

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“DESARROLLO DE UN SISTEMA QUE PERMITA REALIZAR
CONSULTAS BASADAS EN CONTENIDO EN
UNA BASE DE DATOS MULTIMEDIA”.

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN SISTEMAS
COMPUTACIONALES

PRESENTA:

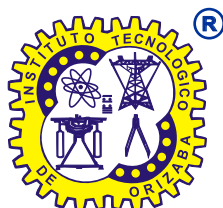
I.S.C. Rodolfo Rojas Ruiz

DIRECTOR DE TESIS:

Dra. Lisbeth Rodríguez Mazahua

CODIRECTOR DE TESIS:

Dr. Asdrúbal López Chau





FECHA: 17/04/2018
DEPENDENCIA: POSGRADO
ASUNTO: Autorización de Impresión
OPCIÓN: I

C. RODOLFO ROJAS RUIZ
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"DESARROLLO DE UN SISTEMA QUE PERMITA REALIZAR CONSULTAS BASADAS EN CONTENIDO EN UNA BASE DE DATOS MULTIMEDIA".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E

RUBEN POSADA GOMEZ
JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO

C.A. TITULACIÓN



SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO
TECNOLÓGICO
DE ORIZABA

ggc



SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



Subsecretaría de Educación Superior
Tecnológico Nacional de México
Instituto Tecnológico de Orizaba

FECHA : 01/03/2018

ASUNTO: **Revisión de Trabajo Escrito**

C. RUBEN POSADA GOMEZ

JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

RODOLFO ROJAS RUIZ

la cual lleva el título de:

"DESARROLLO DE UN SISTEMA QUE PERMITA REALIZAR CONSULTAS BASADAS EN CONTENIDO EN UNA BASE DE DATOS MULTIMEDIA".

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : **DRA. LISBETH RODRIGUEZ MAZAHUA**

FIRMA

SECRETARIO : **DR. ASDRUBAL LOPEZ CHAU**

FIRMA

VOCAL : **M.C. SILVESTRE GUSTAVO SERGIO PELAEZ**

FIRMA

VOCAL SUP. : **M.C. MARIA ANTONIETA ABUD FIGUEROA**

FIRMA

EGRESADO(A) DE LA MAESTRIA EN **SISTEMAS COMPUTACIONALES**

OPCION: **I Tesis**



Agradecimientos

Considero un gran privilegio el haber tenido la oportunidad de pertenecer a este maravilloso sistema de posgrado. Le doy gracias a mis padres, por apoyarme incondicionalmente, así como a mi pareja Fabiola, quien estuvo conmigo a pesar de todo.

Agradezco al comité de la maestría, a la maestra Celia por su paciencia y apoyo, al maestro Luis, quien me apoyó mucho tanto en el plano académico como en el personal y por supuesto, al excelente claustro de profesores por su exigencia, compromiso y dedicación, ya que sin ellos esta experiencia no hubiese sido lo mismo.

Quiero agradecer especialmente a la Dra. Lisbeth sin ella no habría podido terminar el trabajo que aquí se presenta. De igual manera, a la maestra Abud así como el maestro Pelaéz y sus observaciones durante mis revisiones. También tengo mucho que agradecer al Dr. Asdrúbal por su asesoría en la estancia, ya que los resultados de este período fueron muy importantes para toda la investigación.

Estos dos años han enriquecido mi vida, por lo que le agradezco a la vida el haberme permitido estar y formar parte de la generación 2016-2018 de la Maestría en Sistemas Computacionales del Instituto Tecnológico de Orizaba, Veracruz, México.

Resumen

Los datos multimedia son la combinación de diversos tipos de representaciones de información, las imágenes son un ejemplo típico de estas representaciones. Las bases de datos multimedia son el conjunto de datos multimedia, las cuales deben proveer acceso basado en contenido.

CBIR (*“Content-Based Image Retrieval”*, Recuperación de Imágenes Basada en Contenido) es una técnica que permite al usuario extraer una imagen con base en una consulta desde una base de datos multimedia con imágenes almacenadas.

El objetivo de este proyecto es la realización de un sistema de inventario que contenga un sistema CBIR para la ayuda de los empleados de la empresa AGROMAQ en la identificación de diferentes refacciones que los clientes llevan, todo con la finalidad de que el tiempo de atención a los clientes sea menor, ya que actualmente se necesita de un experto para la identificación de estas refacciones, lo cual hace que el tiempo de atención se incremente no sólo en horas sino incluso en días.

El sistema utiliza SURF (*“Speeded Up Robust Features”*, Características robustas para aumento de velocidad) para la extracción de las imágenes similares en la base de datos. La elección de este descriptor se sustenta en el hecho de ser más preciso que otros debido a la naturaleza de su análisis, que no está basado en características como son la textura, color o forma de las imágenes. Su implementación dentro del sistema se realizó mediante la herramienta BoofCV que proporciona bibliotecas para su uso dentro de diferentes lenguajes de programación.

El sistema de inventario se desarrolló con el lenguaje de programación Java, en el IDE (*“Integrated Development Environment”*, ambiente de desarrollo integrado) de trabajo Eclipse. Debido a que la base de datos multimedia está almacenada en el Sistema Gestor de Bases de Datos (SGBD) PostgreSQL, éste es el que se usó para administrar la base de datos del sistema de inventario.

Índice general

Introducción.....	1
Capítulo 1. Antecedentes.....	2
1.1. Marco teórico.....	2
1.1.1 Base de datos.....	2
1.1.2 Bases de datos multimedia.....	2
1.1.3 Recuperación de imágenes basada en contenido.....	2
1.1.3.2 Consulta Rango.....	3
1.1.3.2 Consulta kNN.....	3
1.1.5 Máquina de vectores de soporte.....	3
1.1.6 Descriptor visual.....	3
1.2. Planteamiento del problema.....	3
1.3. Objetivo general y específico.....	4
1.3.1. Objetivo general.....	4
1.3.2. Objetivos específicos.....	4
1.4. Justificación.....	5
Capítulo 2. Estado de la práctica.....	6
2.1. Trabajos relacionados.....	6
2.2. Análisis comparativo.....	16
2.3 Propuesta de solución.....	21
Java.....	21
<i>Framework JavaServer Faces</i>	22
<i>Framework PrimeFaces</i>	22
OOHDM.....	23
NetBeans.....	24
PostgreSQL.....	25

SURF.....	25
BoofCV	26
Capítulo 3. Aplicación de la metodología	27
3.1. Determinación de requerimientos	27
3.2. Diseño conceptual.....	28
3.2.1. Diagrama de clases.....	28
3.2.2. Modelos de base de datos.....	29
3.3. Diseño Navegacional	32
3.3.1. Arquitectura del sistema.....	32
3.3.2. Patrón MVC.....	32
3.3.3. Diagrama de secuencia	34
3.4. Proceso CBIR	36
3.4.1 Registro de producto	36
3.4.2. Búsqueda de producto	37
Capítulo 4. Resultados.....	39
4.1. Caso de estudio	39
4.1.1. Caso de estudio 1.....	41
4.1.2. Caso de estudio 2.....	61
4.2. Evaluación de resultados.....	78
4.2.1. Eficiencia del sistema CBIR	78
4.2.2. Eficacia del sistema CBIR.....	79
Capítulo 5. Conclusiones y recomendaciones	82
5.1. Conclusiones	82
5.2. Recomendaciones	83
Productos académicos	84
Bibliografía.....	85

Índice de tablas

Tabla 2.1 Comparación de trabajos relacionados.....	16
---	----

Índice de Figuras

Figura 3.1 Diagrama de casos de uso	27
Figura 3.2 Diagrama de clases.....	28
Figura 3.3 Modelo lógico de la base de datos	29
Figura 3.4 Modelo conceptual de la base de datos	30
Figura 3.5 Modelo físico de la base de datos	31
Figura 3.6 Arquitectura del sistema.....	32
Figura 3.7 Arquitectura de la aplicación	33
Figura 3.8. Diagrama de secuencia de gestión de producto	34
Figura 3.9 Diagrama de secuencia de consulta de texto y basada en contenido	35
Figura 3.10 Proceso durante el registro de imágenes	36
Figura 3.11 Fragmento de código del proceso de registro 1	36
Figura 3.12 Fragmento de código del proceso de registro 2	36
Figura 3.13 Proceso durante la búsqueda de imágenes	37
Figura 3.14 Fragmento de código del proceso de búsqueda 1.....	37
Figura 3.15 Fragmento de código del proceso de búsqueda 2.....	37
Figura 3.16 Fragmento de código del proceso de búsqueda 3.....	38
Figura 3.17 Fragmento de código del proceso de búsqueda 4.....	38
Figura 4.1 Autenticación de usuario.....	39
Figura 4.2 Interfaz de búsqueda por imagen	40
Figura 4.3 Prueba de búsqueda de imagen	40
Figura 4.4 Resultado de prueba	41
Figura 4.5 Caso de estudio 1 imagen de búsqueda 1	42
Figura 4.6 Caso de estudio 1 resultado 1 de búsqueda 1	42
Figura 4.7 Caso de estudio 1 resultado 2 de búsqueda 1	43

Figura 4.8 Caso de estudio 1 imagen de búsqueda 2.....	43
Figura 4.9 Caso de estudio 1 resultado 1 de búsqueda 2.....	44
Figura 4.10 Caso de estudio 1 resultado 2 de búsqueda 2.....	44
Figura 4.11 Caso de estudio 1 resultado 3 de búsqueda 2.....	45
Figura 4.12 Caso de estudio 1 imagen de búsqueda 3.....	45
Figura 4.13 Caso de estudio 1 resultado 1 de búsqueda 3.....	46
Figura 4.14 Caso de estudio 1 resultado 2 de búsqueda 3.....	46
Figura 4.15 Caso de estudio 1 resultado 3 de búsqueda 3.....	47
Figura 4.16 Caso de estudio 1 imagen de búsqueda 4.....	47
Figura 4.17 Caso de estudio 1 resultado 1 de búsqueda 4.....	48
Figura 4.18 Caso de estudio 1 resultado 2 de búsqueda 4.....	48
Figura 4.19 Caso de estudio 1 resultado 3 de búsqueda 4.....	49
Figura 4.20 Caso de estudio 1 resultado 4 de búsqueda 4.....	49
Figura 4.21 Caso de estudio 1 imagen de búsqueda 5.....	50
Figura 4.22 Caso de estudio 1 resultado 1 de búsqueda 5.....	50
Figura 4.23 Caso de estudio 1 resultado 2 de búsqueda 5.....	51
Figura 4.24 Caso de estudio 1 resultado 3 de búsqueda 5.....	51
Figura 4.25 Caso de estudio 1 resultado 4 de búsqueda 5.....	52
Figura 4.26 Caso de estudio 1 imagen de búsqueda 6.....	52
Figura 4.27 Caso de estudio 1 resultado 1 de búsqueda 6.....	53
Figura 4.28 Caso de estudio 1 resultado 2 de búsqueda 6.....	53
Figura 4.29 Caso de estudio 1 imagen de búsqueda 7.....	54
Figura 4.30 Caso de estudio 1 resultado 1 de búsqueda 7.....	54
Figura 4.31 Caso de estudio 1 resultado 2 de búsqueda 7.....	55
Figura 4.32 Caso de estudio 1 imagen de búsqueda 8.....	55

Figura 4.33 Caso de estudio 1 resultado 1 de búsqueda 8	56
Figura 4.34 Caso de estudio 1 resultado 2 de búsqueda 8.....	56
Figura 4.35 Caso de estudio 1 imagen de búsqueda 9.....	57
Figura 4.36 Caso de estudio 1 resultado 1 de búsqueda 9.....	57
Figura 4.37 Caso de estudio 1 resultado 2 de búsqueda 9.....	58
Figura 4.38 Caso de estudio 1 resultado 3 de búsqueda 9.....	58
Figura 4.39 Caso de estudio 1 imagen de búsqueda 10.....	59
Figura 4.40 Caso de estudio 1 resultado 1 de búsqueda 10.....	59
Figura 4.41 Caso de estudio 1 resultado 2 de búsqueda 10.....	60
Figura 4.42 Caso de estudio 1 resultado 3 de búsqueda 10.....	60
Figura 4.43 Caso de estudio 2 imagen de búsqueda 1	61
Figura 4.44 Caso de estudio 2 resultado 1 de búsqueda 1	61
Figura 4.45 Caso de estudio 2 resultado 2 de búsqueda 1	62
Figura 4.46 Caso de estudio 2 resultado 3 de búsqueda 1	62
Figura 4.47 Caso de estudio 2 imagen de búsqueda 2.....	63
Figura 4.48 Caso de estudio 2 resultado 1 de búsqueda 2.....	63
Figura 4.49 Caso de estudio 2 resultado 2 de búsqueda 2.....	64
Figura 4.50 Caso de estudio 2 imagen de búsqueda 3.....	64
Figura 4.51 Caso de estudio 2 resultado 1 de búsqueda 3	65
Figura 4.52 Caso de estudio 2 resultado 2 de búsqueda 3.....	65
Figura 4.53 Caso de estudio 2 imagen de búsqueda 4.....	66
Figura 4.54 Caso de estudio 2 resultado 1 de búsqueda 4.....	66
Figura 4.55 Caso de estudio 2 resultado 2 de búsqueda 4.....	67
Figura 4.56 Caso de estudio 2 resultado 3 de búsqueda 4.....	67
Figura 4.57 Caso de estudio 2 imagen de búsqueda 5.....	68

Figura 4.58 Caso de estudio 2 resultado 1 de búsqueda 5.....	68
Figura 4.59 Caso de estudio 2 resultado 2 de búsqueda 5.....	69
Figura 4.60 Caso de estudio 2 imagen de búsqueda 6.....	69
Figura 4.61 Caso de estudio 2 resultado 1 de búsqueda 6.....	70
Figura 4.62 Caso de estudio 2 resultado 2 de búsqueda 6.....	70
Figura 4.63 Caso de estudio 2 imagen de búsqueda 7.....	71
Figura 4.64 Caso de estudio 2 resultado 1 de búsqueda 7.....	71
Figura 4.65 Caso de estudio 2 resultado 2 de búsqueda 7.....	72
Figura 4.66 Caso de estudio 2 imagen de búsqueda 8.....	72
Figura 4.67 Caso de estudio 2 resultado 1 de búsqueda 8.....	73
Figura 4.68 Caso de estudio 2 resultado 2 de búsqueda 8.....	73
Figura 4.69 Caso de estudio 2 resultado 3 de búsqueda 8.....	74
Figura 4.70 Caso de estudio 2 resultado 4 de búsqueda 8.....	74
Figura 4.71 Caso de estudio 2 imagen de búsqueda 9.....	75
Figura 4.72 Caso de estudio 2 resultado 1 de búsqueda 9.....	75
Figura 4.73 Caso de estudio 2 resultado 2 de búsqueda 9.....	76
Figura 4.74 Caso de estudio 2 imagen de búsqueda 10.....	76
Figura 4.75 Caso de estudio 2 resultado 1 de búsqueda 10.....	77
Figura 4.76 Caso de estudio 2 resultado 2 de búsqueda 10.....	77
Figura 4.77 Gráfica comparativa de eficiencia usando SIFT y SURF.....	78
Figura 4.78 Gráfica comparativa de la eficacia entre SIFT y SURF.....	79
Figura 4.79 Gráfica de desempeño de filtros.....	80
Figura 4.80 Gráfica de desempeño del sistema CBIR en el primer caso de estudio.....	81
Figura 4.81 Gráfica de desempeño del sistema CBIR en el segundo caso de estudio ...	81
Figura 5.1 Prototipo del sistema para capturar fotografías.....	83

Introducción

El presente trabajo tiene como objetivo realizar un sistema de inventario para la empresa AGROMAQ que utilice CBIR para realizar una búsqueda de refacciones dada una imagen de una pieza que el cliente llevará al establecimiento, todo esto con la finalidad de mejorar los tiempos de atención al cliente al hacer que la dependencia de un experto sea menor a la actual, debido a que en ocasiones el experto se encuentra ocupado con otro cliente o no encontrarse en el momento.

Algunos de los puntos a tomar en cuenta para la realización de este proyecto constan desde el estudio de los conceptos principales de bases de datos multimedia y consultas basadas en contenido y la identificación de las tecnologías a usar en él, hasta la realización del sistema basado en la arquitectura diseñada, de acuerdo con los requerimientos obtenidos por medio de entrevistas con los empleados de AGROMAQ.

En el primer capítulo de este trabajo se dan a conocer los conceptos fundamentales utilizados a lo largo del proyecto, además se muestran los objetivos, el planteamiento del problema y la justificación. En el segundo capítulo se realiza un análisis de diferentes artículos relacionados con el tema (Sistema CBIR) para tener un mejor entendimiento del funcionamiento del mismo o conocer alguna otra posible tecnología. El tercer capítulo presenta el desarrollo del sistema siguiendo la metodología OOHDM; mientras que su evaluación mediante dos casos de estudio se muestra en el cuarto capítulo. Finalmente, el capítulo cinco muestra las conclusiones de la tesis, así como recomendaciones para mejorar el trabajo en un futuro.

Capítulo 1. Antecedentes

1.1. Marco teórico

A continuación, se explicarán los conceptos más importantes relacionados con el trabajo presentado.

1.1.1 Base de datos

Es un repositorio de datos, diseñado para ayudar en el almacenamiento, recuperación y mantenimiento de datos. Existen distintos tipos de bases de datos capaces de almacenar: archivos binarios, imágenes, videos, datos relacionales, datos multidimensionales, datos transaccionales, datos analíticos o datos geográficos, por mencionar algunos [1].

1.1.2 Bases de datos multimedia

Los datos multimedia son la combinación de diversos tipos de representaciones de información, los ejemplos típicos incluyen: imágenes, audio, series de tiempo o video [2].

Las bases de datos multimedia son el conjunto de datos multimedia, las cuales proveen [2]:

- Acceso basado en contenido
- Generación de conocimiento
- Manejo de grandes volúmenes de datos
- Buenos tiempos de respuesta

1.1.3 Recuperación de imágenes basada en contenido

CBIR es una técnica que permite al usuario extraer una imagen con base en una consulta, desde una base de datos con una gran cantidad de imágenes. Usualmente el procedimiento se lleva a cabo realizando una comparación de diferentes características de bajo nivel, tales como: color, textura y forma, extraídas desde las imágenes mismas [3].

La recuperación basada en contenido busca objetos multimedia similares a un objeto consulta [2]. Hay dos tipos de consultas basadas en contenido: Consultas Rango y kNN (“*k Nearest Neighbors*”, K vecinos más cercanos).

1.1.3.2 Consulta Rango

Este tipo de consulta permite recuperar todos aquellos objetos que no se desvían de la consulta más allá de un umbral de tolerancia [2].

1.1.3.2 Consulta kNN

Las consultas kNN obtienen como resultado un número fijo de objetos más similares al objeto consulta [4].

1.1.5 Máquina de vectores de soporte

La idea principal de SVM (*“Support Vector Machine”*, Máquina de vectores de soporte) es la de construir un hiperplano como la superficie de decisión de tal forma que la separación entre ejemplos positivos y negativos se maximiza. Los vectores de soporte constituyen un pequeño subconjunto de datos de entrenamiento extraídos por el algoritmo de aprendizaje de vectores de soporte [3].

1.1.6 Descriptor visual

Se define como la representación matemática del contenido de la imagen. Así, el descriptor visual más básico es un histograma de color, donde un conjunto limitado de valores representa estadísticamente el número de píxeles contabilizados de cada color en una imagen. Además de éste, existen múltiples descriptores que tratan de representar una información de mayor nivel semántico como los bordes presentes o las formas básicas que aparecen en la imagen [5].

1.2.Planteamiento del problema

La empresa AGROMAQ, ubicada en la ciudad de Orizaba, se dedica a la venta y reparación de maquinaria agrícola como motosierras, aspersores, motores eléctricos, por mencionar algunos. AGROMAQ es una empresa líder en su área, sin embargo, no cuenta con un sistema que le permita gestionar su inventario. La mayoría de los clientes de AGROMAQ que compran refacciones llevan una muestra del producto que desean comprar, pero es muy difícil para los empleados saber de qué pieza se trata, por lo que es necesario un mecánico experto que reconozca la pieza para saber si se cuenta con dicha refacción en el inventario o en caso contrario solicitar la pieza con los distribuidores.

En este trabajo se desarrolló un sistema que recupera información sobre una pieza como su precio, nombre, descripción y cantidad con base en su fotografía. Esto permitirá que cualquier empleado sepa si se cuenta con dicha pieza o realizar un pedido en caso contrario. El sistema también gestiona la información de la base de datos multimedia. Además, con este sistema se logrará reducir el tiempo de atención a un cliente, el cual es de veinte minutos en el mejor de los casos.

1.3. Objetivo general y específico

1.3.1. Objetivo general

Desarrollar un sistema que permita realizar consultas basadas en contenido en una base de datos multimedia para la empresa AGROMAQ.

1.3.2. Objetivos específicos

1. Estudiar los conceptos principales de bases de datos multimedia y consultas basadas en contenido para entender el alcance de la investigación.
2. Analizar los métodos para realizar consultas basadas en contenido para seleccionar el más adecuado.
3. Identificar las técnicas y tecnologías que se utilizarán para el desarrollo del sistema: lenguaje de programación, IDE, SGBD, metodología para el desarrollo de software y descriptor visual.
4. Determinar los requerimientos del sistema por medio de entrevistas con los empleados y la dueña de la empresa.
5. Realizar el diseño conceptual, lógico y físico de la base de datos multimedia para gestionar la información requerida.
6. Diseñar la arquitectura del sistema para determinar cada uno de los módulos a desarrollar.
7. Implementar el sistema con base en la arquitectura diseñada.
8. Validar el sistema por medio de un caso de estudio de la empresa AGROMAQ.

1.4. Justificación

Los datos multimedia son de importancia clave en muchas áreas de aplicación como medicina, cartografía, meteorología, seguridad, entre otras. La extracción, clasificación y manipulación automática del contenido multimedia son de importancia crítica para una gestión eficiente de datos multimedia [6]. Los métodos de recuperación de datos multimedia basados en contenido mejoran la precisión de las búsquedas que se realizan en la base de datos [7]. Estos métodos son necesarios cuando las anotaciones textuales son inexistentes o incompletas [8]. Además, los métodos basados en contenido mejoran potencialmente la precisión en la recuperación incluso cuando las anotaciones textuales están presentes, ya que dan un conocimiento adicional a las colecciones de datos multimedia.

El sistema CBIR que se desarrolló permitirá satisfacer distintas necesidades: la primera es que los empleados sabrán identificar de una manera más fácil y rápida los nombres de las piezas que los clientes desean, la segunda es que sabrán si dichas piezas se tienen en el inventario, la tercera es que podrán consultar las características de las piezas en menos tiempo y con menos esfuerzo, la cuarta es que el sistema gestionará el inventario.

Los beneficios que se obtendrán con el desarrollo del sistema son: 1) Reducción del tiempo de atención a los clientes; 2) Menos errores al realizar pedidos; 3) Gestión del inventario eficiente y efectiva; 4) Incremento de las ventas.

Los beneficiados con el desarrollo del sistema son: 1) La propia empresa, ya que obtendrá más ganancias al tener mejor administrado el inventario; 2) Los empleados de la empresa, debido a que tendrán que invertir menos tiempo y esfuerzo al realizar ventas y pedidos; 3) Los clientes, que esperarán menos para obtener una respuesta, y 4) El mecánico experto, ya que ocupará menos tiempo en la atención a los clientes.

Capítulo 2. Estado de la práctica

En esta sección se presenta el análisis del estado de la práctica, es decir, de trabajos que se relacionan con la investigación a desarrollar en este proyecto de tesis.

2.1. Trabajos relacionados

En [3] se propuso una metodología para la recuperación de imágenes basada en contenido (CBIR), la cual se basa en la codificación de cadenas de color (*colour string coding*) y la comparación de cadenas (*String comparison*). Para ello, usaron la metodología siguiente:

1) Clasificación de la base de datos usando una SVM: Los sistemas de CBIR toman mucho tiempo para la búsqueda de imágenes puesto que comparan la imagen que se busca con cada una de las imágenes ingresadas en la base de datos, así que el uso de una SVM para la clasificación de las imágenes en la base de datos se requiere si se desea disminuir el tiempo que toma esta búsqueda.

2) Extracción de características: Extrae varias características de las imágenes tanto de las que están dentro de la base de datos como de la que se provee para la búsqueda.

3) Medida de similitud: Se comparan las características extraídas de las imágenes para encontrar aquellas con mayor compatibilidad.

Por último, para las pruebas se usaron 200 imágenes de diferentes categorías (deportes, flores, frutas, entre otros) y se obtuvo una precisión mucho más elevada en comparación de otros sistemas CBIR con una diferencia de precisión promedio de 85.86%.

En [9] se propuso un método para una clasificación automática rápida y eficaz de diferentes eritrocitos para la obtención de aquellos que presenten infección del parásito *Plasmodium falciparum*, el cual es causante de malaria.

El procedimiento que se sigue para la clasificación de los eritrocitos consta de tres pasos:

1) Preprocesamiento de imagen: Este paso consta de aplicación de un filtro a las imágenes para igualar el color e iluminación a una imagen de eritrocito, esto constó de un proceso de transformación del RGB (Red, Green and Blue, Rojo, Verde y Azul) mediante una transformación YCbCr (Nombrado así por las coordenadas utilizadas Y, Cb y Cr).

2) Reconocimiento de eritrocitos: Se realiza la obtención del objeto de interés, el eritrocito, mediante una clasificación de píxeles que permite etiquetar cada píxel tanto como “fondo” o “primer plano” basado en las características del color. Después una estructura de árbol de inclusión (*Inclusion-Tree structure*) se encarga de poder ver la relación entre el “fondo” y el “primer plano” para eliminar cualquier estructura irrelevante.

3) Clasificación de eritrocitos: Por último, se realiza la comparación de las características de los eritrocitos (color, forma, textura, entre otros) para detectar tanto a aquellos que estén infectados como los que no.

Las pruebas se realizaron a 450 imágenes compuestas por 12,557 eritrocitos y terminó con una sensibilidad del 94% para detectar los eritrocitos infectados.

En [10] se presentó un análisis de diferentes formas para clasificar plantas (forma de la flor/hoja, textura de las hojas y estructura de las venas). La clasificación conlleva un conjunto de dificultades para su realización, entre las cuales la principal es el hecho de que las plantas al no ser “objetos rígidos” resultan en una serie de deformaciones (diferencias entre tamaño y forma dentro del mismo tipo de planta).

La clasificación por medio de la forma de la flor/hoja es la manera más básica de las mencionadas con anterioridad, puesto que incluso las personas no botánicas son capaces de detectar formas similares a una muestra dada. Los sistemas que utilizan esta clasificación como base usan varias aproximaciones diferentes como: análisis de Fourier, dimensiones fractal, firmas de contorno, análisis de punto de referencia (*Landmark analysis*), entre otros.

La siguiente forma de clasificación de acuerdo a su uso es por medio de la estructura de las venas, también conocida como “venación” (*venation*), la forma en la que las venas se desarrollan en una hoja varían para cada planta debido a la manera en que el agua, minerales, azúcares y otras sustancias se distribuyen. Algunos de los métodos de análisis son: algoritmo de análisis escala-espacio y algoritmo de detección de suavizado y borde.

Por último, la textura de la hoja es una forma difícil para clasificar puesto que no hay algún método factible para su realización. Un par de formas son: usar un arreglo de filtros Gabor

en un gran conjunto de datos y otro basado en transformadas de ondícula (*wavelet transforms*) y SVM.

Una vez terminada la comparación de las diferentes formas de clasificación se llegó a la conclusión de que debido a que las plantas son muy diferentes en cuanto a tamaño, forma y color, no existe un método mejor para todos los tipos de hojas.

S. Tomas et al. [4] propusieron un nuevo tipo de consulta para la obtención de imágenes en sistemas CBIR, denominado kDNN (*k Distinct Nearest Neighbors*, k vecinos distintos más cercanos), basado en kNN. El método de consulta kNN es muy efectivo, pero tiene algunas desventajas en cuanto a la escalabilidad de las bases de datos, puesto que mientras más contenido hay en ellas el resultado de las consultas realizadas con kNN da como resultado imágenes similares entre ellas.

El método kDNN soluciona el inconveniente de kNN puesto que su algoritmo excluye a los objetos similares entre otros objetos que ya fueron reportados, lo cual conlleva a que este enfoque sea mucho más robusto con respecto al tamaño de las bases de datos.

Al final se realizó un experimento con una colección de 100 millones de fotos de objetos reales y se aplicó un sistema, basado en Java, a 30 usuarios y los resultados fueron de un 66% de satisfacción de la búsqueda de imágenes en kDNN contra un 26% de satisfacción en kNN.

En [11] se presentó un sistema CBIR para una base de datos de imágenes de tatuajes basado en el descriptor de imagen local SIFT. El objetivo del sistema es la obtención de imágenes “visualmente similares” a la imagen de consulta, así como la determinación de alguna característica extra a la forma y diseño del tatuaje, como el área del cuerpo donde está localizado y etiquetas de clase de los tatuajes. También se introdujeron un conjunto de restricciones entre los puntos clave SIFT en dos imágenes para reducir el número de “falsos positivos”.

El proceso de obtención de imágenes se realiza en una base de datos con imágenes indexadas, en la cual también se incluye la localización en el cuerpo del tatuaje, de estar disponible, así como las etiquetas de clase, de igual forma si hay disponibles.

Las pruebas se realizaron en una base de datos con 63,593 imágenes de tatuajes y dieron como resultado una precisión de 36.4% (Rango 1) y 76.2% (Rango 20) sin aplicación de restricciones y de una precisión de 85.9% y 89.5% con la aplicación de restricciones.

Q. Zheng et al. [12] propusieron un estudio de una rama de CBIR, la Recuperación de imágenes basada en objetos (OBIR, “*Object-Based Image Retrieval*”) y cómo unirlo a la recuperación de texto, dentro de la cual hay técnicas de recuperación que se clasifican en tres grupos: basado en letras-visuales, basado en palabras-visuales y basado en oraciones-visuales. Por último, los autores plantean un nuevo tipo de recuperación de texto, el basado en frases-visuales que resulta un enfoque para un OBIR eficiente y eficaz. Este nuevo enfoque contiene una serie de componentes para su realización: construcción de palabras visuales, construcción de frases visuales, construcción de índice de frases y medida de similitud de imágenes basadas en frases visuales.

Las pruebas que se realizaron para medir la eficiencia y eficacia del sistema OBIR dieron como resultado una eficiencia medida en tiempo de un promedio de 40 milisegundos por imagen para la obtención de frases contenidas en 8707 imágenes, y para la eficacia se comparó el sistema contra otro que usa el mismo algoritmo, en donde se obtuvo una puntuación de 7.27 contra 6.07 logrando un mejor resultado por parte del sistema propuesto.

En [13] se propuso un nuevo sistema de detección y emparejamiento de píldoras de droga que fueron ingeridas, para el sistema se utilizó como base la detección de las píldoras dependiendo de su forma, sello/Figura, color, histograma de color y “*Hu moments*” (que indica el estado de una imagen en un lugar en el tiempo).

El sistema sigue una serie de pasos que son: 1) Preprocesamiento; 2) Construcción de descripción de impresión (“*Imprint descriptor construction*”); 3) Obtención de color; 4) Obtención de forma; 5) Normalización de la rotación; 6) Emparejamiento (“*Matching*”).

Se hicieron pruebas en una base de datos de 15,031 imágenes, incluyendo 1029 píldoras de droga ilícitas, y los resultados fueron de una precisión de 73.17%, con lo cual se llegó a la conclusión de que estos resultados eran muy satisfactorios dada la similitud entre muchos tipos de píldoras en cuanto a color, forma y sello de impresión.

D. Thomas et al. [14] realizaron una comparación experimental de un gran número de descriptores de imágenes para CBIR. La comparación se hace con base en el desempeño de una gran variedad de descriptores visuales para cuatro diferentes tipos de tareas CBIR, las cuales son: 1) recuperación de fotos de catálogo; 2) recuperación de una colección de fotos personales; 3) recuperación de edificios, y 4) recuperación de imágenes médicas.

Las características probadas que se clasifican en diferentes tipos (representación de color, representación de textura, características locales (LF, "*Local Features*") y representación de forma) fueron: 1) Histograma de color; 2) LF SIFT búsqueda global; 3) LF histograma de parches; 4) Histograma de características; 5) MPEG 7: color escalable; 6) LF firma de parches; 7) Histograma Gabor; 8) Imágenes 32x32; 9) MPEG 7: diseño de color; 10) Imágenes Xx32; 11) Histograma de textura Tamura; 12) Histograma de valor gris; 13) LF parches globales; 14) MPEG 7: histograma de bordes; 15) Histograma de características (relacional); 16) Vector Gabor, y 17) Característica de textura global.

Después de su prueba en cinco bases de datos: Wang, UW (Universidad de Washington), IRMA (*Image Retrieval in Medical Applications*, Recuperación de Imágenes en Aplicaciones médicas), UCID (*Uncompressed Colour Image Database*, Base de Datos de Imágenes de Color sin Comprimir) y ZuBuD (*Zurich Buildings Database for Image Based Recognition*, Base de Datos de Edificios de Zurich para el Reconocimiento Basado en Imágenes), se llegó a la conclusión de que los histogramas de color son una buena referencia para las fotos a color, sin embargo, el uso de descriptores de imágenes locales sobrepasa en rendimiento a los histogramas de color algunas veces, aunque a costo de un mayor número de recursos computacionales.

Por último, se recalca el hecho de que ninguna de las características de textura presentadas en el trabajo puede usarse por sí sola de manera óptima en la extracción de propiedades de textura de una imagen, aunque el uso de varias de ellas en conjunto puede obtener resultados muy satisfactorios.

En [15] se propuso una técnica basada en reglas para la extracción de regiones de atención visual (VAR, "*Visual Attention Regions*") que utiliza agrupamiento en tiempo real.

La técnica propuesta consiste en cuatro partes:

- 1) Una técnica de segmentación rápida denominada algoritmo de agrupamiento en tiempo real (RTCA, “*real time clustering algorithm*”).
- 2) Una especificación refinada de VAR, conocida como regiones de atención visual jerárquicas (HVAR, “*hierarchical visual attention regions*”).
- 3) Un algoritmo basado en reglas (RADA, “*rule based detection algorithm*”) que obtiene un conjunto de HVAR en tiempo real.
- 4) Un módulo para presentar imágenes adaptativas y sus correspondientes operaciones de adaptación usando HVAR.

Los experimentos realizados mostraron que RADA cierra la brecha entre las regiones de atención visual y las semánticas de alto nivel (por lo que permite identificar objetos adecuadamente) cuando se aplica a diferentes tipos de imágenes. También se dejó en claro un trabajo a futuro para investigar el desempeño del algoritmo con base en distintas características.

En [16] se presentó un método de dos pasos para un sistema CBIR en bases de datos multimedia grandes, con la finalidad de mejorar tanto la efectividad como la eficiencia del CBIR tradicional mediante la exploración de medios secundarios. El método de recuperación de dos pasos primero clasifica por un medio secundario y después realiza CBIR solo en los K-objetos mejor clasificados.

La manera tradicional de recuperación de imágenes que se sigue en bases de datos multimodales (que tienen varias formas de recuperación) es la de buscar modalidades separadas y después unir los resultados. A pesar de que este método es robusto tiene los siguientes problemas:

- El pesado apropiado de las modalidades y la puntuación de normalización/combinación no son problemas triviales y requieren datos de entrenamiento.
- Si se juzga sólo mediante la similitud visual, la influencia de las puntuaciones textuales puede empeorar la calidad visual de los resultados finales.

La recuperación de dos pasos resuelve los problemas anteriores mediante la delimitación del tamaño de la colección. Tres factores que determinan esta delimitación son:

- 1) El número relevante de objetos en la colección.
- 2) La calidad de la clasificación.
- 3) La medida que se va a optimizar.

Los primeros dos factores dependen de la consulta, por lo tanto, los límites se seleccionan dinámicamente por consulta, no de manera estática como muchos otros métodos proponen.

Los experimentos con la recuperación de imágenes de dos pasos se realizaron en una gran base de datos multimedia y se encontró que el delimitado dinámico es más efectivo y robusto que el delimitado estático, adicionalmente se comparó con la fusión de resultados recuperados visual y textualmente, y se concluyó que el método de dos pasos fue mejor en tres de cuatro medidas de comparación.

La propuesta de una arquitectura para la recuperación de imágenes de forma eficiente, así como el análisis Web desde el punto de vista de imágenes, empleando tecnologías *big data*, es el tema principal que se toca en [5]. Los objetivos de la arquitectura propuesta son tanto aliviar la carga computacional y ser capaces de hacer búsquedas rápidas sobre grandes colecciones de imágenes, como también demostrar que el uso de tecnologías *big data* no requiere de una gran infraestructura de computación para el procesamiento y recuperación de grandes cantidades de imágenes.

Las fases de la arquitectura propuesta son:

- 1) Firma de la imagen de entrada.
- 2) Búsqueda con árboles binarios.
- 3) Computación de distancias distribuida.
- 4) Ordenación de imágenes similares.

Para la fase de pruebas se usó la base de datos *ImageNet* (14 millones de imágenes contenidas), donde se realizaron siete pruebas, en cada prueba se ocuparon conjuntos de datos que incrementaron en cada iteración y dieron como resultado que el *Hashing* de localidad sensible (LSH, "*locality-sensitive hashing*") es el que menos tiempo tardó y su acceso a memoria fue mucho más rápido, a diferencia del uso de árboles de indexación distribuidos, pero aún así no es suficiente para los análisis Web distribuidos, por ello la arquitectura propuesta es la más útil en el ambiente de análisis Web.

S. Jorge et al. [17] propusieron un nuevo método de clasificación para la búsqueda en BoV (“*Bag of visual words*”, Bolsa de palabras visuales), que usa el *marco de trabajo* Fisher Vector (FV). FV tiene diversas ventajas: es eficiente de calcular, lleva a excelentes resultados incluso con clasificadores lineales eficientes y puede comprimirse con un mínimo de pérdida de precisión.

Los experimentos se realizaron en cinco bases de datos diferentes con un total de 9 millones de imágenes y 10 mil clases. Los resultados se compararon con la clasificación k-NN y se mostró que el FV puede comprimirse por un factor de 32 con un impacto muy pequeño en la precisión de clasificación

En [18] se realizó una revisión de los trabajos sobre propuestas de detección y una comparación de los métodos públicos disponibles en un marco de trabajo unificado.

Las contribuciones de este trabajo son las siguientes:

- 1) Se provee una visión general y sistemática de los métodos de propuesta de detección y se definen las bases como puntos de referencia.
- 2) Se introduce la noción de la repetitividad de propuesta (*proposal repeatitibility*).
- 3) Se evalúa la influencia de diferentes métodos de propuestas de detección en el desempeño de detección.

Se analizaron varios métodos de detección como: Bing, CPMC (“*Constrained Parametric Min-Cuts*”, mínimo de cortes paramétricas con restricciones), “*EdgeBoxes*”, entre otros. La conclusión fue que los mejores métodos de acuerdo a su desempeño son:

“*SelectiveSearch*”, “*Rigor*”, MCG (“*Multiscale combinatorial grouping*”, Agrupamiento combinado multiescala) y “*EdgeBoxes*”.

La creciente necesidad de sistemas CBIR eficientes en muchas áreas de aplicación tales como Internet, biomedicina, comercio y educación, conllevó a la realización de un nuevo conjunto de descriptores apropiados para la indexación y recuperación de imágenes propuesto en [19]. La estructura de los descriptores combina información de textura y color, extraídos usando enfoques difusos.

Para evaluar el desempeño de los descriptores propuestos se usó ANMRR (“*Average Normalized Modified Retrieval Rank*”, Rango promedio normalizado de recuperación

Modificado). Los resultados experimentales mostraron que el desempeño de los descriptores propuestos es mejor que el desempeño de descriptores MPEG-7 de tamaño similar.

En los sistemas CBIR, el descriptor de imagen es un elemento muy importante. Este es responsable de la evaluación de las similitudes entre las imágenes. Los descriptores se clasifican dependiendo de la propiedad de la imagen analizada, por ejemplo, los descriptores de color o textura, que analizan las propiedades de color o textura, respectivamente. En [20] se realizó un estudio comparativo sobre diferentes descriptores de color y textura para su uso en entornos Web.

Para el estudio se tomaron en cuenta 24 descriptores de color y 28 descriptores de textura, en los cuales se aplicó una evaluación de dos niveles: un análisis teórico en términos de la complejidad de algoritmo y una comparación experimental tomando en consideración aspectos de eficiencia y efectividad.

La comparación experimental se llevó a cabo en un ambiente Web con una base de datos que contiene más de 230 mil imágenes, en donde durante el análisis de los algoritmos, se mostró que la mayoría de los descriptores tienen una complejidad de $O(n)$, donde n es el número de píxeles en una imagen, así como en el segundo nivel de la evaluación se concluyó que los descriptores de color BIC (“*border/interior pixel classification*”, clasificación de píxeles de borde/interior) y ACC (“*Image indexing using color correlograms*”, Indexación de imágenes usando correlogramas de color) son las mejores opciones para ambientes heterogéneos dentro del escenario Web.

Navel et al. [21] propusieron un SURF mejorado para los sistemas CBIR, esto se logra extrayendo los momentos H_u y los valores propios de la imagen, formando así un descriptor SURF de dimensión 88. Los experimentos con esta versión mejorada de SURF demostraron que esta versión es mejor que el SURF original.

Ronald et al. [22] presentaron un sistema CBIR, Plantcyclopedia, que permite identificar las plantas proporcionadas por el usuario y utiliza SURF. El sistema se divide en tres pasos principales: 1) Recuperar la información de la imagen; 2) Definir la orientación de la imagen,

y 3) Obtener puntos de características de ambas imágenes usando *K-Means* como una técnica de clasificación.

Pradnya y Pravin [23] desarrollaron un sistema SBIR (*“Sketch Based Image Retrieval”*, Recuperación de imagen basado en bosquejo), este sistema usa SURF con una desviación estándar para recuperar las imágenes de la base de datos.

Kaur y Jindal [24] realizaron la extracción de características en imágenes médicas usando SURF bajo la plataforma OpenCV, utilizada para la búsqueda del iris en los ojos, huesos y de IRM (Imagen por Resonancia Magnética). Los pasos del sistema son cinco:

- 1) Cargar imágenes de prueba.
- 2) Extraer puntos clave, contornos y datos texturizados.
- 3) Almacenar las características en un vector.
- 4) Mostrar imágenes resaltando los puntos clave.
- 5) Calcular el rendimiento.

Alexandra et al. [25] propusieron un método para CBIR y clasificación, el método utiliza SURF combinado con BoVW (*“Bag of Visual Words”*, Bolsa de Palabras Visuales). Esta combinación tiene buenos resultados de recuperación y clasificación en comparación con otros métodos. SURF tiene cuatro pasos principales: 1) Crear la representación integral de la imagen; 2) Usar la determinante de la matriz Hessiana; 3) Extraer el descriptor SURF, y 4) Almacenar las características principales. BoVW consta de tres pasos principales: 1) Extraer automáticamente los puntos de interés y el descriptor de las imágenes; 2) Cuantificar los puntos clave y los descriptores para formar el diccionario visual, y 3) Encontrar las ocurrencias de cada palabra visual en la imagen para construir el histograma de BoW.

2.2. Análisis comparativo

La Tabla 2.1 concentra la información del análisis comparativo realizado a los artículos relacionados con el trabajo desarrollado en este proyecto de Tesis.

Tabla 2.1 Comparación de trabajos relacionados

Artículo	Objetivo	Tecnologías	Resultado
[3]	Obtener una metodología CBIR que use la comparación de cadenas de color.	-Sistema CBIR -Codificación de cadenas de color -SVM	Se obtuvo un porcentaje de precisión mucho más elevado comparado con otros sistemas CBIR.
[9]	Crear un método para la detección de eritrocitos que presenten infección del parásito <i>Plasmodium falciparum</i> .	-Transformación YCbCr -Sistema CBIR -Estructura de árbol de inclusión	Se obtuvo una sensibilidad del 94% para detectar los eritrocitos infectados.
[10]	Realizar una comparación de diversos métodos para la clasificación de plantas.	-Sistema CBIR -Photoshop -SVM	Debido a que las plantas son muy diferentes en cuanto a tamaño, forma y color, no existe un método mejor para todos los tipos de hojas.

Artículo	Objetivo	Tecnologías	Resultado
[4]	Proponer kDNN, un nuevo tipo de consulta basado en kNN.	-kNN -kDNN	66% de satisfacción de la búsqueda de imágenes en kDNN contra un 26% de satisfacción en kNN.
[11]	Crear un sistema CBIR para la identificación de tatuajes.	-Sistema CBIR - SIFT	Las pruebas se realizaron en una base de datos con 63,593 imágenes de tatuajes y dieron como resultado una precisión de 36.4% (Rango 1) y 76.2% (Rango 20), sin aplicación de restricciones, y de una precisión de 85.9% y 89.5% con la aplicación de restricciones.
[12]	Proponer un sistema de recuperación de imágenes basado en objetos que usa la recuperación de texto basada en frases-visuales.	-Sistema OBIR -Recuperación de texto -SIFT	Una eficiencia medida en tiempo de un promedio de 40 milisegundos por imagen para la obtención de frases contenidas en 8707 imágenes, y para la eficacia se comparó el sistema contra otro que usa el mismo algoritmo, en donde se obtuvo una puntuación de 7.27 contra 6.07 logrando un mejor resultado por parte del sistema propuesto.
[13]	Crear un nuevo sistema de detección y emparejamiento de píldoras de droga.	-Sistema CBIR -Filtrado Gaussiano -SIFT -MLBP (<i>Multi-Scale Local Binary Pattern</i>)	Se hicieron pruebas en una base de datos de 15,031 imágenes, incluyendo 1029 píldoras de droga ilícitas y los resultados fueron de una precisión de 73.17%.

Artículo	Objetivo	Tecnologías	Resultado
[14]	Realizar una comparación de características en sistemas CBIR para ver su desempeño en diferentes tareas.	-Sistema CBIR	Los histogramas de color son una buena referencia para las fotos a color, sin embargo, el uso de descriptores de imágenes locales sobrepasa en rendimiento a los histogramas de color algunas veces.
[15]	Proponer una técnica basada en reglas para la extracción VAR usando agrupamiento en tiempo real.	-RTCA -RADA -HVAR	Los experimentos realizados mostraron que RADA permite identificar objetos adecuadamente en diferentes tipos de imágenes.
[16]	Proponer un método de dos pasos para un sistema CBIR en bases de datos multimedia grandes.	-Sistemas CBIR	Los experimentos se realizaron en una gran base de datos multimedia y se encontró que el delimitado dinámico es más efectivo y robusto que el delimitado estático.
[5]	Proponer una arquitectura para la recuperación de imágenes de forma eficiente, así como el análisis Web desde el punto de vista de imágenes, empleando tecnologías <i>big data</i> .	-Big data -LSH -Sistema CBIR	Para la fase de pruebas se usó la base de datos <i>ImageNet</i> y se obtuvo como resultado que el LSH es el que menos tiempo tardó y su acceso a memoria fue mucho más rápido, a diferencia del uso de árboles de indexación distribuidos.

Artículo	Objetivo	Tecnologías	Resultado
[18]	Realizar una comparativa de diferentes métodos de propuesta de detección.	-Métodos de propuesta de detección como : Bing, CPMC, “EdgeBoxes”, entre otros.	Los mejores métodos de acuerdo a su desempeño son: “SelectiveSearch”, “Rigor”, MCG y “EdgeBoxes”.
[19]	Proponer un nuevo conjunto de descriptores apropiados para la indexación y recuperación de imágenes.	-Sistema CBIR -ANMRR	El desempeño de los descriptores propuestos es mejor que el desempeño de descriptores MPEG-7 de similar tamaño.
[20]	Realizar un estudio comparativo sobre diferentes descriptores de color y textura para su uso en entornos Web.	-Sistema CBIR	La comparación experimental se llevó a cabo en un ambiente Web con una base de datos que contiene más de 230 mil imágenes, en donde se mostró que la mayoría de los descriptores tienen una complejidad de $O(n)$, donde n es el número de píxeles en una imagen, así como en el segundo nivel de la evaluación se concluyó que los descriptores de color BIC y ACC son las mejores opciones para ambientes heterogéneos dentro del escenario Web.

Artículo	Objetivo	Tecnologías	Resultado
[21]	Proponer una mejora para el descriptor SURF y demostrar que es más eficiente.	- Enhanced SURF	El descriptor SURF mejorado supera a SURF tanto en precisión (9.2% mejor) como en <i>recall</i> (3% mejor).
[22]	Realizar una aplicación que provea la información de las plantas, dicha aplicación tendrá un sistema CBIR para realizar la búsqueda por imagen.	- SURF -CBIR	El Sistema CBIR desarrollado tiene una precisión en un rango entre 69.51% y 91.19%.
[23]	Desarrollar un sistema que recupere imágenes de rostros con consultas realizadas con bosquejos.	-SBIR -SURF	SURF con desviación estándar funciona mejor para encontrar imágenes en una base de datos con bosquejos.
[24]	Construir un sistema que realice la extracción de características en imágenes médicas usando SURF.	-SURF -OpenCV -C++	La precisión del sistema fue del 80% para IRM, 75% para iris y 90% para huesos.
[25]	Se presenta un trabajo en progreso para un método CBIR y clasificación.	BoVW SURF	Se obtuvo una precisión promedio del 78.97%.

Después de haber revisado los artículos anteriores se obtiene como conclusión que los sistemas CBIR son importantes debido a que facilitan la tarea de identificación de objetos a personas que no son expertas en la materia. Así como el hecho de que estos no están limitados a una sola área como la medicina, botánica, medicina forense, entre otras.

Finalmente, es posible decir que SIFT y SURF tienen buenos resultados (precisión) en los sistemas CBIR como se muestra en la Tabla 2.1. La sección 4 mostrará que el tiempo y la recuperación son mejores en SURF que en SIFT, por lo tanto, se utilizó SURF en el sistema CBIR de la empresa AGROMAQ.

2.3 Propuesta de solución

A continuación, se muestran las diferentes tecnologías utilizadas para el desarrollo del sistema propuesto en esta tesis.

Java

Java [26] es un lenguaje de programación de alto nivel orientado a objetos desarrollado por *Sun Microsystems*, empresa adquirida recientemente por *Oracle Corporation*. Algunas características de este lenguaje se describen a continuación.

- **Orientado a objetos.** Los objetos agrupan en estructuras encapsuladas tanto sus datos como sus métodos.
- **Distribuido.** Java proporciona clases para su uso en aplicaciones de red, lo que facilita la creación de aplicaciones distribuidas.
- **Robusto.** Proporciona numerosas comprobaciones en compilación y tiempo de ejecución. Se ha prescindido por completo de los punteros y elimina la necesidad de liberación explícita de memoria.
- **Portable.** Java es un lenguaje compilado y ejecutado y esto es lo que permite la portabilidad.
- **Seguro.** Se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.
- **Independiente de la arquitectura.** El compilador de Java genera *bytecodes*, un formato diseñado para transportar el código eficientemente a múltiples plataformas hardware y software.

- **Multihilo.** Java soporta la sincronización de múltiples hilos de ejecución, especialmente útiles en la creación de aplicaciones de red distribuidas.
- **Dinámico.** Las clases solo se enlazan a medida que se necesitan.

Framework JavaServer Faces

JSF [27] es un *framework* basado en el API de *Servlets* que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante *Facelets*.

JSF proporciona las siguientes características destacables:

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Conexión de los componentes gráficos con los datos de la aplicación mediante los denominados *beans* gestionados.
- Conversión de datos y validación automática de la entrada del usuario.
- Navegación entre vistas.
- Internacionalización

Framework PrimeFaces

PrimeFaces [28] es una biblioteca de componentes JSF de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones Web.

Características:

- Conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte de AJAX (“*Asynchronous JavaScript And XML*”, JavaScript asíncrono y XML) con despliegue parcial, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.
- Componente para aplicaciones Web para teléfonos móviles, especiales para iPhones, Palm, Android y teléfonos móviles Nokia.

OOHDM

OOHDM (*“Object Oriented Hypermedia Design Models”*, Modelo de Diseño de Hipermedia Orientado a Objetos) [29] es una metodología de desarrollo para la elaboración de aplicaciones multimedia y tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia. OOHDM se basa en HDM (*“Hypermedia Design Models”*, Modelo de Diseño de Hipermedia), en el sentido de que toma muchas de las definiciones, sobre todo en los aspectos de navegación, planteadas en HDM. Sin embargo, OOHDM supera con creces a su antecesor, ya que no es simplemente un lenguaje de modelado, sino que define unas pautas de trabajo, centrado principalmente en el diseño, para desarrollar aplicaciones multimedia de forma metodológica. OOHDM ha evolucionado bastante desde su nacimiento. También se ha utilizado por sus autores para el desarrollo de aplicaciones en la Web.

OOHDM como ya se comentó es una metodología de desarrollo para aplicaciones multimedia. Antes de comenzar a detallar cada una de las fases que propone, es necesario resaltar algunas de sus características.

La primera de ellas es que OOHDM está basada en el paradigma de la orientación a objetos. En esto se diferencia de su antecesor HDM.

Otra característica de OOHDM es que, a diferencia de HDM, no sólo propone un modelo para representar a las aplicaciones multimedia, sino que propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo.

Fundamentalmente OOHDM toma como partida el modelo de clases que se obtiene en el análisis del Proceso Unificado de UML (*“Unified Modeling Language”*, Lenguaje de modelado unificado). A este modelo lo denomina modelo conceptual.

Partiendo de este modelo conceptual, OOHDM propone añadir características que permitan incorporar a esta representación del sistema todos los aspectos propios de las aplicaciones multimedia. En una segunda etapa de diseño, se parte de ese modelo conceptual y se añade a éste todos los aspectos de navegación, obteniéndose un nuevo modelo de clases denominado modelo navegacional. Por último, este modelo sirve como base para definir lo que en el argot

de OOHDm se denomina modelo de interfaz abstracta. El modelo de interfaz abstracta representa la visión del sistema que tendrá cada usuario del mismo.

OOHDm como técnica de diseño de aplicaciones hipermedia, propone un conjunto de tareas que pueden resultar costosas a corto plazo, pero a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reutilización de diseño, y así simplificar el costo de evoluciones y mantenimiento.

Esta metodología plantea el diseño de una aplicación de este tipo a través de cinco fases que se desarrollan de un modo iterativo. Estas fases son:

- Determinación de Requerimientos
- Diseño Conceptual
- Diseño Navegacional
- Diseño de Interfaz Abstracto
- Implementación

NetBeans

NetBeans [30] es un entorno de desarrollo integrado libre, desarrollado principalmente para el lenguaje de programación Java. Es un proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento.

Su plataforma permite desarrollar aplicaciones a partir de un conjunto de módulos, que pueden implementarse independientemente. Las aplicaciones construidas a partir de estos se extienden agregándoles nuevos módulos.

Las aplicaciones son capaces de instalar módulos dinámicamente; e incluso algunas agregan un módulo de actualización para permitir a los usuarios descargar, a su vez, actualizaciones de firma digital y nuevas características directamente dentro de la aplicación en ejecución; reinstalando una actualización o una nueva versión, no obligando a los usuarios a descargar toda la aplicación de nuevo.

Esta plataforma ofrece servicios reutilizables comunes para las aplicaciones de escritorio, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. Entre sus características destacan: gestión de la interfaz de usuario (menús y barras de herramientas);

gestión de configuración de usuario; gestión de almacenamiento; gestión de ventana; marco asistente; biblioteca visual de NetBeans y herramientas de desarrollo integrado.

PostgreSQL

PostgreSQL [31] es un sistema de gestión de bases de datos relacional orientado a objetos y libre, sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Algunas de sus principales características son:

- Alta concurrencia.
- Amplia variedad de tipos nativos.
- Claves ajenas también denominadas llaves ajenas o claves foráneas.
- Cuenta con: disparadores, vistas, integridad transaccional, herencia de tablas.

Sus ventajas son:

- Es gratuito.
- Hace más sencillo el análisis de datos.
- Tiene mejor soporte que los proveedores comerciales.
- Diseñado para ambientes de alto volumen.
- Herramientas gráficas de diseño y administración de bases de datos.
- Tiene una mayor seguridad.
- Es multiplataforma.

SURF

SURF [32] es otro de los algoritmos más utilizado en la literatura para la extracción de puntos de interés invariantes. La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen. Posteriormente se representa la vecindad del punto de interés como un vector de características (por defecto con un tamaño

de 64, aunque es posible aumentarlo). El algoritmo SURF utiliza una aproximación básica de la matriz *Hessiana* para reducir el tiempo de computación. La matriz *Hessiana* se utiliza debido a su buena relación entre la precisión y el costo temporal.

La característica principal de los puntos de interés SURF es la repetibilidad, si el punto es considerado fiable, el detector encontrará el mismo punto bajo distintas perspectivas (diferente escala, orientación, entre otras).

BoofCV

BoofCV [33] es una biblioteca Java de código abierto para aplicaciones de visión artificial y robótica en tiempo real. Escrito desde cero para facilidad de uso y alto rendimiento. Su funcionalidad cubre una amplia gama de temas, incluyendo rutinas optimizadas de procesamiento de imágenes de bajo nivel, calibración de cámaras, detección / seguimiento de características, estructura desde el movimiento y reconocimiento. BoofCV fue lanzado bajo una licencia Apache 2.0 para uso académico y comercial.

BoofCV se organiza en varios paquetes: procesamiento de imágenes, características, visión geométrica, calibración, reconocimiento, visualización e IO (*“Input/ Output”*, Entrada / Salida). El procesamiento de imágenes contiene funciones de procesamiento de imágenes de uso común que operan directamente en píxeles. El paquete de características contiene algoritmos de extracción de características para usar en operaciones de nivel superior.

Capítulo 3. Aplicación de la metodología

En este capítulo se muestra el desarrollo de la aplicación siguiendo las fases que componen la metodología OOHDm mostrada en el Capítulo 2.

3.1. Determinación de requerimientos

Su objetivo es encontrar los requisitos funcionales de la aplicación Web para representarlos como casos de uso, los cuales se muestran en la Figura 3.1.

Es mediante este diagrama de casos de uso que se representan las interacciones de los actores con la aplicación identificando así los requerimientos funcionales.

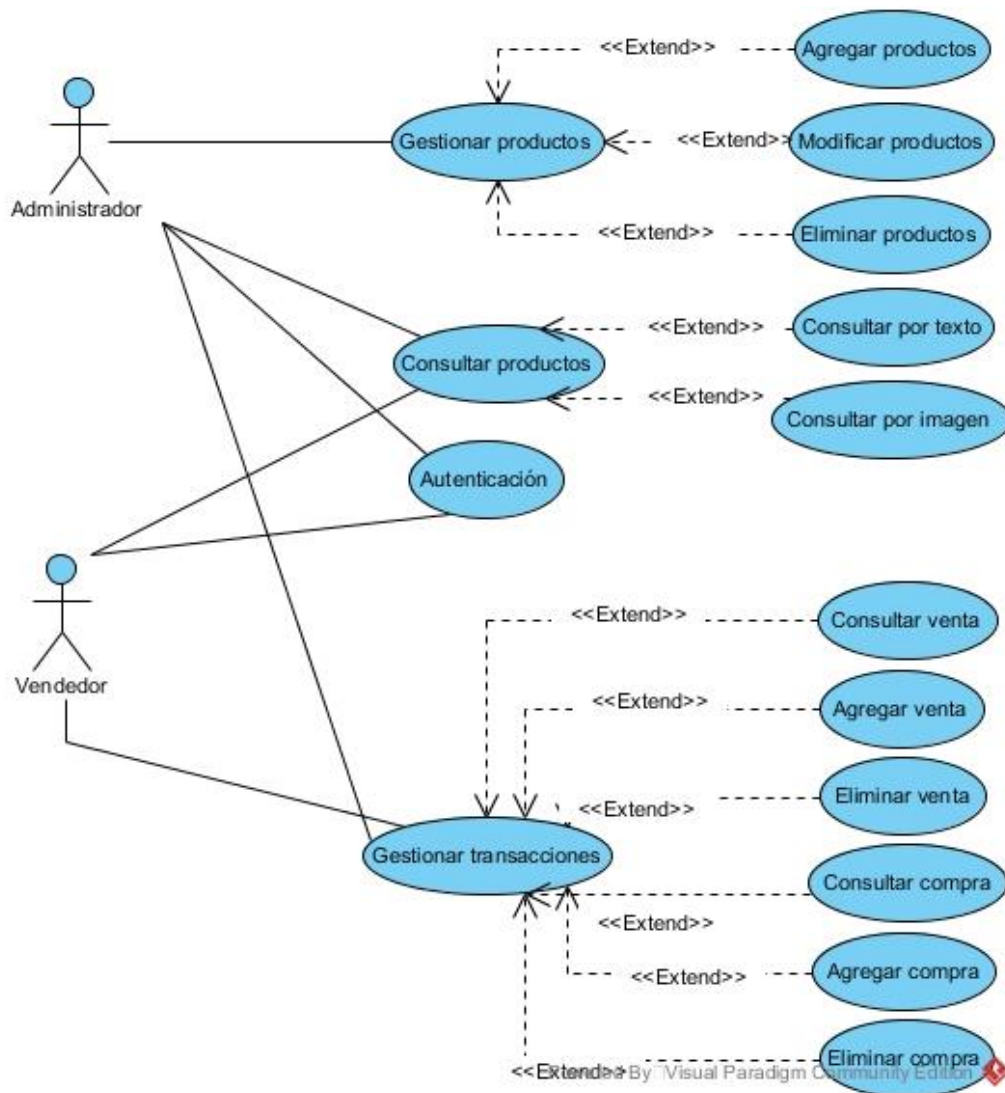


Figura 3.1 Diagrama de casos de uso

Los actores *Administrador* y *Vendedor* deben autenticarse en el sistema para realizar sus operaciones, de igual forma pueden finalizar su sesión en cualquier momento. El *Administrador* es el encargado de gestionar los productos, por lo que tiene permisos para insertar uno nuevo y modificar, consultar o eliminar uno existente. El *Vendedor* junto con el *Administrador* gestionan la información con las transacciones que se realicen, ya sea una compra de productos o una venta de los mismos, por último, la búsqueda de los productos por texto o por imagen es posible para ambos.

3.2. Diseño conceptual

Durante el diseño conceptual se construyen las relaciones y colaboraciones existentes en los objetos del dominio.

3.2.1. Diagrama de clases

El diagrama de clases del sistema representado en la Figura 3.2 fue diseñado para corresponder con la base de datos que ya existía para la empresa AGROMAQ.

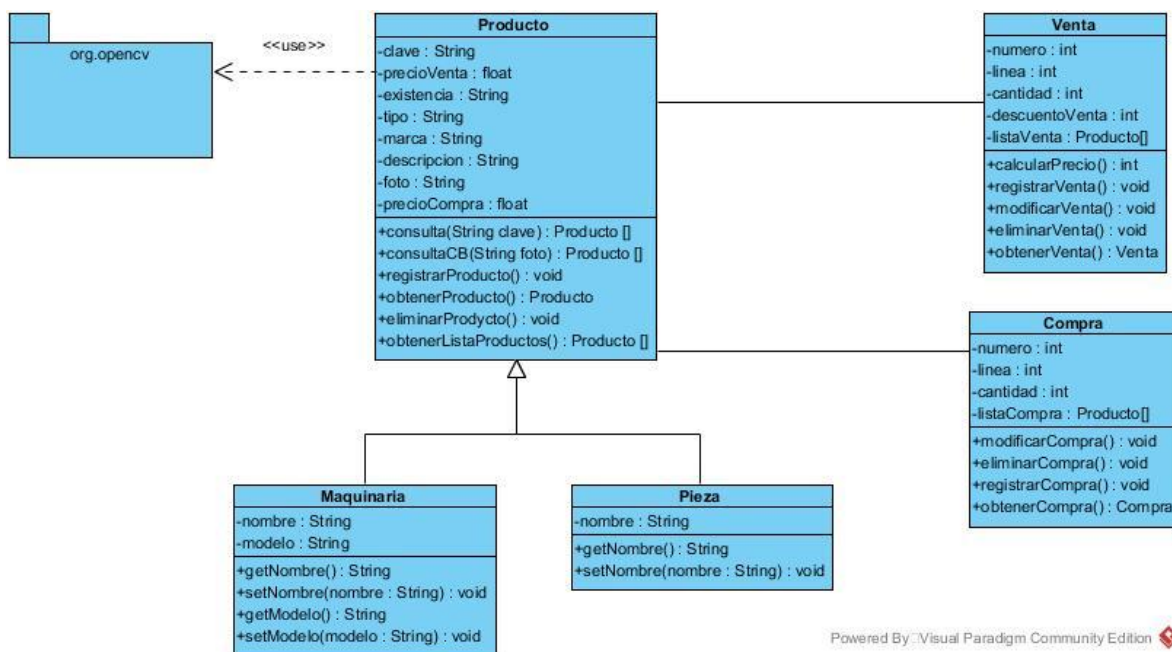


Figura 3.2 Diagrama de clases

3.2.2. Modelos de base de datos

El modelo lógico de la base de datos se muestra en la Figura 3.3.

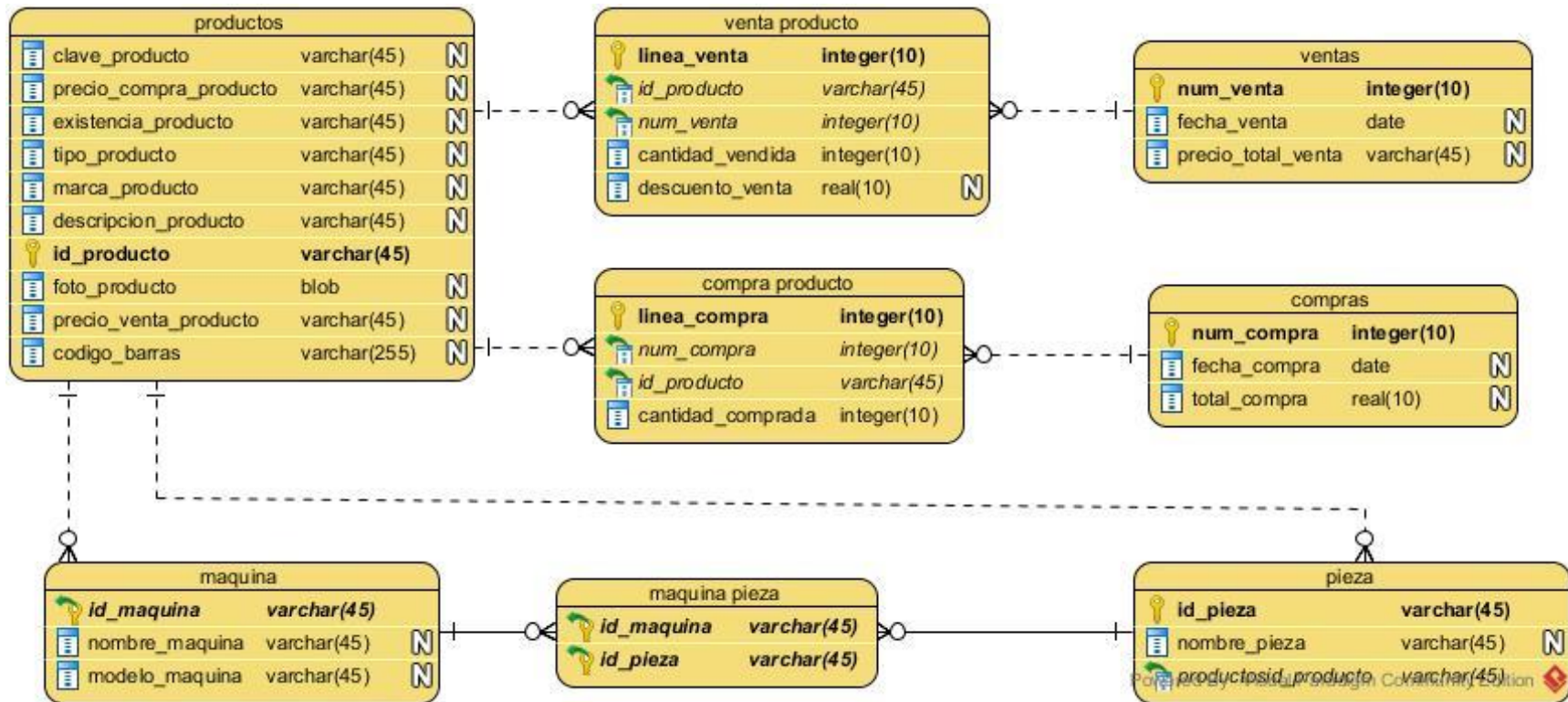


Figura 3.3 Modelo lógico de la base de datos

El modelo conceptual de la base de datos se visualiza en la Figura 3.4.

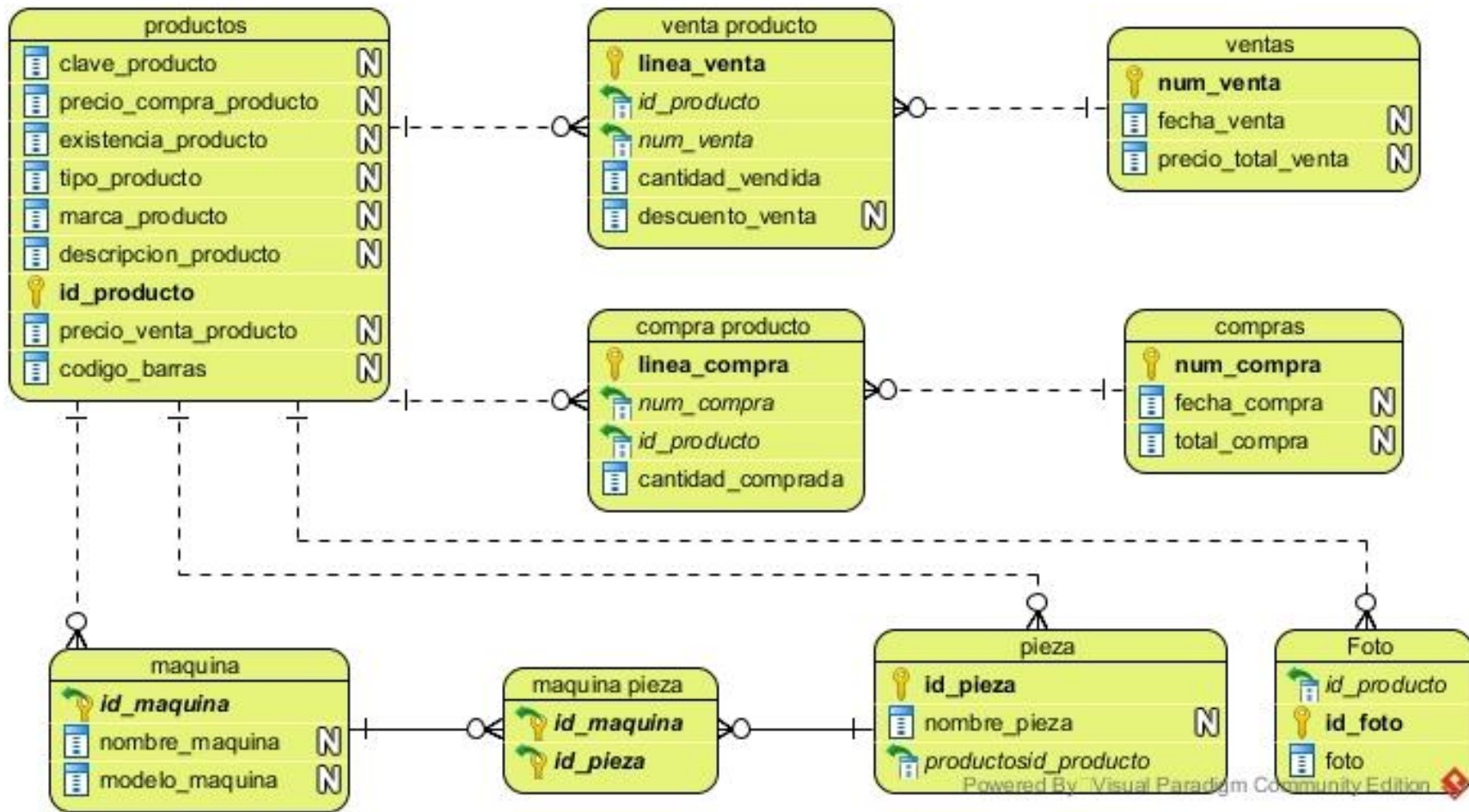


Figura 3.4 Modelo conceptual de la base de datos

El modelo físico de la base de datos se muestra en la Figura 3.5.

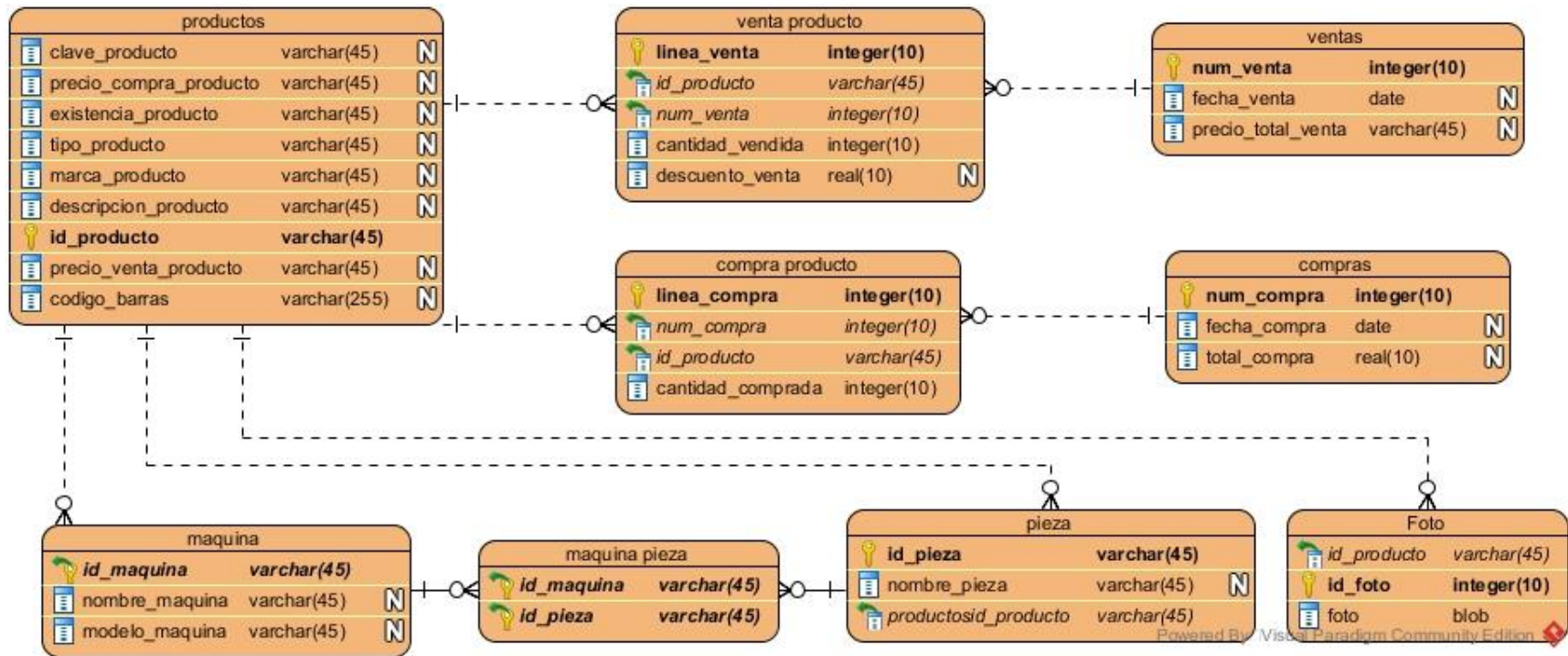


Figura 3.5 Modelo físico de la base de datos

3.3. Diseño Navegacional

En esta fase se debe tener en mente que la generación de aplicaciones Web fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermedia.

3.3.1. Arquitectura del sistema

La arquitectura del sistema se diseñó con base en el patrón arquitectónico MVC (*“Model View Control”*, Modelo Vista Controlador), la cual distribuye los componentes del sistema de tal forma que facilite su mantenimiento y se representa de manera abstracta en la Figura 3.6.

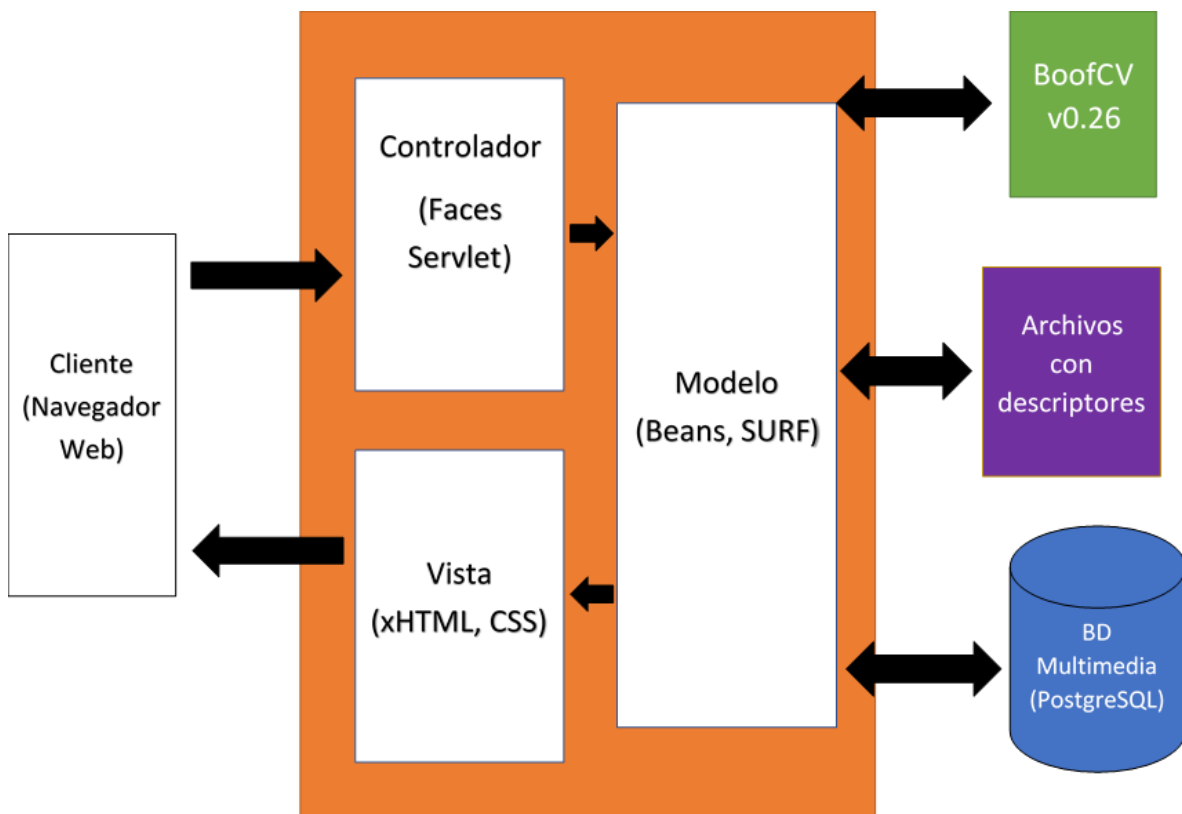


Figura 3.6 Arquitectura del sistema

3.3.2. Patrón MVC

El patrón arquitectónico MVC define tres capas dentro de la aplicación separando la lógica de negocio de la representación y la persistencia, como se observa en la Figura 3.7.

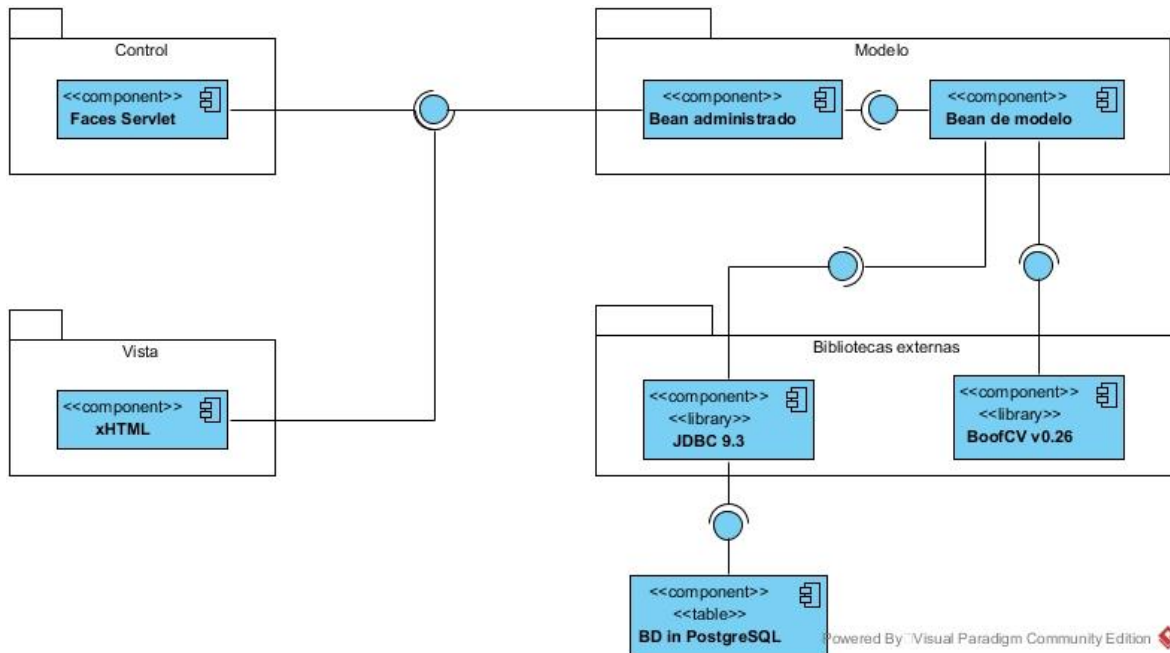


Figura 3.7 Arquitectura de la aplicación

Modelo: Agrupa la lógica del sistema representada en los *beans* administrados que tienen acceso a los componentes de la interfaz y pasan información a los *beans* de modelo, estos últimos representan las clases importantes del dominio y usan las bibliotecas, como la de BoofCV para realizar la búsqueda basada en contenido (imágenes) y de JDBC para controlar el acceso al gestor de base de datos PostgreSQL y manipular la información.

Vista: Mediante archivos xHTML (“*eXtensible HyperText Markup Language*”, Lenguaje de Marcado de Hipertexto Extensible) se representa el modelo y maneja la interacción con el usuario; para este propósito se utilizan etiquetas propias de JSF (“*JavaServer Faces*”), en conjunto de hojas de estilo (CSS) para dar una mejor visualización al usuario.

Controlador: El servlet de JSF es el vínculo entre el modelo y la vista. Se ocupa de gestionar las peticiones de los recursos accediendo al modelo requerido en la petición del usuario y seleccionando la vista adecuada para representarlo.

Como se observa, la estructura y organización que propone el MVC proporciona un buen acoplamiento entre los componentes y los cambios solo serán perceptibles para las partes directamente involucradas.

3.3.3. Diagrama de secuencia

En la Figura 3.8 se ilustra el proceso de gestión de los productos que va desde el registro hasta la modificación y eliminación de los mismos.

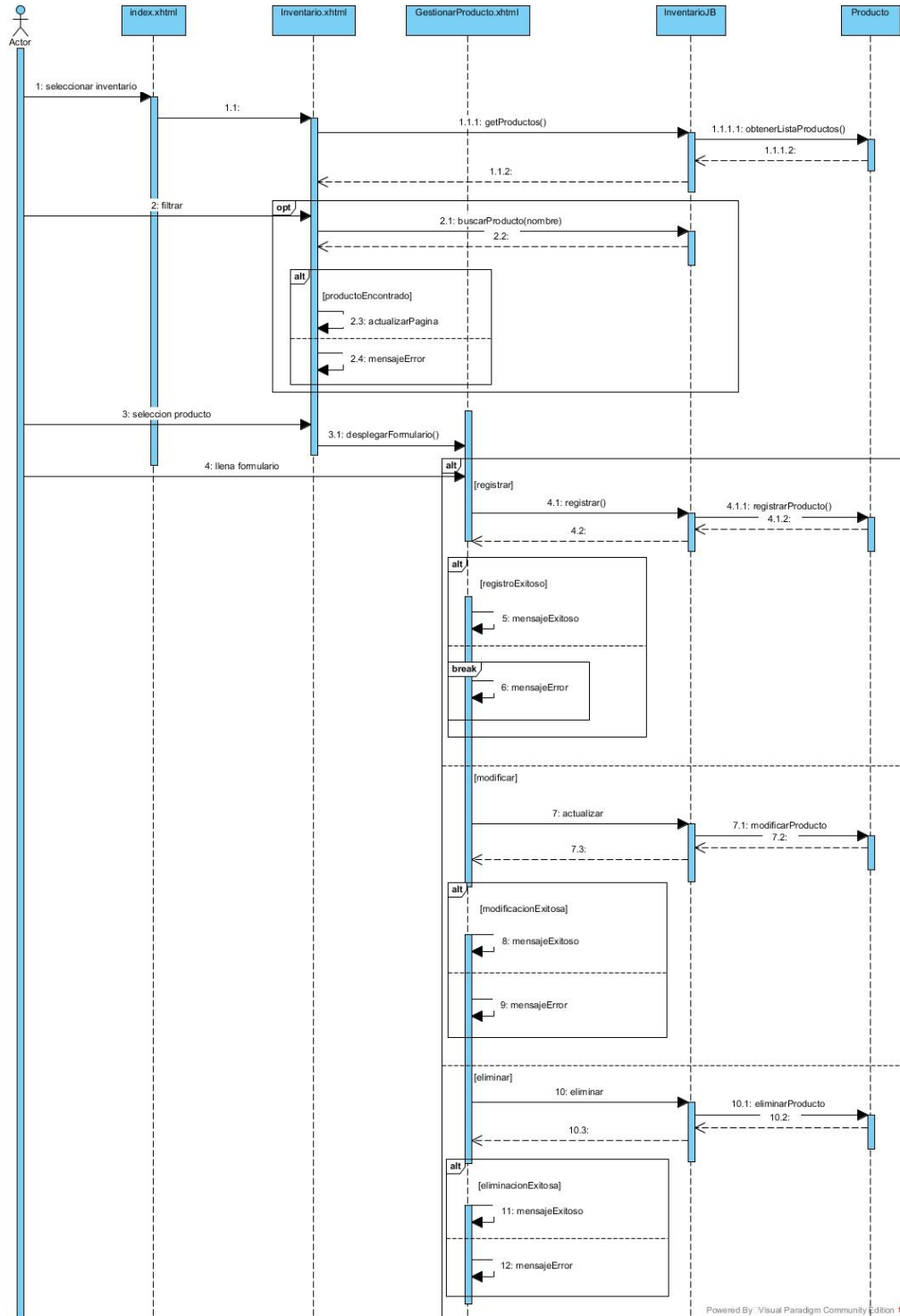


Figura 3.8 Diagrama de secuencia de gestión de producto

En la Figura 3.9 se muestra la forma en la que el sistema realiza la búsqueda de los productos, ya sea por un medio de texto o por imágenes similares a la imagen consulta dada por el usuario.

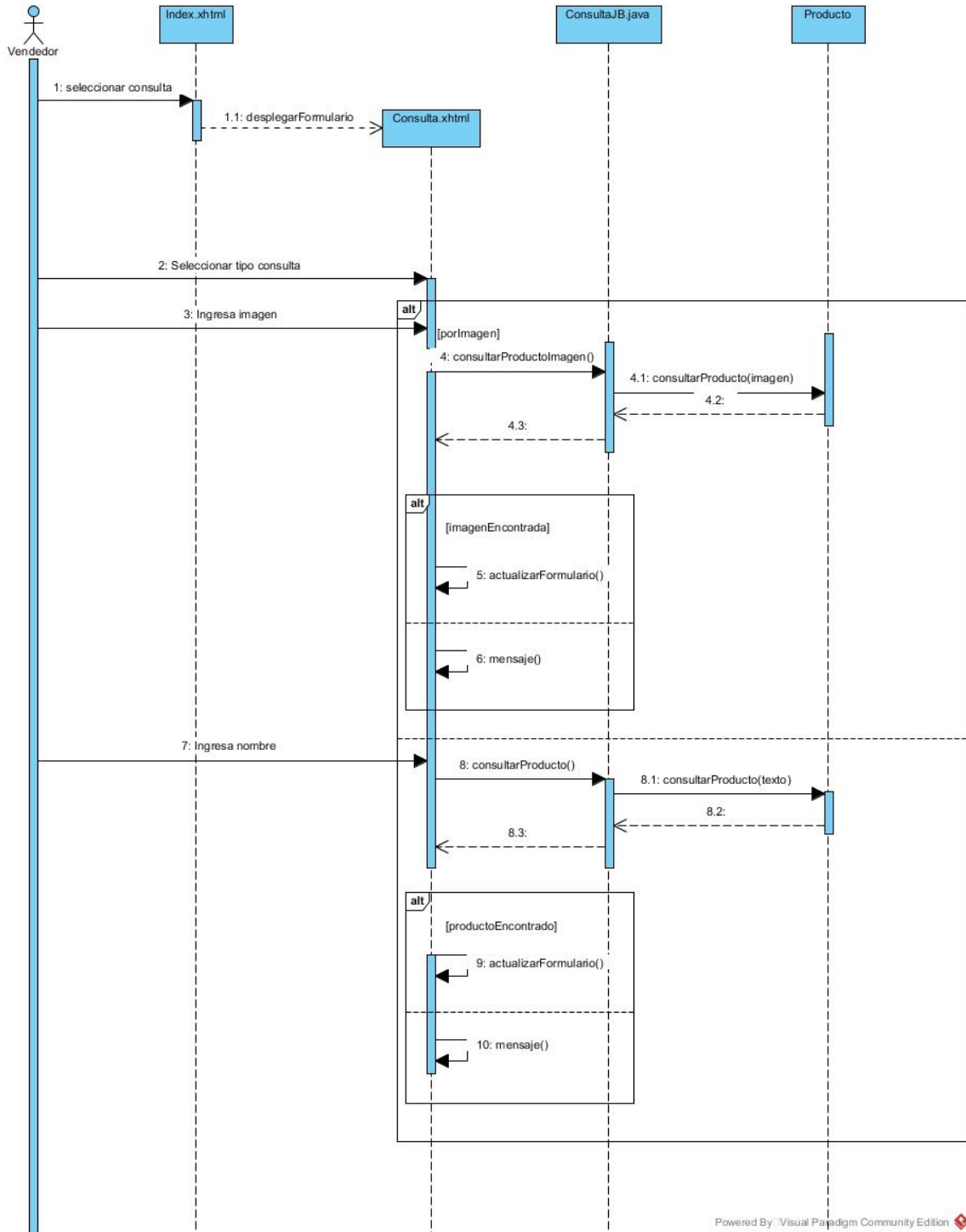


Figura 3.9 Diagrama de secuencia de consulta de texto y basada en contenido

3.4. Proceso CBIR

3.4.1 Registro de producto

La Figura 3.10 representa el proceso que sigue el sistema para la extracción y almacenaje de los descriptores cada vez que una imagen es registrada para ahorrar tiempo en la búsqueda.

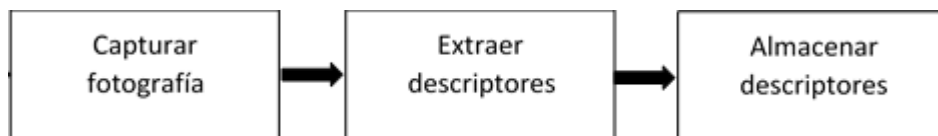


Figura 3.10 Proceso durante el registro de imágenes

A continuación, se explica de una manera más detallada el proceso de registro de un producto:

- 1) El usuario toma una o varias fotografías al producto a registrar y las sube al sistema.
- 2) Una vez que se registra el producto, el sistema extrae los puntos de interés de cada una de las imágenes ingresadas, después de eso extrae los descriptores con la ayuda de SURF. Como se visualiza en la Figura 3.11 en la línea 171 el sistema extrae los puntos de interés y calcula los descriptores de las imágenes (una a la vez como se muestra en la línea 166).

```
164 public void registrar(String nombreImagen){
165
166     BufferedImage imagenBD = UtilImageIO.loadImage(UtilIO.pathExample("C:\\imagenes\\"+nombreImagen+".jpg"));
167
168     ImageGray inputA = ConvertBufferedImage.convertFromSingle(imagenBD, null, GrayF32.class);
169     FastQueue descA = UtilFeature.createQueue(detDesc, 100);
170     List<Point2D_F64> pointsA = new ArrayList<>();
171     detectarDescriptores(inputA, pointsA, descA); //extracción y detección de los descriptores
```

Figura 3.11 Fragmento de código del proceso de registro 1

- 3) Por último, estos descriptores son almacenados en forma de objetos en archivos (.txt) para su futuro uso como se visualiza en la Figura 3.12.

```
172
173     AdmArchivos fileObj = new AdmArchivos(); //clase para el manejo de archivos
174     FastQueue descB = UtilFeature.createQueue(detDesc, 100);
175     fileObj.guardar1(descB, nombreImagen);
176
177 }
```

Figura 3.12 Fragmento de código del proceso de registro 2

3.4.2. Búsqueda de producto

La Figura 3.13, por otra parte, representa al proceso que sigue el sistema para la imagen consulta y las imágenes en la base de datos durante el proceso de búsqueda.

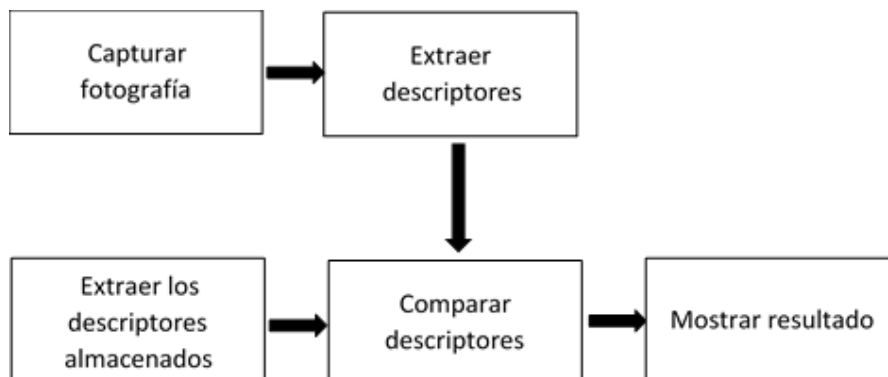


Figura 3.13 Proceso durante la búsqueda de imágenes

A continuación, se explica de una manera más detallada el proceso de búsqueda de un producto:

1) El usuario debe ingresar una fotografía del producto a buscar, en la Figura 3.14 se visualiza cómo se le extraen los puntos de interés y se calculan los descriptores con SURF (línea 74).

```
68 private Map<String, Integer> emparejar(BufferedImage imageBusqueda, String[] listaNombreProd, Producto pB) {
69
70     Map<String, Integer> listaProductos = new LinkedHashMap<>();
71     ImageGray entradaA = ConvertBufferedImage.convertFromSingle(imageBusqueda, null, GrayF32.class);
72     FastQueue descA = UtilFeature.createQueue(detDesc, 100); //Descriptores de la imagen buscada
73     List<Point2D_F64> pointsA = new ArrayList<>(); //puntos de interes de la imagen buscada
74     detectarDescriptores(entradaA, pointsA, descA);
75     pB.setPuntosInteres(descA.getSize());
```

Figura 3.14 Fragmento de código del proceso de búsqueda 1

2) Después el sistema procederá a extraer los descriptores almacenados en objetos de cada una de las imágenes registradas en la base de datos como se muestra en la Figura 3.15, donde se aprecia cómo con sólo una lista de nombres extraen los descriptores (línea 81) sin tener que acceder a la base de datos.

```
76
77     for (String p : listaNombreProd) {
78
79         AdmArchivos fileObjb = new AdmArchivos();
80         FastQueue descB = UtilFeature.createQueue(detDesc, 100);
81         fileObjb.extraer1(descB,p); //extracción del descriptor de la "imagen en BD" desde un archivo
82
```

Figura 3.15 Fragmento de código del proceso de búsqueda 1

3) Una vez se tengan los descriptores tanto de las imágenes en la base de datos como de la imagen consulta, se procede a realizar la comparación de esta última con cada una de las imágenes de la base de datos para así obtener los puntos donde el emparejamiento es correcto. En la Figura 3.16 se muestra (línea 85) cómo se comparan los descriptores de la imagen búsqueda (A) y los de la imagen en la base de datos (B) y el resultado es almacenado para su uso posterior (línea 86).

```
83 |         associate.setSource(descA);  
84 |         associate.setDestination(descB);  
85 |         associate.associate();//se comparan los descriptores de ambas imágenes  
86 |         listaProductos.put(p, associate.getMatches().size);  
87 |     }
```

Figura 3.16 Fragmento de código del proceso de búsqueda 3

4) Finalmente se procede a ordenar las imágenes con respecto a la cantidad de emparejamientos correctos como se muestra en la Figura 3.17.

```
89 |     List<Entry<String, Integer>> entryList = new ArrayList<>(listaProductos.entrySet());  
90 |     Collections.sort(entryList, new Comparator<Entry<String, Integer>>() {  
91 |         @Override  
92 |         public int compare(Entry<String, Integer> obj1, Entry<String, Integer> obj2) {  
93 |             return obj2.getValue().compareTo(obj1.getValue());  
94 |         }  
95 |     });  
96 |     listaProductos.clear();  
97 |  
98 |     for (int i=0;i<kNN;i++) {  
99 |         listaProductos.put(entryList.get(i).getKey(), entryList.get(i).getValue());  
100 |     }  
101 |  
102 |     return listaProductos;  
103 | }
```

Figura 3.17 Fragmento de código del proceso de búsqueda 4

Capítulo 4. Resultados

Como parte del trabajo realizado se desarrolló una aplicación Web que permite tener un control del inventario, así como la búsqueda de productos ya sea de manera convencional como es la búsqueda por texto y por medio de consultas con imágenes, este proceso de búsqueda se realiza mediante la detección y extracción de puntos de interés tanto en la imagen consulta como en las imágenes almacenadas en la base de datos, esto gracias al descriptor SURF que forma parte de la biblioteca de BoofCV.

En este capítulo se describe el funcionamiento de la aplicación con base en los dos casos de estudio propuestos, en el primero de ellos se realizan consultas con imágenes borrosas y en el segundo consultas con imágenes recortadas parcialmente, para así medir la precisión y el *recall* del sistema que son las medidas más ocupadas de acuerdo con el estado de arte revisado durante el capítulo 2.

4.1. Caso de estudio

Antes de realizar alguna acción dentro del sistema se inicia sesión ya sea como un vendedor o como el administrador, teniendo el primero algunas limitaciones en cuanto a las opciones disponibles, en la Figura 4.1 se muestra la interfaz para el inicio de sesión.

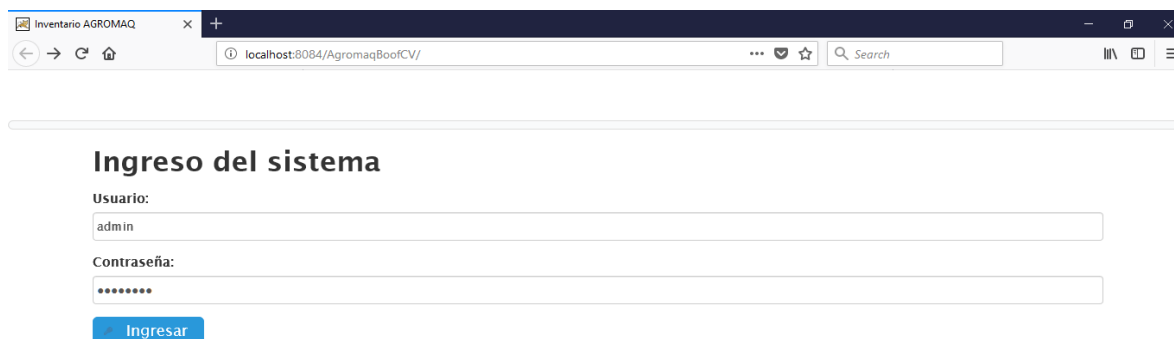


Figura 4.1 Autenticación de usuario

La sección ocupada para los casos de estudio es la de *Buscar productos por imagen* en donde se ingresa el número de imágenes similares a la imagen consulta que se deseen conseguir, así como la imagen consulta en sí, la Figura 4.2 muestra la interfaz para la búsqueda por imagen.

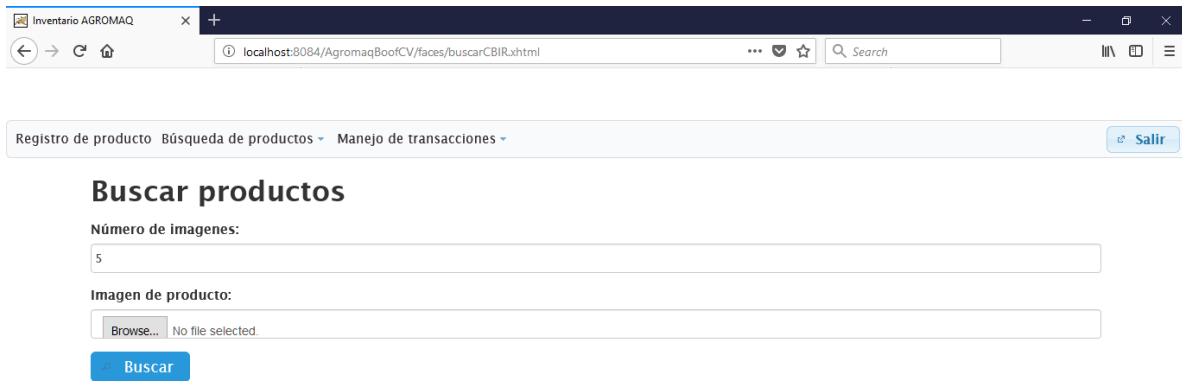


Figura 4.2 Interfaz de búsqueda por imagen

En la Figura 4.3 se visualiza la consulta de una imagen normal (sin recortes ni borrosa) como prueba del sistema y posteriormente se muestran sus resultados en la Figura 4.4, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos y se aprecia que el porcentaje de similitud es del 100% en las imágenes recuperadas.

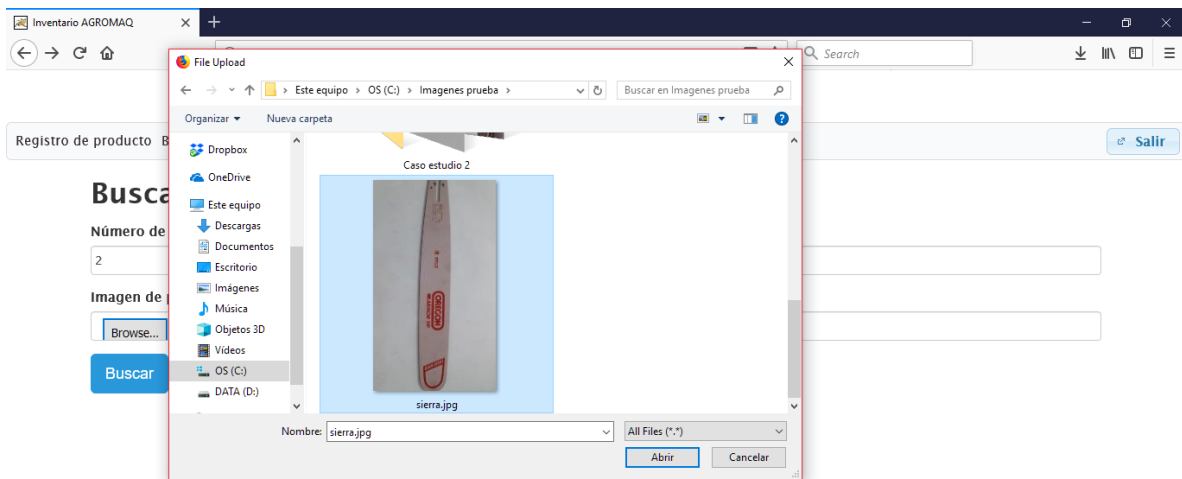


Figura 4.3 Prueba de búsqueda de imagen

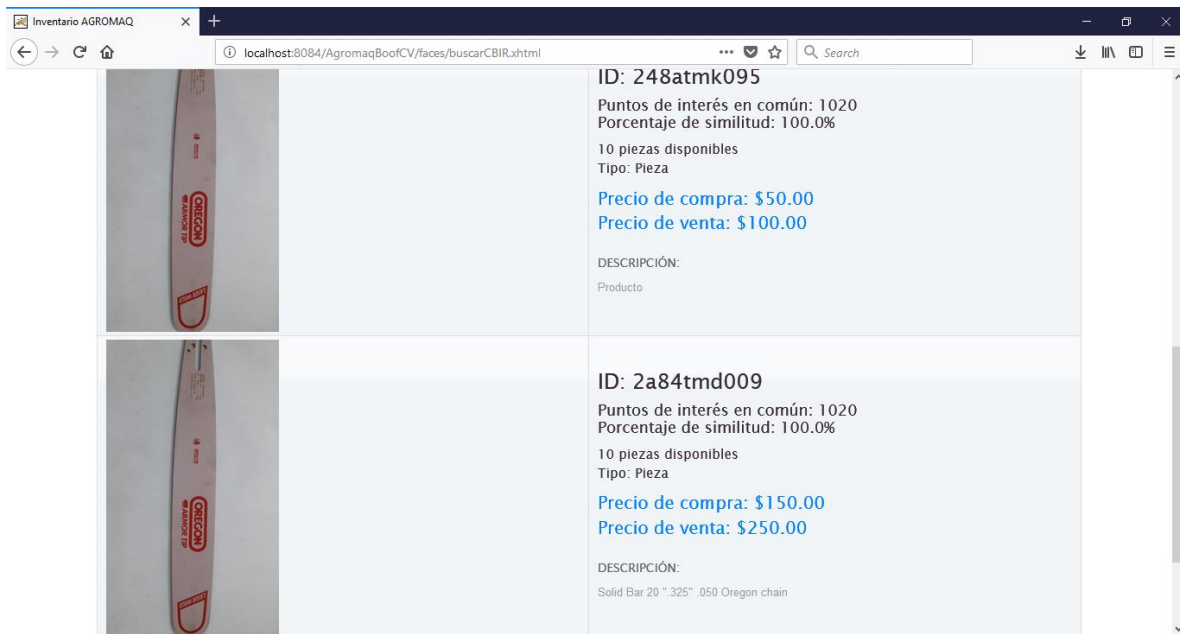


Figura 4.4 Resultado de prueba

4.1.1. Caso de estudio 1

A continuación, se muestran los resultados de las 10 consultas realizadas para el primer caso de estudio, que como ya se mencionó anteriormente consiste en la búsqueda con imágenes borrosas como entrada.

En la Figura 4.5 se visualiza la imagen consulta 1 y posteriormente se muestran sus resultados en las Figuras 4.6 y 4.7, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

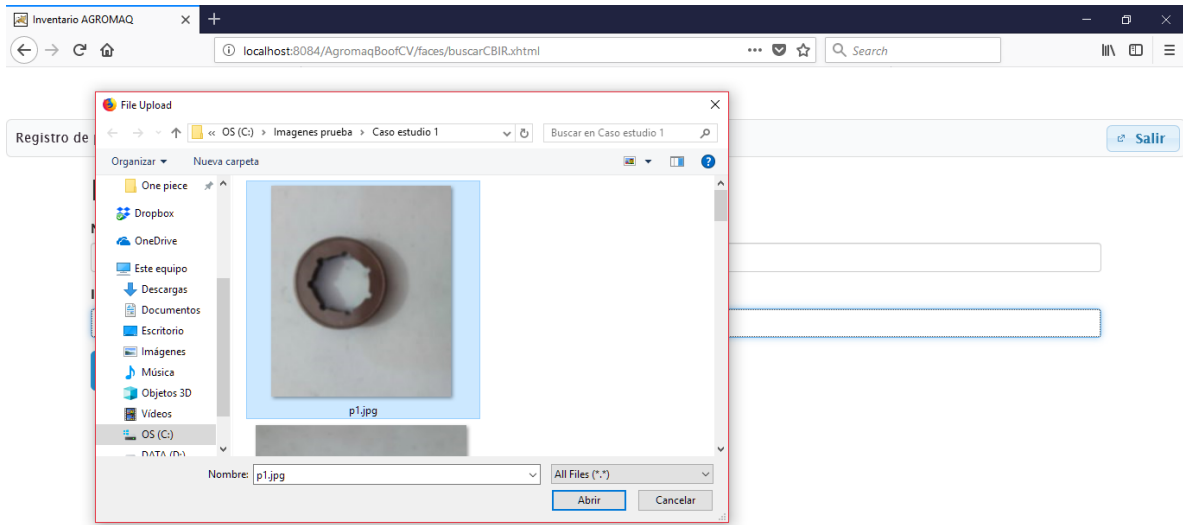


Figura 4.5 Caso de estudio 1 imagen de búsqueda 1

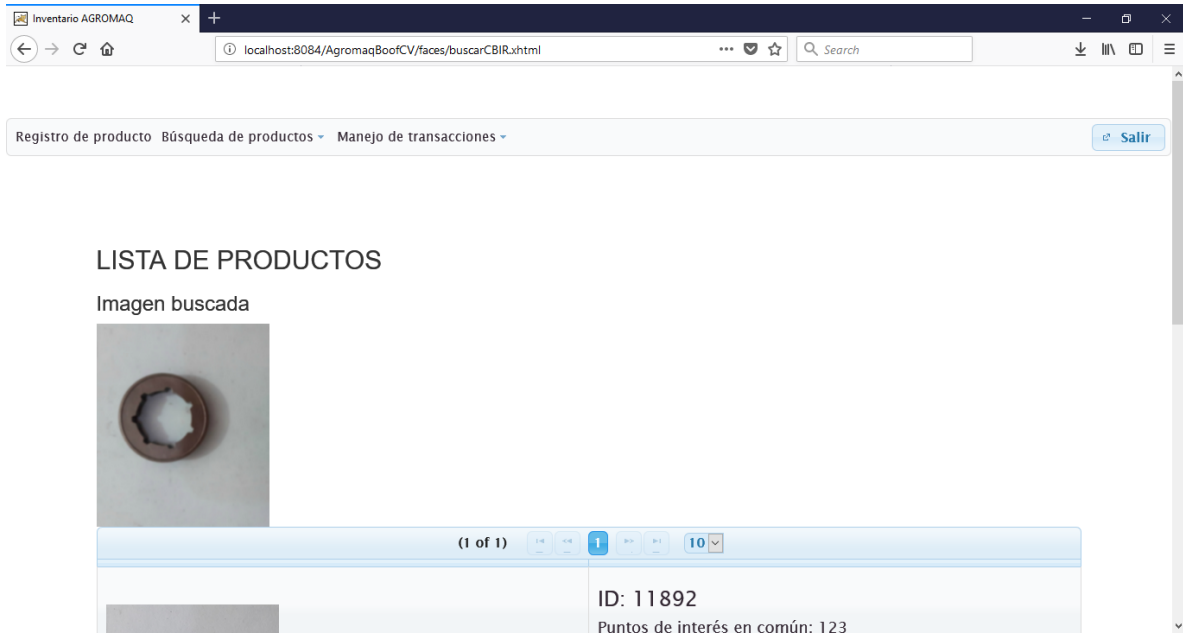


Figura 4.6 Caso de estudio 1 resultado 1 de búsqueda 1



Figura 4.7 Caso de estudio 1 resultado 2 de búsqueda 1

En la Figura 4.8 se observa la imagen consulta 2 y posteriormente se muestran sus resultados en las Figuras 4.9, 4.10 y 4.11, la consulta se realizó con $k=4$ y 2 como número de imágenes relevantes en la base de datos.

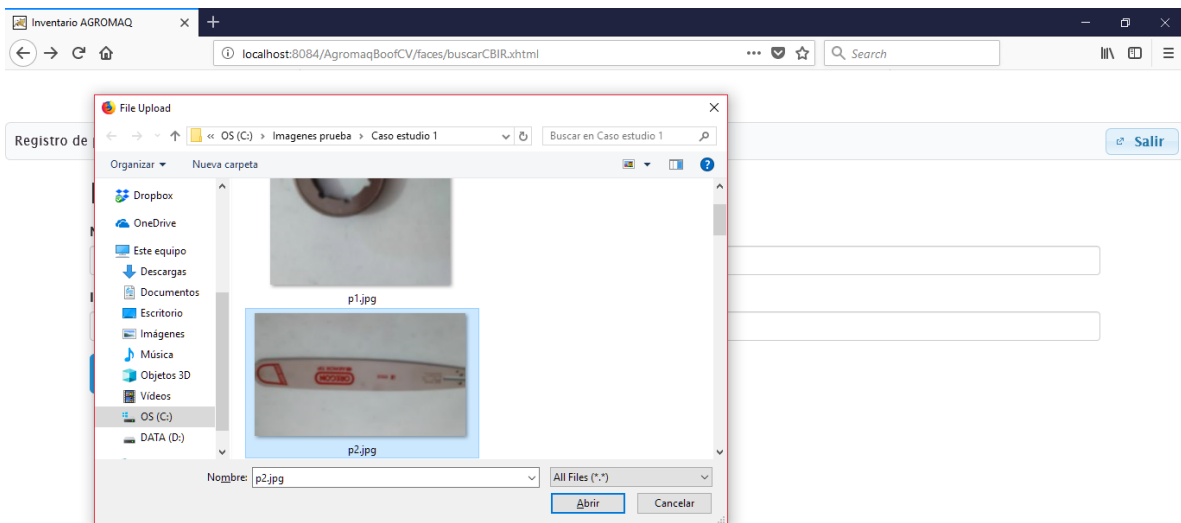


Figura 4.8 Caso de estudio 1 imagen de búsqueda 2

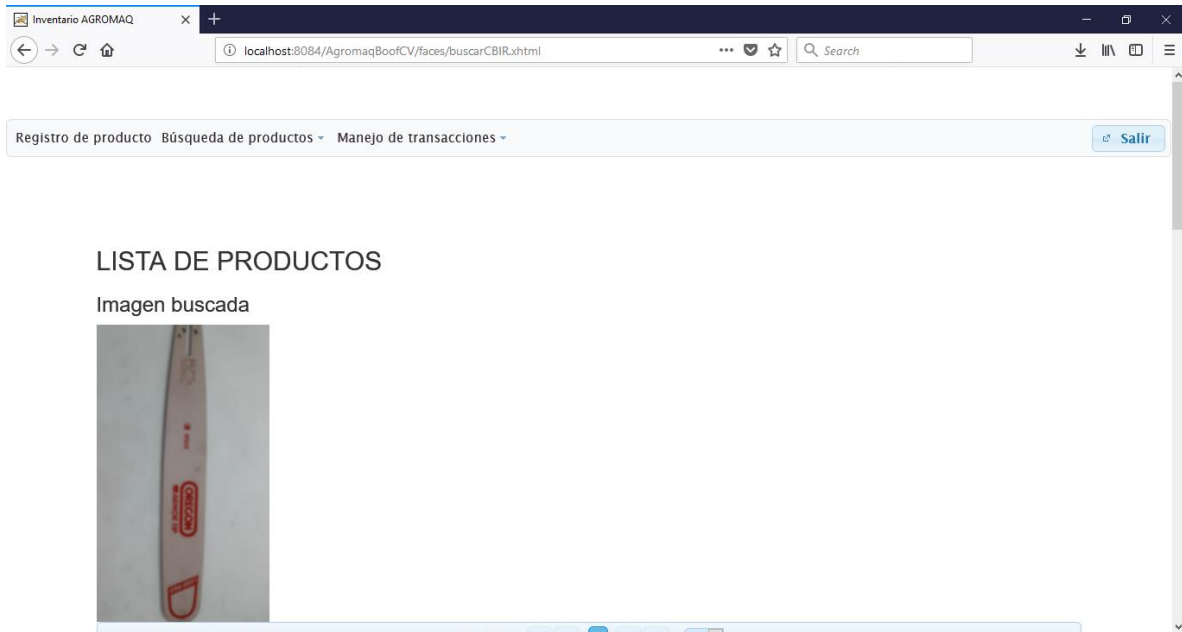


Figura 4.9 Caso de estudio 1 resultado 1 de búsqueda 2

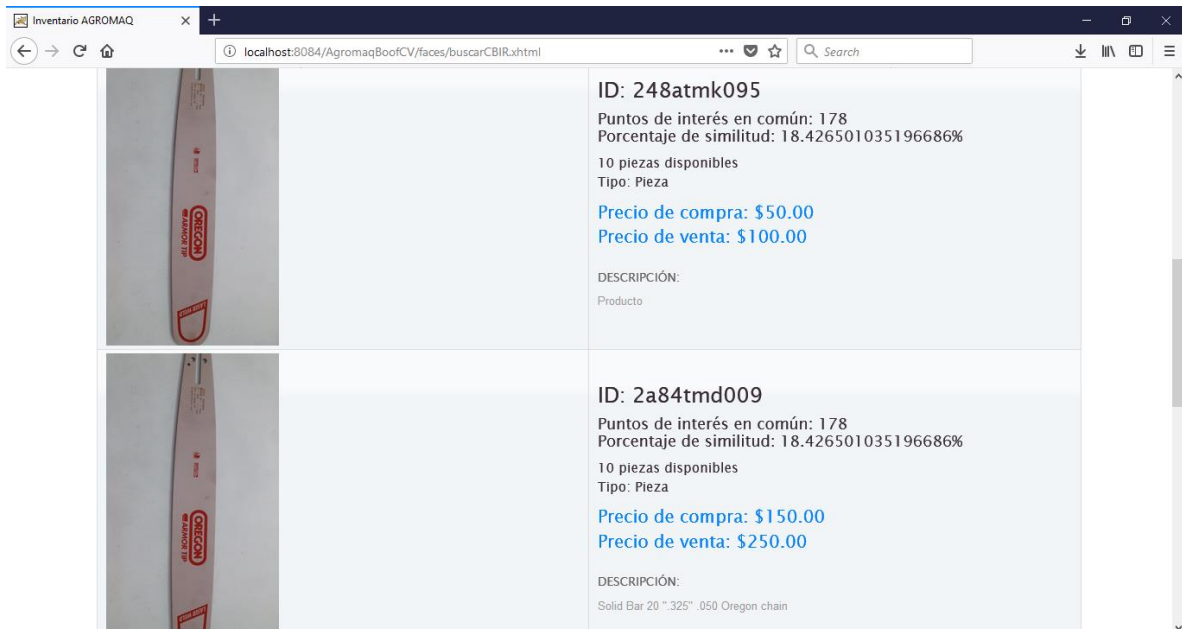


Figura 4.10 Caso de estudio 1 resultado 2 de búsqueda 2

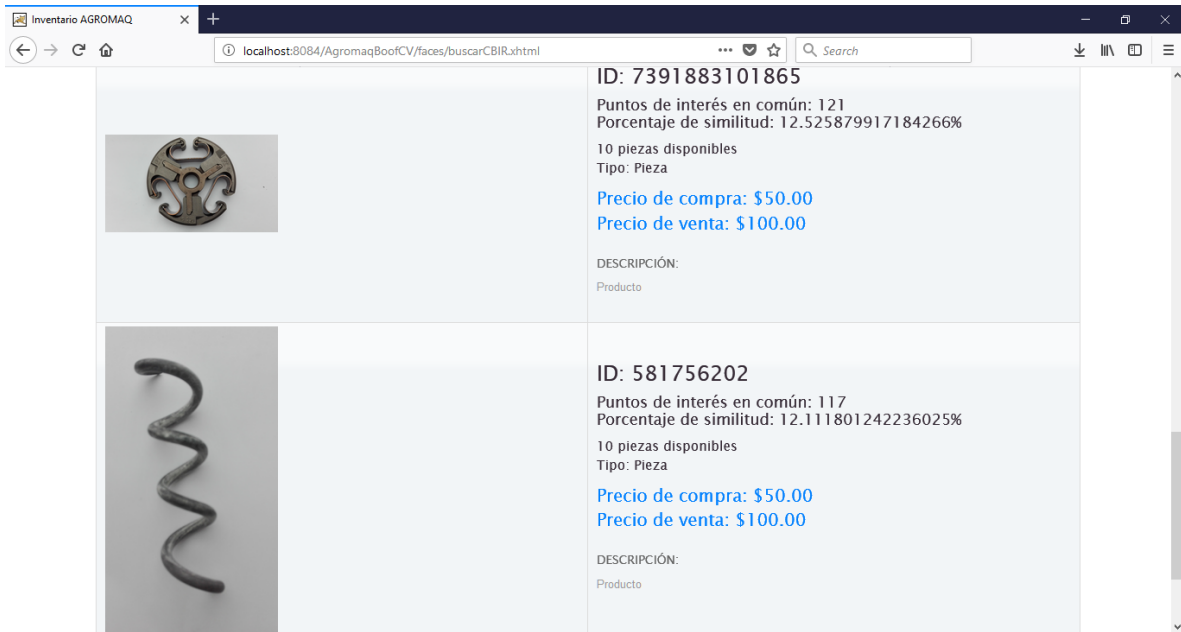


Figura 4.11 Caso de estudio 1 resultado 3 de búsqueda 2

En la Figura 4.12 se presenta la imagen consulta 3 y posteriormente se muestran sus resultados en las Figuras 4.13, 4.14 y 4.15, la consulta se realizó con $k=4$ y 2 como número de imágenes relevantes en la base de datos.

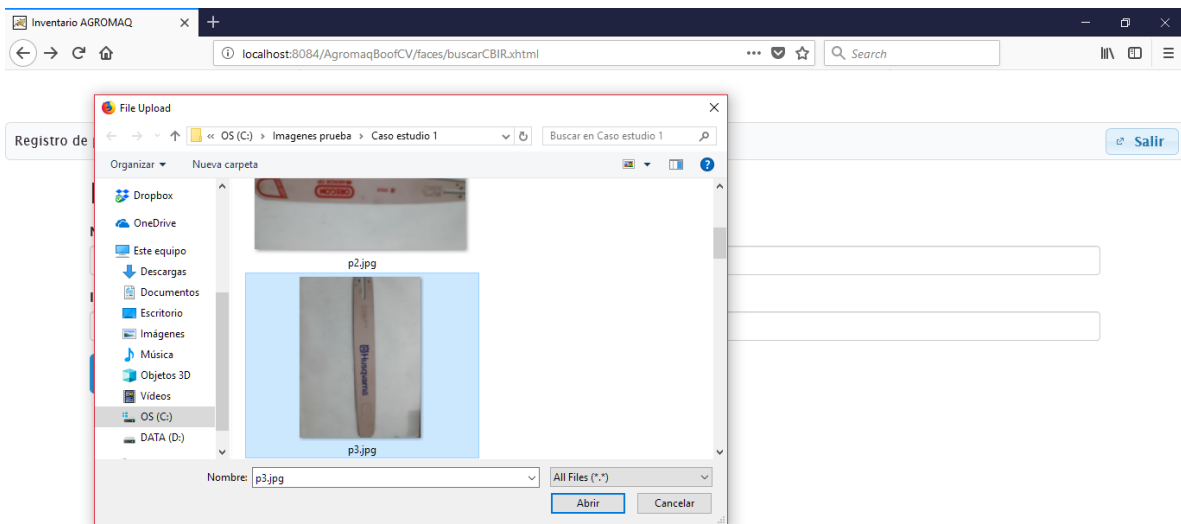


Figura 4.12 Caso de estudio 1 imagen de búsqueda 3

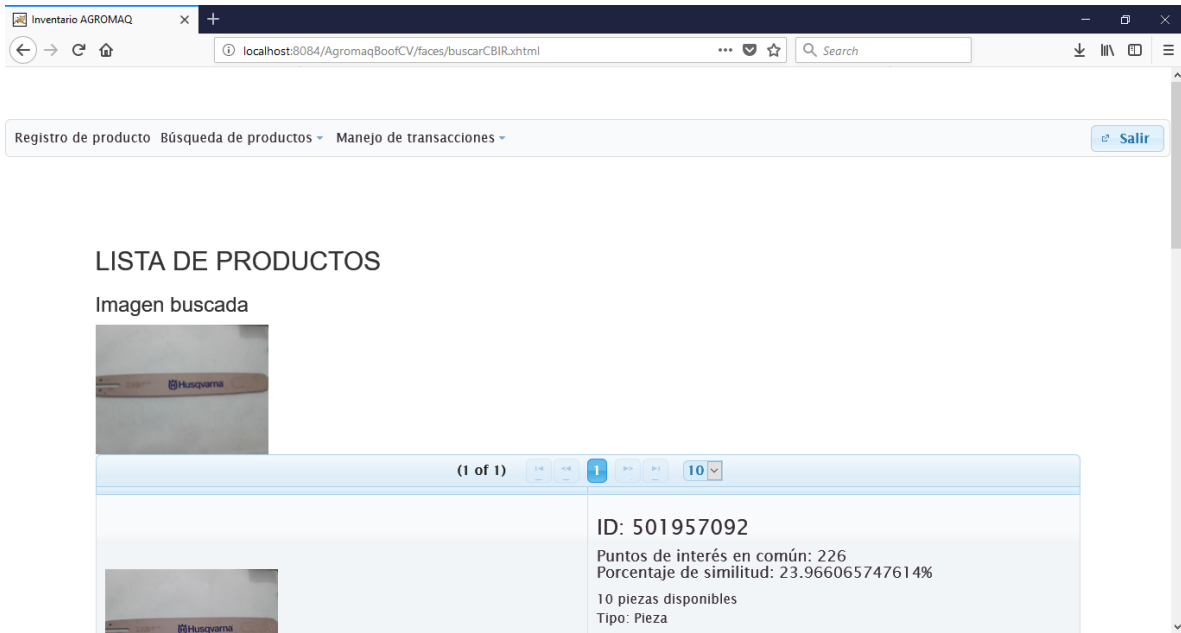


Figura 4.13 Caso de estudio 1 resultado 1 de búsqueda 3

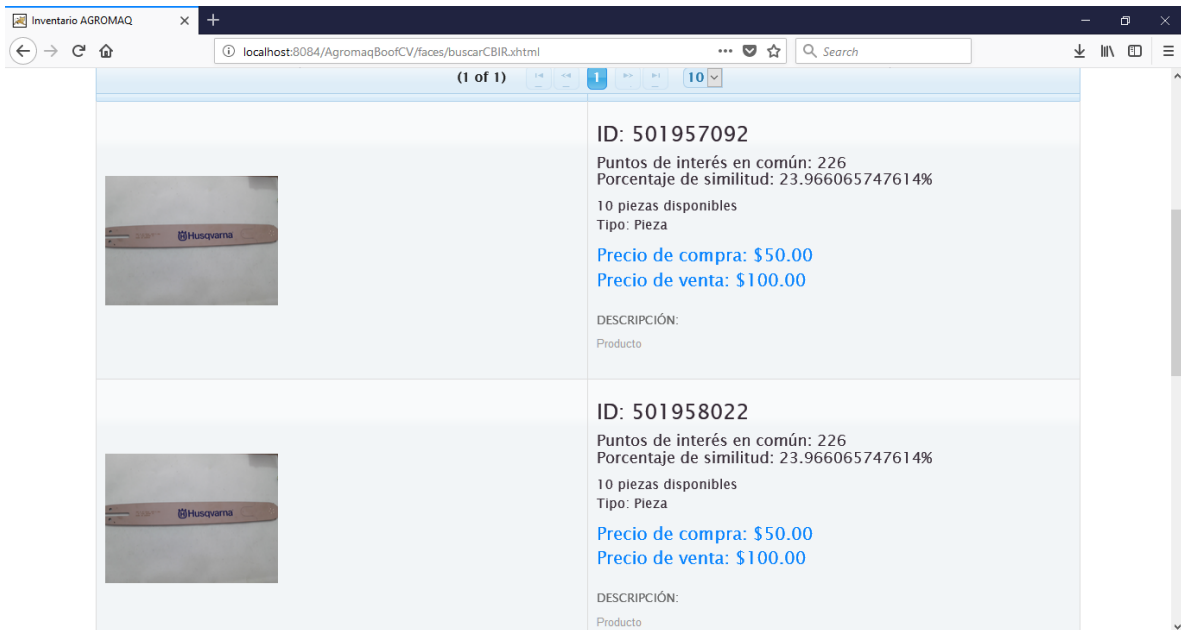


Figura 4.14 Caso de estudio 1 resultado 2 de búsqueda 3

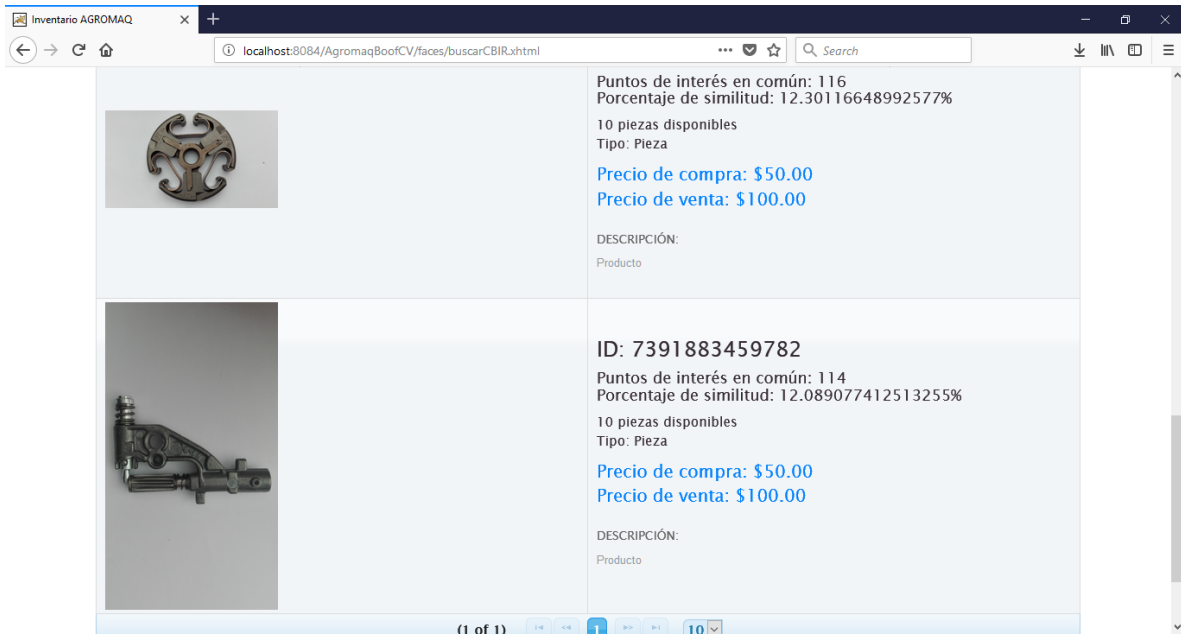


Figura 4.15 Caso de estudio 1 resultado 3 de búsqueda 3

En la Figura 4.16 se ilustra la imagen consulta 4 y posteriormente se muestran sus resultados en las Figuras 4.17, 4.18, 4.19 y 4.20, la consulta se realizó con $k=5$ y 4 como número de imágenes relevantes en la base de datos.

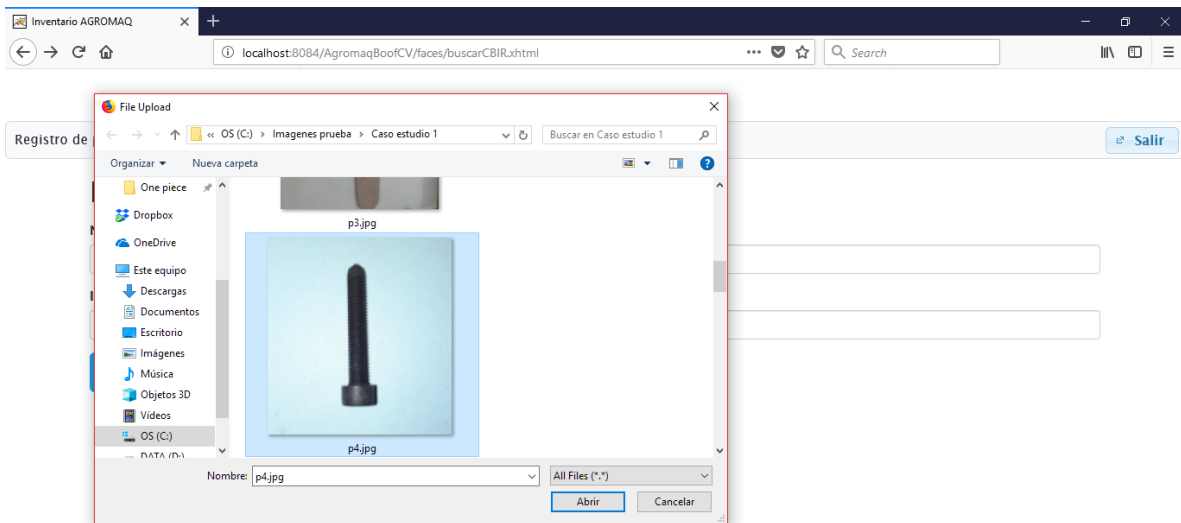


Figura 4.16 Caso de estudio 1 imagen de búsqueda 4

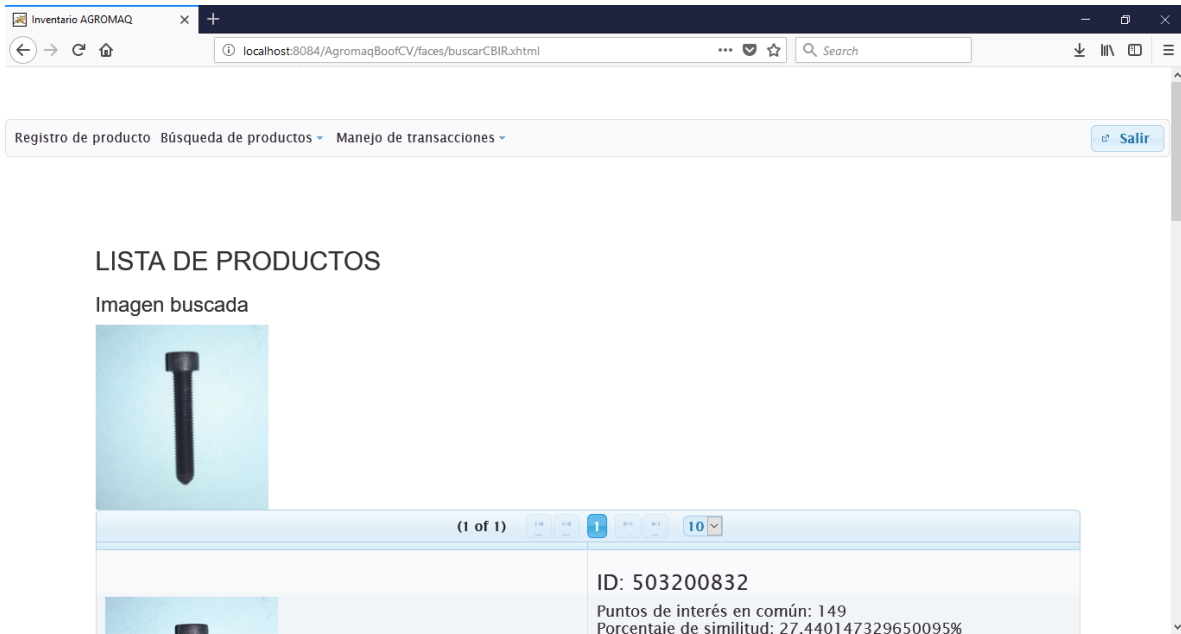


Figura 4.17 Caso de estudio 1 resultado 1 de búsqueda 4

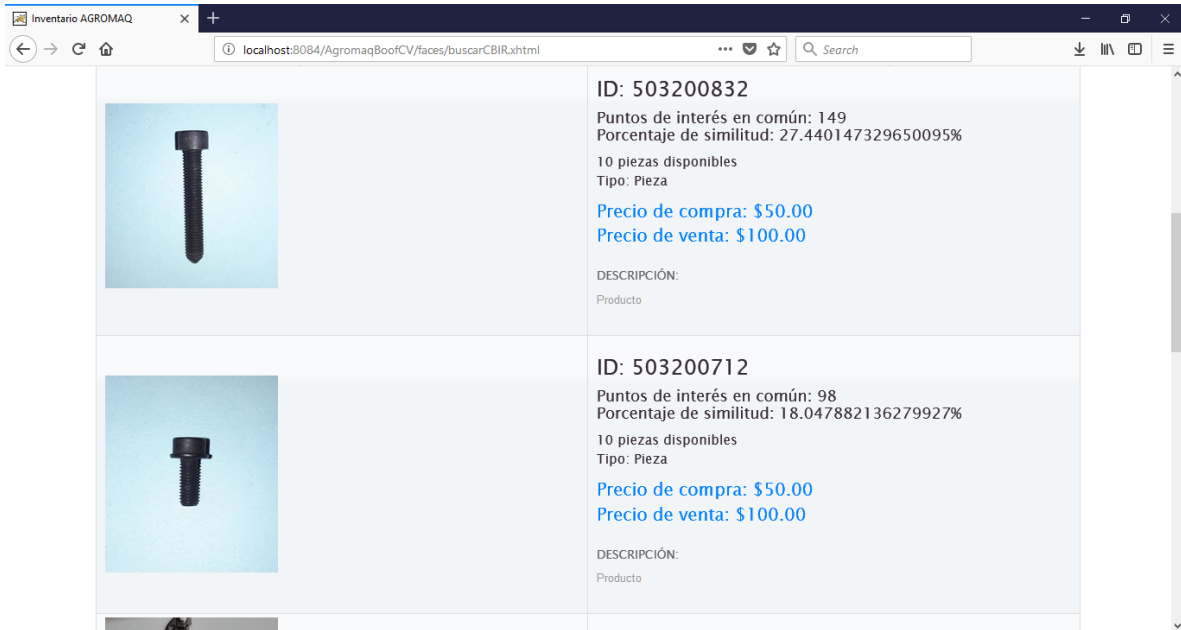


Figura 4.18 Caso de estudio 1 resultado 2 de búsqueda 4

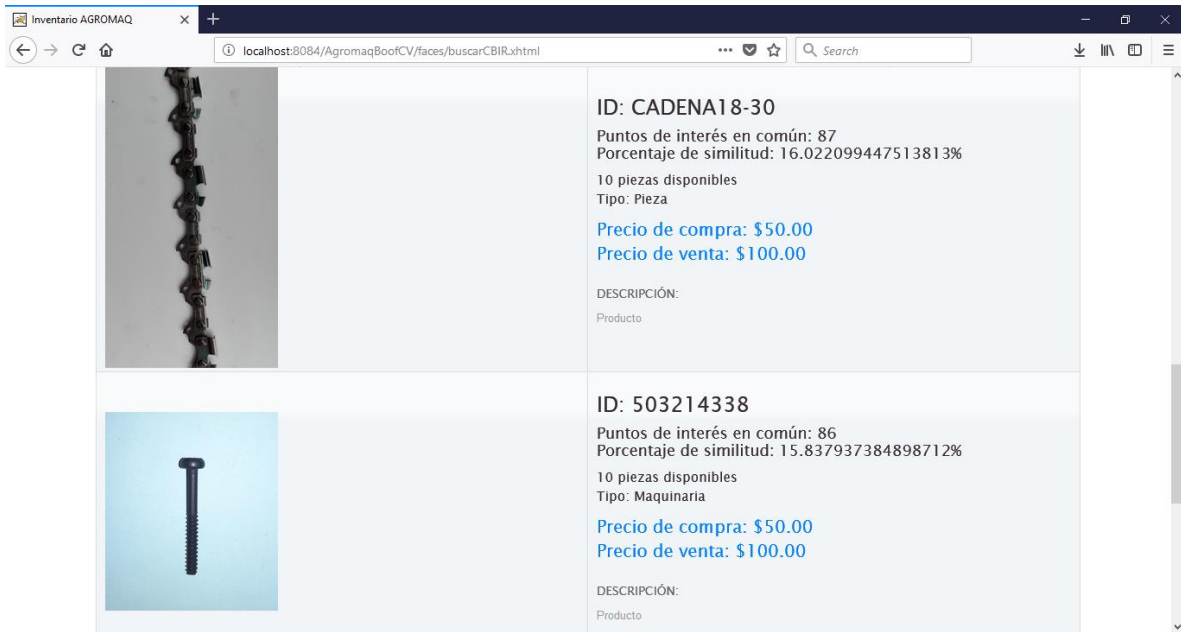


Figura 4.19 Caso de estudio 1 resultado 3 de búsqueda 4

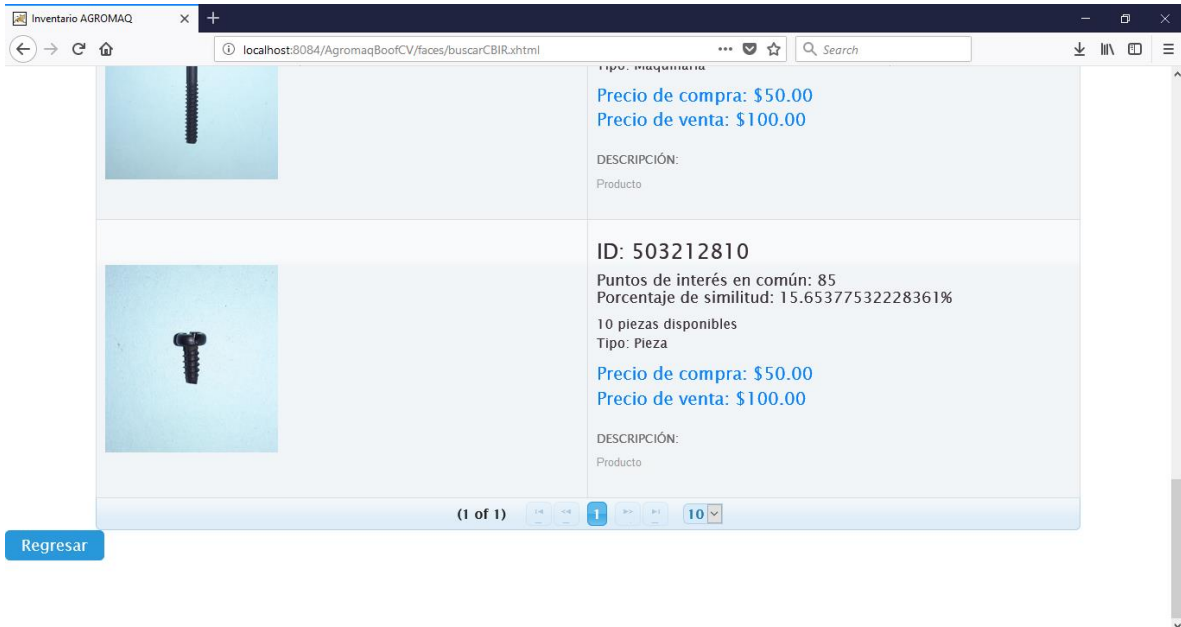


Figura 4.20 Caso de estudio 1 resultado 4 de búsqueda 4

En la Figura 4.21 se observa la imagen consulta 5 y posteriormente se muestran sus resultados en las Figuras 4.22, 4.23, 4.24 y 4.25, la consulta se realizó con $k=5$ y 4 como número de imágenes relevantes en la base de datos.

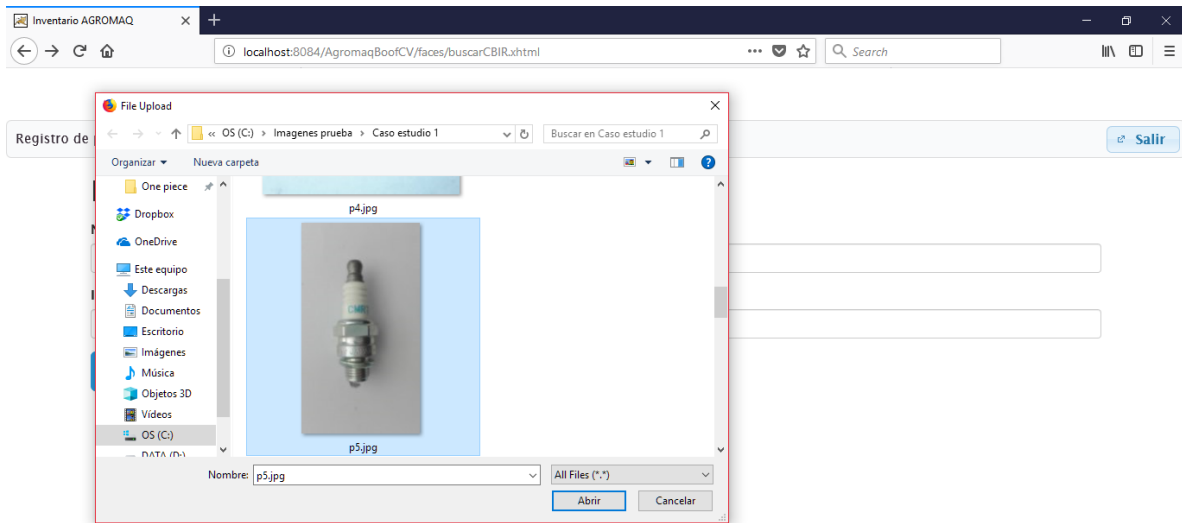


Figura 4.21 Caso de estudio 1 imagen de búsqueda 5

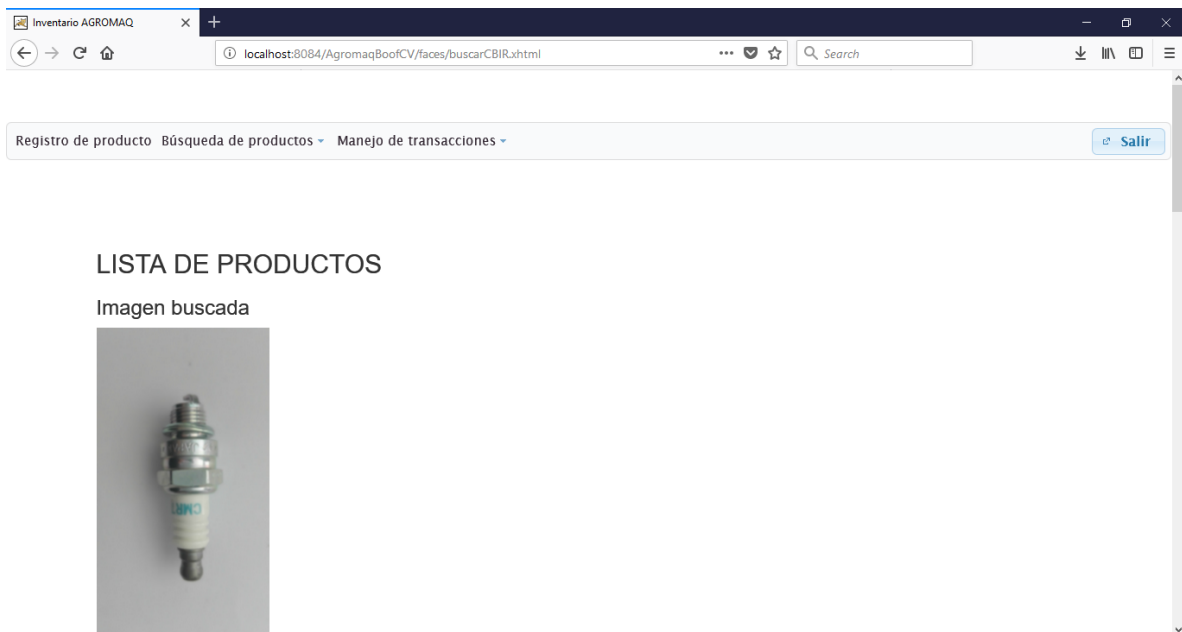


Figura 4.22 Caso de estudio 1 resultado 1 de búsqueda 5

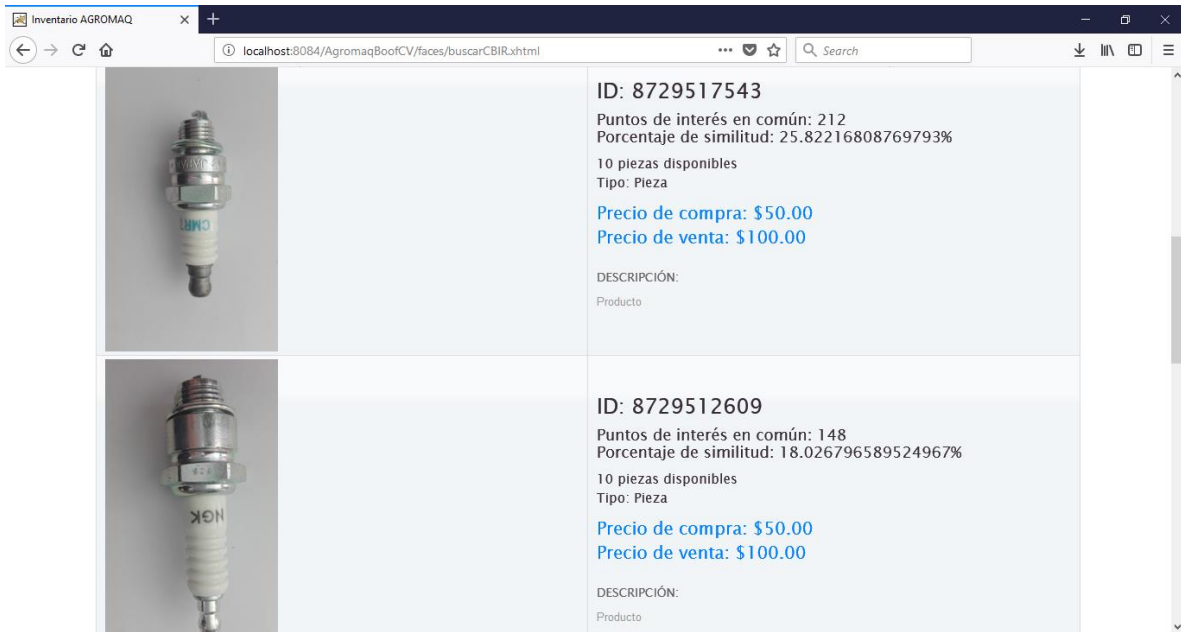


Figura 4.23 Caso de estudio 1 resultado 2 de búsqueda 5

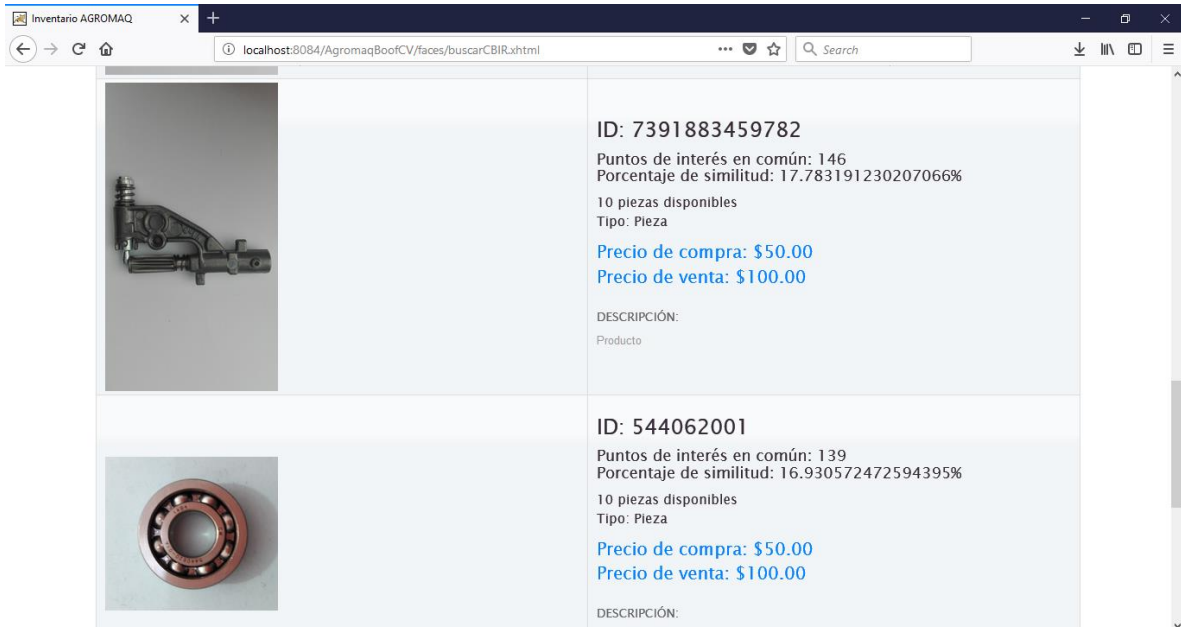


Figura 4.24 Caso de estudio 1 resultado 3 de búsqueda 5

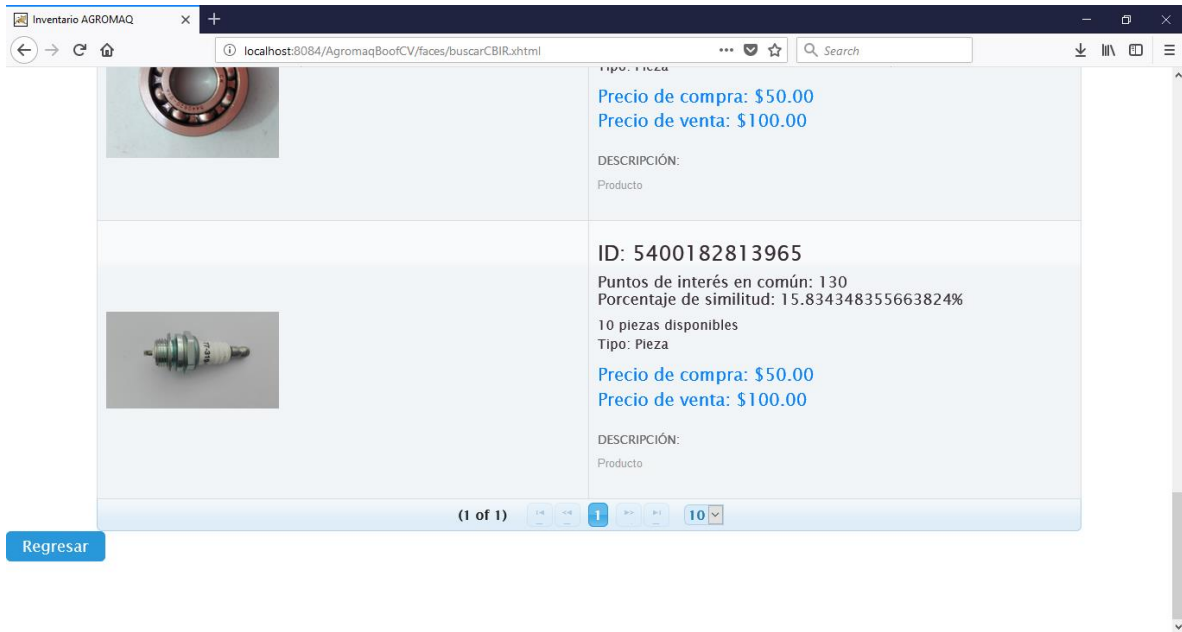


Figura 4.25 Caso de estudio 1 resultado 4 de búsqueda 5

En la Figura 4.26 se observa la imagen consulta 6 y posteriormente se muestran sus resultados en las Figuras 4.27 y 4.28, la consulta se realizó con $k=1$ y 1 como número de imágenes relevantes en la base de datos.

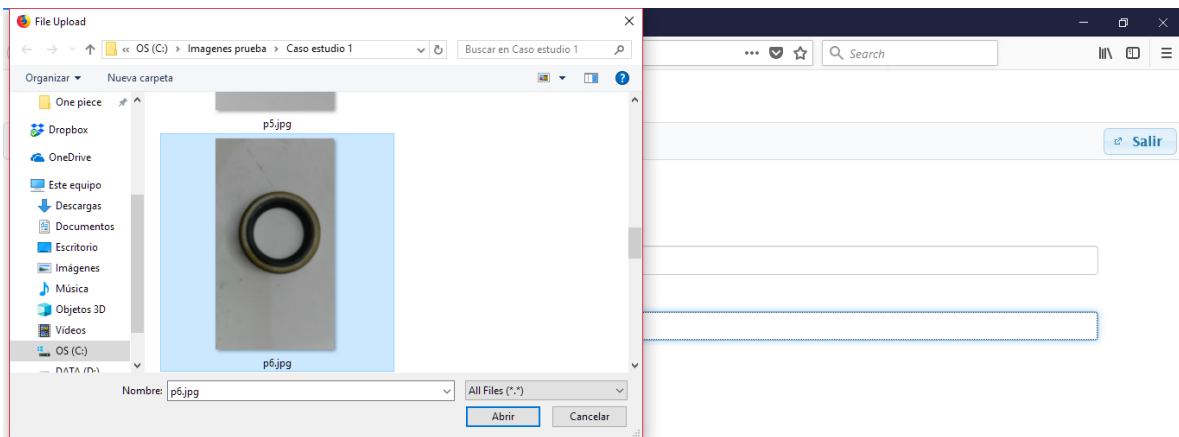


Figura 4.26 Caso de estudio 1 imagen de búsqueda 6

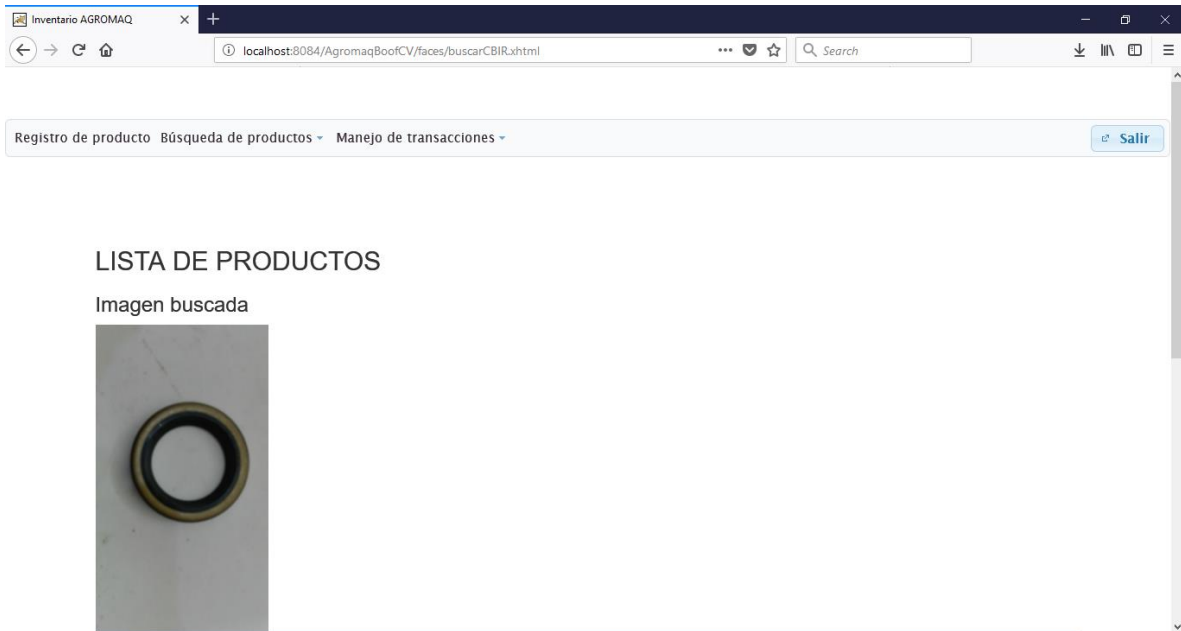


Figura 4.27 Caso de estudio 1 resultado 1 de búsqueda 6

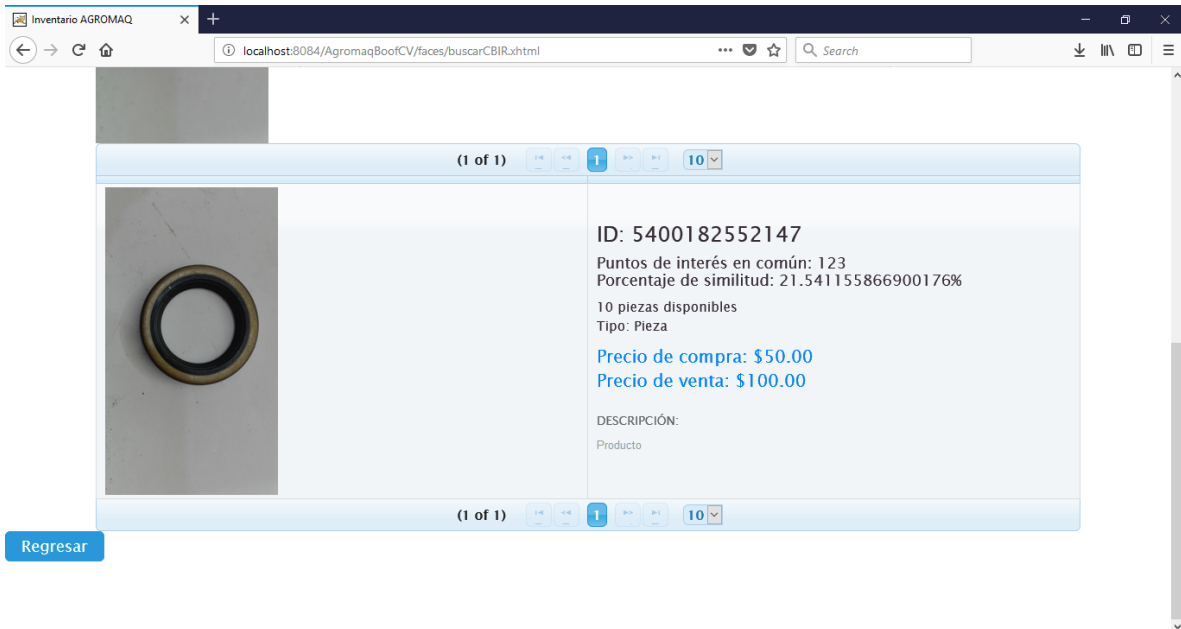


Figura 4.28 Caso de estudio 1 resultado 2 de búsqueda 6

En la Figura 4.29 se observa la imagen consulta 7 y posteriormente se muestran sus resultados en las Figuras 4.30 y 4.31, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

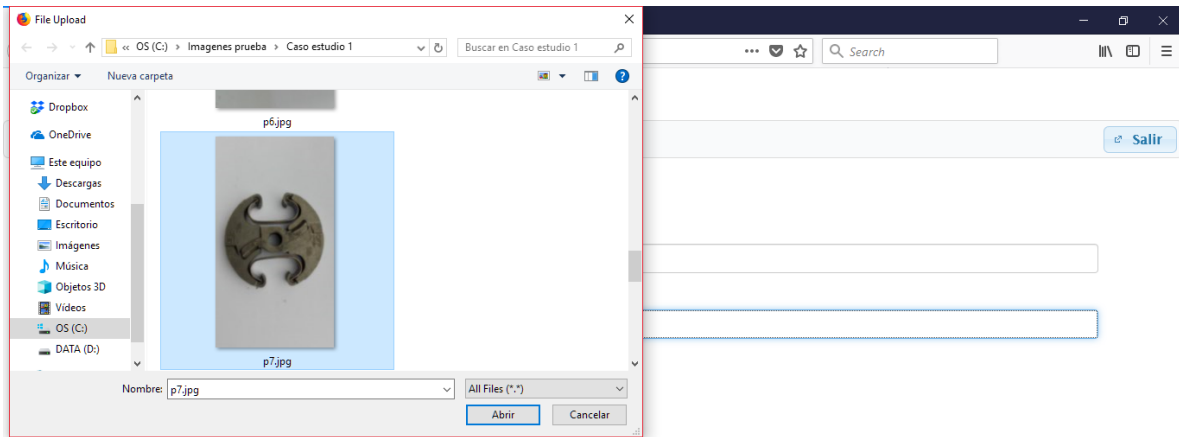


Figura 4.29 Caso de estudio 1 imagen de búsqueda 7

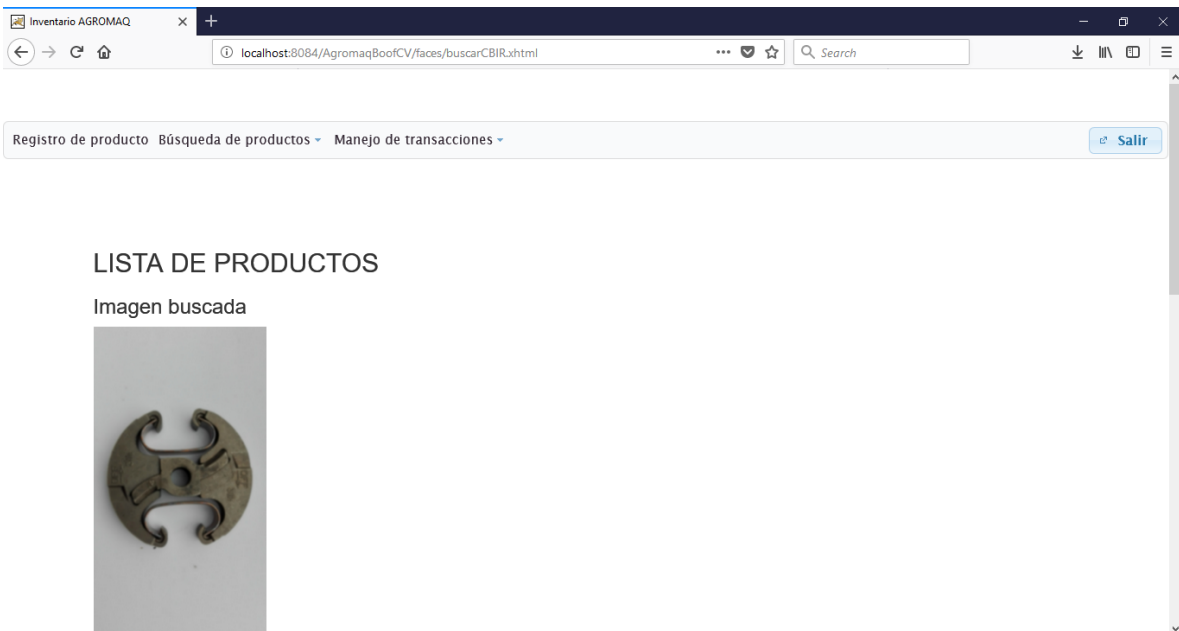


Figura 4.30 Caso de estudio 1 resultado 1 de búsqueda 7

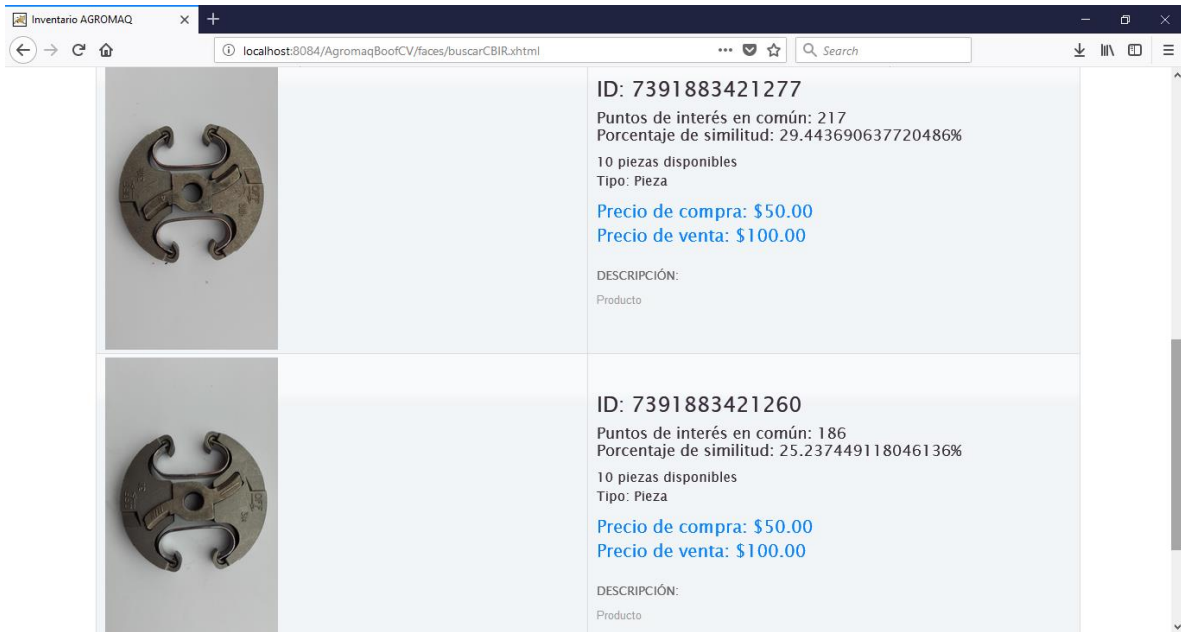


Figura 4.31 Caso de estudio 1 resultado 2 de búsqueda 7

En la Figura 4.32 se presenta la imagen consulta 8 y posteriormente se muestran sus resultados en las Figuras 4.33, y 4.34, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

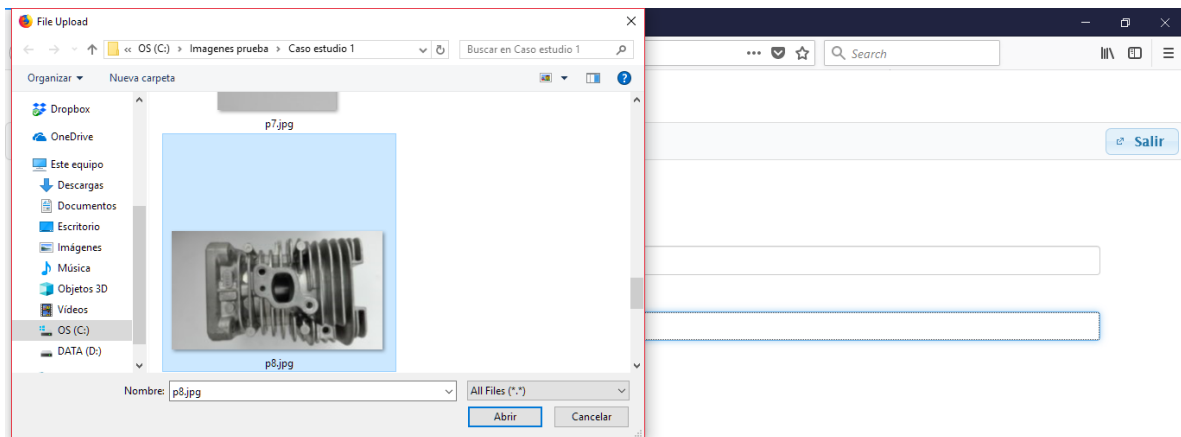


Figura 4.32 Caso de estudio 1 imagen de búsqueda 8

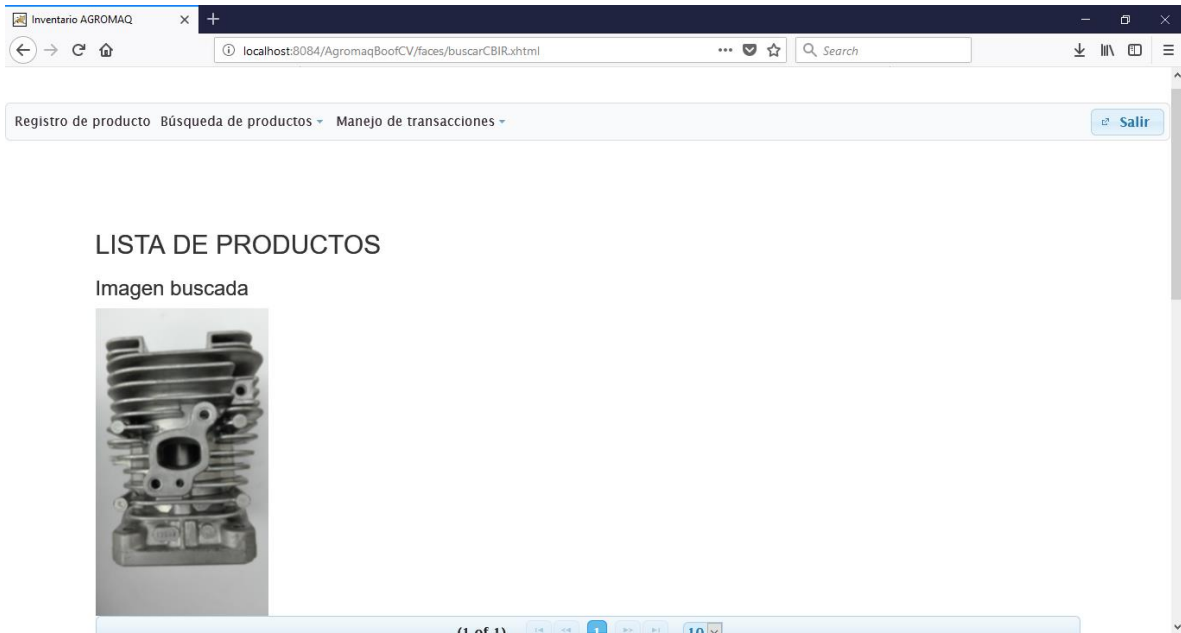


Figura 4.33 Caso de estudio 1 resultado 1 de búsqueda 8

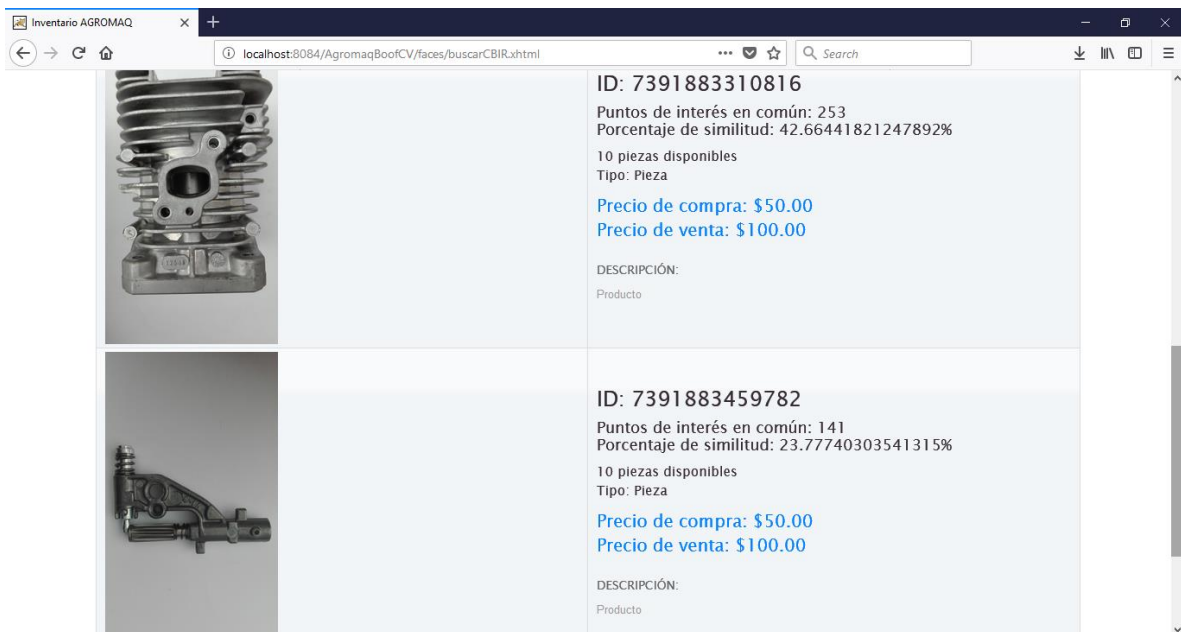


Figura 4.34 Caso de estudio 1 resultado 2 de búsqueda 8

En la Figura 4.35 se observa la imagen consulta 9 y posteriormente se muestran sus resultados en las Figuras 4.36, 4.37 y 4.38, la consulta se realizó con $k=4$ y 3 como número de imágenes relevantes en la base de datos.

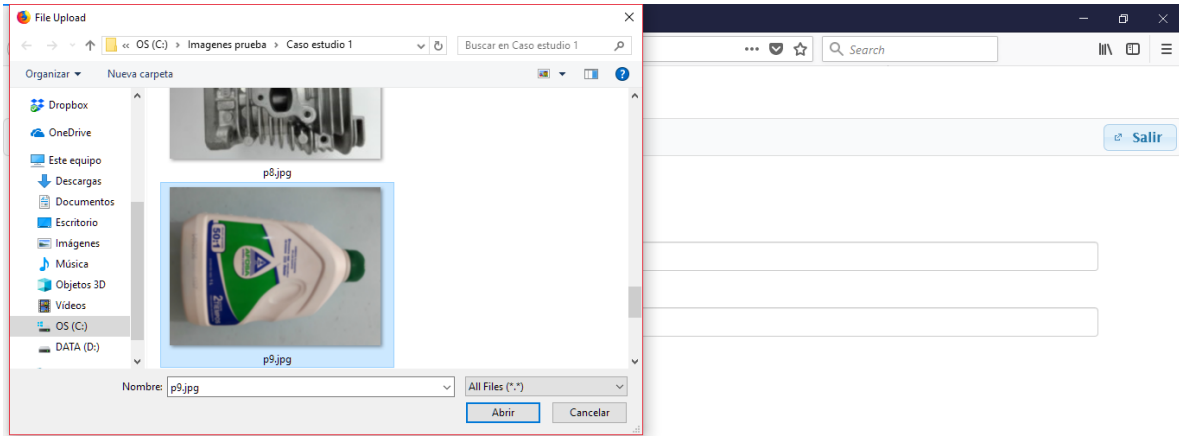


Figura 4.35 Caso de estudio 1 imagen de búsqueda 9

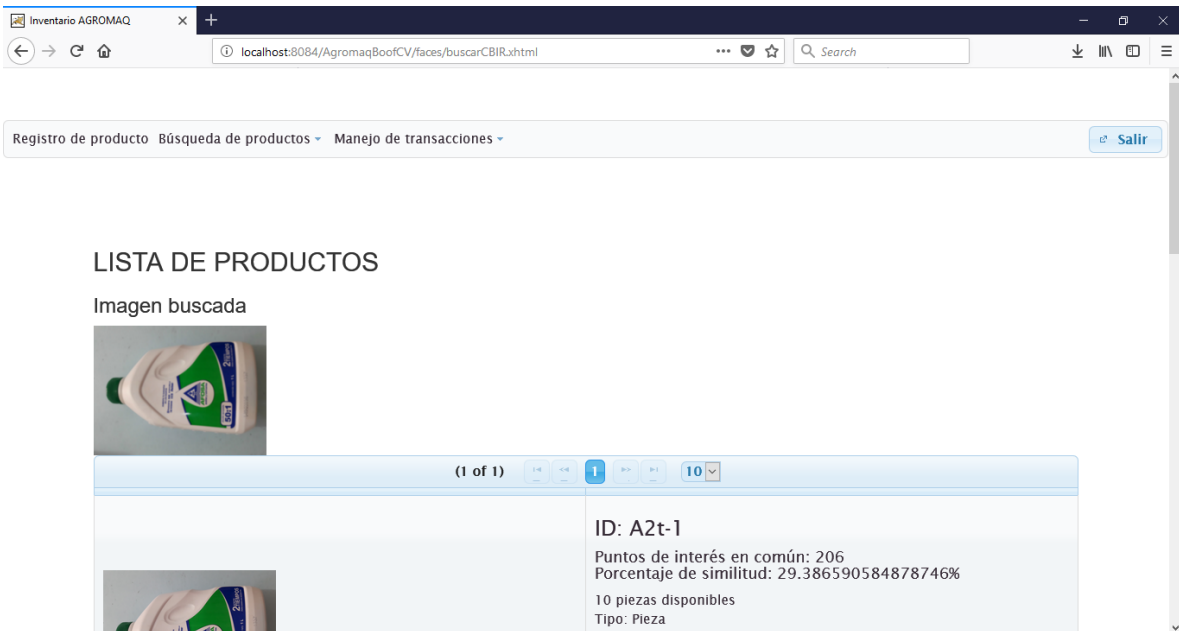


Figura 4.36 Caso de estudio 1 resultado 1 de búsqueda 9

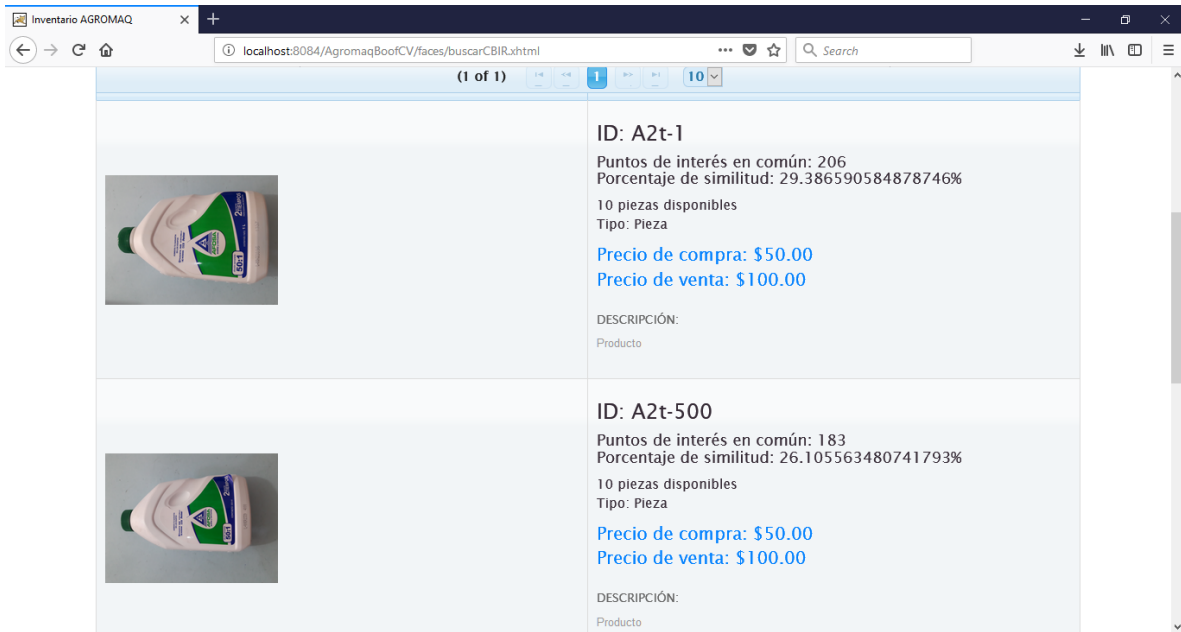


Figura 4.37 Caso de estudio 1 resultado 2 de búsqueda 9



Figura 4.38 Caso de estudio 1 resultado 3 de búsqueda 9

En la Figura 4.39 se visualiza la imagen consulta 10 y posteriormente se muestran sus resultados en las Figuras 4.40, 4.41 y 4.42, la consulta se realizó con $k=3$ y 3 como número de imágenes relevantes en la base de datos.

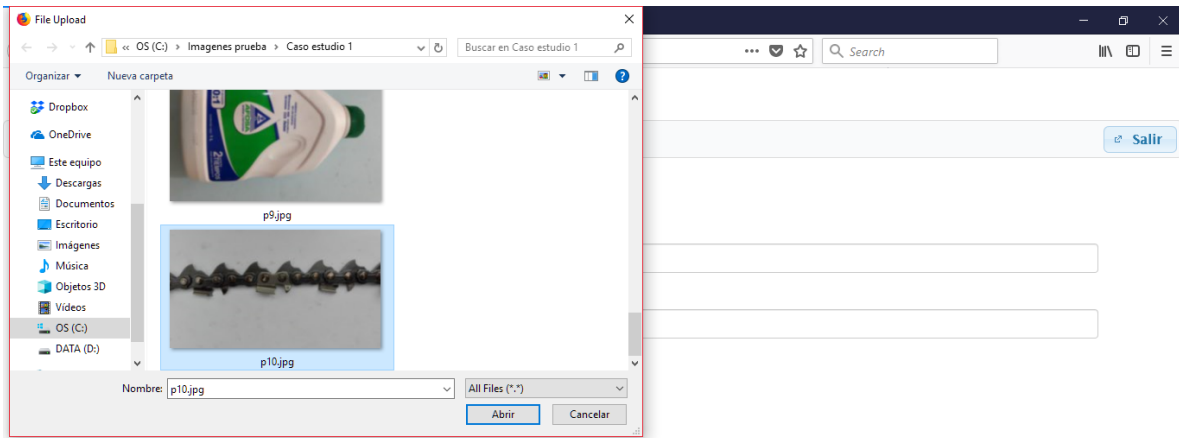


Figura 4.39 Caso de estudio 1 imagen de búsqueda 10

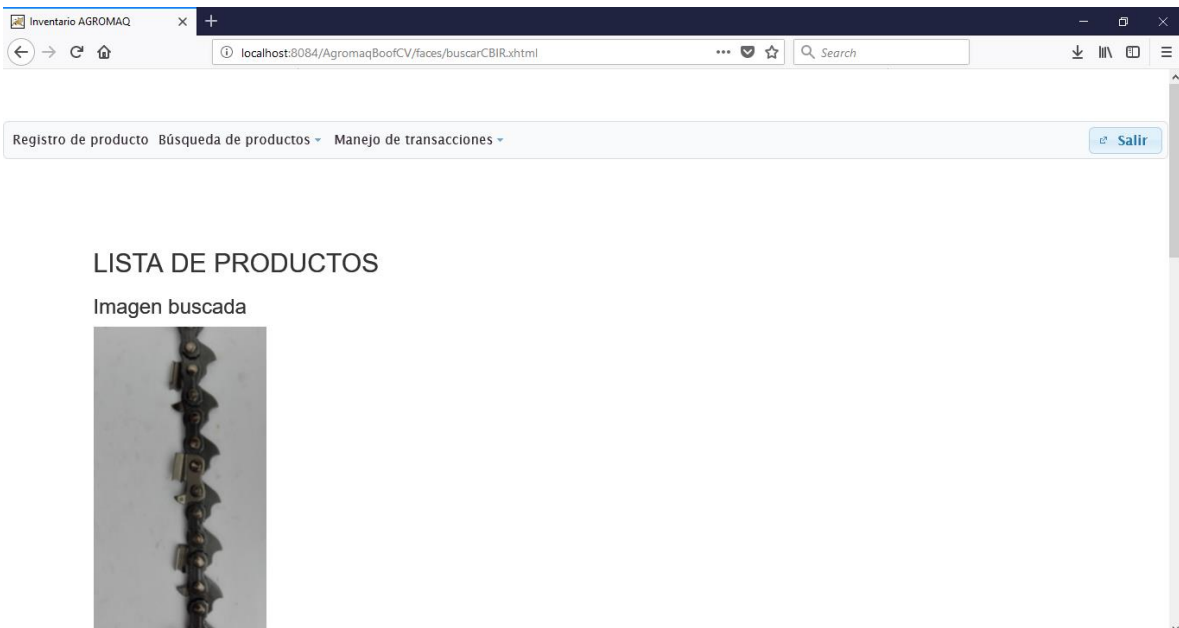


Figura 4.40 Caso de estudio 1 resultado 1 de búsqueda 10

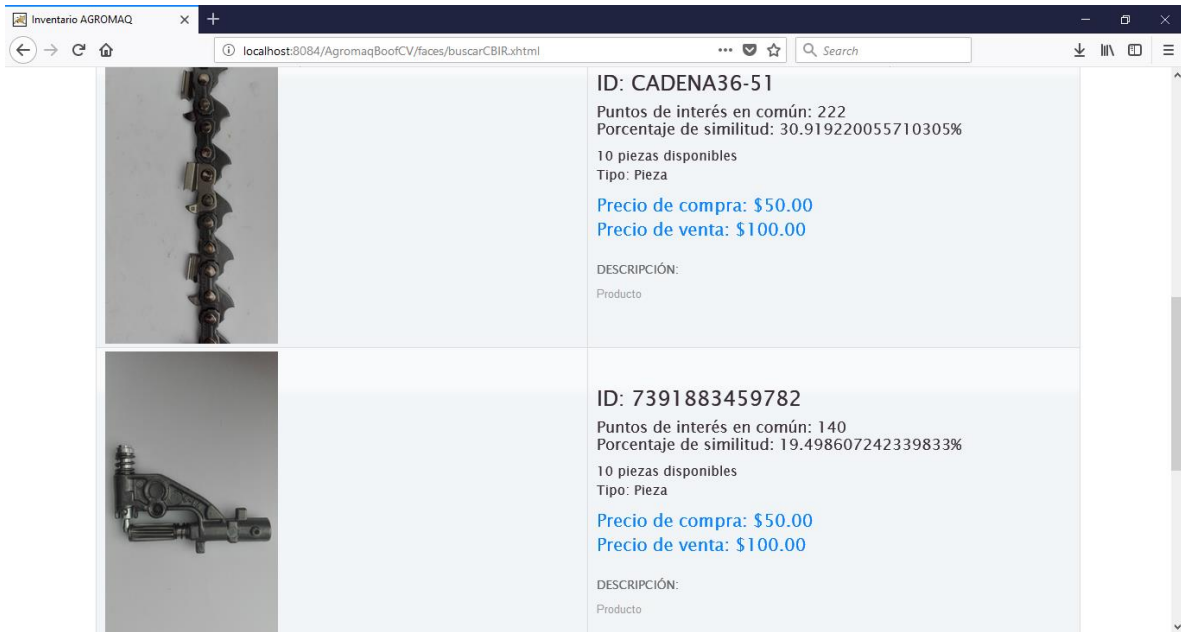


Figura 4.41 Caso de estudio 1 resultado 2 de búsqueda 10



Figura 4.42 Caso de estudio 1 resultado 3 de búsqueda 10

4.1.2. Caso de estudio 2

A continuación, se muestran los resultados de las 10 consultas realizadas para el segundo caso de estudio, que como ya se mencionó anteriormente consiste en la búsqueda con imágenes recortadas como entrada.

En la Figura 4.43 se observa la imagen consulta 1 y posteriormente se muestran sus resultados en las Figuras 4.44, 4.45 y 4.46, la consulta se realizó con $k=4$ y 4 como número de imágenes relevantes en la base de datos.

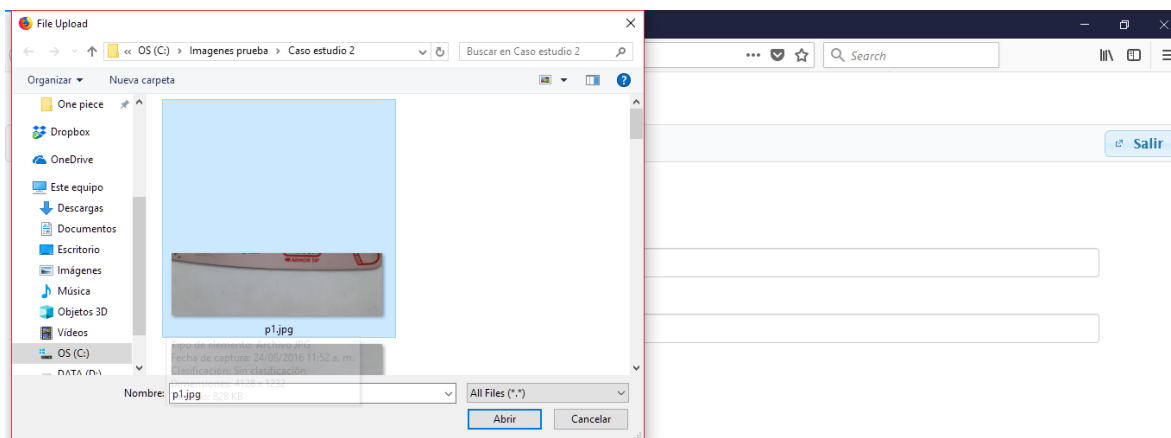


Figura 4.43 Caso de estudio 2 imagen de búsqueda 1

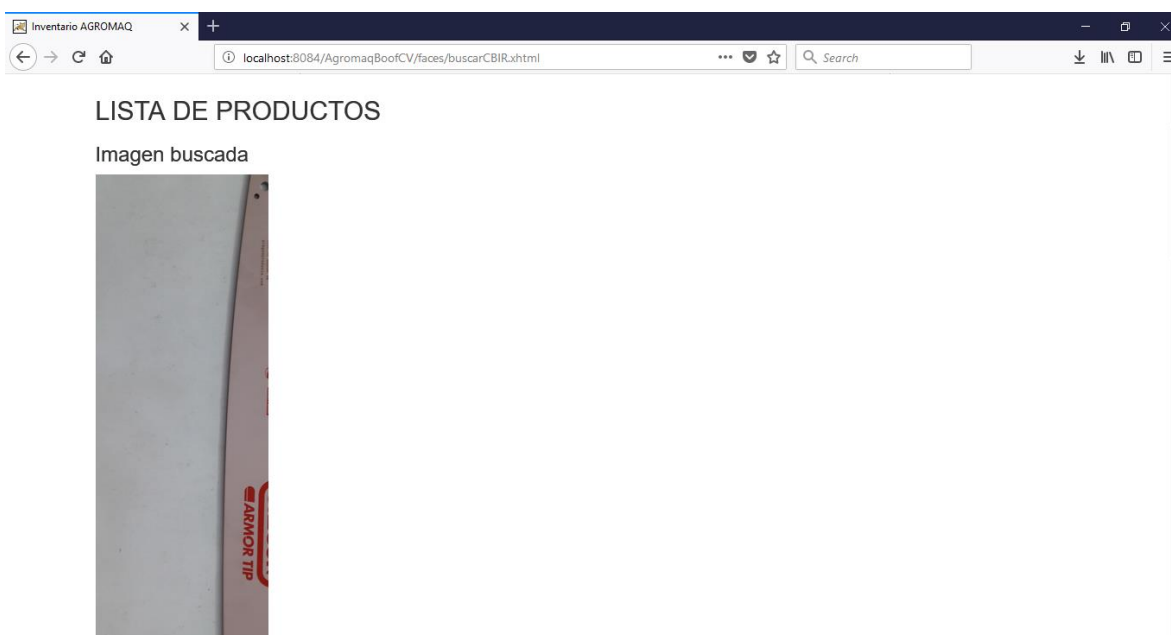


Figura 4.44 Caso de estudio 2 resultado 1 de búsqueda 1

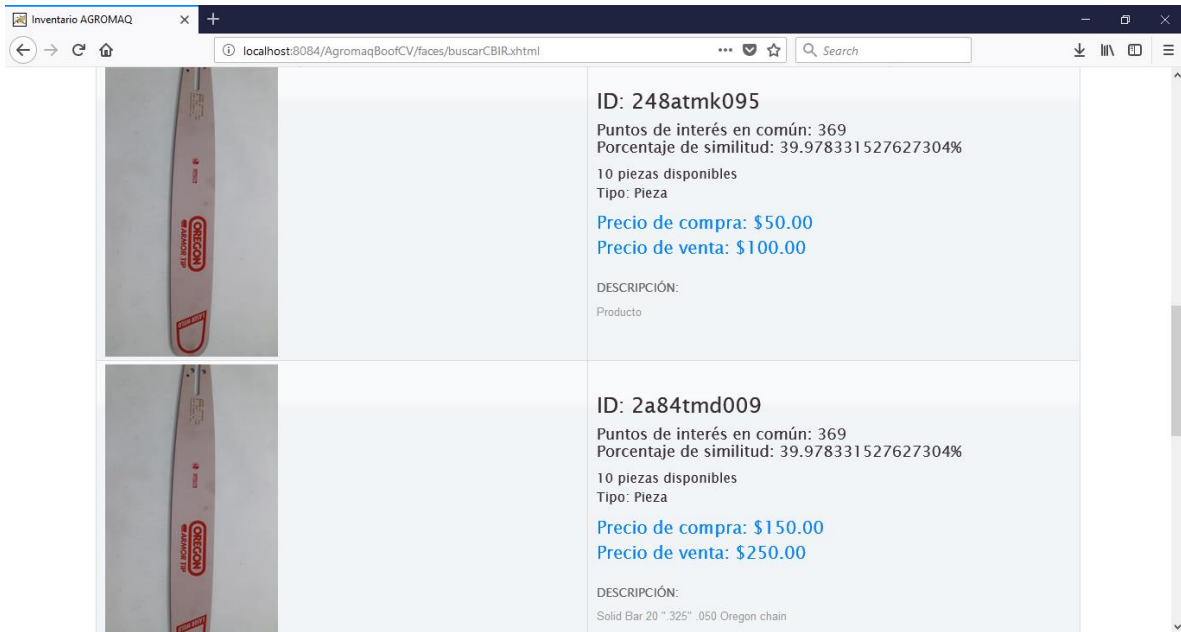


Figura 4.45 Caso de estudio 2 resultado 2 de búsqueda 1

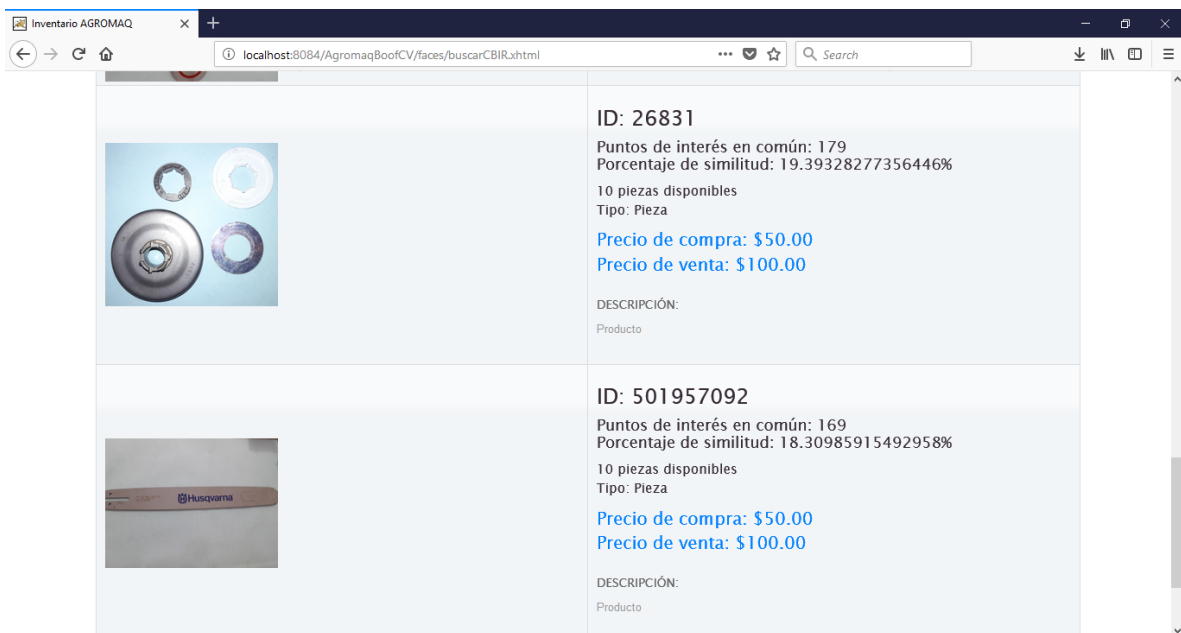


Figura 4.46 Caso de estudio 2 resultado 3 de búsqueda 1

En la Figura 4.47 se presenta la imagen consulta 2 y posteriormente se muestran sus resultados en las Figuras 4.48, y 4.49, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

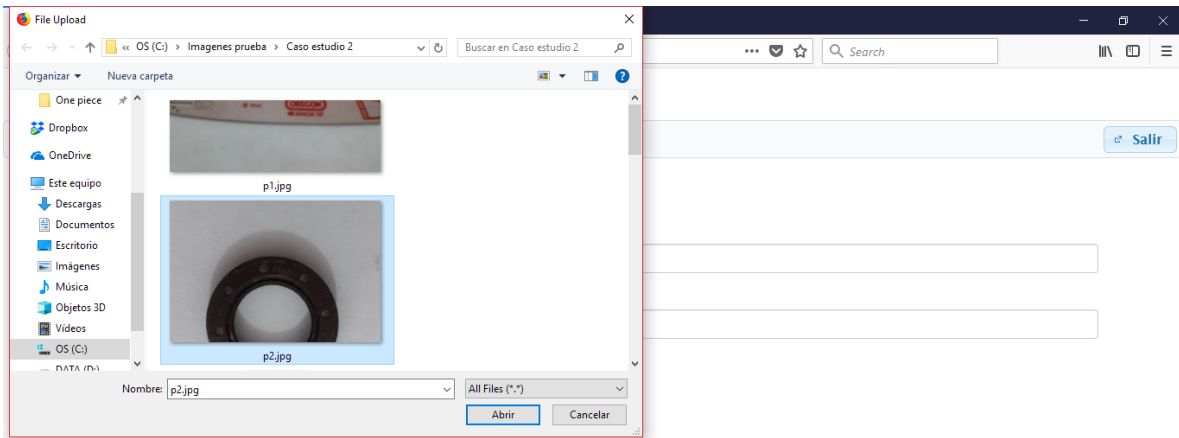


Figura 4.47 Caso de estudio 2 imagen de búsqueda 2

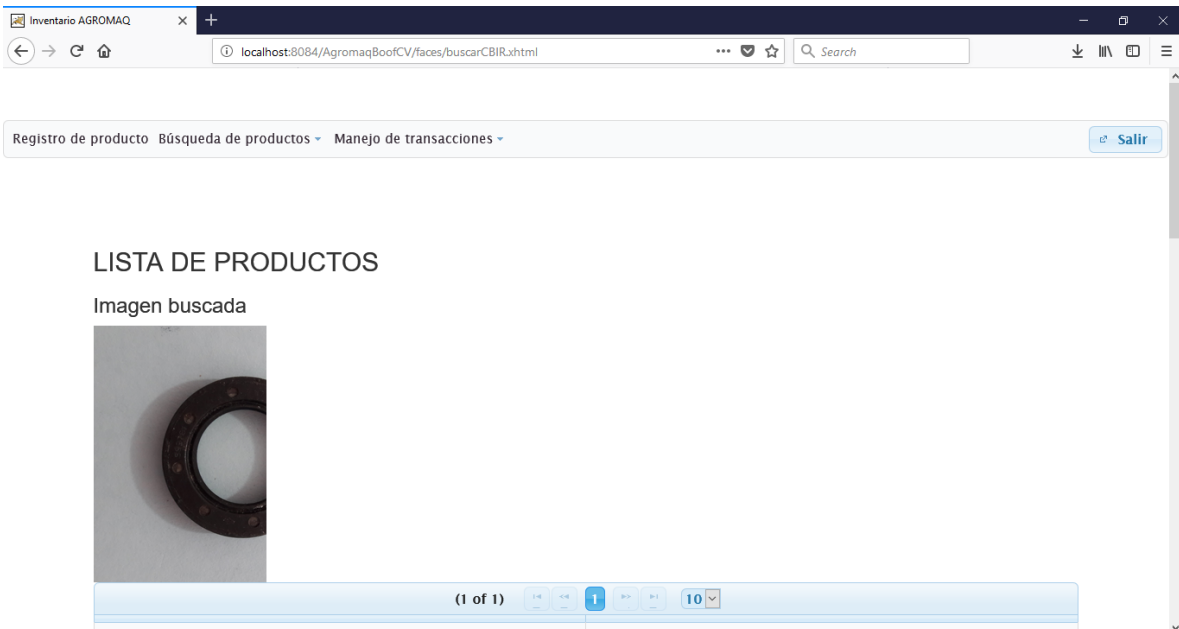


Figura 4.48 Caso de estudio 2 resultado 1 de búsqueda 2

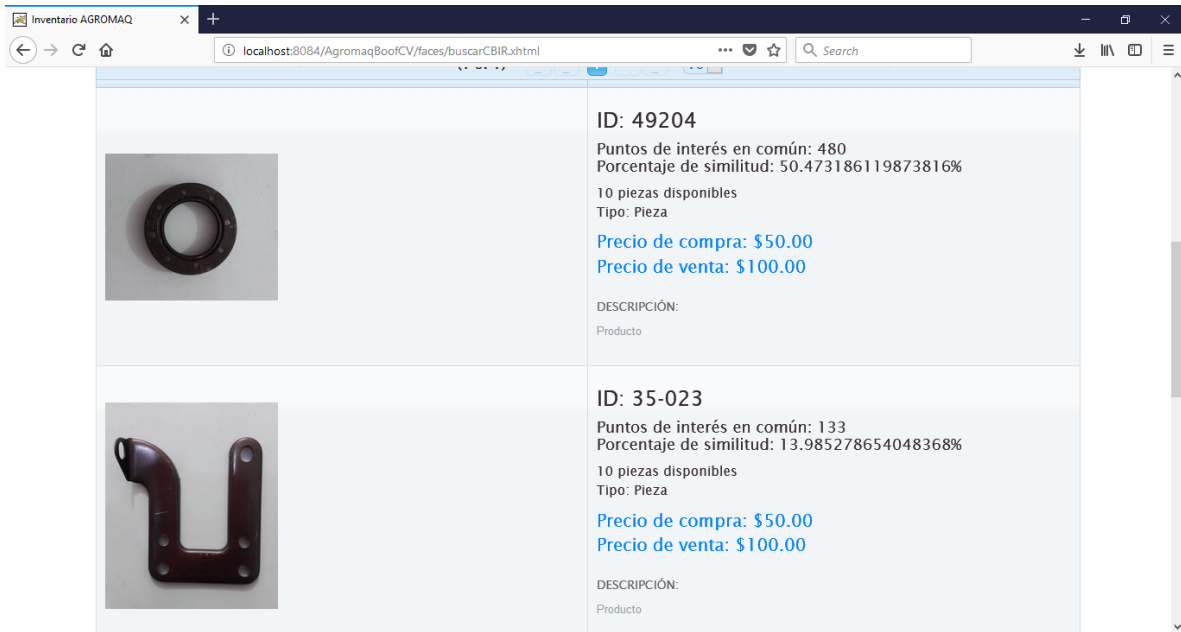


Figura 4.49 Caso de estudio 2 resultado 2 de búsqueda 2

En la Figura 4.50 se visualiza la imagen consulta 3 y posteriormente se muestran sus resultados en las Figuras 4.51 y 4.52, la consulta se realizó con $k=1$ y 1 como número de imágenes relevantes en la base de datos.

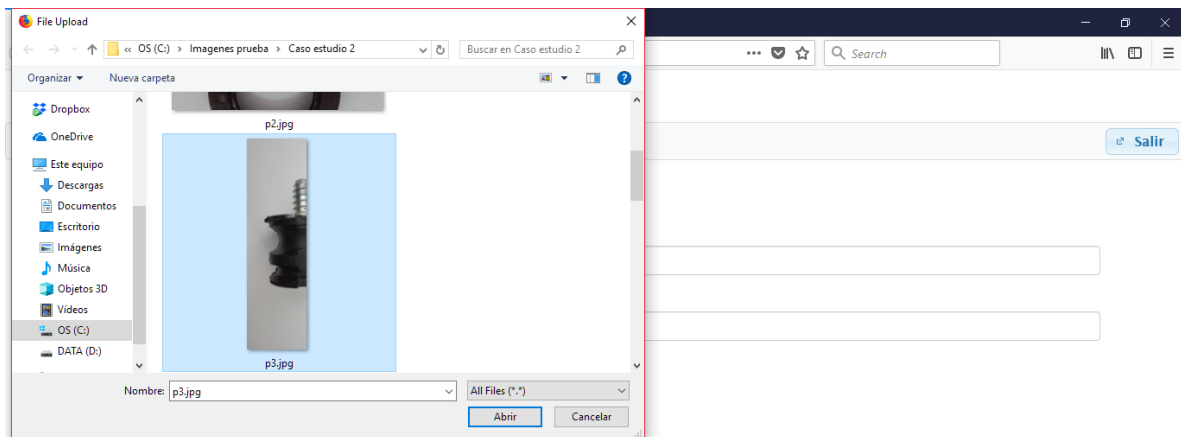


Figura 4.50 Caso de estudio 2 imagen de búsqueda 3

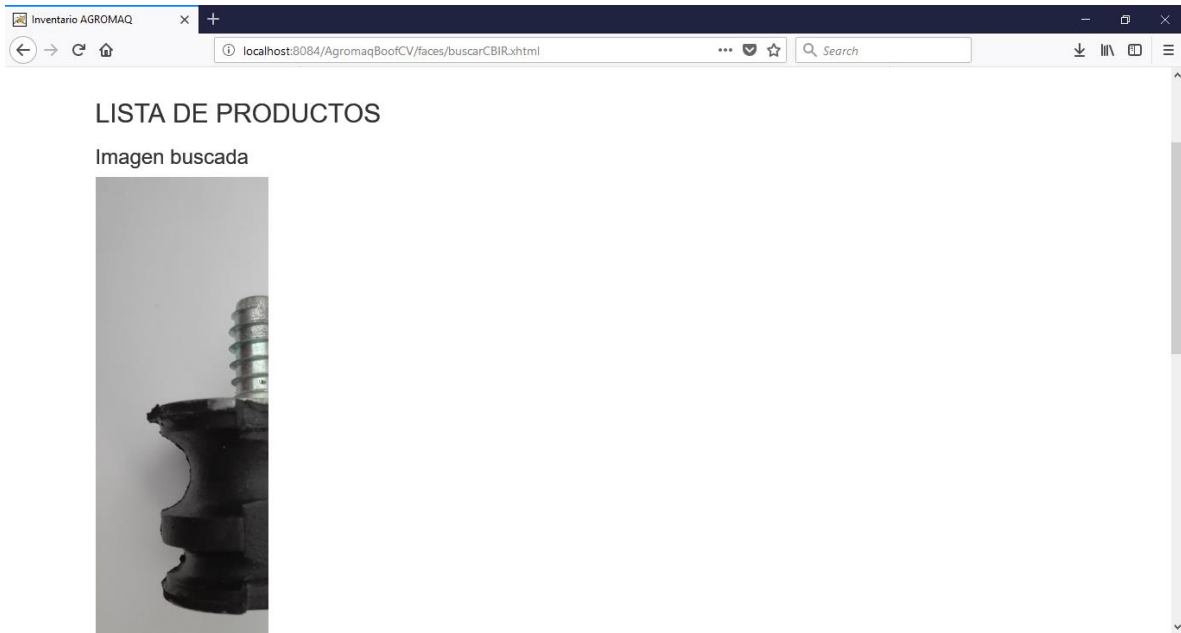


Figura 4.51 Caso de estudio 2 resultado 1 de búsqueda 3



Figura 4.52 Caso de estudio 2 resultado 2 de búsqueda 3

En la Figura 4.53 se presenta la imagen consulta 4 y posteriormente se muestran sus resultados en las Figuras 4.54, 4.55 y 4.56, la consulta se realizó con $k=5$ y 5 como número de imágenes relevantes en la base de datos.

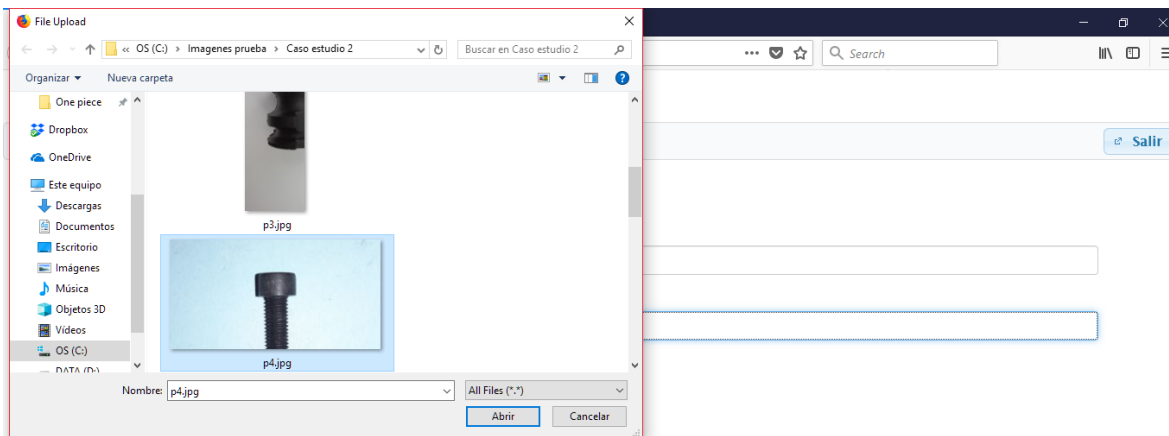


Figura 4.53 Caso de estudio 2 imagen de búsqueda 4

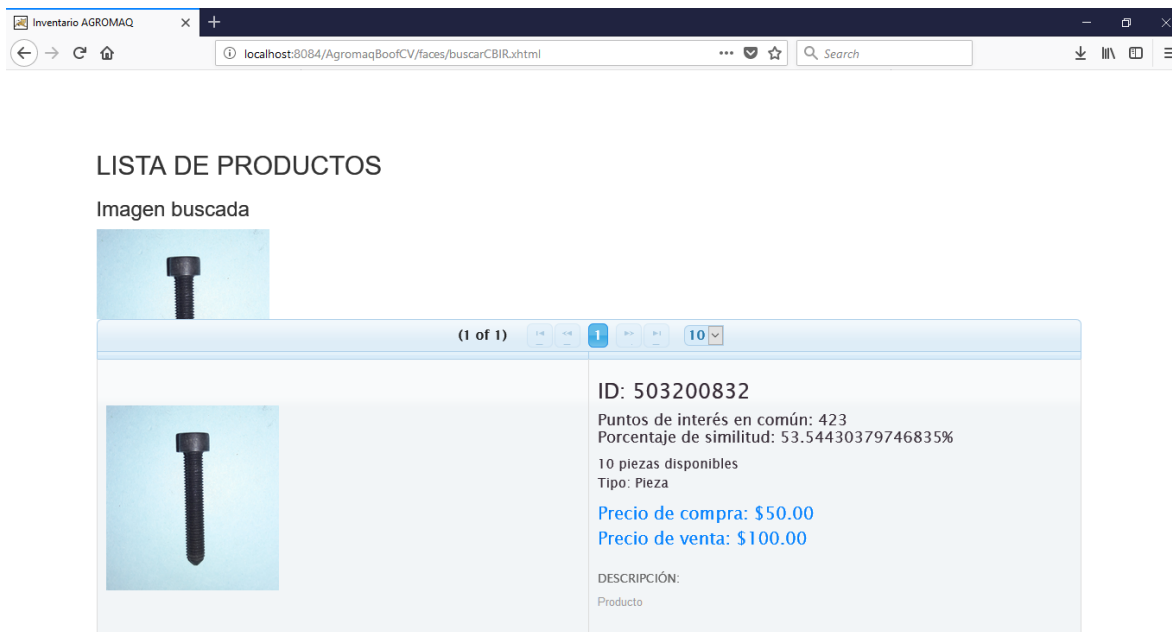


Figura 4.54 Caso de estudio 2 resultado 1 de búsqueda 4

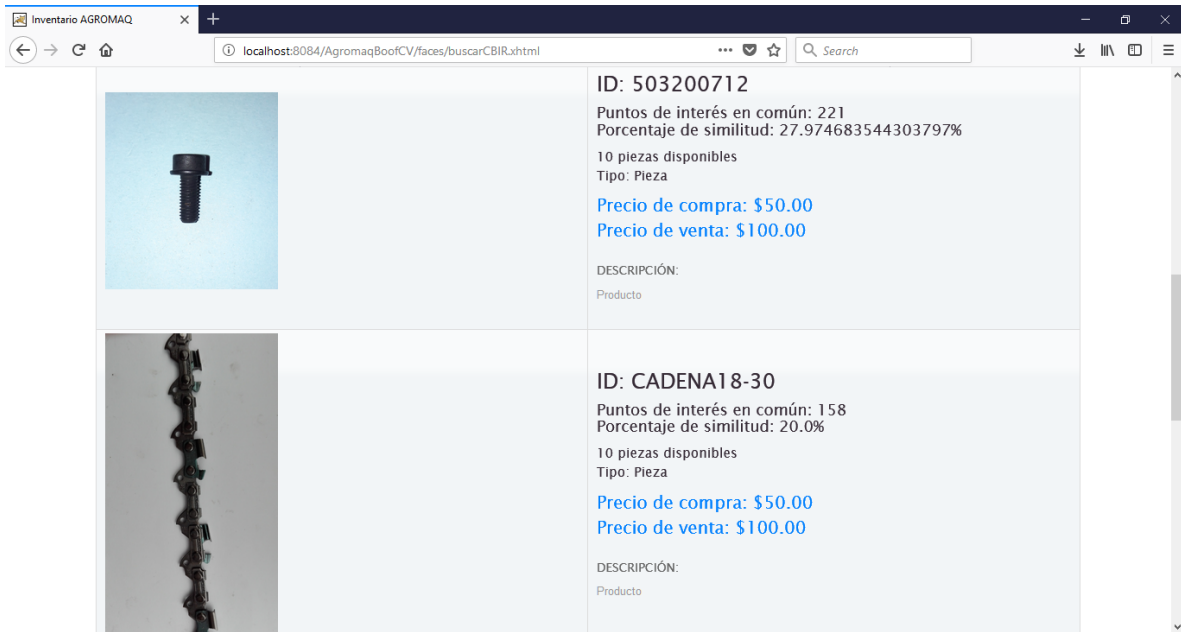


Figura 4.55 Caso de estudio 2 resultado 2 de búsqueda 4

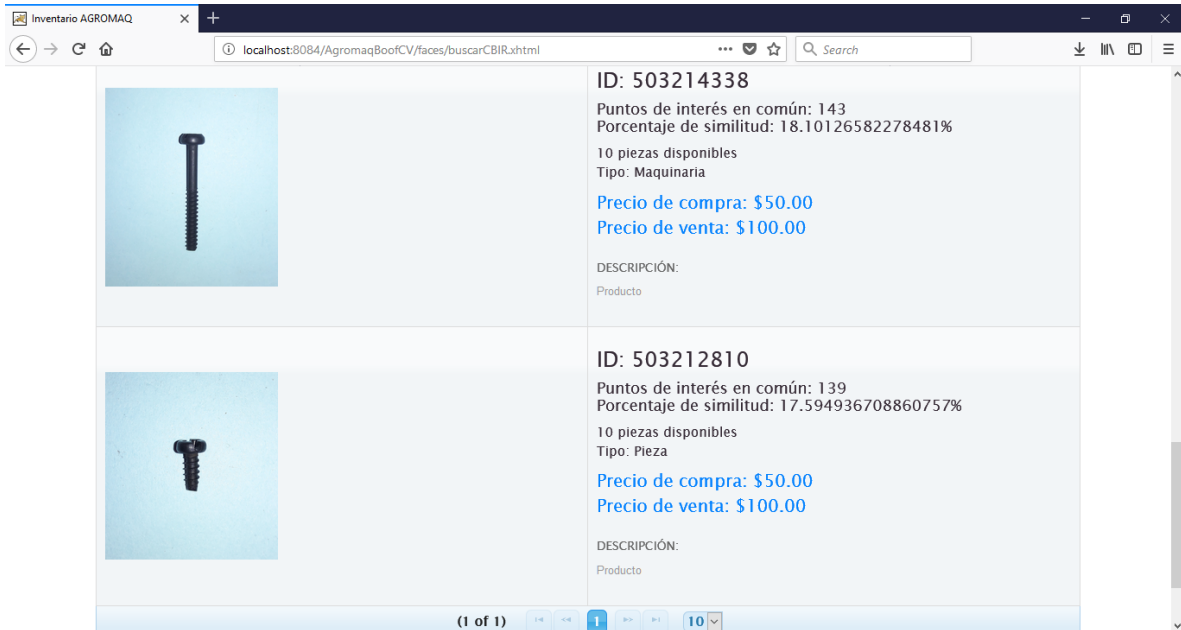


Figura 4.56 Caso de estudio 2 resultado 3 de búsqueda 4

En la Figura 4.57 se observa la imagen consulta 5 y posteriormente se muestran sus resultados en las Figuras 4.58 y 4.59, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

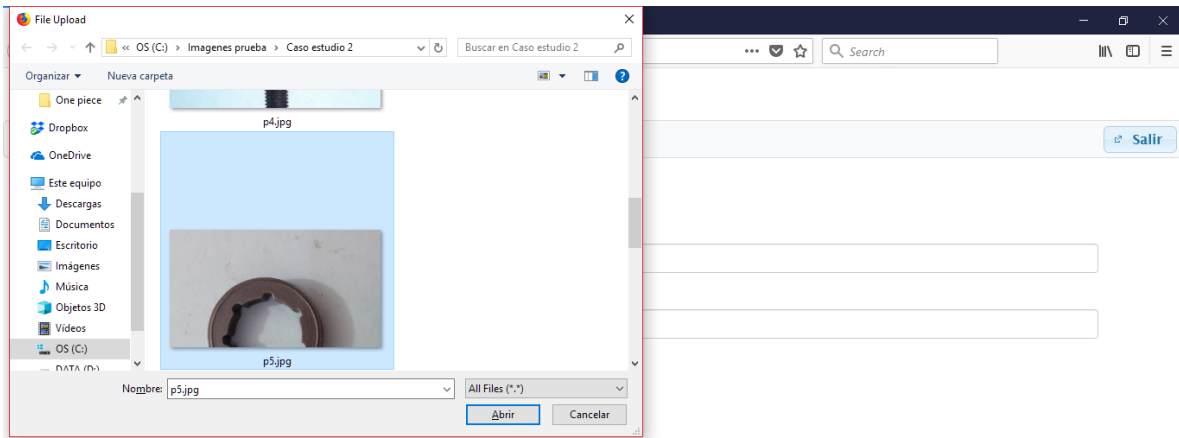


Figura 4.57 Caso de estudio 2 imagen de búsqueda 5

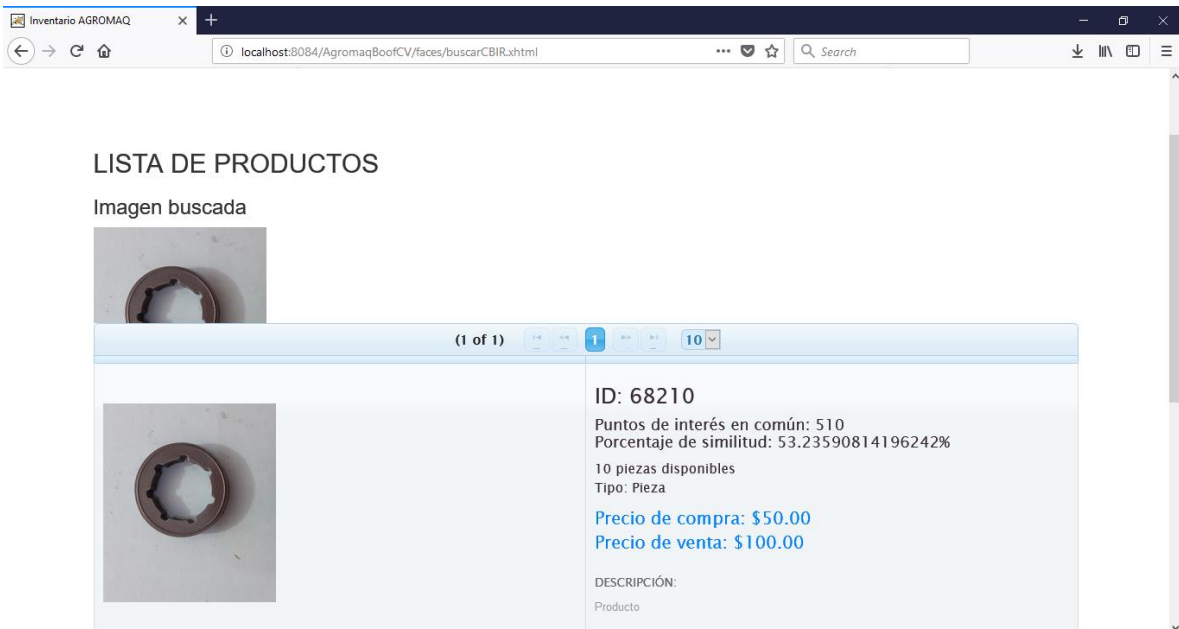


Figura 4.58 Caso de estudio 2 resultado 1 de búsqueda 5

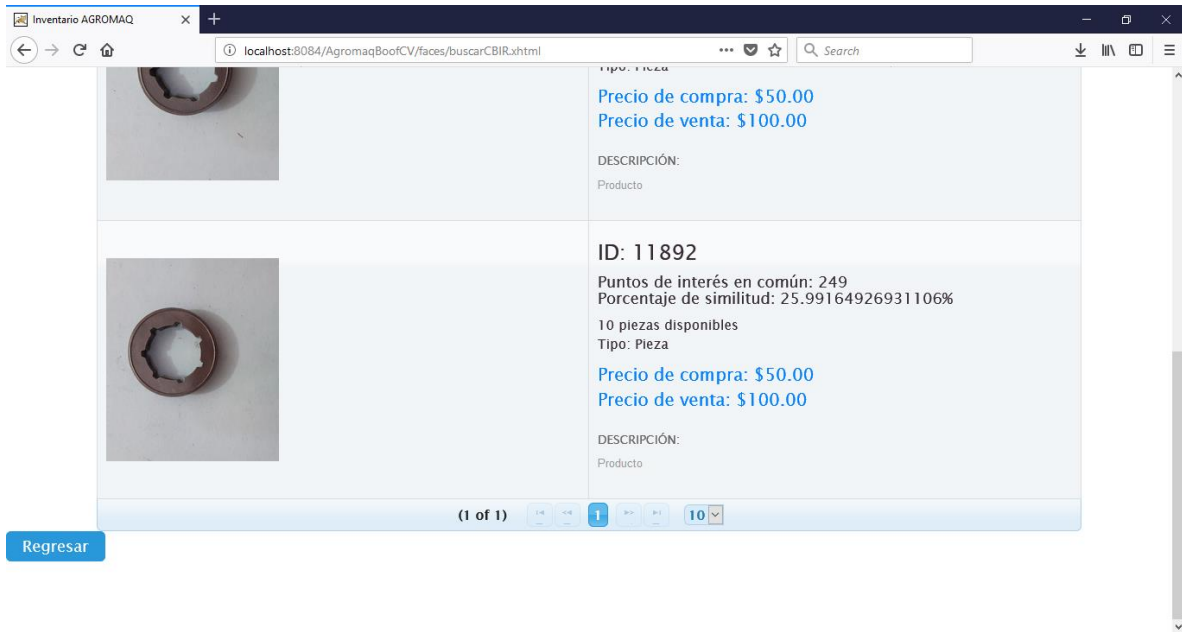


Figura 4.59 Caso de estudio 2 resultado 2 de búsqueda 5

En la Figura 4.60 se presenta la imagen consulta 6 y posteriormente se muestran sus resultados en las Figuras 4.61 y Figura 4.62, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

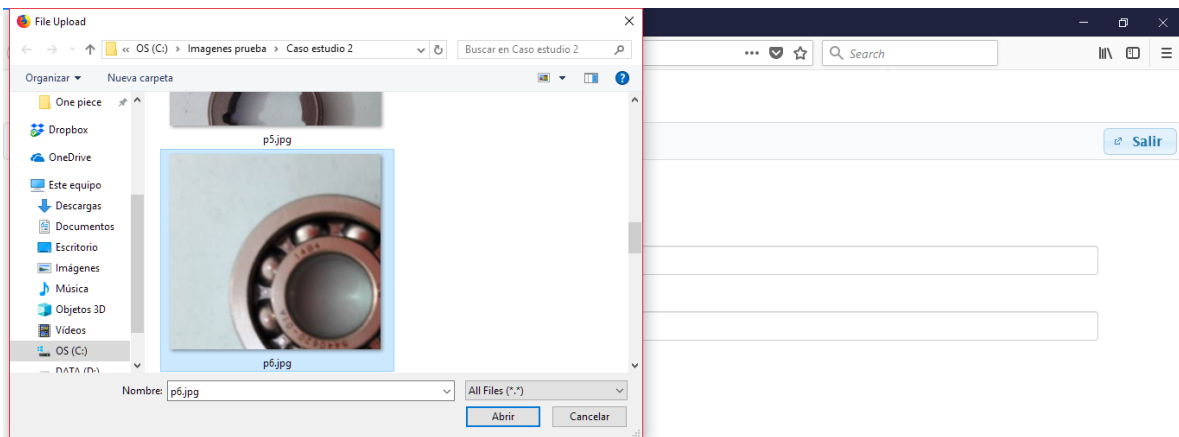


Figura 4.60 Caso de estudio 2 imagen de búsqueda 6

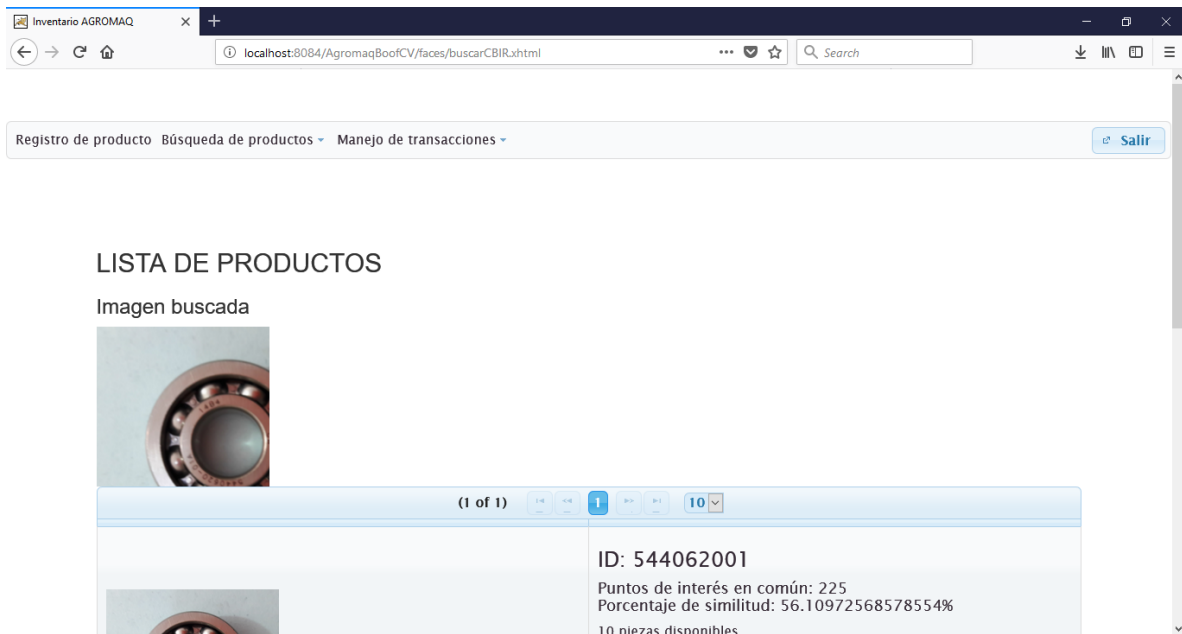


Figura 4.61 Caso de estudio 2 resultado 1 de búsqueda 6

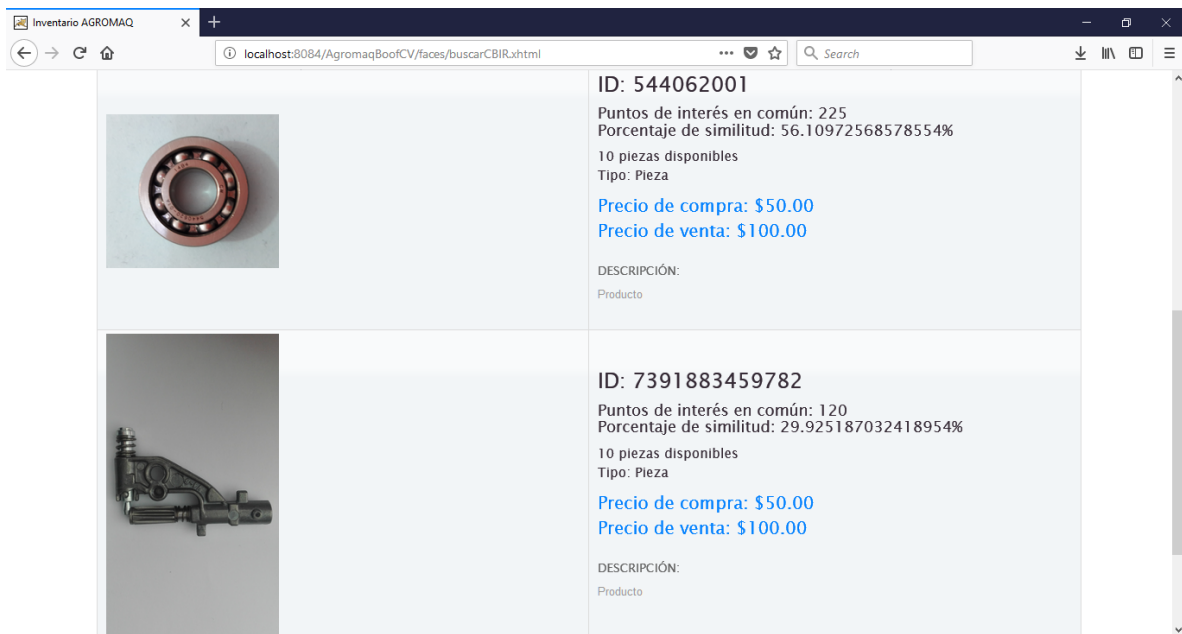


Figura 4.62 Caso de estudio 2 resultado 2 de búsqueda 6

En la Figura 4.63 se observa la imagen consulta 7 y posteriormente se muestran sus resultados en las Figuras 4.64 y 4.65, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

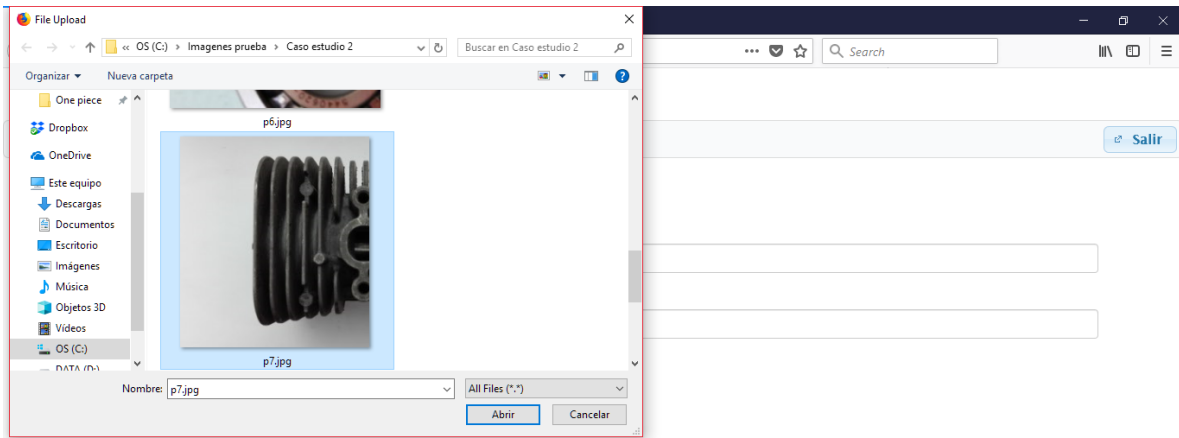


Figura 4.63 Caso de estudio 2 imagen de búsqueda 7

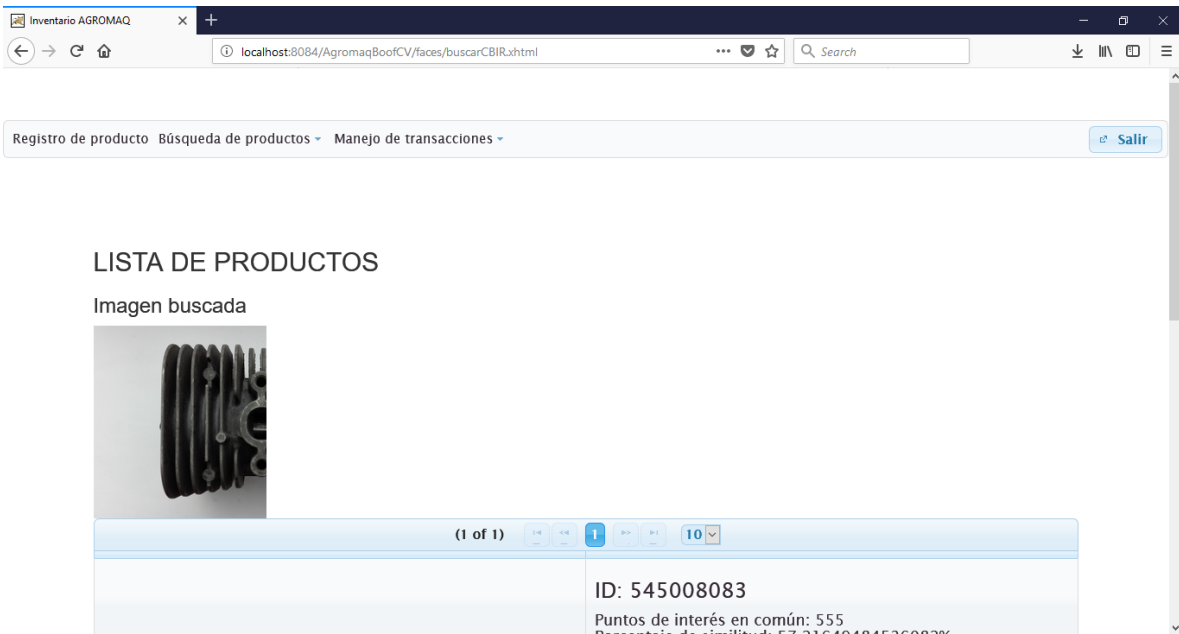


Figura 4.64 Caso de estudio 2 resultado 1 de búsqueda 7

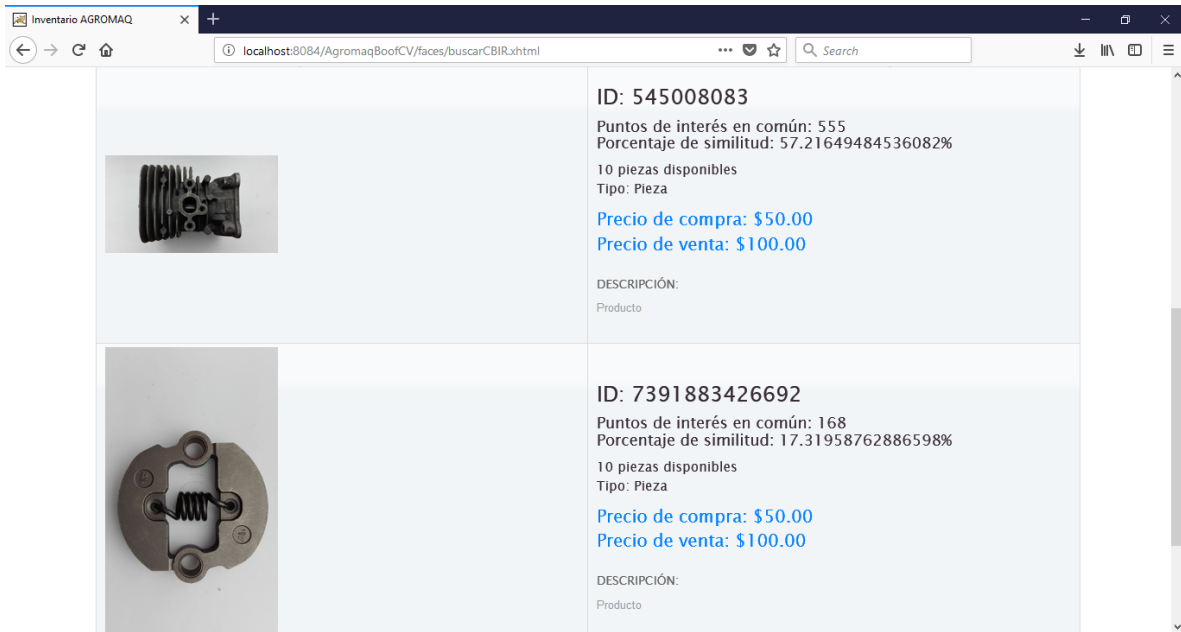


Figura 4.65 Caso de estudio 2 resultado 2 de búsqueda 7

En la Figura 4.66 se presenta la imagen consulta 8 y posteriormente se muestran sus resultados en las Figuras 4.67, 4.68, 4.69 y 4.70, la consulta se realizó con $k=5$ y 4 como número de imágenes relevantes en la base de datos.

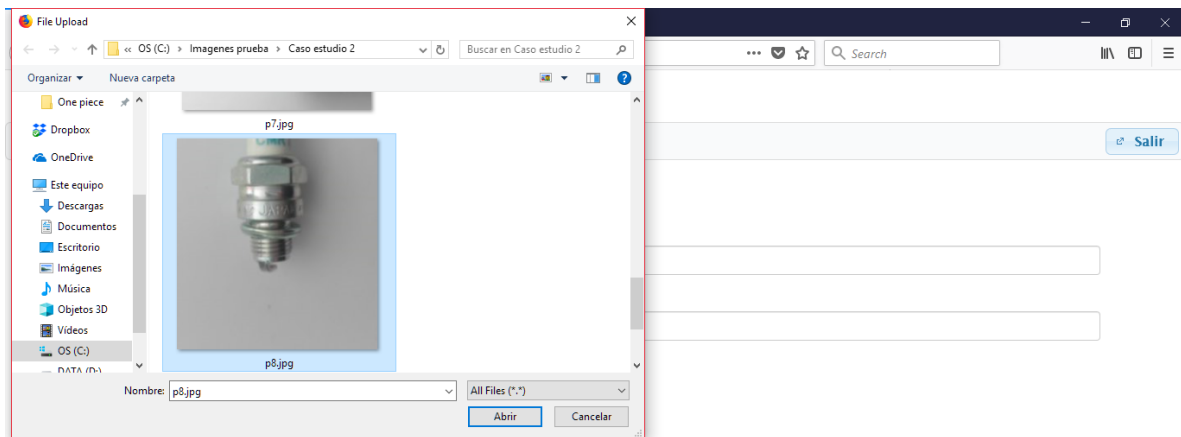


Figura 4.66 Caso de estudio 2 imagen de búsqueda 8

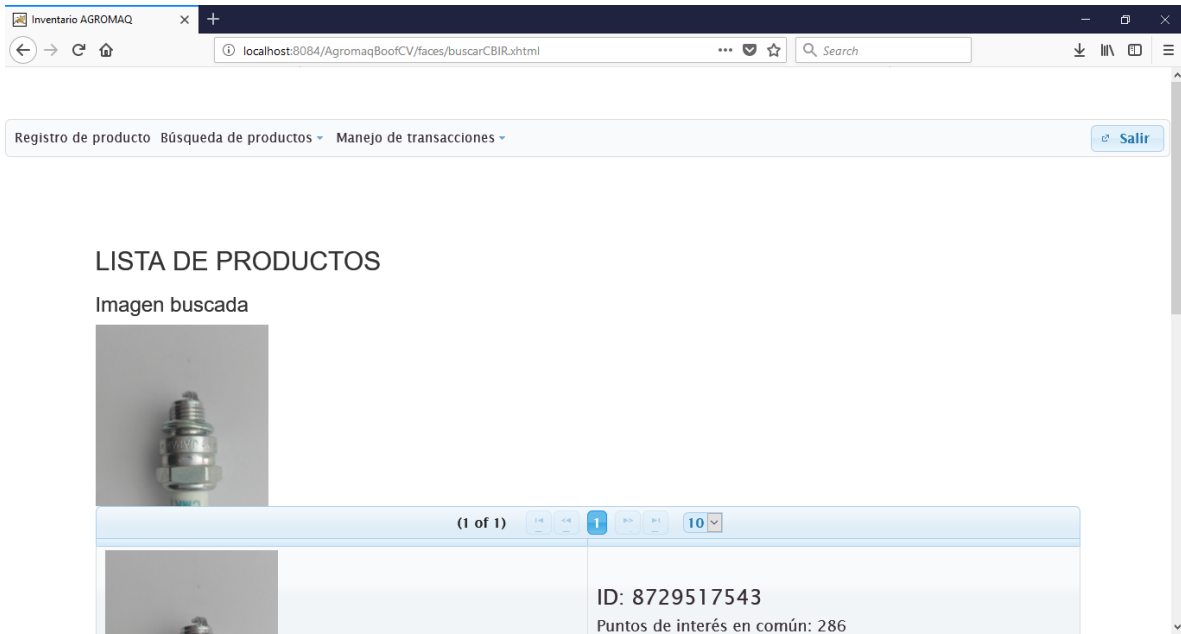


Figura 4.67 Caso de estudio 2 resultado 1 de búsqueda 8

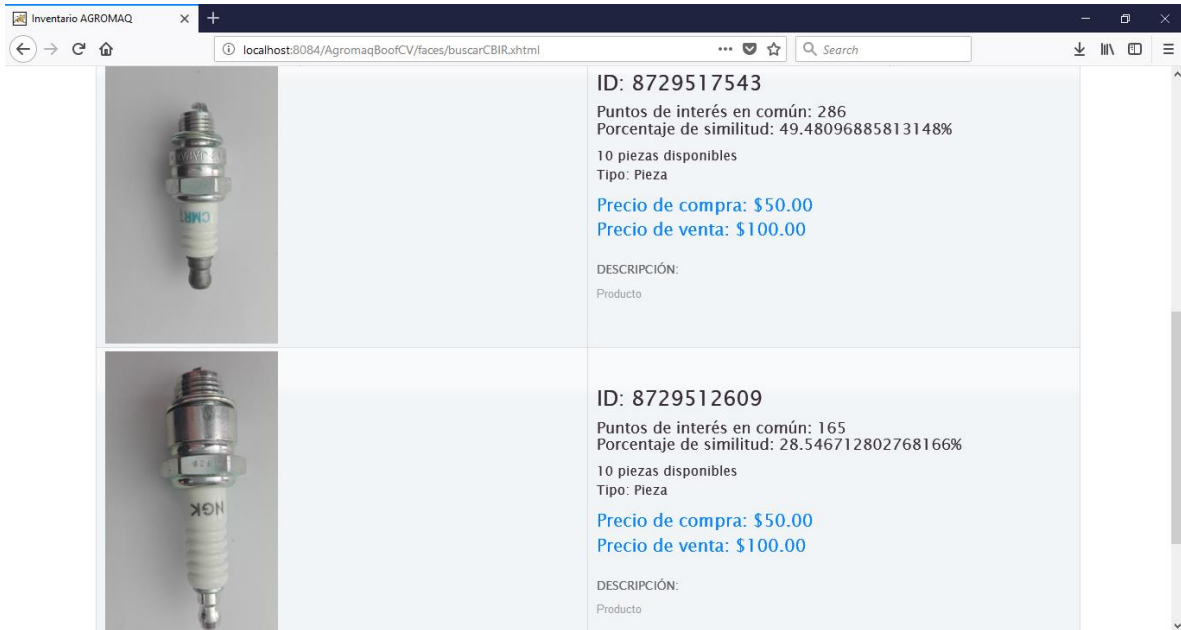


Figura 4.68 Caso de estudio 2 resultado 2 de búsqueda 8

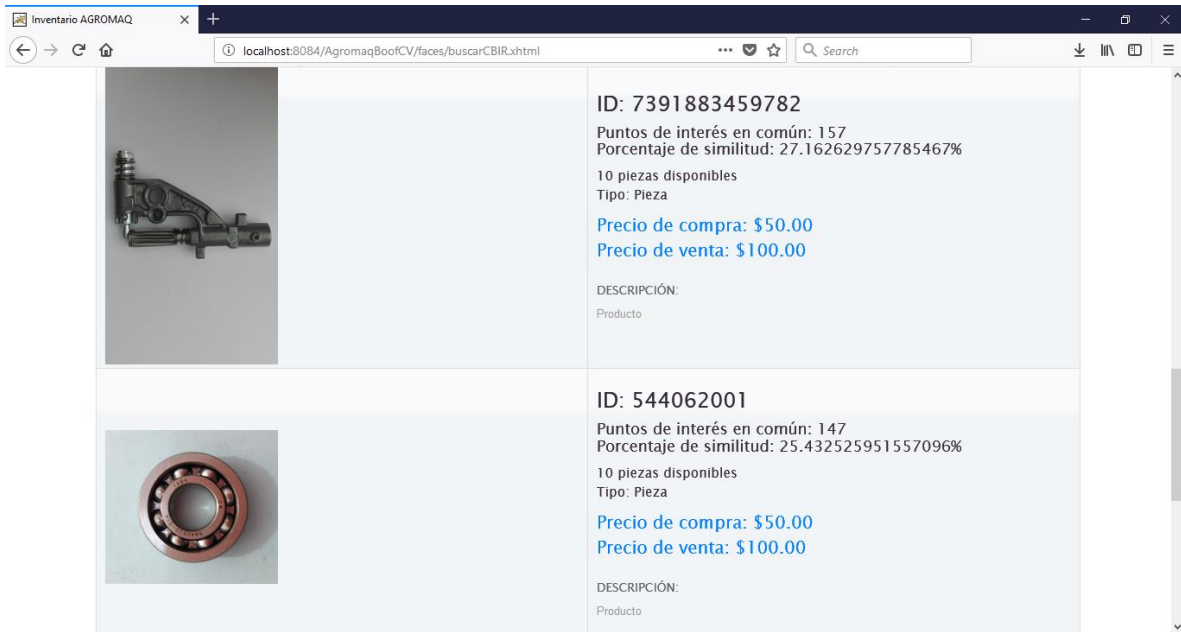


Figura 4.69 Caso de estudio 2 resultado 3 de búsqueda 8

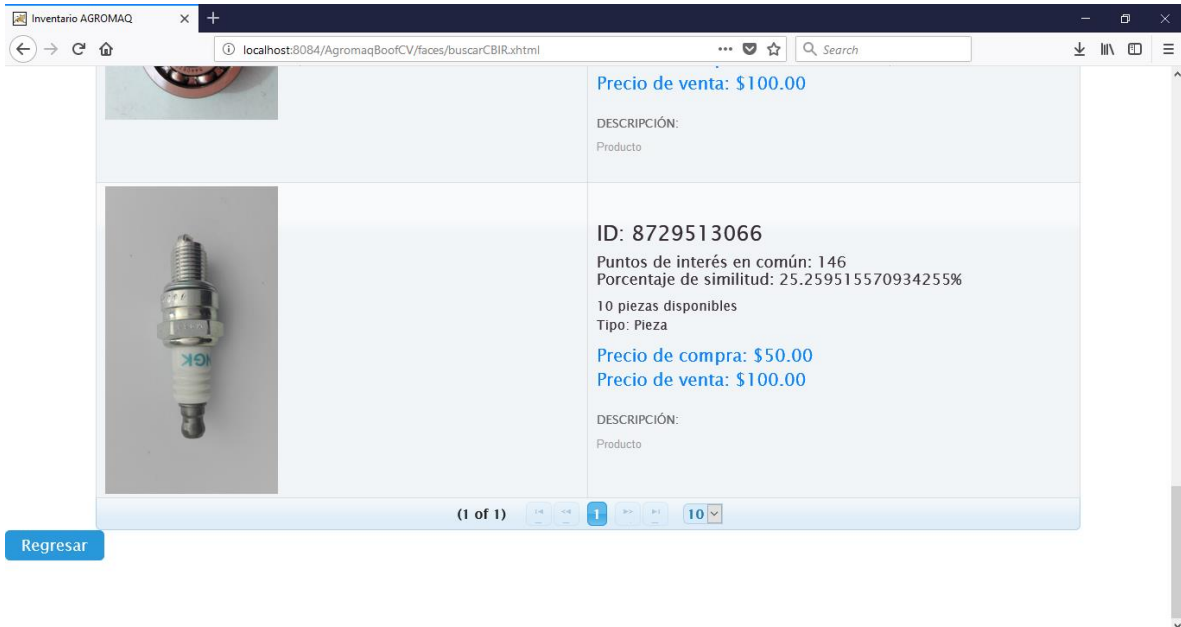


Figura 4.70 Caso de estudio 2 resultado 4 de búsqueda 8

En la Figura 4.71 se observa la imagen consulta 9 y posteriormente se muestran sus resultados en las Figuras 4.72 y 4.73, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

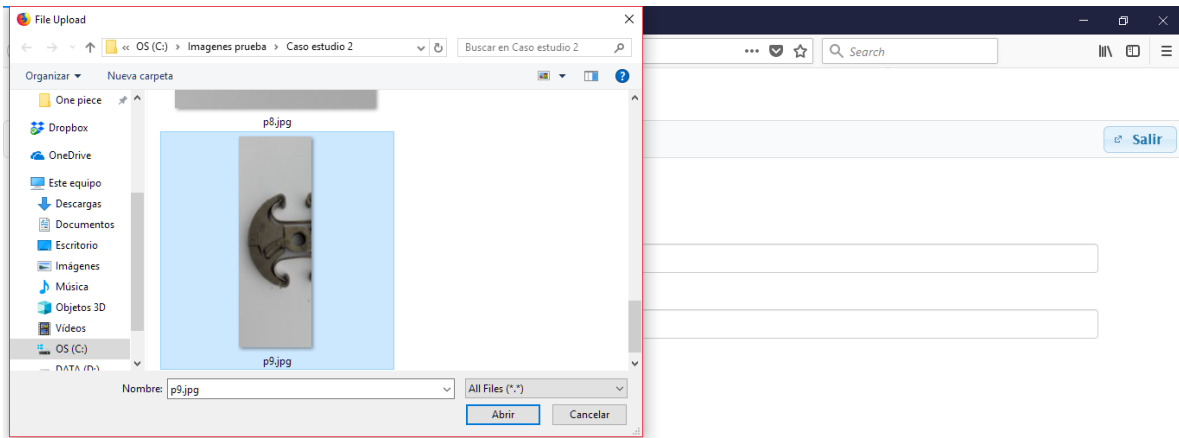


Figura 4.71 Caso de estudio 2 imagen de búsqueda 9

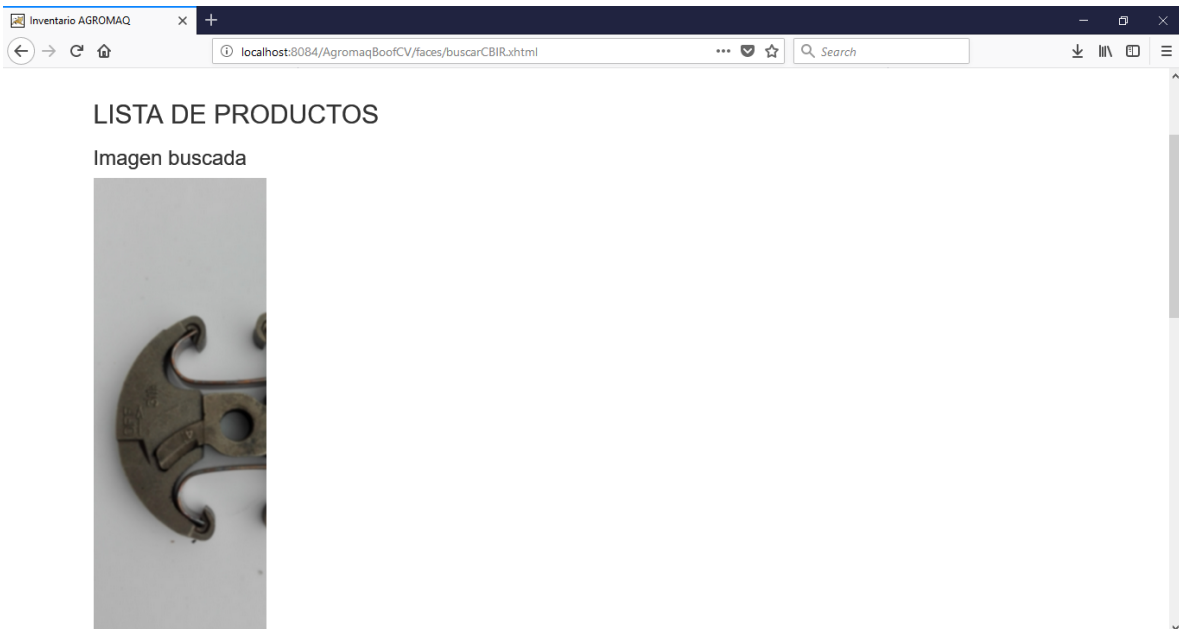


Figura 4.72 Caso de estudio 2 resultado 1 de búsqueda 9

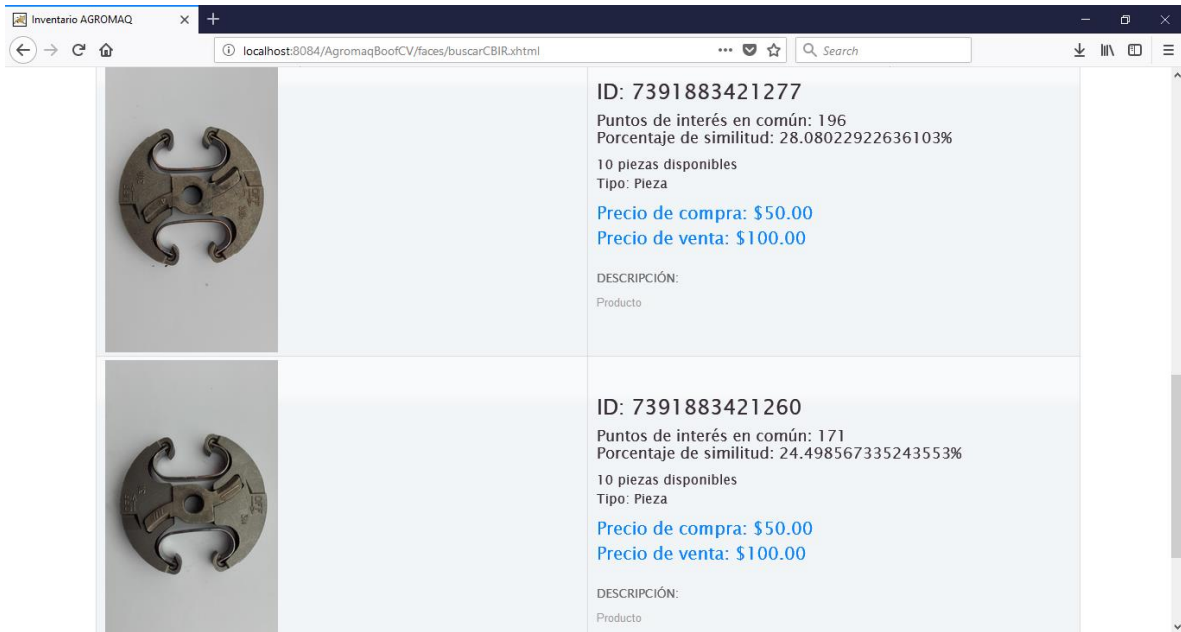


Figura 4.73 Caso de estudio 2 resultado 2 de búsqueda 9

En la Figura 4.74 se ilustra la imagen consulta 10 y posteriormente se muestran sus resultados en las Figuras 4.75 y 4.76, la consulta se realizó con $k=2$ y 2 como número de imágenes relevantes en la base de datos.

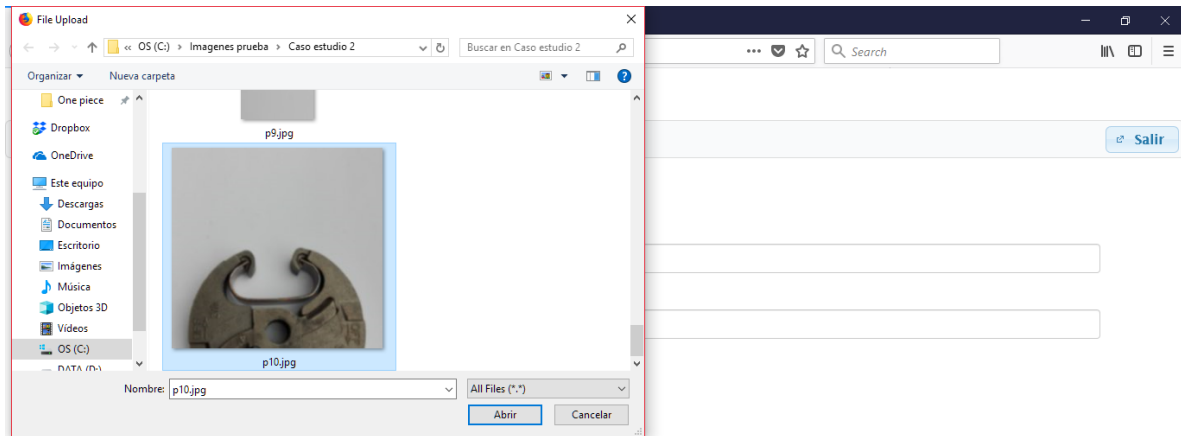


Figura 4.74 Caso de estudio 2 imagen de búsqueda 10

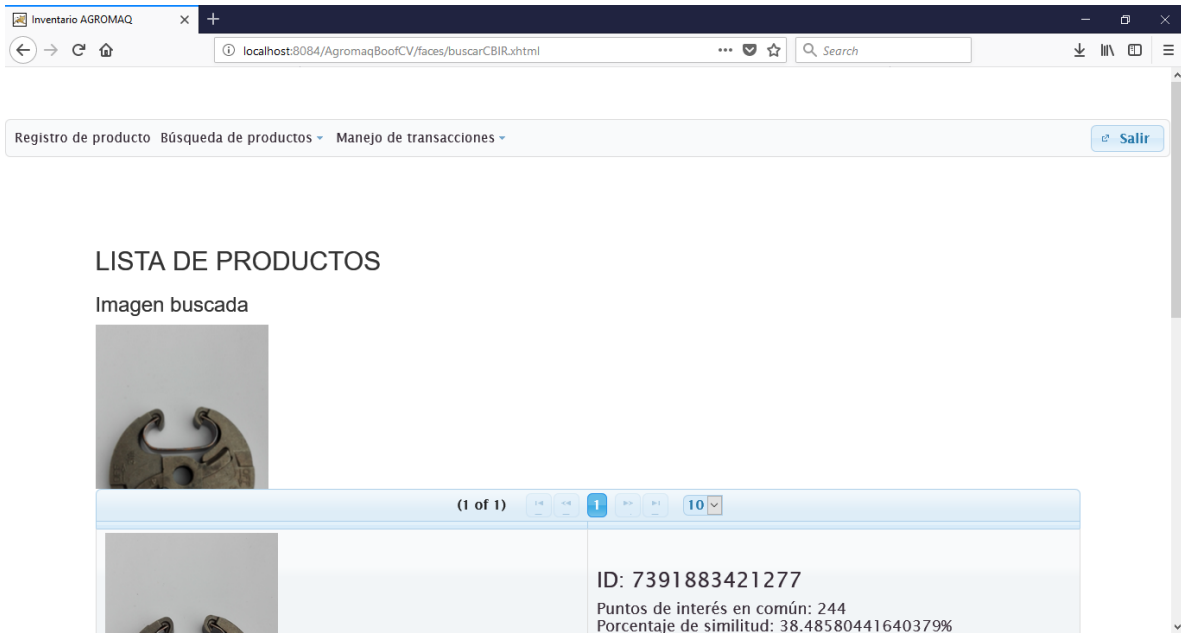


Figura 4.75 Caso de estudio 2 resultado 1 de búsqueda 10

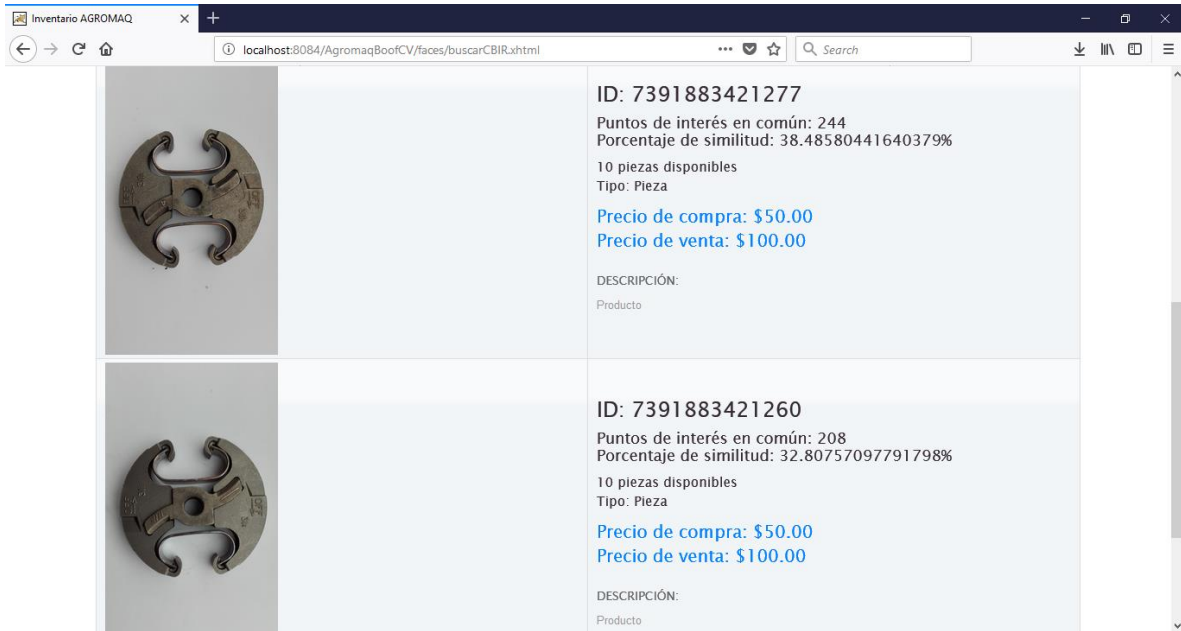


Figura 4.76 Caso de estudio 2 resultado 2 de búsqueda 10

4.2. Evaluación de resultados

Hay dos parámetros principales utilizados para medir el rendimiento de un sistema CBIR: eficiencia y eficacia. La eficiencia se refiere principalmente al tiempo de respuesta del sistema, mientras que la eficacia se relaciona con la capacidad del sistema para recuperar elementos relevantes y descartar elementos irrelevantes [25]. Con el fin de evaluar el rendimiento del sistema CBIR presentado en este documento, los siguientes experimentos se realizaron con el descriptor SURF de BoofCV v0.26 y el descriptor SIFT de OpenCV 2.4.13, en los casos de estudio sólo se utilizó el descriptor SURF.

4.2.1. Eficiencia del sistema CBIR

En la Figura 4.77 se observa el tiempo de ejecución de las consultas con el sistema que utiliza el descriptor SURF contra uno que tiene el descriptor SIFT, el cual es parte de la biblioteca de OpenCV [34]. La evaluación de desempeño se ejecutó en diferentes casos que variaron entre sí por el número de imágenes en la base de datos, desde 50 imágenes hasta 100 imágenes aumentaron 10 imágenes en cada caso, tanto para SIFT como para SURF. En todos los casos, el número de imágenes recuperadas fue 5.

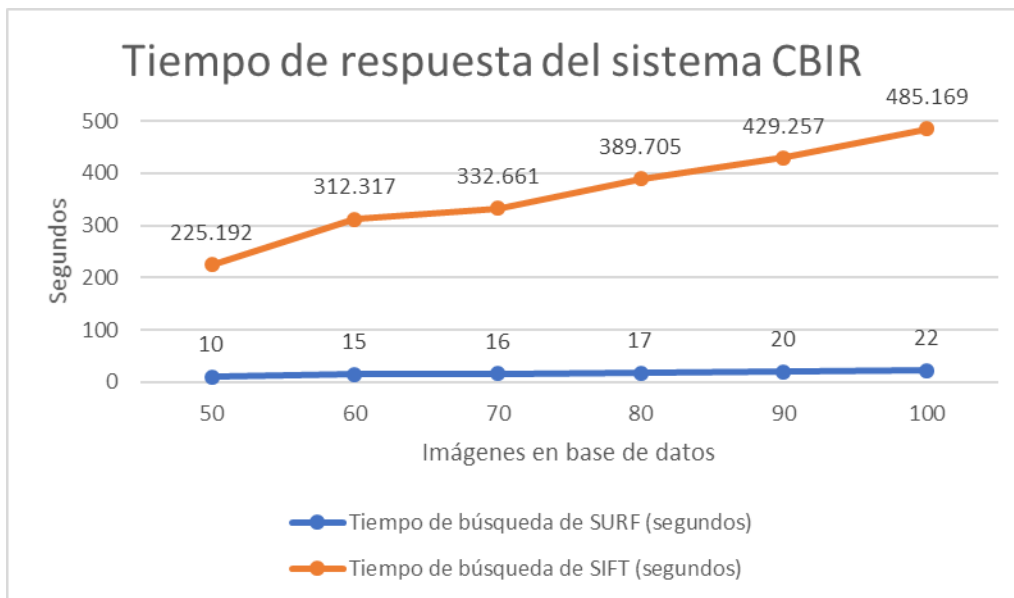


Figura 4.77 Gráfica comparativa de eficiencia usando SIFT y SURF

La Figura 4.77 muestra claramente que el sistema CBIR que aplica el descriptor SURF tiene una mejor eficiencia que aquel donde se implementa SIFT.

4.2.2. Eficacia del sistema CBIR

Con base en el estado del arte presentado en el capítulo dos, el *recall* y la precisión son las medidas más utilizadas para evaluar la eficacia de los sistemas CBIR. El *recall* mide la capacidad de recuperar elementos relevantes, y la precisión mide la capacidad de rechazar elementos irrelevantes. Un buen sistema debería tener valores de recuperación y precisión elevados [32, 36]. Por lo tanto, se realizaron pruebas en el sistema a dos grupos diferentes de imágenes (Piezas y Maquinaria) en 10 iteraciones con imágenes en perfecto estado (sin recortes ni borrosas), las consultas se hicieron a 100 imágenes en la base de datos, los resultados que se obtuvieron de la precisión y el *recall* mediante las ecuaciones (1) y (2) [35] se muestran en la Figura 4.78.

$$\text{Recall} = \frac{\text{Número de imágenes relevantes recuperadas}}{\text{Número de imágenes relevantes}} \quad (1)$$

$$\text{Precisión} = \frac{\text{Número de imágenes relevantes recuperadas}}{\text{Número total de imágenes recuperadas}} \quad (2)$$

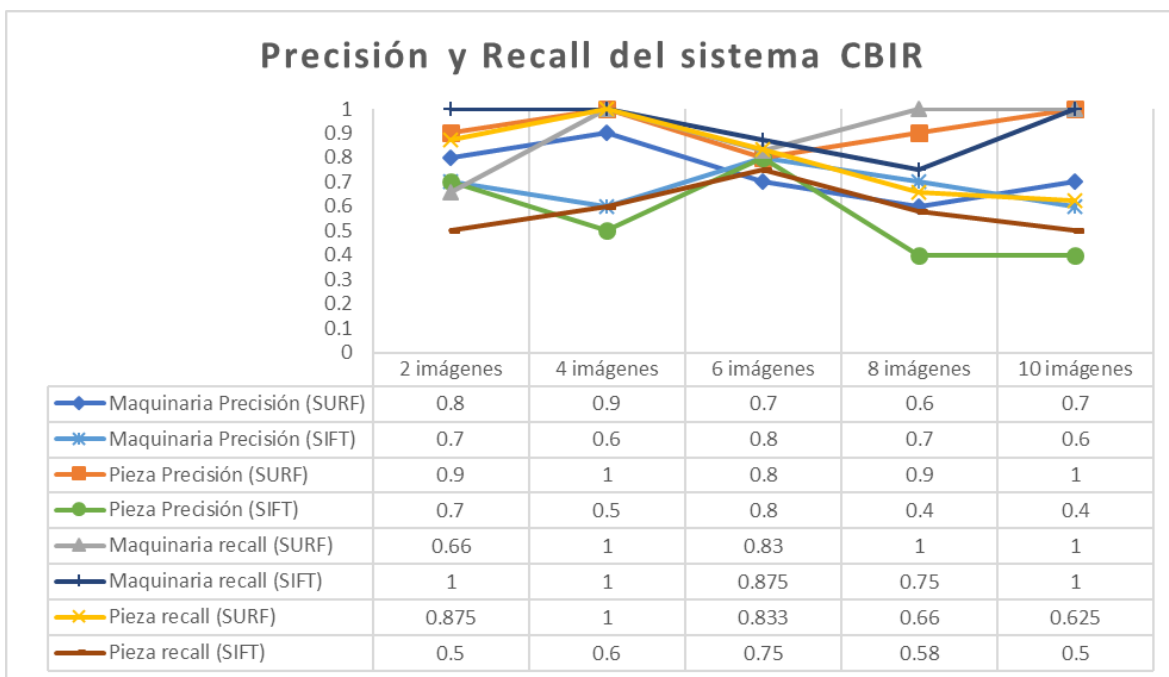


Figura 4.78 Gráfica comparativa de la eficacia entre SIFT y SURF

Los resultados de la Figura 4.78 muestran que SIFT tiene una precisión promedio del 68% en la búsqueda de maquinaria y del 56% en la de piezas, mientras que SURF tiene una precisión del 72% en la búsqueda de maquinaria y del 92% en la de piezas.

También se aprecia que en cuanto al *Recall* durante la búsqueda de maquinaria y piezas, SIFT tiene un promedio del 92.5% y 58.6%, respectivamente, mientras que SURF tiene un promedio de 89.8% en maquinaria y 79.86% en piezas.

En la Figura 4.79 se muestra una comparativa de los resultados obtenidos con la aplicación de diferentes filtros a las imágenes de búsqueda para así determinar cuál filtro era el que mejor trabajaba, llegando a la conclusión de que el descriptor SURF tiene un desempeño muy bueno en las imágenes sin la aplicación de ningún filtro, debido a la diferencia mínima entre los puntos de interés en común detectados entre la imagen consulta y las imágenes en la base de datos.

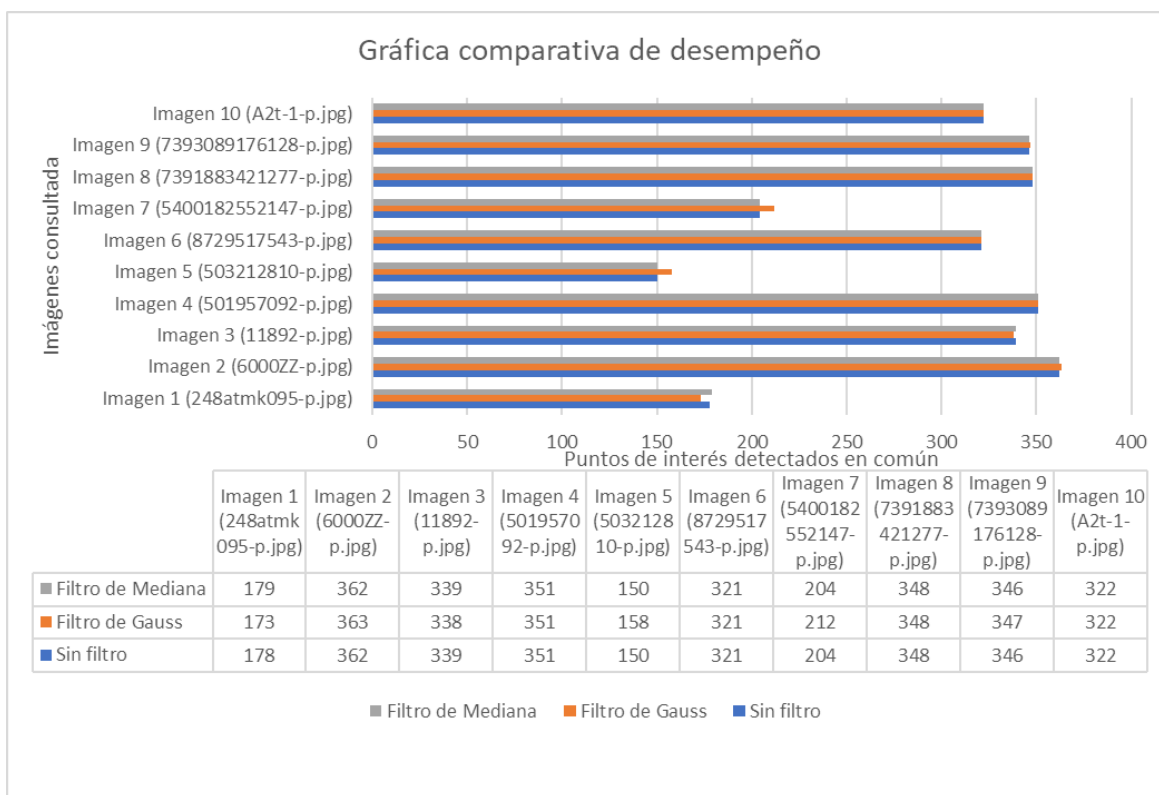


Figura 4.79 Gráfica de desempeño de filtros

En la Figura 4.80 se observa el desempeño calculado con base en el primer caso de estudio donde se muestra que el sistema CBIR tiene una precisión que va del 50% hasta el 100% y un promedio del 70.6%, de igual forma se obtiene un *recall* que va del 50% hasta el 100% con un promedio del 83.7% cuando la imagen consulta es borrosa.

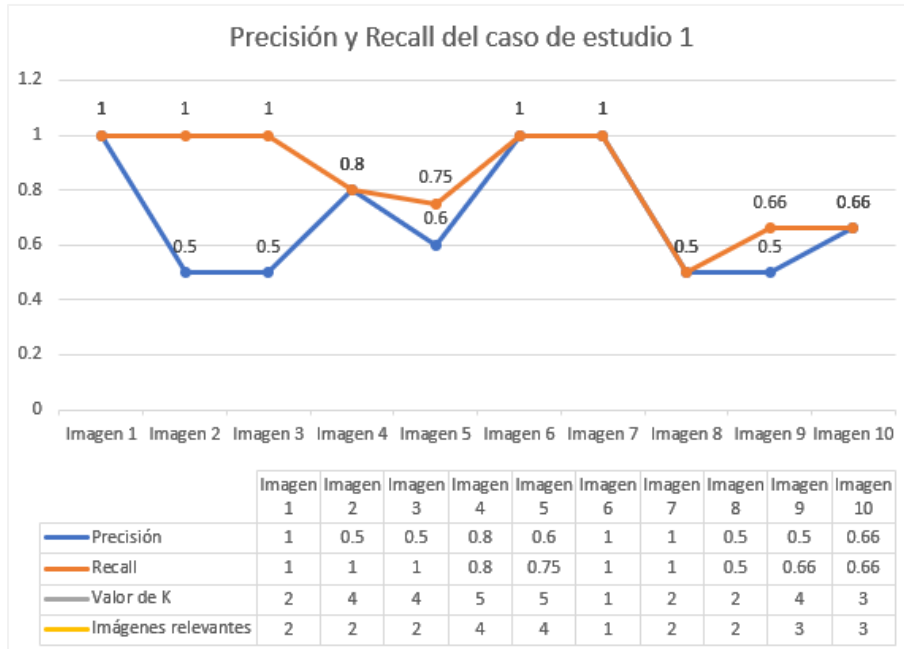


Figura 4.80 Gráfica de desempeño del sistema CBIR en el primer caso de estudio

En la Figura 4.81 se observa el desempeño calculado con base en el segundo caso de estudio donde se muestra que el sistema CBIR tiene una precisión que va del 50% hasta el 100% y un promedio del 76.5%, de igual forma se obtiene un *recall* que va del 50% hasta el 100% con un promedio del 83% cuando la imagen consulta está recortada.

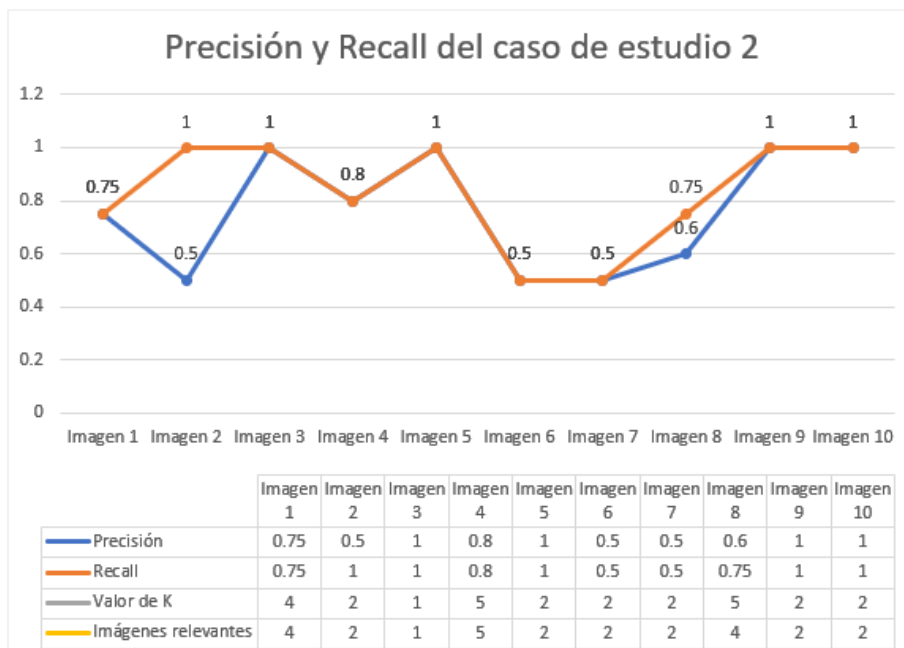


Figura 4.81 Gráfica de desempeño del sistema CBIR en el segundo caso de estudio

Capítulo 5. Conclusiones y recomendaciones

5.1. Conclusiones

Los sistemas CBIR son importantes a nivel comercial debido a que hacen más fácil la tarea de identificación de objetos, refacciones en el caso de este proyecto, a personas que no son expertas en la materia, así como el hecho de que ya no se dependería tanto de personas expertas en el área, con esto se busca hacer que los tiempos de atención al cliente sean menores.

El objetivo de este trabajo fue realizar una aplicación que permita realizar consultas basadas en contenido, en la cual mediante una imagen dada por el usuario se recuperan las imágenes más similares, esto mediante la detección de los puntos de interés en las imágenes, tanto la de consulta como las almacenadas en la base de datos, con la ayuda del descriptor SURF que proporciona la plataforma de BoofCV.

Dependiendo de la finalidad del sistema algunas técnicas serán más recomendables para su uso en un sistema CBIR como ya se vio en este documento, el descriptor SURF se usó en el proyecto debido a que se basa en la obtención de los objetos dentro de las imágenes y así compararlos, en este caso las refacciones que se requieren, además de que su precisión es más alta en comparación con SIFT como se vio en los resultados obtenidos y presentados en el capítulo 4.

De igual forma, la diferencia drástica del tiempo de ejecución de las consultas es gracias a que BoofCV proporciona los objetos de los descriptores obtenidos que son más fáciles de manejar dentro de Java, esto permite su almacenamiento en archivos (.txt) para así disminuir este tiempo de cálculo durante cada consulta.

El descriptor SURF de BoofCV obtuvo buenos resultados a la hora de realizar consultas con imágenes sin la aplicación de algún tipo de filtro como se observó en la Figura 4.79, donde se muestra que la diferencia entre cada una de las consultas (con o sin filtro) es mínima, así mismo los resultados de cada caso de estudio muestran una precisión del 70.6% y un *recall* del 83.7% en el primer caso de estudio y una precisión del 76.5% y un *recall* del 83% en el segundo caso de estudio.

5.2. Recomendaciones

A pesar de que este trabajo cumplió con el objetivo planteado, sería beneficiosa la implementación de un sistema para tomar las fotografías con la finalidad de controlar la iluminación de las mismas, en la Figura 5.1 se visualiza el prototipo del sistema.

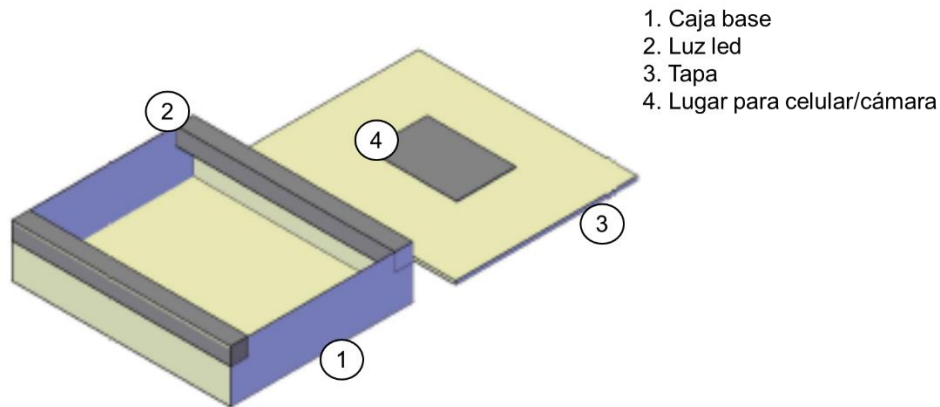


Figura 5.1 Prototipo del sistema para capturar fotografías

De igual manera, otra recomendación es intentar mejorar el tiempo de ejecución de las consultas buscando alguna otra forma para guardar y manejar los descriptores extraídos de las imágenes, también se espera que se actualice la biblioteca de BoofCV, pues actualmente se está implementando la versión 0.26, pero se estima que el algoritmo de SURF o alguno de sus componentes tendrá una mejoría en un futuro. Por último, se espera agregar un lector de código de barras para así facilitar la búsqueda por texto al sistematizar el ingreso de dicho dato.

Productos académicos



Rojas Ruiz Rodolfo, Rodríguez Mazahua Lisbeth, López Chau Asdrúbal, Peláez Camarena S. Gustavo, Abud Figueroa Ma. Antonieta, “Diseño de un sistema para realizar consultas basadas en contenido en una base de datos multimedia”, ANIEI 2017.



Rojas Ruiz Rodolfo, Rodríguez Mazahua Lisbeth, López Chau Asdrúbal, Peláez Camarena S. Gustavo, Abud Figueroa Ma. Antonieta, Machorro Cano Isaac, “A CBIR System for the Recognition of Agricultural Machinery”, WIDSSI 2017.

Bibliografía

- [1] S. Neeraj et al, “Database information models” in Database fundamentals, 1st ed, Markham, Canada, 2010, ch. 1, sec. 1.1, pp. 23.
- [2] I. Assent, “Efficient adaptive retrieval and mining in large multimedia databases” in Dissertation, Fakultät für Mathematik, Informatik und Naturwissenschaften, RWTH Aachen.
- [3] J. Kommineni et al, “Content Based Image Retrieval Using Colour Strings Comparison” in Procedia Computer Science, Vol. 50, 2015, pp. 374-379.
- [4] S. Tomas et al, “Distinct Nearest Neighbors Queries for Similarity Search in Very Large Multimedia Databases” in Proceedings of the eleventh international workshop on Web information and data management, New York, 2009.
- [5] R. Sergui et al, “Tecnologías big data para análisis y recuperación de imágenes Web” en El profesional de la información, vol. 23, No. 6, pp 567-574, 2014.
- [6] Lew, M. S. et al, “Content-Based Multimedia Information Retrieval: State of the Art and Challenges. ACM Trans.” in Multimedia Computing, Communications and Applications, Vol. 2, No. 1, pp. 1–19, 2006.
- [7] Suckchul, K. et al, “mClover: mobile Content-based Leaf Image Retrieval System” In the MM, ACM, 2005.
- [8] Springmann, M. et al, “QbS-Searching for Known Images using User- Drawn Sketches” in the MIR, ACM, 2010.
- [9] D. Gloria et al., “A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images” in Journal of Biomedical informatics, Vol. 42, pp. 296-307, 2009.
- [10] S. C. James et al, “Plant species identification using digital morphometrics” in Expert Systems with Applications, Vol 39 ,No. 8, 2012, pp. 7562–7573.
- [11] J. Lee et al, “Image Retrieval in Forensics: Tattoo Image Database Application” in Multimedia in Forensics, Security and Intelligence, Vol 19 , No.1, 2012, pp. 2-11.

- [12] Q. Zheng et al, “Effective and efficient object-based image retrieval using visual phrases” in Proc. 14th ACM International Conference on Multimedia, Santa Barbara, 2006, pp. 77-80
- [13] Y. Lee et al, “Pill-ID: Matching and retrieval of drug pill images” in Pattern Recognition Letters, Vol 33, 2012, pp. 904-910.
- [14] D. Thomas et al, “Features for Image Retrieval: An Experimental Comparison” in Journal Information Retrieval, 2007, Vol. 11, No. 2, pp 77-107.
- [15] Z. Yu and H. Wong, “A Rule Based Technique for Extraction of Visual Attention Regions Based on Real-Time Clustering” in IEEE Transaction on multimedia, Vol 9, No. 4, 2007, pp. 766-784.
- [16] A. Avi et al, “Dynamic two-stage image retrieval from large multimedia databases” in Information Processing and Management, Vol 49, pp. 274-285, 2013.
- [17] S. Jorge and et al, “Image Classification with the Fisher Vector: Theory and Practice” in International Journal of Computer Vision manuscript, Vol 105, No. 3, 2013.
- [18] H. Jan et al, “What makes for effective detection proposals?” in Journal IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, No. 4, pp 814-830, 2015.
- [19] Z. Konstatinos et al, “Accurate image retrieval based on compact composite descriptors and relevance feedback information” in International Journal of Pattern Recognition and Artificial Intelligence, Vol. 24, No. 2, pp. 1-38, 2010.
- [20] P. Otávio A. B. et al, “Comparative Study of Global Color and Texture Descriptors for Web Image Retrieval” in Journal of Visual Communication and Image Representation, Vol. 23, No. 2, pp. 359–380, 2011.
- [21] G. Navel et al., “Content Based Image Retrieval using Enhanced SURF” in Computer Vision, Pattern Recognition, Image Processing and Graphics, India, 2015.

- [22] A. Ronald et al., "Plant Encyclopedia with Image Matching on a Web-Based Application" in International Conference on Information Management and Technology, Indonesia, 2016, pp 89-92.
- [23] V. Pradnya and K. Pravin, "ENHANCING CONTENT BASED IMAGE RETRIEVAL BY USING SKETCHES" in Information Communication and Embedded Systems, India, 2014.
- [24] B. Kaur and S. Jindal, "An implementation of Feature Extraction over medical images on OPEN CV Environment" in Devices, Circuits and Communications, India, 2014
- [25] A. Alexandra et al., "Content Based Image Retrieval And Classification Using Speeded-Up Robust Features (SURF) and Grouped Bag-of-Visual-Words (GBoVW)" in International Conference on Technology, Informatics, Management, Engineering & Environment, Indone-sia, 2013, pp. 77–82.
- [26] Oracle, "Java Developer Center" [Online]. Available: <http://www.oracle.com/technetwork/es/Java/index.html>. Accessed: Oct. 30, 2016.
- [27] JavaServer Faces, "*Introducción a JavaServer Faces*". [online] Available at: <http://www.jtech.ua.es/>, Accessed 24 Nov. 2017
- [28] PrimeFaces, *Why PrimeFaces* [online] Available at: <https://www.primefaces.org/whyprimefaces/>, Accessed 24 Nov. 2017.
- [29] Schwabe, d. Rosi, et al, "S. Systematic Hypermedia Application Design with OOHDMM" in Proceedings of the ACM International Conference on Hypertext, Washington DC, March, pp. 16-20, 1996.
- [30] NetBeans, "NetBeans Docs & support," 2016. [Online]. Available: <https://netbeans.org/kb/index.html>. Accessed: Oct. 30, 2016.
- [31] PostgreSQL, "PostgreSQL: Manuals," 1996. [Online]. Available: <https://www.postgresql.org/docs/manuals/>. Accessed: Oct. 30, 2016.

[32] M. Romero et al, “Comparativa de detectores de características visuales y su aplicación al SLAM” in Comparativa de detectores de características visuales y su aplicación al SLAM, Cáceres, pp. 55-62, 2009.

[33] Boofcv.org, “BoofCV” [online] Available at: https://boofcv.org/index.php?title=Main_Page [Accessed 24 Nov. 2017].

[34] OpenCV, "DOCUMENTATION OpenCV" 2016. [Online]. Available: <http://opencv.org/documentation.html>. [Accessed: Oct. 30, 2016].

[35] D. Sunita y K.Hemachandran, “Retrieval and Recognition of faces using Content-Based Image Retrieval (CBIR) and Feature Combination method” in IEEE International Conference on Computational Intelligence and Computing Research, India, 2016

[36] G. Lu, "Multimedia Database Management Systems", Artech House Computing Library, 1999.