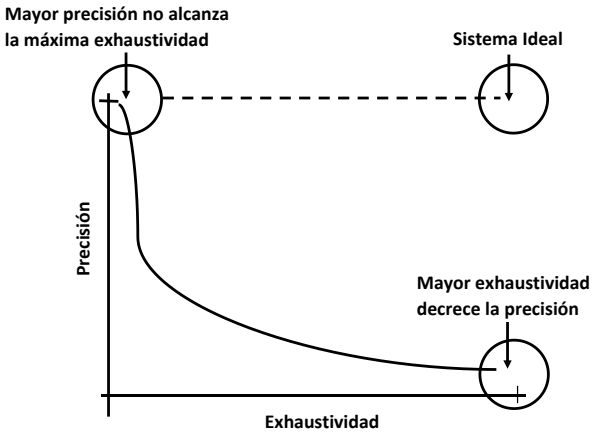

Ilustración 5. Relación entre Precisión y Exhaustividad [8].

2.2 Extracción de la información en archivos en formato PDF

El formato de archivos PDF se desarrolló entre 1991 y 1992 por Adobe. El objetivo consistía en que todo el mundo pudiera capturar documentos de cualquier aplicación y enviar la versión electrónica de dichos documentos a donde quisiera, así como verlos e imprimirlos desde cualquier máquina [8]. Debido a que son independientes del software de aplicación, el hardware y los sistemas operativos, los archivos PDF se han convertido en una forma popular de compartir documentos.

A diferencia de otros archivos como por ejemplos los XML [9], que gracias a sus etiquetas definen la estructura que posee un documento, y que establecen una sintaxis para la codificación de documentos, que tanto los usuarios (humanos) como las propias máquinas

en sí puedan ser capaces de leer, los archivos en formato PDF carecen de esta característica o alguna otra que permita estructurar la información contenida. Resulta una tarea desafiante estructurar y comprender los datos contenidos en un archivo en formato PDF. Un software para extraer datos de archivos PDF es más una necesidad para aquellas organizaciones que tienen una gran cantidad de archivos PDF y no pueden perder tiempo en la extracción manual de datos [10].

Actualmente existen muchas herramientas para la extracción de la información de los archivos PDF, todo depende del enfoque que se le desee dar a la extracción, como por ejemplo se encuentran herramientas que extraen la información de los archivos y su salida es un archivo en otro formato, ya sea formato de texto (txt) archivos de Excel, etc.

Después de una investigación realizada (ver anexo #3) sobre las herramientas existentes, se ha escogido una herramienta de tipo biblioteca, la cual sirve con la unión de un lenguaje de programación para extraer y procesar la información de los archivos PDF.

La herramienta escogida es Apache PDFBox. Entre las razones por la cual se escogió sobre el resto de las herramientas investigadas, es porque fue implementada en el mismo lenguaje que se escogió para la realización del proyecto [11], y los abundantes ejemplos existentes en la red.

2.3 Building Automation and Control Networks (BACnet)

Es el protocolo de control de red y automatización de edificios creado por la Sociedad Americana de Ingenieros de Calefacción, Refrigeración y Aire Acondicionado (ASHRAE), ha sido diseñado específicamente para satisfacer las necesidades de comunicación de los sistemas de control y automatización de edificios para aplicaciones como control de calefacción, ventilación y aire acondicionado, control de iluminación, control de acceso y sistemas de detección de incendios. El protocolo BACnet proporciona mecanismos mediante los cuales equipos computarizados de función arbitraria pueden intercambiar información, independientemente del servicio particular que realice. Como resultado, el protocolo BACnet puede ser utilizado por servidores, controladores digitales directos de propósito general y controladores unitarios o específicos de la aplicación con el mismo efecto [12].

BACnet define una serie de protocolos y estándares (servicios) que se utilizan para comunicarse entre los dispositivos del edificio. Los servicios de protocolo incluyen Who-Is, I-Am, Who-Has, I-Have, que se utilizan para el descubrimiento de dispositivos y objetos. Los servicios como Read-Property y Write-Property se utilizan para compartir datos. A partir de ANSI / ASHRAE 135-2016, el protocolo BACnet define 60 tipos de objetos sobre los que actúan los servicios.

Cada dispositivo BACnet es una combinación de hardware y software, encontrándolo como norma general en forma de controlador, puerta de enlace (gateway) o interfaz de usuario.

Cada uno de ellos dispone de un identificador o número de instancia único que lo identifica y diferencia de los existentes en la red.

Dentro del protocolo BACnet se encuentran dos conceptos claramente diferenciados: objetos y propiedades. Toda la información contenida dentro de un dispositivo BACnet es ordenada como objetos, el cual le convierte en un protocolo orientado a objetos. Cada uno de los objetos representan un componente del propio dispositivo o un conjunto de información, que otro dispositivo puede solicitar a través de medios como Ethernet, IP, RS-485, etc. [13].

El protocolo define un total de 60 tipos de objetos para los usos más comunes cubriendo así la mayoría de las necesidades. Sin embargo, da la posibilidad de crear otros nuevos para no interferir sobre aquellos usos más concretos o aplicaciones propietarias. Cada objeto tiene un identificador de 32 bits el cual contiene un código para el tipo de objeto y el número de instancia de dicho objeto.

Con independencia del propósito o función, cada objeto posee un conjunto de propiedades. Cada una de ellas contiene dos partes, un identificador y un valor. El identificador son números que la definen de forma única en el ámbito del tipo de objeto; y el valor, la magnitud de este. Dicha información podrá ser tanto de lectura como de y escritura. Si unificamos ambos conceptos, podemos decir que cada objeto incluye propiedades que determinan sus capacidades, operación e información asociada.

Cada dispositivo envía las peticiones empleando mensajes convenientemente estructurados tanto en la solicitud como en la respuesta. Los mensajes son codificados en caracteres numéricos definiendo las funciones que deben llevarse a cabo.

Ya que BACnet es un estándar, los distintos fabricantes de equipos deben garantizar que sus productos soportan ciertos requisitos para poder operar unos con otros. Para ello, deben cumplir con lo definido en lo que se denomina "Protocol and Implementation and Conformance Statement", *PICS*; una ficha técnica con distinta información [14]:

- Información básica que identifica al proveedor y describe el dispositivo BACnet.
- Los bloques de construcción de interoperabilidad BACnet admitidos por el dispositivo.
- El perfil de dispositivo BACnet estandarizado al que se ajusta el dispositivo, si lo hay.
- Todos los servicios de aplicaciones no estándar que son compatibles junto con una indicación para cada servicio de si el dispositivo puede iniciar la solicitud de servicio, responder a una solicitud de servicio, o ambas.
- Una lista de todos los tipos de objetos estándar y patentados que son compatibles.
- Para cada tipo de objeto admitido:
    - Cualquier propiedad opcional que sea compatible.
    - En qué propiedades se puede escribir utilizando los servicios BACnet.
    - Si los objetos se pueden crear o eliminar dinámicamente utilizando los servicios BACnet.
    - Cualquier restricción en el rango de valores de datos para propiedades.

- Se admiten las opciones de la capa de enlace de datos, tanto reales como virtuales.
- Si se admiten solicitudes segmentadas.
- Si se admiten respuestas segmentadas.
- Puede descargar una copia de un PICS editable [15].

En las Ilustraciones 6 y 7 se muestran dos ejemplos de PICSs:

GREYSTONE
ENERGY SYSTEMS INC

BACnet Protocol Implementation Conformance Statement (PICS)

Date : April 24, 2014
Vendor Name : Greystone Energy Systems
Product Name : Carbon Dioxide Detector
Product Model Number : CDD3
Application Software Version : 1.0
Firmware Revision : 1.4
BACnet Protocol Revision : 7

Product Description : The Greystone CO2 Detector uses Infrared Technology to monitor CO2 levels and features a native BACnet MS/TP protocol for network communication. It measures CO2 levels and reports this value back to a building automation system (BAS). The device features has an LCD to display measured values. Options include a control relay, RH and temperature sensors

BACnet Standardized Device Profile (Annex L) : BACnet Application Specific Controller (B-ASC)

BACnet Interoperability Building Blocks Supported (Annex K) : DS-RP-B, DS-WP-B, DM-DDB-B, DM-DOB-B DM-DCC-B

Segmentation Capability : Not supported

Standard Object Types Supported :

Object Type	Dynamically Creatable	Dynamically Deletable	Optional Properties Supported	Writable Properties
Device	No	No	Location, Description, Max_Master, Max_Info_Frames	Object_Identifier, Object_Name, Location, Description, APDU_Timeout, Max_Master, Number_Of_APDU_Retries
Analog Input	No	No	Description, Reliability, Device_Type	
Analog Value	No	No	Description	Present_Value
Binary Input	No	No	Description, Reliability Device_Type	

Data Link Layer Options : MS/TP master (Clause 9), baud rates : 9600, 19200, 38400, 76800

Device Address Binding : Not supported

Networking Options : None

Character Set Supported : ANSI X3.4

150 English Drive, Moncton, New Brunswick Canada E1E 4G7
Phone: +1-506-853-3057 Fax: +1-506-853-6014 Email: mail@greystoneenergy.com

Ilustración 6. Ejemplo de un PICS de la marca Greystone.

BACnet® Protocol Implementation Conformance Statement

Vendor		Listing Status
WattMaster Controls, Inc. 8500 NW River Park Drive, Suite 108A Parkville, MO 64152 USA		Listed Product
Test Requirements	**BACnet® Protocol Revision**	**Date Tested**
Requirements as of December 2011	Revision 12 (135-2010)	April 2013

Product Name	Model Number	Software Version
DDC Controller	OE377-26B-00001	1.00

BACnet® Standardized Device Profile (Annex L)
BACnet Application Specific Controller (B-ASC)

BIBBs Supported		
Data Sharing	ReadProperty-B	DS-RP-B
	ReadPropertyMultiple-B	DS-RPM-B
	WriteProperty-B	DS-WP-B
Device and Network Management	Dynamic Device Binding-B	DM-DDB-B
	Dynamic Object Binding-B	DM-DOB-B
	DeviceCommunication Control-B	DM-DCC-B

Object Type Support		
Device	Analog Input	Analog Value
Binary Input	Binary Value	
Device does not support CreateObject, DeleteObject, and there are no Proprietary Properties.		

Data Link Layer Options	
Media	**Options**
MS/TP Master	9600,19200, 38400, 57600, 76800

Character Set Support
ANSI X3.4

Ilustración 7. Ejemplo de un PICS de la marca WattMaster.

3. Implementación

En este capítulo se referencia los detalles pertinentes sobre la realización e implementación de los algoritmos.

3.1 Recuperación automática de PICS

Tratamiento previo de la información

Estudiando el modelo vectorial y el booleano, la literatura consultada aconseja realizar un preprocesado del texto en la etapa inicial del proceso, con el fin de poder determinar qué términos pueden utilizarse como términos índices [6].

En esta etapa se realizaron dos pasos:

1. Análisis léxico del texto con el objetivo de determinar el tratamiento que se realizará sobre números, guiones, signos de puntuación, tratamiento de mayúsculas y/o minúsculas, nombres propios, etc. Como parte de este apartado se pasaron todas las palabras a minúsculas, y se omitieron tiras de caracteres que cuentan con más de tres puntos (…), números con punto o guiones que significan numeración (1., 2-, 3.1. etc.), tiras compuestas con solo letras y guiones se dejaron intactas.

2. Eliminación de "stop-words" con el objetivo de reducir el número de términos con valores muy pocos discriminatorios para la recuperación. Para determinar las palabras vacías se realizó una ejecución inicial y se estudiaron las palabras que aparecían en al menos el 95% de los documentos, y se realizó una lista de estas palabras con el fin de filtrarlas en todos los documentos que el sistema vaya a analizar en el futuro. La lista completade palabras se encuentran en el anexo #1.

Detalles de implementación

La implementación consiste en un programa que lea todos los archivos PDF (PICs) de una carpeta y analice palabra por palabra de cada documento, apenas encuentra una palabra nueva la guarda en la base de datos. En dicha tabla cada entrada cuenta con algunas columnas que se utilizan para realizar el cálculo del modelo vectorial y el booleano.

La biblioteca utilizada para la lectura de los PDFs es Apache PDFBox.

En la ilustración 8 se muestran las columnas de la tabla “words”.

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default
ABC word	1	varchar(64)	[v]	[ ]		
123 repetitions	2	int(11)	[v]	[ ]		
123 pics	3	int(11)	[v]	[ ]		
123 vector	4	decimal(10,4)	[ ]	[ ]		0.0000

Ilustración 8. Columnas de la tabla "words".

En esencia cada vez que se encuentra una palabra nueva, se guarda la palabra y se inicializan los demás campos en cero, luego cuando encuentra de nuevo esa palabra, solo actualiza los campos necesarios, a continuación, se hace una breve descripción de los valores:

Repetitions: cuenta cuantas veces se ha encontrado en total la palabra.
Pics: es el identificador del PICS que cuenta con dicha palabra.
Vector: es el cálculo que realiza el modelo vectorial para cuantificar la relevancia de dicha palabra, el cálculo se realiza de esta manera:

$$peso = log(N/n_i)$$

Siendo:
N: número total de documentos en el sistema
n_i: número de documentos en los cuales parece el término k_i

El cálculo de la frecuencia de los términos, no se ha tomado en cuenta, ya que en la implementación particular realizada, resulta más importante la existencia de los términos que la cantidad de veces que aparecen en cada documento.

Como parámetros para la búsqueda de los PICs se establecieron 3: marca, modelo y firmware. Se utilizaron estos criterios porque después de varias pruebas se llegó a la conclusión que era la cantidad mínima necesaria de criterios, con los cuales se podría realizar un reconocimiento positivo de los PICSs.

Como salida para el algoritmo de recuperación automática de PICSs según la información obtenida en el tráfico de red suministrado, se realizó un ajuste, el cual se basa en los siguientes escenarios:

1. Si el método vectorial devuelve resultados con un ranking mayor o igual a 3, se devuelve el resultado con mayor valor.
2. Si el método vectorial devuelve resultados con un ranking menor a 3, se asume como salida que no se encontró resultados.

3.2 Recuperación de la información de los objetos y propiedades implementadas en los dispositivos BACnet

Tratamiento previo de la información

Antes de poder realizar la lectura de los PICSs para obtener sus objetos y propiedades, se necesita homologar el nombre de los objetos y propiedades. Los distintos fabricantes escriben el nombre de los objetos y propiedades de formas diferentes en ciertas ocasiones, causando así problemas de reconocimiento.

Además, se realiza un normalizado entre mayúsculas y minúsculas, tomando como estándar que la primera letra de cada palabra sea mayúscula y las demás minúsculas.

La escogencia del estándar para normalizar el nombre de los objetos se basó en la aparición más común encontrada en el análisis de mas de 200 archivos PICS de los diferentes fabricantes.

El estándar utilizado en el nombre de las propiedades no es el mismo que el utilizado en los objetos, y su razón radica en que al igual que en la escogencia del estándar en los objetos, se basó en la forma más común de encontrar el nombre de las propiedades en los distintos documentos PICS. Ambas escogencias se realizaron luego de analizar los mismos 200 PICS escogidos de manera aleatoria.

Aunque para el análisis de texto en general el algoritmo transforma a todas las palabras en minúscula, el nombre homologado puede expresar en mayúscula algunos caracteres, con el fin de mostrar el nombre del objeto o propiedad formateado de una manera específica según sea el caso en los archivos de salida.

En la tabla 1 se muestra la tabla de homologación de nombre de objetos:

Tabla 1. Tabla de homologación de nombre de los objetos de los PICS.

Posible nombre encontrado	Nombre homologado
Analogue-Input / Analogue Input / Analog_Input	Analog Input
Analogue-Output / Analogue Output / Analog_Output	Analog Output
Analogue-Value / Analogue Value / Analog_Value	Analog Value
Binary_Input	Binary Input
Binary_Lighting_Output	Binary Lighting Output
Binary_Value	Binary Value
BitString_Value	BitString Value
CharacterString_Value	CharacterString Value
Credential_Data_Input	Credential Data Input
Date_Pattern_Value	Date Pattern Value
Date_Value	Date Value
DateTime_Pattern_Value	DateTime Pattern Value
DateTime_Valuet	DateTime Value
Elevator_Group	Elevator Group
Event_Enrollment / Event Enrolment	Event Enrollment
Event_Log	Event Log
Global_Group	Global Group
Integer_Value	Integer Value
Large_Analog_Value	Large Analog Value
Life Safety Point	Life-Safety-Point
Life Safety Zone	Life-Safety-Zone
Load_Control	Load Control
Multistate Input	Multi-state Input
Multistate Output	Multi-state Output
Multistate Value	Multi-state Value
Network_Port	Network Port
Notification_Class	Notification Class
Notification_Forwarder	Notification Forwarder
Octetstring_Value	Octetstring Value
Positive_Integer_Value	Positive Integer Value
Pulse Converter	Pulse-Converter
Structured_View	Structured View
Time_Pattern_Value	Time Pattern Value
Time_Value	Time Value
Trend_Log	Trend Log
Trend_Log_Multiple	Trend Log Multiple

Así que como primer paso lo que se hace es guardar un diccionario con los nombres de los objetos y BACnet, el cual sirve de referencia, y a partir de él se homologaran las diferentes formas de expresar los objetos y propiedades. Las tablas completas de homologación de las propiedades se encuentran en los anexos.

Detalles de implementación

Con el fin de poder leer toda la información requerida, es necesario revisar la estructura de los archivos, con el fin de poder clasificarlos y así aplicar las técnicas necesarias para cada tipo.

Independientemente del tipo de clasificación de cada archivo, se usa una misma estructura de datos para almacenar los objetos y propiedades. La estructura de datos es una lista de objetos, donde cada objeto contiene una lista en donde se guardan sus respectivas propiedades. En la Ilustración 9 se muestra un ejemplo de la estructura de datos.

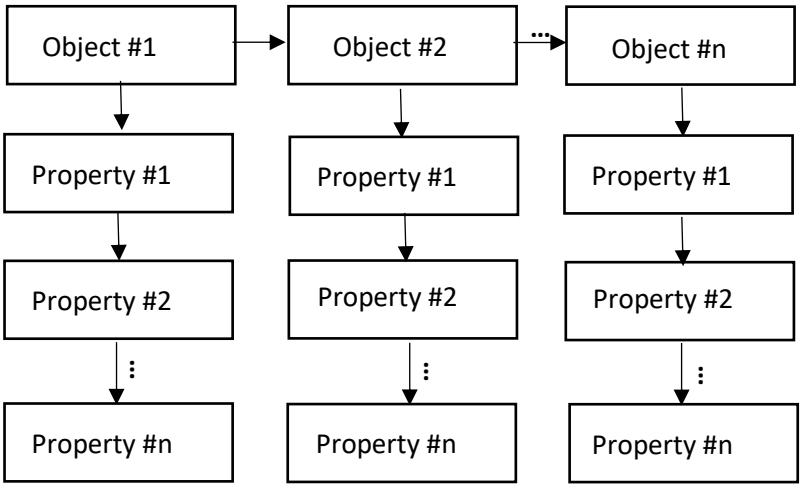

Ilustración 9. Estructura de datos donde se guardan los objetos y propiedades de los PICS.

A continuación, se muestran las clasificaciones hechas:

Clasificación #1

Objetos referenciados por una equis (x), o un check (V), dentro de un cuadro. Entre sus características es que enlistan objetos que no se encuentran en el dispositivo, y la diferenciación se realiza entre los cuadros vacíos, y lo que contienen la equis, o el check. En la Ilustración 10 se muestra un ejemplo de este tipo de documentos:

Object type	Supported
Object#1	[x]
Object#2	[ ]
……..	[x]
Object#n	[ ]

Object type	Supported
Object#1	[√]
Object#2	[ ]
……..	[√]
Object#n	[ ]

Ilustración 10. Ejemplo de aparición de objetos dentro de los archivos PICs bajo el método de clasificación #1.

El algoritmo detecta el símbolo con el cual se está trabajando (equis, o un check), también la posición del símbolo. En algunas tablas existen hasta cuatro columnas referentes a otras propiedades. Independientemente de la cantidad de columnas, el algoritmo solo debe verificar que la primera columna contenga un cuadro, ya sea con algún símbolo dentro o que se encuentre vacía. En la Ilustración 11 se muestra un ejemplo de una tabla con más de dos columnas.

Object type	Supported	Can be created dynamically	Can be deleted dynamically
Object#1	[x]	[ ]	[ ]
Object#2	[ ]	[ ]	[ ]
……..	[x]	[x]	[x]
Object#n	[ ]	[ ]	[ ]

Ilustración 11. Ejemplo de tabla de más de dos columnas en el método de clasificación #1.

Luego de que el algoritmo obtiene el objeto, lo guarda en la estructura de datos (Ilustración 9). Al inicio y por la estructura de este tipo de documentos, se encuentran solo los objetos que contiene cada dispositivo, luego en un apartado posterior, este tipo de documentos muestra el nombre del objeto y enlista sus propiedades. En ese punto se busca el objeto de la lista y se agregan a sus respectivas propiedades. En la Ilustración 12 se muestra un ejemplo de cómo se encuentran enlistadas las propiedades por objeto.

analog-value

☐ Dynamically Creatable ☐ Dynamically Deletable

Property	Read	Write	optional
object-identifier	☑		
object-name	☑	☑	
object-type	☑		
present-value	☑		
status-flags	☑		
event-state	☑		
out-of-service	☑		
units	☑		

binary-input

☐ Dynamically Creatable ☐ Dynamically Deletable

Property	Read	Write	optional
object-identifier	☑		
object-name	☑		
object-type	☑		
present-value	☑	☑	
status-flags	☑		
event-state	☑		
out-of-service	☑		
polarity	☑		
inactive-text	☑		☑
active-text	☑		☑

Ilustración 12. Ejemplo de la aparición de las propiedades en el método de clasificación #1.

Cuando se alcanza el final del documento, se revisa que la cantidad de nombres de objetos acompañados de sus respectivas propiedades sea la misma que la existente en la lista, de no ser así, el objeto que no cuente con sus propiedades se le carga las propiedades mínimas que BACnet dictamina que ese objeto deba contener.

Clasificación #2

Objetos que se encuentran dentro de una tabla, y dentro de la misma (lado derecho) se encuentran sus propiedades. En la Ilustración 13 se muestra un ejemplo de este tipo de documentos:

Object Type	Properties	
Object #1	property #1 property #2 property #3 property #4	property #5 property #6 property #7 property #8
Object #2	property #1 property #2 property #3 property #4	property #5 property #6 property #7 property #8

Ilustración 13. Ejemplo de aparición de objetos dentro de los archivos PICs bajo el sistema de clasificación #2.

La particularidad de este tipo de documento es que cuando se recorra el archivo, el algoritmo va a encontrar primero algunas propiedades antes que el nombre del objeto. En este caso, cuando se encuentren nombres de propiedades, el algoritmo las guarda en una lista temporal, y apenas se encuentre el nombre de un objeto lo inserta en la misma estructura de datos usada en el método de clasificación #1, y le agrega en su lista de propiedades la existentes en la lista temporal.

En el momento que encuentre el nombre de una propiedad que ya insertó para el objeto actual, el algoritmo supone que se encontró la primera propiedad del siguiente objeto, y desde ese punto comienza a guardar las propiedades en la lista temporal, hasta que de nuevo encuentre el nombre de un nuevo objeto.

Clasificación #3

Objetos que se encuentran enlistados, y debajo del nombre del objeto se encuentra una tabla con sus propiedades. En la Ilustración 14 se aprecia un ejemplo de este tipo de documentos:

- Object #1

Properties name	Required	Optional	Writable
object_identifier	√		
object_name	√		
object_type	√		
reliability		√	√
event_state	√	√	
profile_name		√	
profile_location		√	

Ilustración 14. Ejemplo de aparición de objetos dentro de los archivos PICs bajo el sistema de clasificación #3.

Este caso es una versión simplificada del método #1, ya que no se enlistan primero los objetos, y luego aparecen con sus respectivas propiedades. El algoritmo cuando encuentra el nombre de un objeto lo inserta en la lista, y comienza a guardar los nombres de las propiedades que encuentre hasta que aparezca el nombre del siguiente objeto o un nuevo título.

Clasificación #4

Es el modo genérico de tratar los documentos, en este caso se espera el nombre del objeto, y luego se enlistan sus propiedades, no necesariamente dentro de una tabla.

Como característica esta clasificación se cuenta que, para llegar a este punto, el sistema tuvo que realizar dos verificaciones del documento, y deberá realizar una tercera para extraer la información requerida.

A continuación, se describen las verificaciones al documento realizadas:

1. Análisis previo de la estructura del documento, es con el fin de deducir si encaja en alguna de las 3 clasificaciones anteriores.
2. Búsqueda del índice que especifiquen la localización de la información requerida.
3. Recorrido del documento como método para buscar indicios de la estructura. La técnica consiste en leer el documento buscando ciertos aspectos claves que pueden ayudar a deducir donde se encuentran los objetos. Por ejemplo, ubicándolos debajo de algún título especial, ejemplo “Object Type Support”, o palabras que coincidan con los nombres de objetos y propiedades juntas.

En la Ilustración 15 se da un ejemplo de este tipo de documentos:

- Object #1

1 Creatable? NO
2 Deletable? NO
3 Optional properties supported:
    acked_transitions
    cov_increment
    deadband
    description
    device_type
    event_enable
    event_time_stamps
4 Writeable properties:
    cov_increment
    deadband
    description
    device_type

notification_class
5 Proprietary properties: None
6 Range restrictions:
description: limited to 50 octets in length
device_type: limited to 50 octets in length

Ilustración 15 Ejemplo de aparición de objetos dentro de los archivos PICs

A diferencia de los métodos anteriores, en muchas ocasiones estos documentos carecen de una índice o tabla de contenidos, el cual ayuda a delimitar por títulos hasta donde hay que comenzar o terminar de leer o seguir esperando objetos o propiedades.

Si el algoritmo encuentra el nombre de un objeto lo guarda en una lista temporal, y si luego encuentra el nombre de una propiedad agrega el objeto y las propiedades que encuentre en la estructura de datos existente para esto.

Si se encuentra otro nombre de objeto y ya la lista temporal contiene otro objeto y ninguna propiedad, reemplaza el objeto guardado con el nuevo en espera de sus propiedades.

En este caso el algoritmo continua la búsqueda de objetos y propiedades hasta el final del archivo, teniendo como verificación, que, si se encuentra el nombre de un objeto ya procesado, lo ignora y no limpia la lista temporal.

3.4 Interfaz gráfica

Detalles de implementación

Ya que los algoritmos anteriormente presentados se desarrollaron en el lenguaje JAVA, para minimizar cualquier posible barrera por uso de diferentes lenguajes de programación, la interfaz web fue desarrollada en el lenguaje JSP, y con el servidor web Tomcat versión 9.0.55.

A nivel de funcionabilidad, la interfaz gráfica se compone de dos partes: la primera en una ventana donde el usuario puede subir el tráfico de red, y segunda donde el sistema luego de procesar el tráfico de red muestra la lista de PICSs que se relacionan a los dispositivos encontrados.

Como parte principal, se ha ponderado la usabilidad y fácil acceso a los datos, así que la interfaz se ha implementado de manera que sea de la forma más simple y funcional posible.

En la Ilustración 16 se muestra la interfaz de la pantalla donde el usuario debe subir el tráfico de red.

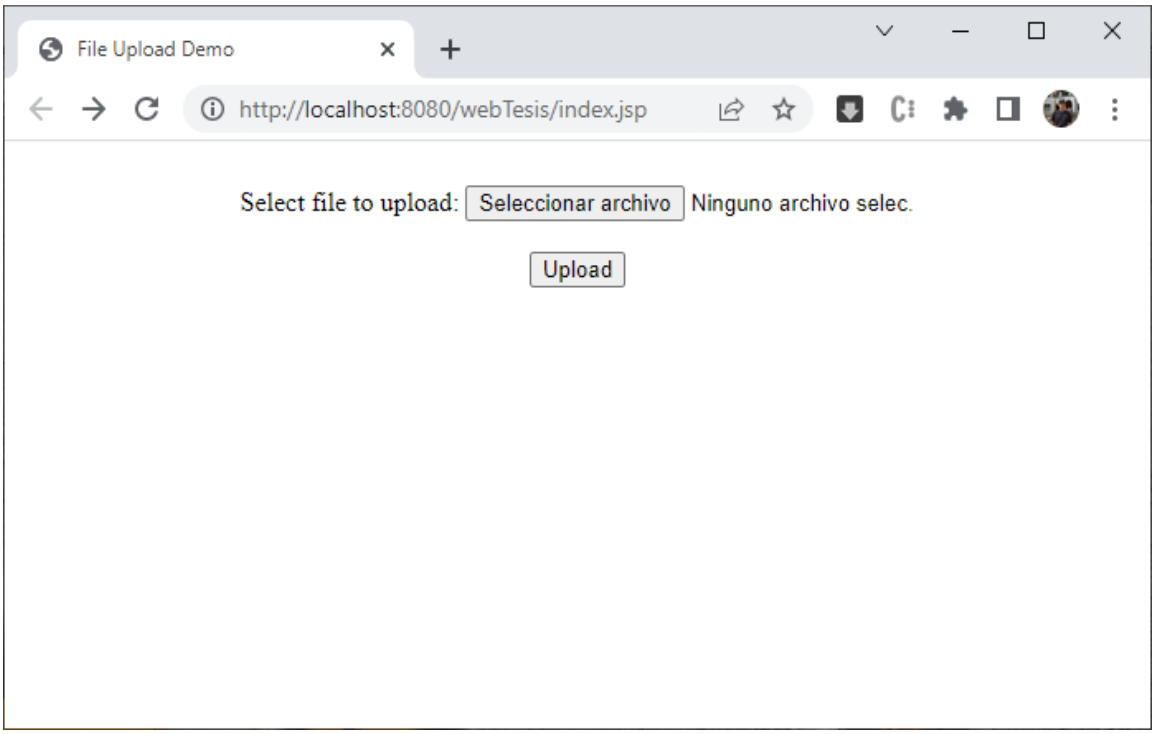

Ilustración 16. Pantalla principal de la interfaz gráfica.

Luego en la segunda y final parte de la interfaz, el sistema muestra el listado de los PICSs encontrados, mostrando una descripción de estos, además en las dos últimas columnas le da al usuario la posibilidad de poder descargar el archivo PDF del correspondiente PICS, además el archivo de texto que contiene la extracción de los objetos y propiedades del PICS, en la Ilustración 17 se muestra un ejemplo.

File Analysis: plugfest-delta-2b.pcap

Model name	Vendor name	Firmware revision	Ranking	PICS	Processed objects
SPAK MSTP	Lithonia Lighting	SY20V200	4.3354	BTL-Listing-Synergy-MSTP.pdf	BTL-Listing-Synergy-MSTP.pdf.txt
SYSC MLX	Lithonia Lighting, Inc.	2x65j	4.8132	SynergyPICS-MLX.pdf	SynergyPICS-MLX.pdf.txt
DNT-T221	Delta Controls	0028	3.4858	BTL Listing_LU18013_Delta_BACStatII_03272008.pdf	BTL Listing_LU18013_Delta_BACStatII_03272008.pdf.txt
DSC-1616E	Delta Controls	15604	3.1179	BTL_Listing_23809_Delta_entelliBUS.pdf	BTL_Listing_23809_Delta_entelliBUS.pdf.txt
DAC-T305	Delta Controls	15604	3.1179	BTL_Listing_23816_Delta_DAC-AAC.pdf	BTL_Listing_23816_Delta_DAC-AAC.pdf.txt
DNT-T103	Delta Controls	3319	3.4858	BTL Listing_LU18013_Delta_BACStatII_03272008.pdf	BTL Listing_LU18013_Delta_BACStatII_03272008.pdf.txt
Total PICSs: 6					Go back!

Ilustración 17. Página que muestra el análisis del tráfico de red.

Al archivo de texto que contiene la extracción de los objetos y propiedades de cada PICS se le proporcionó un formato básico y simple de leer con el fin de que pueda ser fácilmente manipulado. También para su fácil búsqueda y referencia cada archivo de texto se llama igual

que el archivo PDF que representa, por ejemplo, para el PICS representado en el archivo PDF "DESIGO PICS V5.10.pdf", el archivo de texto se llama "DESIGO PICS V5.10.pdf.txt"

A continuación, en la Ilustración 18, se muestra un extracto del contenido de un archivo:

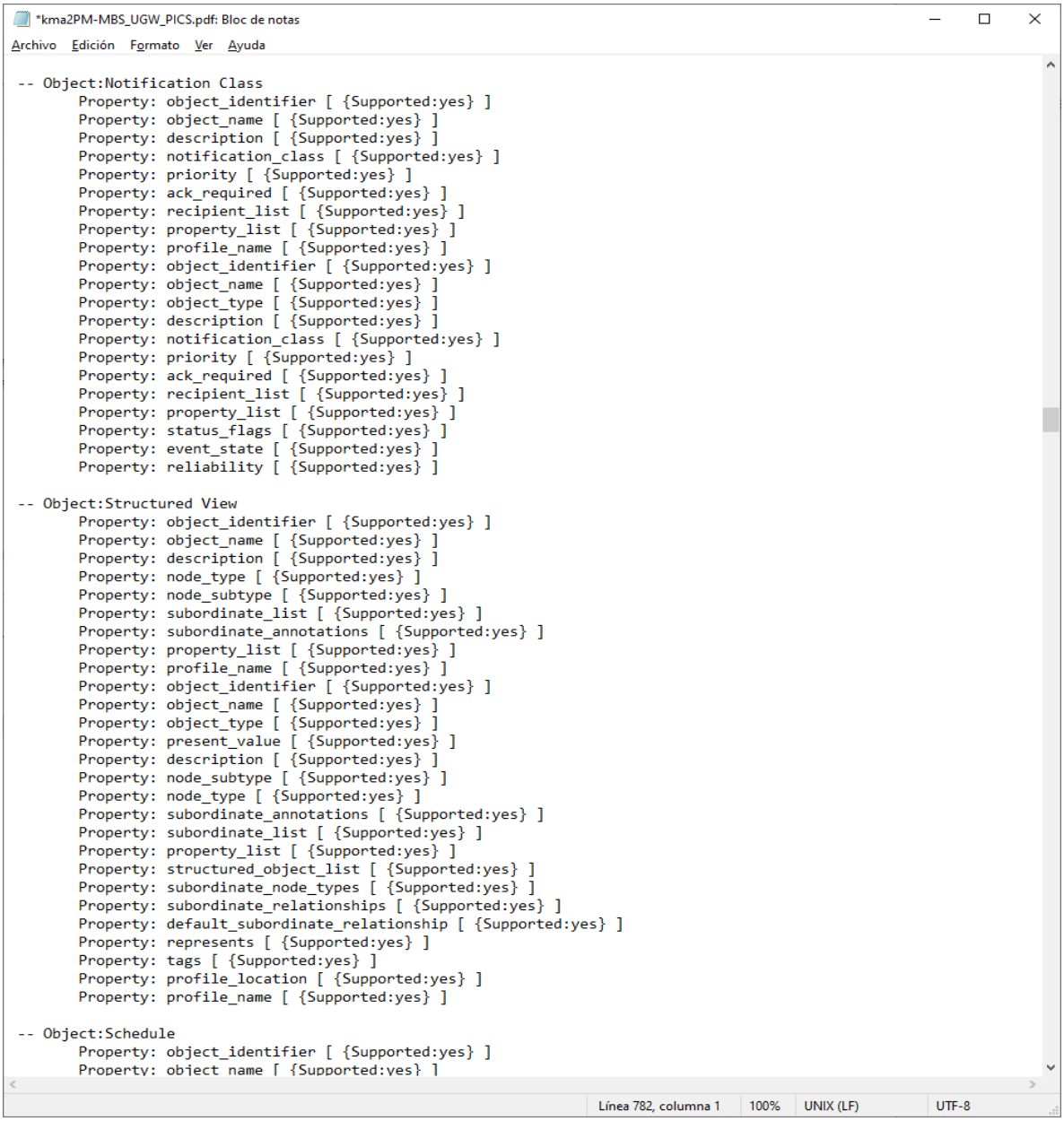

Ilustración 18. Contenido parcial que refleja la extracción de objetos y propiedades de un PICS.

A nivel funcional la interfaz cumple el objetivo de manera completa, además contiene todos los componentes necesarios para realizar su función sin permitir confusiones acerca de su manipulación.

Aparte de procesar el tráfico de red, se guardan todos los archivos de tráfico de red en una carpeta de nombre y ruta configurable por el administrador del sistema, por cualquier tipo de estudio o posterior análisis si se requiere.

Las interacciones entre el usuario y el sistema son únicamente las necesarias con el fin de resultar intuitivo. En la primera ventana solo se cuenta con dos botones:

- Seleccionar archivo: abre un explorar de archivos para escoger el archivo a subir.
- Subir archivo: inicia el proceso de procesamiento del archivo.

La ventana final que muestra el resultado del procesamiento del archivo de trafico de red, muestra toda la información relativa de los PICSs, además interactuar con el usuario dándole la posibilidad de descargar el PICS o el archivo de texto que contiene los objetos y propiedades obtenidos del PICS seleccionado, y por último al final de la página, con un botón de “volver atrás”, por si se desea subir un nuevo archivo de trafico de red.

4. Evaluación

En este capítulo se presenta cuáles fueron los resultados obtenidos luego de la realización de cada uno de los objetivos específicos que dan como resultado el cumplimiento del objetivo general y razón de ser del proyecto.

Con el fin de contar con el conjunto más completo de PICSs, se descargaron todos los archivos que se encontraban listados en la página de https://www.bacnetinternational.org/, la descarga se realizó el 2 de junio del 2021, de manera automática con la herramienta wget [16]. El universo total de PICSs con los que se cuentan son 2504.

Es importante señalar que la pruebas realizas a cada algoritmo fueran hechas de manera separada. La razón de dicha separación es la naturaleza de cada algoritmo. Para la recuperación automática de PICSs se realizó la búsqueda de 100 PICSs de manera aleatoria. Para la extracción de objetos y propiedades de los PICSs en este caso se examinaron 50 PICSs. Finalmente, para la comparación del algoritmo de extracción de objetos y propiedades con el estado del arte, se realizaron dos pruebas: una con el tráfico de red incluida y la otra con un muestreo aleatorio de 50 PICSs.

A continuación, se describe a profundidad los resultados que han sido encontrados.

4.1 Recuperación automática de PICS

Este apartado se encuentra relacionado directamente con el objetivo #1, el cual se refiere a: la recuperación automática de PICS según la información obtenida en el tráfico de red suministrado.

Resultados obtenidos

Después de la formación de los índices en la base de datos, se creó un programa en Java, que a partir de los PICSs contenidos en la carpeta, elije 100 a azar, y escribe su nombre en un archivo de texto. Posteriormente se utilizaron esos nombres con el fin de comprobar el nivel de precisión de búsqueda bajo la implementación de los modelos vectorial y el booleano.

Después de realizar las 100 pruebas se determina que si se realiza la búsqueda con los datos escritos de manera correcta ambos modelos (vectorial y el booleano) como primer resultado devuelven el resultado correcto.

Como punto importante se determina que si se altera alguno de los parámetros el modelo booleano no devuelve ningún resultado (como es de esperarse), pero esta característica se

convierte en una desventaja para la recuperación de los PICS, ya que en el tráfico de red se pueden presentar casos que la versión del firmware no esté disponible, o sea diferente, y aunque se cuenten con dos de los tres criterios de búsqueda, los resultados estarán vacíos.

A modo de ejemplo se muestra una búsqueda realizada de un PICS de la marca DEOS, sus criterios de búsqueda son:

Marca:	DEOS
Modelo:	Air
Firmware:	1.040

En la Ilustración 19 se muestran los resultados de ambos modelos:

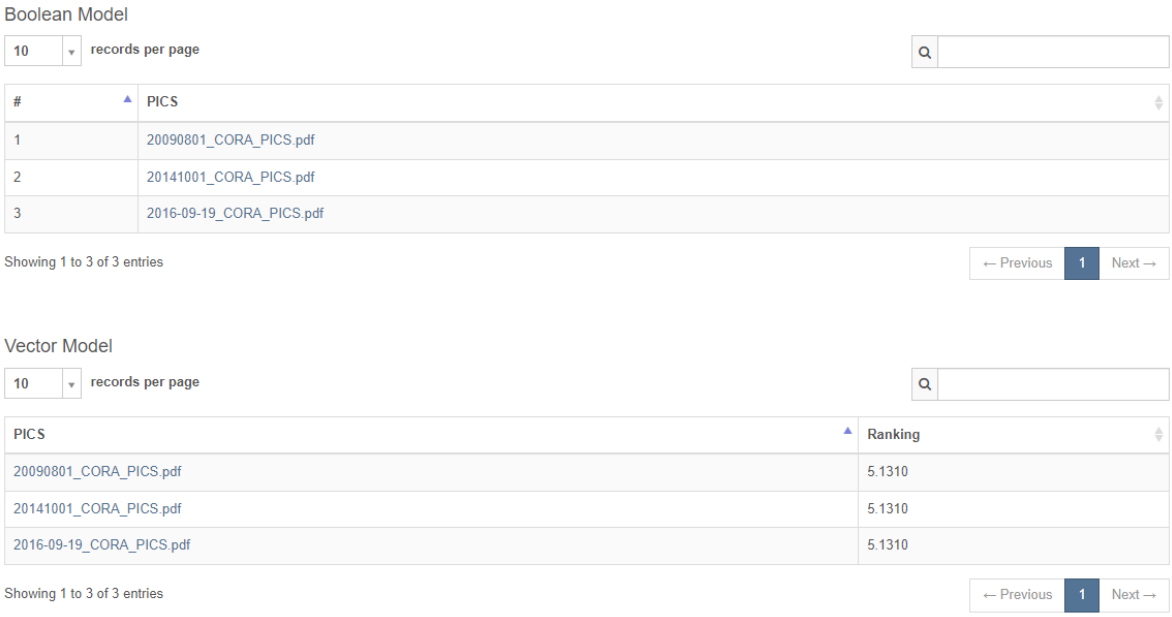

Ilustración 19. Ejemplo de una búsqueda de PICSs, bajo el modelo booleano y el vectorial.

Es importante señalar que en varios casos las empresas hacer revisiones posteriores de un mismo equipo, generando en el repositorio varios PICS que se refieren al mismo equipo, así que, en el caso particular, la aparición de tres documentos cuando se realiza la búsqueda no se considera como error o falta de precisión.

En el siguiente ejemplo se modificó negativamente el modelo del dispositivo, cambiando "Air" por "the", nótese que el cambio no representa ninguna palabra raíz o relevante en el criterio de búsqueda. Los resultados se aprecian en la Ilustración *20*.

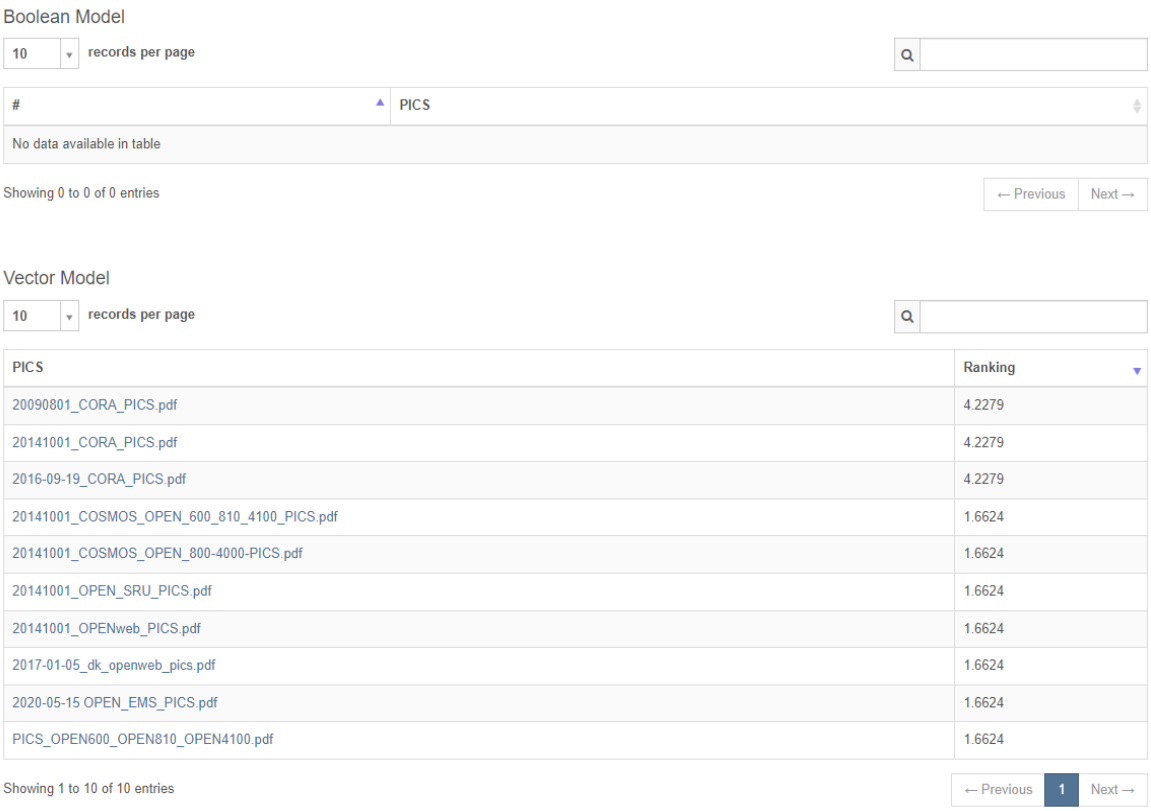

Ilustración 20. Ejemplo de una búsqueda de PICs, bajo el modelo booleano y el vectorial, con afectación negativa en los criterios de búsqueda.

En este caso el modelo booleano no devuelve ningún resultado, en cambio el modelo vectorial logra posicionar como el mejor resultado el resultado correcto. Aparte los tres resultados que se pueden catalogar como correctos cuentan con una ponderación notablemente más alta que el resto de los valores.

En resumen, en las 100 pruebas realizadas al azar, el 100% de las búsquedas ponderaban como primer resultado el correcto en el modelo vectorial, aunque se cambiara uno de sus criterios de búsqueda, y en el modelo booleano solo cuando los criterios eran escritos de manera exacta.

Después de los resultados obtenidos de ambos métodos, se optó por usar solo el método vectorial como algoritmo de búsqueda. Y se concluye una tasa de éxito de un 100%.

Precision obtenida:

Como premisa inicial se toma que se debe recuperar un sólo documento, y éste fue correcto, el cálculo de la precisión es el siguiente:

$$\text{Precision} = \frac{|\{\text{documentos relevantes}\} \cap \{\text{documentos recuperados}\}|}{|\{\text{documentos recuperados}\}|}$$

Precisión = 1/1 = 1

Exhaustividad obtenida:

Tomando en cuenta que la cantidad de documentos relevantes devueltos es exactamente uno, y este fue el correcto el cálculo de la exhaustividad es el siguiente:

$$\text{Exhaustividad} = \frac{|\{\text{documentos relevantes}\} \cap \{\text{documentos recuperados}\}|}{|\{\text{documentos relevantes}\}|}$$

Exhaustividad = 1/1 = 1

4.2 Recuperación de la información de los objetos y propiedades implementadas en los dispositivos BACnet

Este apartado se relaciona directamente con el objetivo #2 del proyecto.

Resultados obtenidos

Por el tamaño y el tiempo requerido para realizar las pruebas, se realizaron muestreos en 50 documentos escogidos de manera aleatoria, y se verificó tanto la recuperación de los objetos como de sus propiedades, a continuación, se muestran los resultados:

Objetos

Cantidad de objetos que se debían encontrar 723:
Cantidad de objetos encontrados: 729
En este caso específico se obtuvo una precisión de 0.99177.

Es importante señalar que el sistema encontró 729 objetos, encontró todos los que tenía que encontrar, pero además agregó el objeto “Network Security” erróneamente en 6 casos, el sistema interpretó las palabras “Network Security”, y no supo diferenciar que se trataba de la palabra en una oración y no en un listado de objetos. El error se encontró en el método de clasificación #4 (el método genérico).

Propiedades

Cantidad de propiedades que se debían encontrar 17 328.
Cantidad de propiedades encontradas: 17 328.
Nivel de efectividad: 100%

4.3 Comparación del nivel de precisión y exhaustividad del algoritmo de la extracción de la información de los dispositivos BACnet con el estado del arte.

Se realizaron dos tipos diferentes de pruebas, ya que después de analizar los resultados, se constató que cuando se analizó el tráfico de red de una misma empresa, la probabilidad de que la mayoría de los dispositivos sean de una misma marca es muy alta. Esta particularidad genera que muchos dispositivos referencien a un mismo PICS, provocando un análisis de muchos gigabytes de información, pero de PICSs de un solo fabricante.

La primera prueba consiste en la búsqueda de los PICSs mediante el análisis de tráfico de red, y a partir de los PICSs recuperados se realiza su correspondiente extracción de objetos y propiedades. La segunda prueba corresponde a la escogencia aleatoria de 50 PICSs, y su correspondiente extracción de objetos y propiedades. Ambas pruebas obtuvieron una efectividad del 100%.

Prueba #1

Las pruebas se realizaron con archivos de tráfico real, concretamente se utilizaron 8 archivos de extensión PCAP. Los archivos en conjunto contaban con un tamaño de 9.82 gigabytes.

Con el propósito de poder llevar un registro detallado tanto de los PICSs recuperados, así como los aciertos y fallos obtenidos, se crea una rutina en el lenguaje Java con el fin de que automatice las consultas, y registre todos los datos necesarios para su posterior estudio.

Como salida por cada lectura se obtiene el modelo, vendedor y versión del firmware de cada dispositivo, y el PICS recuperado después de usar estos criterios para su búsqueda.

En la Ilustración 21 se muestra un ejemplo de cada salida realizada por el programa:

```
| Device id: [redacted] | model_name: SYSC MLX | vendor_name: Lithonia Lighting, Inc. | firmware_revision: 2x65j|
| Criterio:  | model_name: sysc | vendor_name: lithonia | firmware_revision: 2x65j|
| Device id: *[redacted]* PDF: *SynergyPICS-MLX.pdf Value: 9.2312*|
```

Ilustración 21. Ejemplo de salida para un dispositivo buscado en el repositorio de PICSs.

En total se encontraron 200 dispositivos que referenciaban a 9 PICSs diferentes, todos del mismo fabricante. Se obtuvo un nivel de recuperación de objetos y propiedades del 100%, y un nivel de recuperación de los PICSs basado en el tráfico de red también de un 100%.

Prueba #2

Esta prueba tiene como objetivo, hacer una búsqueda entre los diferentes documentos realizados por los fabricantes. Es importante mencionar que sobre los 2 504 PICs existentes en el repositorio oficial, ante la imposibilidad de tiempo para examinarlos todos de manera manual se realizó un muestreo aleatorio de 50 PICSs.

Con respecto esos 50 PICSs se encontraron un total de 723 objetos y 17 328 propiedades.

En la tabla 2 se realiza la comparación de los resultados obtenidos con el nuevo algoritmo de recuperación de objetos y propiedades con el estado del arte.

Tabla 2. Comparación de los resultados obtenidos con el nuevo algoritmo de recuperación de objetos y propiedades con respecto al estado del arte.

	Recuperación de objetos y propiedades	Porcentaje de recuperación de objetos y propiedades	Recuperación de los PICSs basado en el tráfico de red.	Porcentaje de recuperación de los PICSs basado en el tráfico de red.
Estado del arte	-136 objetos. -2 064 propiedades.	-100% -99.85%	-10 PICSs. -4.5 gigabytes analizados.	100%
Algoritmo implementado	-723 objetos. -1 7328 propiedades	-99,17% -100%	-200 PICSs. -9.82 gigabytes analizados.	100%

A nivel de conclusión general se considera que se cumplió el objetivo de alcanzar al menos el nivel de precisión y recuperación del nuevo algoritmo con respecto al estado del arte.

5. Conclusiones y trabajo futuro

En este capítulo se resumen los aspectos claves del trabajo realizado, así como los principales hallazgos y recomendaciones por si se decidiera avanzar sobre esta línea de trabajo.

5.1 Conclusiones

La presente investigación cumplió los objetivos impuestos y contestó las preguntas de investigación, las siguientes son las conclusiones que dan respuesta a las preguntas planteadas:

1. Para recuperar los objetos y propiedades de los documentos PICSs automáticamente, una de las posibles soluciones es aplicar dos técnicas diferentes. Primero agrupar los documentos PDFs a procesar según su estructura y utilizar un método para cada estructura en particular. Segundo para los documentos que de manera general no se pueden agrupar, se puede utilizar un método por aproximación. El método consiste en leer el documento buscando ciertos aspectos claves que pueden ayudar a deducir donde se encuentran los objetos, ubicándolos debajo de algún título especial o palabras que coincidan con los nombres de objetos y propiedades juntas; sirviendo esto como un indicio que en qué parte del documento se encuentra la información deseada.

2. Apoyarse en buenas prácticas de calidad del software como por ejemplo la usabilidad, facilitan la adopción de un sistema por parte de los usuarios haciéndolo más intuitivo de manipular.

3. El uso del método vectorial en lugar del booleano como método de recuperación de la información es más beneficioso cuando se hace uso del tráfico de red para obtener los criterios de búsqueda; ya que éstos no siempre son exactos. Por ejemplo, en el tráfico de red se puede encontrar el nombre del vendedor y modelo del dispositivo de manera exacta, pero una versión de firmware distinta al existente en el PICS, de esta manera la búsqueda se debe basar en la mejor aproximación y no en búsqueda exacta.

4. Asignar un peso mayor al criterio "*nombre del vendedor*" puede mejorar la calidad de los resultados de las búsquedas.

5. Automatizar los procesos de ejecución de un software al máximo, disminuye la posibilidad del error humano en la manipulación de mismo y mejora la usabilidad del sistema [17].

5.2 Trabajo futuro

La investigación desarrollada logró cumplir los objetivos planteados en esta tesis, no obstante, siempre existen áreas que se pueden desarrollar para futuros trabajos. A continuación, una lista de posibles mejoras que se podrán trabajar en el futuro:

1. Extracción de objetos y propiedades de los PICS: Aunque el algoritmo propuesto tiene una tasa de éxito del 100%, puede necesitar recorrer hasta 3 veces el archivo, en busca de algún patrón para realizar la extracción de manera correcta. En estos casos, aunque el algoritmo es efectivo, se considera importante aplicar técnicas de reconocimiento de entidades nombradas (NER por sus siglas en inglés) para generalizar la extracción sin tener que clasificar los documentos previamente.

2. Extracción de objetos y propiedades de los PICS: Por razones de tiempo se realizaron pruebas aleatorias para comprobar la efectividad del algoritmo, en promedio se necesitó 20 horas de trabajo para verificar 50 PICSs. Tomando en cuenta el tamaño del universo a la hora de efectuar este trabajo (2504 PICSs en el repositorio oficial), se considera recomendable buscar la manera de aumentar el tamaño de la muestra, en al menos un 20% del universo.

3. Interfaz gráfica: Aunque la interfaz es completamente funcional, con el potencial del trabajo realizado, se considera que se puede aumentar sus funcionabilidades en aspectos como reportes estadísticos, o manipulación en el lado del sistema, como la reconstrucción de los índices de búsquedas de los PICs, incorporación de nuevos PICSs al repositorio, entre otros.

6. Anexos

Anexo #1. Lista completa de stop-words

2.5
232
878.1
878.1
10646
(annex
(b-aac)
(b-asc)
(b-bc)
(bbmd)
(b-ows)
(b-sa)
(b-ss)
(clause
(ucs-2)
(ucs-4)
8802-3
8802-3
8859-1
actuator
address
advanced
all
analog
and
annex
application
arcnet
are
bacnet
bacnet/ip
baud
bbmd
be
binary
binding
blocks
broadcast
building
by
can
character
clause
communication
conformance
controller
data
dbcs
description
device
devices?
dm-dcc-b
dm-ddb-b
dm-dob-b
does
ds-rp-b
eia
ethernet
firmware
for
foreign
h,
h
if
implementatio
n
imply
in
indicating
input
interoperability
ip
is
iso
j),
j)
jis
layer
link
list
lontalk
management
master
model
modem
ms/tp
multiple
network
networking
no
not
object
of
operator
optional
other
output
over
point-to-point
product
profile
properties
property
protocol
registrations
router
segmentation
segmented
sensor
sets
simultaneously.
slave

smart
software
specific
standard
standardized
statement
static
support
supported
supported?
that
the
they
this
to
tunneling
type
types
value
vendor
window
with
workstation
writable
yes

Anexo #2. Tabla de homologaciones de los nombres de las propiedades de los PICS.

Posible nombre encontrado	Nombre homologado
absentee-limit	absentee_limit
acked-transitions	acked_transitions
activation-time	activation_time
active cov subscriptions	active_cov_subscriptions
active text	active_text
authentication-factors	authentication_factors
belongs-to	belongs_to
buffer-size	buffer_size
client-cov-increment	client_cov_increment
cov increment	COV_Increment
credential-disable	credential_disable
credential-status	credential_status
event-enable	event_enable
event-state	event_state
event-time-stamps	event_time_stamps
expiry-time	expiry_time
global-identifier	global_identifier
inactive text	inactive_text
last-access-event	last_access_event
last-access-point	last_access_point
last-notify-record	last_notify_record
last-use-time	last_use_time
log-buffer	log_buffer
log-device-object-property	log_device_object_property
log-interval	log_interval
max info frames	Max_Info_Frames
max master	Max_Master
multi state	multi-state
multistate-input	multistate input
multistate-output	multistate output
multistate-value	multistate value
notification-class	notification_class
notification-threshold	notification_threshold
notification-threshold	notification_threshold
notify§-type	notify§_type
object identifier	Object_Identifier

object-identifier	object_identifier
object name	Object_Name
object-name	Object_Name
object-type	object_type
present value	Present_Value
profile name	Profile_Name
property list	property_list
reason-for-disable	reason_for_disable
record-count	record_count
records-since-notification	records_since_notification
records-since-notification	records_since_notification
relinquish default	Relinquish_Default
state text	state_text
status-flags	status_flags
stop-time	stop_time
stop-when-full	stop_when_full
total-record-count	total_record_count
trace-flag	trace_flag
uses-remaining	uses_remaining

Anexo #3. Comparación de herramientas para la extracción de información en archivos en formato PDF.

Nombre	Licencia	Breve descripción	Lenguaje
Apache PDFBox	Apache	Biblioteca de desarrollador de Java para crear, ver, extraer e imprimir archivos PDF.	Java
Biblioteca de Adobe PDF	Propietaria	C ++, .NET, API Java con soporte para edición, visualización, impresión y extracción de texto de PDF.	C, Java, .Net
Camelot	MIT	Biblioteca que facilita la extracción de tablas de archivos PDF.	Python
iText	Propietario / AGPL	Biblioteca para crear y manipular archivos PDF, RTF y HTML.	Java, C#
JPedal	Propietario / GNU LGPL	Biblioteca de desarrollador de Java para ver, extraer e imprimir archivos PDF.	Java
jPDFText	Licencias por núcleo	Biblioteca que le permite procesar documentos para extraer el contenido textual de los archivos PDF.	Java
OpenPDF	GNU LGPLv3 / MPLv2.0	Biblioteca de código abierto para crear y manipular archivos PDF. Bifurcación de una versión anterior de iText, pero con la licencia LGPL / MPL original.	Java
Proyecto BIRT	Licencia pública de Eclipse (EPL)	De código abierto basado en Java Business Intelligence and Reporting Tools (BIRT) que puede crear una salida en PDF, HTML, Web Viewer, Microsoft XLS, XLSX, DOC, DOCX, PPT, PPTX, ODT, ODS, PAO, Postscript, valores separados por comas y archivos XML y pueden integrarse en sitios web o ampliarse para formatos individuales y salida de base de datos.	Java
PDFsharp	MIT	Biblioteca de desarrollador para crear, extraer y editar archivos PDF.	C#
Podofo	GNU LGPL	Biblioteca de código abierto para leer y escribir archivos PDF.	C ++

Referencias

[1] H. Esquivel-Vargas, M. Caselli, and A. Peter, “Automatic deployment of specification-based intrusion detection in the BACnet Protocol,” *CPS-SPC 2017 - Proc. 2017 Work. Cyber-Physical Syst. Secur. PrivaCy, co-located with CCS 2017*, pp. 25–36, 2017, doi: 10.1145/3140241.3140244.

[2] bacnet.org, “Vendor Gallery.” http://www.bacnet.org/Gallery/ (accessed Mar. 06, 2022).

[3] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, no. July 1999. 1999.

[4] G. H. Tolosa and F. R. a. Bordignon, “Introducción a la Recuperación de Información Conceptos , modelos y algoritmos básicos,” *Univ. Nac. Luján, Argentina*, pp. 1–149, 2008.

[5] J. Araya, “Apuntes de clase. Recuperacion textual de la información.” Cartago, 2006.

[6] Á. F. Z. Rodríguez, G. Carlos, F. Paniagua, J. L. A. Berrocal, and R. G. Díaz, “Recuperación de información utilizando el modelo vectorial. Participación en el taller CLEF- 2001,” 2002.

[7] F. J. Martínez Méndez, *Recuperación de información: modelos, sistemas y evaluaciones*, vol. 1. 2004.

[8] Adobe, “PDF: tres letras que continúan cambiando el mundo,” 2020. https://acrobat.adobe.com/es/es/acrobat/about-adobe-pdf.html (accessed Jun. 13, 2021).

[9] MDN, “Introducción a XML - XML: Extensible Markup Language | MDN,” 2021. https://developer.mozilla.org/es/docs/Web/XML/XML_introduction (accessed Apr. 25, 2022).

[10] E. Extract and S. Demo, “PDF Scraping : Guía para extraer datos no estructurados de PDF,” 2021. https://www.astera.com/es/type/blog/pdf-scraping/ (accessed Jun. 13, 2021).

[11] Apache Software Foundation, “Apache PDFBox | A Java PDF Library.” 2014, [Online]. Available: http://pdfbox.apache.org/.

[12] ASHRAE, “ASHRAE 135-2020,” 135–2020, 2020.

[13] “en-RED-ando con la HISTORIA.” https://diarium.usal.es/enredandoconlahistoria/ (accessed Jun. 13, 2021).

[14] BACnet, “BACnet,” 2021. http://www.bacnet.org/ (accessed Jun. 13, 2021).

[15] ASHRAE SSPC 135, “PICS.” http://www.bacnet.org/DL-Docs/ (accessed Apr. 25, 2022).

[16] GNU, “GNU wget,” *GNU wget*, 2003. https://www.gnu.org/software/wget/manual/html_node/Contributors.html (accessed Feb. 17, 2022).

[17] ISO/IEC 25000, “Usabilidad,” *2020*, 2020. https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad (accessed Apr. 10, 2022).