

Trabajo final de grado

Universidad Nacional de Río Negro

Sede Atlántica

Licenciatura en Sistemas



DETECCIÓN AUTOMÁTICA DE ENFERMEDADES EN ABEJAS MELÍFERAS

Director: Dra. Britos Paola Verónica.

Codirector: Ing. Cambarieri Mauro German.

Tesista: Castro, Patricio Nicolas.

Viedma, Río Negro, 24 de Noviembre de 2017

Agradecimientos

Quiero agradecer en primera instancia a mis Padres, mi Familia y Afectos, Mis Amigos los que están cerca y no tanto. También a quienes directa o indirectamente han colaborado a lo largo de estos años, desde mi comienzo en la Universidad hasta la actualidad.

A mi Directora por enseñarme a escribir el presente trabajo de grado, por su tiempo y dedicación.

Al Coordinador de la Carrera de la Licenciatura en Sistemas por su constante motivación y desafíos presentados.

Por último, a los compañeros del Laboratorio de Informática Aplicada y a los compañeros de trabajo de Rectorado.

Resumen

El presente trabajo propone una solución basada en técnicas de visión artificial para automatizar el proceso de detección de enfermedades en abejas melíferas. La herramienta permite generar apiarios con su información correspondiente y asociar imágenes al mismo. Luego se procesan dichas imágenes donde los resultados informan el número de esporas y varroas presentes en cada imagen y en el total del apiario para cada una de las enfermedades consideradas.

En el proyecto se diseña e implementa una Aplicación Web que sirve de ayuda a los laboratoristas y los apicultores para el análisis y el diagnóstico de dos enfermedades que afectan a las abejas melíferas llamadas nosemosis y varroasis. El trabajo se realizó en colaboración con los biólogos: el Dr. Sergio Damian Abate y la Lic. Paola Andrea Pizzingrilli, que participaron en la adquisición de las imágenes y en la posterior validación de los resultados.

El marco de trabajo utilizado es Scrum ya que permite adaptarse de manera fácil a los cambios y responder de manera rápida a los requerimientos solicitados. Además, para la implementación se emplean tecnologías basadas en el lenguaje de programación Java junto con la librería OpenCV, que implementa los algoritmos de visión artificial.

Palabras Clave: Detección de enfermedades, esporas, varroas, visión artificial, desarrollo web, Java.

Índice

Capítulo 1: Introducción	6
1.1 Apicultura	6
1.2. Enfermedades de las abejas	8
1.2.1. Nosemosis	8
1.2.2. Varroa	11
1.3. Objetivos	13
1.4. Organización del trabajo	15
Capítulo 2: Estado del arte	16
2.1 Etapas de un sistema de visión artificial	16
2.1.1 Adquisición de la imagen	17
2.1.2 Procesamiento digital de las imágenes	18
2.1.3 Segmentación de las imágenes	18
2.1.4 Reconocimiento o clasificación de imágenes	19
2.2 Librerías de visión artificial	19
2.3 Detección de enfermedades utilizando visión artificial	20
Capítulo 3 - Método de trabajo	22
3.1 Posible solución	22
3.2 Metodología de trabajo	23
3.3 Plan de trabajo	24
Capítulo 4: Propuesta de solución	27
4.1 Metodología de desarrollo a utilizar	27
4.2 Tecnologías a utilizar	33
4.2.1 Arquitectura de la aplicación	33
4.2.1.1 Arquitectura en capas	33
4.2.2 Librería de visión artificial a utilizar	37
4.2.3 Lenguaje de programación a utilizar	38

4.2.4 Otras tecnologías	38
4.2.5 Flujo de trabajo	39
4.2.6 Técnicas de programación	40
4.3 Planificación y ejecución del proyecto	41
4.3.1 Sprint Nro. 0, instalación del entorno de desarrollo, repositorio y servidor de la aplicación	46
4.3.1.1 Publicación automática	51
4.3.1.2 Estructura del proyecto	52
4.3.1.3 Interfaces del proyecto base	56
4.3.2 Sprint Nro. 1 Administración del Sistema Web	59
4.3.3 Sprint Nro. 2 Módulo de identificación y conteo de esporas de nosemas	62
4.3.4 Sprint Nro. 3 Módulo de identificación y conteo de ácaros de varroa	70
Capítulo 5: Validación de los resultados	79
5.1 Casos de prueba	79
Caso de prueba: C.P. 1 - Creación de usuario	80
Caso de prueba: C.P. 2 - Creación de apiario	82
Caso de prueba: C.P. 3 - Cargar imágenes en un apiario	84
Caso de prueba: C.P. 4 - Configuración del algoritmo	87
Caso de prueba: C.P. 5 - Detección y conteo de nosemas	89
Caso de prueba: C.P. 6 - Detección y conteo de varroas	96
Capítulo 6: Conclusiones y líneas futuras	107
Capítulo 7: Bibliografía	110

Capítulo 1: Introducción

El presente documento forma parte del proyecto de tesina necesario para obtener el título de grado de la carrera de Licenciatura en Sistemas perteneciente a la Universidad Nacional de Río Negro. En este trabajo se presenta un proyecto cuyo nombre es “Detección Automática de Enfermedades en Abejas Melíferas” (DAEAM).

El objetivo del proyecto es diseñar e implementar una Aplicación Web que permita, mediante la utilización de técnicas de visión artificial, detectar automáticamente dos tipos de enfermedades pertenecientes a las abejas melíferas “apis mellifera”: la nosemosis y la varroasis.

La solución propuesta se centra en analizar imágenes digitales microscópicas de los intestinos de las abejas para realizar el conteo de las esporas de nosemas y detectar nosemosis. Se busca realizar una tarea similar a partir de imágenes captadas por una cámara digital o un dispositivo móvil del microscopio de los pisos trampas de los apiarios para realizar el conteo de ácaros de varroa, que son visibles a simple vista, para la detección de varroasis. Ambos métodos de detección se desarrollan en base a dos descriptores morfológicos como: la forma y el tamaño.

Se espera lograr una herramienta que permita realizar un análisis y seguimiento de las imágenes obteniendo resultados representativos en un tiempo inferior en comparación con el método tradicional realizado a través del conteo manual.

1.1 Apicultura

El problema de detección de enfermedades en las abejas melíferas es un tema perteneciente a la apicultura, actividad dedicada a la crianza y cuidado de las abejas. Esta actividad es importante en el país ya que por un lado brinda ganancias económicas con un gran valor en el mercado apícola, y por otro impacta positivamente en el medio ambiente, contribuyendo en la mantención y preservación

del mismo. El principal producto que se obtiene de la producción apícola es la miel, producto antimicrobiano rico en carbohidratos, proteínas, lípidos, azúcares, varios minerales y vitaminas.

Para mejorar la utilización de este producto y su comercialización es necesario llevar los controles sanitarios correspondientes tanto en la miel como en las abejas, que son quienes la elaboran. Este control permite generar mayores lucros económicos y aumentar la calidad de la misma.

Aunque en la Argentina existen aproximadamente 2,8 millones de colmenas, se estima que su verdadero potencial sería alrededor de 10 millones. Nuestro país es uno de los principales productores y exportadores de miel del mundo: es reconocido internacionalmente por la calidad de sus mieles, dadas las grandes extensiones de pasturas naturales, la abundante flora autóctona y las enormes superficies implantadas con diversos cultivos agrícolas (Hervias & Mogni, 2005). Se destaca también, que produce entre 60 y 80 mil toneladas de miel al año (según datos de la última década), que se exportan casi en su totalidad por lo que la producción de miel es una actividad altamente rentable. Además del interés económico, la producción apícola es relevante por su efecto en la polinización. En la provincia de Río Negro, la producción y comercialización de fruta, es una de las principales fuentes de ingreso en la balanza comercial y requiere la actividad polinizante de las abejas para asegurar un nivel de producción óptimo.

En los últimos años, se observó en la provincia de Río Negro un incremento en la mortalidad de colmenas: aunque no se conocen todas las causas, se ha demostrado que las cenizas del volcán Puyehue generaron un debilitamiento general por causas directas e indirectas, predisponiendo que otras enfermedades se manifiesten con mayor gravedad.

1.2. Enfermedades de las abejas

1.2.1. Nosemosis

La nosemosis es una parasitosis intestinal presente en las abejas melíferas, de tipo multifactorial y de distribución mundial, ocasionada por los microsporidios *Nosema Apis* (Zander) y *Nosema Ceranae* (Fries) (Fries, 2010). En nuestro país el agente etiológico necesario es el microsporidio *Nosema Apis*, aunque se ha descrito a *Nosema Ceranae* como agente de nosemosis en infestaciones puras y mixtas en otros países. Dicha enfermedad afecta a las abejas adultas y a sus crías, causando graves pérdidas económicas para la producción apícola (SENASA, 2014).

Durante su ciclo de vida, *Nosema Apis* produce esporas ovaladas, de 7µm de largo por 3-4µm de ancho aproximadamente con un contorno oscuro (OIE, 2013). Estas esporas se reproducen en las células epiteliales del aparato digestivo de las abejas, destruyéndolas, alterando el normal proceso de secreción enzimática del intestino medio, dificultando el proceso de digestión y absorción de nutrientes, produciendo debilidad general como síndrome de mala absorción, envejecimiento prematuro y mortandad precoz; las abejas enfermas viven la mitad del tiempo que las no afectadas.

La demostración de esporos en lisados de abejas por medio de observación microscópica (ver Figura 1.1, Figura 1.2 y Figura 1.3) es el método diagnóstico para demostrar la infestación de una colmena (OIE, 2013). Debido a que esta enfermedad puede causar graves daños económicos cuando el nivel de infestación es elevado (≥ 500.000 esporos/abeja), se realiza una estimación del grado de infestación mediante el recuento de esporos en cámara de Neubauer, luego de morterear un número representativo de abejas de una colmena.¹ Este método de diagnóstico es de escaso nivel tecnológico y posee un costo relativamente bajo, no obstante presenta varias limitaciones:

¹ Moler o machacar a mano con un mortero.

- a) Reducida capacidad operativa: el recuento microscópico de esporos en cámara de Neubauer, por duplicado, insume entre 2 y 5 minutos, sin contar el procesamiento previo de las abejas; esto trae como consecuencia que no se realice el procesamiento de un número elevado de muestras y, en los apiarios se estile el procesamiento de muestras provenientes solamente de algunas colmenas. Esta situación relativiza los hallazgos de laboratorio: dado que en un mismo apiario se pueden encontrar simultáneamente colmenas con un elevado grado de infestación subclínica, y otras con un grado bajo de infestación, el submuestreo de algunas colmenas puede no representar el estado sanitario de cada una de las colmenas de un apiario.
- b) La observación microscópica durante más de 1 hora constituye una causa de agotamiento visual y posible motivo de errores durante la lectura, interpretación y recuento de esporos en cámara de Neubauer.
- c) Falta de concentración al realizar demasiado tiempo la misma tarea.
- d) Es un trabajo tedioso que consume mucho tiempo y genera resultados imprecisos.
- e) Es un método subjetivo porque los resultados que genera dependen en gran parte del criterio del observador.

Dentro de la cadena de producción de miel el diagnóstico de enfermedades en las abejas es una etapa de gran importancia ya que permite el monitoreo y control del desarrollo de los apiarios. En este marco, resulta de gran utilidad poder contar con un método diagnóstico que permita el procesamiento de un mayor número de colmenas de un apiario. Lo ideal sería 100% de las colmenas de cada apiario o en su defecto la mayor cantidad posible, en la menor cantidad de tiempo, con la mayor sensibilidad y especificidad en sus resultados lo que permitiría el diagnóstico temprano: la rapidez en la obtención de resultados haría posible tomar medidas sanitarias adecuadas y oportunas, evitando la muerte de la colmena y las posteriores consecuencias económicas.

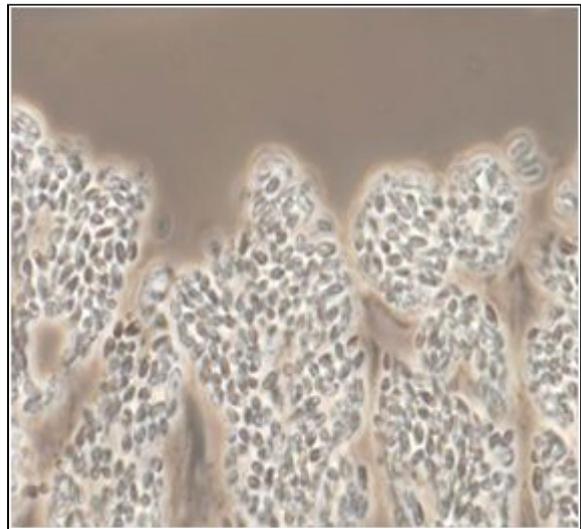


Figura 1.1. Procesamiento de las abejas para la obtención de esporas



Figura 1.2. Operador observando la muestra

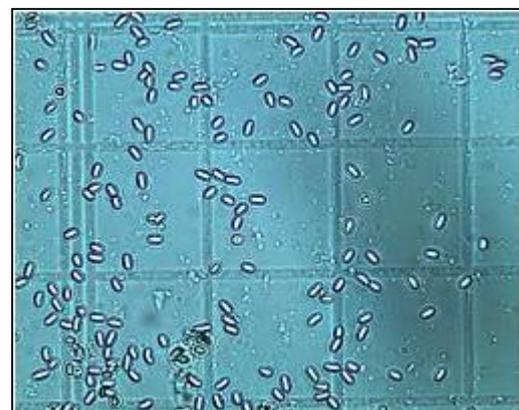


Figura 1.3. Muestra con esporas de nosemas

1.2.2. Varroa

La varroosis es una enfermedad causada por un ácaro, un parásito externo que afecta a las abejas productoras de miel (OIE, 2008). Dicha enfermedad afecta a las abejas adultas y a sus crías. Esto al igual que la nosemosis causa graves pérdidas económicas para la producción apícola (SENASA, 2014).

En esta afección los ácaros succionan la hemolinfa (sangre) de las abejas. Los primeros signos de infección normalmente pasan desapercibidos y solo cuando la infección es masiva se hacen aparentes, etapa en la cual se pueden observar ácaros adultos en las abejas. También es uno de los factores responsables del síndrome de colapso de colmenas y es una de las mayores amenazas presentes en la apicultura debido a que se encuentra mundialmente distribuida.²

La varroa tiene dimorfismo sexual: hembra y macho se diferencian en forma y tamaño. La varroa hembra posee un tamaño de 1,1mm x 1,5mm aproximadamente con un color marrón claro a castaño oscuro con forma elíptica (ver Figura 1.4), mientras que la varroa macho tiene un tamaño de 1mm con un color marrón claro con forma esférica (ver Figura 1.5) (Cardozo, 2014; González, 2013). Se trata además del único parásito de la abeja que se puede ver a simple vista (OIE, 2008).

Los ácaros que se observan parasitando abejas son únicamente hembras, el macho solo tiene un papel reproductivo.

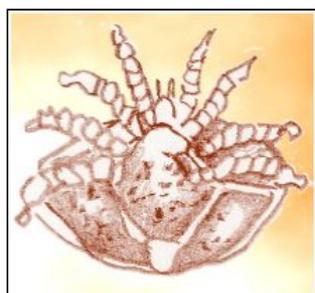


Figura 1.4. Ácaro de varroa hembra adulta



Figura 1.5. Ácaro de varroa macho adulta

² Fenómeno en el que las abejas obreras de una colmena desaparecen bruscamente quedando únicamente la reina y las crías.

Método de diagnóstico a través de pisos trampa

Cómo esta enfermedad se encuentra en todas o la mayoría de las colmenas, en general no se realiza un diagnóstico de la misma sino que se aplica la cura directamente, mayormente mediante el uso de antiparasitarios. En aquellos casos donde se realiza el diagnóstico se utiliza la llamada “prueba del frasco” sugerida por SENASA (Recomendaciones para el control de varroasis, 2010).³ Dicha prueba tiene como riesgos:

- Atrapar a la reina accidentalmente.
- Sacrificar las abejas atrapadas en la muestra (entre 300 y 500).

Una alternativa a este método es la utilización de pisos trampa como método de diagnóstico (ver Figura 1.6 y Figura 1.7) que permite realizar un muestreo y seguimiento sin afectar la producción. Dicho control de la varroasis implica cambiar el piso de la colmena por una superficie con aberturas donde los ácaros caen y no pueden regresar a incorporarse a la colonia ni a las abejas en particular, ya que se utilizan pisos untados con una base de aceite o vaselina.

La utilización de este método posibilita la recolección de los ácaros en forma permanente durante cualquier intervalo de tiempo, sin la necesidad de sacrificar abejas para realizar el conteo manual de los mismos.



Figura 1.6. INTA columna 1 exterior



Figura 1.7. Colocación de nuevo piso

³ Prueba de diagnóstico que permite conocer qué porcentaje de varroas hay en las colmenas.

Conteo manual de varroas

Al igual que ocurre con las esporas de nosemas aquí se realiza un conteo manual de los ácaros de varroas con la diferencia de que no se precisa un microscopio para poder visualizarlos ya que pueden ser vistos a simple vista. Este proceso además de ser tedioso, necesita de tiempo y de entrenamiento para hacerlo.

Es por ello que se busca desarrollar una solución para analizar las imágenes digitales de los pisos trampa y realizar el conteo de los ácaros de las varroas automáticamente, en base también a descriptores morfológicos como los mencionados anteriormente.

1.3. Objetivos

Esta tesina de grado busca investigar, estudiar y proponer algoritmos, que a partir de las imágenes obtenidas permitan detectar ambas enfermedades con un alto porcentaje de certeza y en un menor tiempo que el actual.

Para el caso de la nosemosis se trabaja en conjunto con el laboratorio de la Fundación Barrera Zoofitosanitaria Patagónica - FunBAPA, laboratorio LR: 0016 (Red Nacional de Laboratorios del SENASA) - y el Dr. Sergio Damian Abate, profesor en la Sede Atlántica de la Universidad Nacional de Río Negro, quien colabora en la obtención de las muestras de abejas de diferentes apiarios dentro de la región atlántica, en las etapas de la captura de las imágenes y la evaluación de los resultados obtenidos. A partir de estas imágenes se intenta detectar el número de esporas de nosemas a través del método automático planteado, para posteriormente realizar una comparación con el diagnóstico realizado en forma manual. En aquellos casos en los que no se pueda detectar ninguna espora de nosema se descarta la imagen aunque se registren dichos casos.

Para el caso de la varroa se trabaja interdisciplinariamente en conjunto con la Lic. Paola Pizzingrilli - Lic. en Biología, profesora de la Sede Andina de la Universidad

Nacional de Río Negro y el alumno Daniel Bazcur. Ambos colaboran en la obtención de las muestras de abejas de diferentes apiarios dentro de la zona andina, en las etapas de la captura de las imágenes y la evaluación de los resultados obtenidos. En el mismo momento que el alumno obtiene las muestras y prueba los pisos trampa captura imágenes digitales que me son enviadas para su procesamiento (ver Figura 1.8 y Figura 1.9). A partir de las mismas se efectúa la detección y conteo automático de las varroas encontradas en el piso que posteriormente son comparadas con los resultados del conteo manual para medir la eficacia de la herramienta.



**Figura 1.8. Enfermedades de las abejas -
Conteo manual**



**Figura 1.9. Enfermedades de las abejas -
Inta columna 1 piso**

Para ambos casos se proponen diferentes etapas para poder elaborar el algoritmo de detección automática de enfermedades mediante la aplicación de técnicas de visión artificial, entre las que se encuentran la adquisición de la imagen, el pre-procesamiento, la segmentación, la búsqueda de contornos, la detección de elipses y, por último, el conteo de estas elipses que en este contexto son las formas que tienen tanto las esporas de nosemas como los ácaros de varroa.

1.4. Organización del trabajo

A continuación, se presenta como se organiza el desarrollo de esta tesina:

- Capítulo 2 - Estado del arte: Aquí se indica el estado de situación de los diferentes ámbitos que aborda esta tesina y el contexto que llevaron a la elaboración de este trabajo. Se presentan los distintos algoritmos existentes para cada uno de los pasos mencionados en el método que se aplica, haciendo una breve descripción de cada uno, evaluando sus ventajas y desventajas.
- Capítulo 3 - Método de trabajo: Se describe la forma como se aplica la solución, el método que se lleva a cabo el presente trabajo y la planificación del mismo.
- Capítulo 4 - Propuesta de solución: En este capítulo se detalla la solución propuesta y las decisiones que se toman en el desarrollo del proyecto.
- Capítulo 5 - Validación de los resultados: En conjunto con el experto del dominio se realiza un análisis de los resultados obtenidos para poder establecer restricciones y condiciones necesarias para un uso eficiente de la aplicación.
- Capítulo 6 - Conclusiones y futuras líneas de investigación: Como su nombre lo indica, en este capítulo se detallan las conclusiones finales del proyecto y las futuras líneas de investigación.
- Capítulo 7 - Bibliografía: En este capítulo, se encuentra la bibliografía utilizada a lo largo del trabajo de tesina.

Capítulo 2: Estado del arte

En este apartado se define lo que es visión artificial y se menciona que se hace en la actualidad para detectar estas dos enfermedades en las abejas melíferas aplicando técnicas de visión artificial.

2.1 Visión artificial

La visión artificial es una rama de la inteligencia artificial que permite simular el sistema visual humano extrayendo información a partir de imágenes digitales, ayudando a automatizar procesos que manualmente son lentos, agotadores y propensos a errores humanos (Gonzalez & Woods, 2008; Szeliski, R. 2011). La misma, se utiliza en distintas áreas como la industria, la medicina, la seguridad, la automotriz, el deporte, los videojuegos, etc.

2.2 Etapas de un sistema de visión artificial

El conteo de algún objeto de interés es un problema recurrente en el área de las ciencias naturales. En general este procedimiento se realiza manualmente y no se utiliza algún software de respaldo, esto se relaciona con la exactitud y la velocidad con la que la persona realiza el conteo, lo que resulta tedioso y cansador al efectuar un gran número de conteos.

Los seres humanos reconocen objetos en imágenes fácilmente (después de tener el entrenamiento adecuado), a pesar del hecho que el objeto dentro de la imagen puede variar según el punto de vista, en tamaño o escala, incluso cuando están trasladados o rotados. Sin embargo, dicha tarea representa un alto grado de complejidad para los sistemas de visión artificial. Por lo tanto, el análisis automático en imágenes plantea varios retos técnicos:

- Iluminación no homogénea.
- Oclusión de objetos.
- Variación en la forma, el tamaño y la orientación de los objetos.
- Variaciones de intensidades.

Debido a la particularidad de cada problema los métodos usados en cada fase difieren considerablemente, pasando por aplicaciones de teoría de conjuntos, filtros de imágenes, métodos de segmentación, métodos de aprendizaje, etc. Se pueden citar cuatro etapas principales que comprende un sistema de visión artificial:

- Adquisición de la imagen: Obtención de la imagen a través de un sensor.
- Procesamiento digital de las imágenes: En dicha etapa se aplican filtros o transformaciones, para eliminar ruidos o partes no deseadas, o por el contrario realzar partes deseadas. Es decir, tiene como objetivo obtener una imagen de mejor calidad a partir de otra, eliminando el ruido o partes no deseadas (González, R. C., Woods, R. E. 2008); para luego extraer información relevante.
- Segmentación: Consiste en reconocer y extraer los objetos de interés. Busca determinar qué regiones de una imagen comparten algún tipo de atributo determinado, dichas regiones serán los objetos de interés en la escena.
- Reconocimiento o clasificación: Esta última etapa está profundamente relacionada con el aprendizaje automático (boosting, redes neuronales, support vector machine, entre otras), aquí se obtienen diferentes clases dependiendo de las características establecidas para diferenciarlas. Es donde se utilizan diferentes técnicas o métodos que determinan de manera automática en qué clase o categoría se encuentra un objeto. Algunas de estas técnicas se basan únicamente en la presencia de características o sus

posiciones relativas, mientras otras implican la segmentación en la imagen en regiones semánticamente significativas.

Cabe destacar que estas etapas no se realizan siempre de forma secuencial, por lo que se puede volver a una etapa anterior, realimentando las mismas hacia atrás. A continuación, se presentan cada una de las etapas para las metodologías desarrolladas utilizadas en el reconocimiento tanto de las esporas de nosemas como de los ácaros de varroas.

2.2.1 Adquisición de la imagen

En la obtención de la imagen intervienen diferentes elementos como el sensor de captura y la iluminación. El sensor de captura es generalmente una cámara y se encarga de tomar la imagen y digitalizarla, transformarla en una matriz de valores que luego interpreta la computadora. El objetivo de la iluminación es brindar la luz necesaria para obtener una imagen de calidad.

2.2.2 Procesamiento digital de las imágenes

Esta etapa consiste en transformar la imagen original en otra en la cual se han eliminado problemas de ruidos. Aquí se utilizan principalmente las siguientes técnicas:

- Eliminación del ruido: El ruido en las imágenes puede ser producto del sensor de captura o el medio transmisión de la señal. Aquí se aplican filtros para eliminar el ruido tratando de conservar las características más importantes de la imagen, como bordes, texturas o intensidad.
- Conversión a escala de grises: En esta etapa se convierte la imagen a color en escala de grises. El valor de los píxeles varían entre negro o blancos, o entre tonalidades de estos para representar la intensidad, brillo o densidad de la imagen.

2.2.3 Segmentación de las imágenes

La segmentación de una imagen, es un proceso por el cual una imagen se subdivide en partes, zonas u objetos disjuntos que la constituyen; es decir, se agrupan los píxeles en función del nivel de intensidad luminosa, el nivel al que se lleva a cabo la subdivisión depende del problema a resolver. En general se busca separar el fondo de la imagen del objeto de interés.

El filtrado es una técnica que facilita la segmentación y sirve para modificar o mejorar la imagen, que puede resaltar o atenuar algunas características.

- **Umbralización:** Es el más sencillo de los filtros implementados, es conocido como umbralización fija (threshold).
- **Erosión:** Provoca en las imágenes el afinado las regiones de píxeles negros. Esto es de utilidad para adelgazar los bordes de los objetos. Dicha función toma como parámetro la imagen a erosionar I y un elemento estructural B , que determina la forma de un entorno de píxeles, del cual el mínimo es elegido y colocado en el lugar del píxel procesado.

La detección de formas elípticas es otra de las técnicas utilizadas en la etapa de segmentación. Existen diversos métodos para realizar dicha tarea, que se pueden dividir en dos: clustering y optimización. En este proyecto se utiliza el algoritmo propuesto en Fitzgibbon que envuelve el objeto de interés con una elipse, devuelve la posición del centro y el ángulo de inclinación de la misma.

2.2.4 Reconocimiento o clasificación de imágenes

El reconocimiento de objetos en imágenes consiste en agrupar aquellos que tienen una representación similar. Este problema pertenece al área de reconocimiento de patrones. En este trabajo los rangos de dimensiones de los parásitos sirven de información adicional para incorporar a la solución del problema de reconocimiento y clasificación.

2.2 Librerías de visión artificial

Antes de realizar la implementación de la aplicación se estudian las diferentes herramientas más utilizadas para esta tarea. Se optó por Java como lenguaje de programación ya que cuenta con las herramientas necesarias para cumplir con el objetivo del proyecto, por lo tanto se ven las herramientas que se integran con este lenguaje. A continuación, se indica un breve resumen de las mismas:

- ImageJ. Es un software libre para el procesamiento de imágenes desarrollado en Java, de dominio público y de código abierto. Está dirigido hacia las áreas de la ciencia de la salud.
- JavaCV. Es un librería Java que permite acceder a las funciones nativas de OpenCV. Ofrece un conjunto de librerías que permite desarrollar aplicaciones de visión artificial, contiene un catálogo de funciones que incorporan cientos de algoritmos de procesamiento y visión por computadora en tiempo real. Es una herramienta libre y de código abierto. Integra otras herramientas para el procesamiento como: FFmpeg, Geometric Calibrator, Image Transformer, Object Finder, CameraDevice y otras que no están por defecto en OpenCV
- OpenCV. Es una librería especializada en los campos de visión artificial, su acrónimo significa “Open Source Computer Vision” que contiene más de 500 algoritmos implementados. Es utilizada con fines académicos y comerciales, se encuentra bajo licencia BSD (Berkeley Software Distribution) y está ampliamente documentada.

2.3 Detección de enfermedades utilizando visión artificial

Existen herramientas de escritorio disponibles para realizar tareas de procesamiento de imágenes para la detección, diagnóstico y clasificación de enfermedades en áreas de la ciencias naturales y la salud como IMAge Pro Plus y CellProfiler.

En lo siguiente se mencionan los proyectos investigados donde se utiliza la visión artificial para realizar tareas de detección de nosemosis o varroasis.

En Alvarez-Ramos, 2010 se propone un método simple y directo de clasificación, medición y reconocimiento de esporas usando visión artificial. En este proyecto se desarrolla un plugin para la plataforma ImageJ que permite automatizar el proceso de clasificación de esporas desde el procesamiento de la imagen hasta la medición de la longitud y la anchura de los parásitos retornando un reporte con la clasificación de las esporas analizadas. Este resultado sirve para ayudar a diagnosticar las enfermedades parasitarias.

En el trabajo de (Álvarez-Ramos, 2013) se presenta un método para identificar y clasificar las esporas aplicando técnicas de visión artificial y de aprendizaje supervisado para detectar enfermedades en las colonias de abejas. En este método se aplican diferentes etapas para mejorar las imágenes y obtener la información relevante, luego se utilizan técnicas de segmentación para identificar las esporas que posteriormente se clasifican por medio de máquinas de vectores de soporte multiclase.

En (López Pérez, 2016) se propone un prototipo de sistema laser de eliminacion de varroas. Mediante dos cámaras montadas en el interior de la colmenas se identifican y eliminan las varroas a través de un rayo láser. Esto permite reducir y eliminar el uso de fármacos y de químicos en el tratamiento de la enfermedad lo que posibilidad una recuperación de la infección de una forma menos invasiva.

Capítulo 3 - Método de trabajo

En este capítulo se presenta el método de trabajo del proyecto, mencionando la posible solución a desarrollar, la metodología de trabajo y la planificación estimada para llevar a cabo el proyecto.

3.1 Posible solución

La automatización en el método de identificación y conteo de esporas de nusemas y ácaros de varroas torna el diagnóstico más rápido y menos trabajoso a la forma como se realiza en la actualidad.

El propósito de este trabajo es poder implementar una solución web apoyándose en el uso de distintos frameworks de desarrollo dentro del entorno empresarial basados en el lenguaje de programación Java (Oracle, n.d.) y utilizando la librería de procesamiento de imágenes OpenCV (OpenCV, 2017), dado que es factible su integración (Baggio, 2015). La arquitectura que se utiliza se basa en los frameworks más populares dentro de la plataforma Java. Su uso en este proyecto facilita diversas tareas de desarrollo, entre otras, la gestión de la base de datos, la implementación de la vista de la aplicación y la autenticación y autorización de los usuarios.

Por un lado, la estructura de la aplicación se define utilizando el gestor de proyectos Maven. Para publicar la aplicación se utiliza el contenedor de servlets Apache Tomcat que se utiliza en la referencia oficial de la implementación para JavaServer y JavaServer Pages.

Por otro lado, la estructura de la arquitectura se divide en capas de presentación, negocio y persistencia. En la capa de presentación de las páginas web se usa JavaServer Faces en conjunto con la librería de componentes Primefaces (Ultimate UI Framework for Java EE). En la capa de negocio se encuentran los servicios que

son implementados con tecnologías Spring. Además, Spring permite la integración de los componentes de las diferentes capas y también la inyección de dependencias. En la capa de persistencia el framework Hibernate es el elegido para gestionar el acceso de la base de datos, MySQL, con la ayuda de phpMyAdmin y MySQL Workbench. Spring Security se encarga de la autorización y autenticación de los usuarios, que es transversal a todas las capas.

3.2 Metodología de trabajo

Para poder lograr los objetivos planteados se llevan adelante las siguientes tareas:

- Análisis bibliográfico: Se consultan los principales repositorios mundiales que proveen artículos relacionados a la temática IEEE Xplore, Scopus, entre otros.
- Creación de la base de datos de imágenes:
 - Para los nosemas: las imágenes a proveer son capturadas por la cámara digital para microscopio marca Mikoba modelo DCM900 (USB2.0) de 9 Megapíxeles, con chip CMOS, montada a un microscopio óptico marca Leica modelo DM500. Las mismas se obtienen con objetivo de microscopio seco de 40X, adquiriendo 400 aumentos totales y se digitalizan con el software propio de la cámara fotográfica.

- Para las varroas: las imágenes a proveer son capturadas por una cámara de un celular LG 50 con una resolución de 5 megapíxeles y un aumento de 10 x 20, es decir, 200 de aumento. Las muestras pertenecen a un apiario del INTA ubicado en el Campo Forestal General San Martín localizado en el paraje Las Golondrinas, Chubut.
- Construcción de la primera versión del software: se utiliza la metodología de desarrollo Scrum. El código fuente es manejado por el sistema de control de versiones a través de la herramienta Git. Como entorno de trabajo se utiliza IntelliJ. Finalmente, se utiliza el plugin de Maven integrado dentro del entorno; el cual administra el ciclo de vida de un proyecto Java y permite integrar fácilmente el resto de los frameworks. Como base de datos se utiliza MySQL la cual es Open Source y Apache Tomcat que aporta como ventajas la facilidad de uso y el soporte de las características estándar en el contexto de aplicaciones web, por ejemplo: el fichero Deploy War.
- Validación del software a través de pruebas y comparación de resultados entre el conteo manual y el conteo automático para un conjunto de imágenes determinado.
- Publicación de los resultados.

3.3 Plan de trabajo

Se utiliza una metodología de trabajo que incluye las siguientes actividades:

1. Estudiar el estado de la investigación – realizar la revisión de bibliografía de trabajos publicados en congresos nacionales e internacionales y revistas de prestigio internacional a fin de estudiar metodologías, marcos de trabajo y soluciones con las tecnológicas adoptadas para la detección de varroas y nosemas.
2. Estudiar el estado de la práctica – conducir un relevamiento de casos prácticos de desarrollo de aplicaciones sobre visión artificial para la detección automática de enfermedades en abejas melíferas.

3. Identificar el gap entre investigación y práctica – analizar los desarrollos teóricos que no se aplican en la práctica y/o los desarrollos prácticos que no cuentan con sustento teórico que sean relevantes para el desarrollo de aplicaciones sobre visión por computador en la detección automática de enfermedades en abejas melíferas, a fin de determinar un espacio de contribución científica y definir requerimientos específicos para las soluciones tecnológicas a proponer.
4. Especificar la solución – en base al gap identificado en el paso anterior, definir requerimientos funcionales y no funcionales para las soluciones tecnológicas a desarrollar. Para esta etapa, se utilizan metodologías ágiles de especificación de requerimientos y desarrollo de software. También se definen los criterios de aceptación de los productos a elaborar.
5. Implementar la solución – en base a los requerimientos especificados, seleccionar las tecnologías más adecuadas a utilizar para el desarrollo de la solución que mejor resuelva las necesidades de los distintos actores o roles existentes. En esta tarea, la solución se implementa bajo el paradigma orientado a objetos, con las herramientas de software ya mencionadas.
6. Prueba piloto de la solución – la solución desarrollada se valida en pruebas por los expertos del dominio. En el caso de la nosemosis en conjunto con el laboratorio FunBAPA y Dr. Sergio Damián Abate; en el caso de la varroasis la Lic. Paola Pizzingrilli y el alumno Daniel Bazcur.
7. Diseminación de resultados – publicar en congresos nacionales e internacionales y en revistas con referato los resultados de investigación producidos durante el proyecto.

La propuesta del plan de trabajo se puede ver de forma resumida en la siguiente tabla: En la Tabla 3.1 dentro del apartado de actividades se nombran las siete actividades mencionadas anteriormente. El apartado de meses se divide en los doce

meses en los que se planificó el proyecto, indicando en qué meses se realiza cada actividad.

Cronograma de actividades (Consignar sucesivamente cada actividad unitaria):												
Actividad	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
Estudiar el estado de investigación	X	X										
Estudiar el estado de practica		X	X									
Identificar el gap entre investigación y práctica			X	X								
Especificar la solución				X	X	X						
Implementar la solución						X	X	X				
Prueba Piloto de la Solución y Validación de resultados								X	X	X		
Diseminación de Resultados										X	X	X

Tabla 3.1. Cronograma de actividades

Capítulo 4: Propuesta de solución

Aquí se cuenta la solución propuesta y la metodología aplicada para la implementación de dicha solución. Se detalla la arquitectura y diseños utilizados, y las tecnologías con las que se llevó a cabo el proyecto. También, se explican las distintas etapas de los algoritmos de identificación de enfermedades en abejas melíferas, y el proceso seguido para la marcación y discriminación de casos al utilizar esta herramienta.

4.1 Metodología de desarrollo a utilizar

Para la gestión del proyecto se utiliza el marco de trabajo llamado Scrum (Schwaber, K., & Sutherland, J., 2016), que permite trabajar de manera iterativa-incremental para optimizar la predictibilidad y el control de riesgos. Scrum es útil por su facilidad de aplicación en equipos pequeños, flexibilidad a los cambios, colaboración con el cliente y orientación a los resultados. Permite lograr desarrollos en cortos períodos de tiempo y además posibilita comenzar a trabajar sin tener los requerimientos definidos completamente desde un principio sino que define un conjunto inicial de funcionalidades que son priorizadas según los objetivos de negocio. Dicho marco no se aplica rigurosamente sino que se adapta a las necesidades del equipo y del proyecto.

Scrum utiliza el concepto de “Sprint”, un período de entre cuatro y seis semanas, dentro del cual se implementan un conjunto de requerimientos definidos según la necesidad del usuario, también llamado Incremento. Un Sprint permite entregas frecuentes y continuas al cliente de forma que pueda disponer de una funcionalidad básica en un tiempo mínimo con la mejora continua del sistema.

En la metodología mencionada se tienen los siguientes roles:

- **El Dueño del Producto (Product Owner)**, representa al cliente, aunque también puede ser un miembro del equipo con suficiente conocimiento del negocio.
- **Scrum Master**, es el encargado de que se adopte y entienda Scrum, además se encarga de verificar que el equipo funcione correctamente, así como de organizar reuniones al final de los Sprints.
- **El Equipo de Desarrollo (Development Team)**, se encarga tanto del análisis y diseño como del desarrollo, pruebas y documentación, realiza el trabajo de entregar un Incremento del producto.

Los artefactos de Scrum se refieren a los elementos de trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos que se utilizan en Scrum son:

- **Lista de Producto (Product Backlog)**, es una lista que contiene todos los requerimientos y funcionalidades deseables, ordenados según la prioridad del negocio. El mismo es propenso a modificaciones en el transcurso del proyecto. Es el Dueño del Producto (Product Owner) el encargado de su contenido, disponibilidad y ordenación.
- **Lista de pendientes del Sprint (Sprint Backlog)**, en el cual se detalla la actividad del siguiente Sprint creado a partir de elementos elegidos del Product Backlog, más un plan para entregar el Incremento del producto y alcanzar el Objetivo del Sprint. En esta lista se refleja el trabajo que el Equipo de Desarrollo cree necesario, sin asignar las tareas a personas específicas.
- **Incremento**, es el resultado del Sprint. Un incremento es la suma de todas las tareas, historias de usuarios y cualquier otro elemento desarrollado durante el Sprint y se entrega al Dueño del Producto en forma de software.

Los eventos que se llevan a cabo en Scrum son las siguientes:

a) Planificación de la iteración (Sprint Planning)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

- Selección de requisitos (4 horas máximo). El cliente presenta el proyecto y los requisitos. El equipo pregunta al cliente las dudas que surgen y selecciona junto al cliente cuales son los requisitos más prioritarios.
- Planificación de la iteración (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a los que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas. El conjunto de tareas de una iteración incluyendo el recurso asignado se denomina Sprint Backlog.

b) Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización de 15 minutos máximo. Aquí cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la ejecución de la iteración el Scrum Master tiene un rol importante ya que ayuda a eliminar obstáculos que el equipo no puede resolver por sí mismo, y lo protege al equipo de interrupciones externas. Se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

c) Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración en conjunto con el equipo. Tiene dos partes:

- Demostración (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, el equipo por su parte se encarga de atender sugerencias, redefinir requerimientos y/o prioridades y re planificar el proyecto.
- Retrospectiva (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Scrum Master se encargará de ir eliminando los obstáculos identificados.

Un punto a destacar es que este marco adopta y supone un compromiso de parte del cliente: debe estar continuamente involucrado para garantizar que se esté construyendo software de valor y se esté priorizando el trabajo. En resumen esto se puede ver gráficamente en la Figura 4.1.

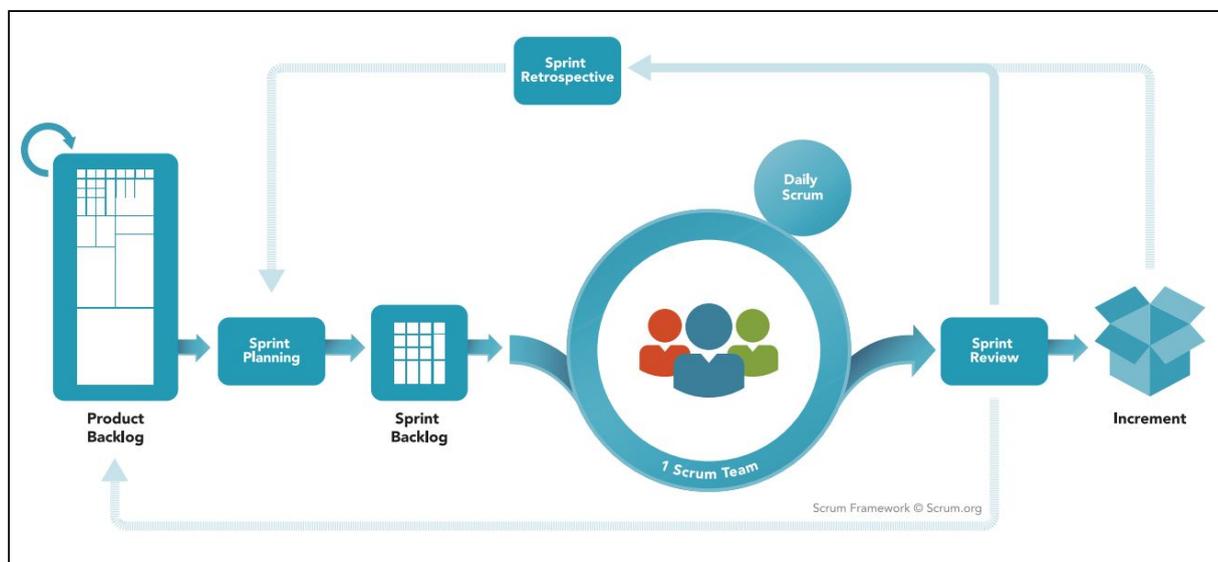


Figura 4.1. Resumen del proceso Scrum (Scrum.org, 2016)

Historias de usuario y criterios de aceptación

La historia de usuario se utiliza para describir funcionalidad que es útil para el usuario. Estas se dividen en tareas por parte del equipo durante la reunión de Planificación del Sprint. En paralelo otras historias pueden ser modificadas en su enunciado, objetivo o criterios de aceptación, re priorizadas u ordenadas según el criterio del cliente. También pueden ser quitadas del Product Backlog si se modifica el alcance o las tareas a desarrollar.

Básicamente una historia se divide en dos apartados, el enunciado y los criterios de aceptación:

- El enunciado se escribe en un lenguaje coloquial según el dominio del cliente y a partir de una conversación con el mismo, teniendo en cuenta que sirve de acuerdo formal para desarrollar la funcionalidad esperada.
- Los criterios de aceptación se encuentran en la propia historia de usuario y describen las pruebas que deben superar para ser aceptada como completa; dicha aceptación es binaria: se acepta o se rechaza, no hay puntos medios.

El formato de escritura de una historia a partir del enunciado es: *Como _____ Quiero _____ Para _____ y seguir por los criterios de aceptación de forma concreta.* En la Tabla 4.1 se puede ver un ejemplo de una historia de usuario.

Deben cumplir las siguientes características:

- Independientes: Atómicas en su definición, que no dependa de otras historias.
- Negociables: Ambiguas en su enunciado y concretas en sus criterios de aceptación.
- Valoradas: Valoradas por el cliente para saber cuánto valor aporta a la aplicación y junto con la estimación convertirse en un criterio de prioridad.
- Estimables: Tener su alcance definido como para poder suponer una medida de trabajo en la que pueda ser completada.
- Pequeñas: Mayores de dos días y menores de dos semanas.

- Verificables: Se verifican que los criterios de aceptación han sido cumplidas con las funcionalidades descritas y esperadas.

Enunciado	Criterios de aceptación
<ul style="list-style-type: none"> • Como usuario registrado. • Quiero dar de alta un apiario y asociarle un conjunto de imágenes. • Para ver el número de esporas de noseemas. 	<ul style="list-style-type: none"> • Quiero dar de alta un apiario completando el formulario de apiarios. • Quiero asociar imágenes al apiario. • Quiero saber el número de esporas presentes en cada una de las imágenes del conjunto. • Quiero registrar toda la información asociada.

Tabla 4.1. Ejemplo de historia de usuario

4.2 Tecnologías a utilizar

4.2.1 Arquitectura de la aplicación

Para el desarrollo de la Aplicación Web se utiliza una arquitectura web en capas basada en los principales frameworks dentro de la plataforma Java y utilizando la librería de procesamiento de imágenes OpenCV, dado que es factible su integración (Baggio, 2015). Esta arquitectura fue definida por el Laboratorio de Informática Aplicada de la Universidad Nacional de Río Negro (Vivas, 2013) del que forma parte el co-director de la tesina, el Ing. Cambarieri Mauro German y el tesista. Por un lado, está comprobada la utilidad, capacidad de escalabilidad y robustez de dicha arquitectura ya que se usó en otros proyectos exitosamente como son el Voto Nominal Electrónico - LEGISVOTO para la Legislatura de Río Negro y el Sistema de Administración de Proyectos de Información de Investigadores y Proyectos de Investigación - SAIPI para la Universidad de Río Negro. Por otro lado, facilita las diversas tareas de desarrollo como la gestión de la base de datos, el desarrollo de la lógica del negocio, el desarrollo de las vistas de la aplicación, y la autenticación y autorización de los usuarios.

4.2.1.1 Arquitectura en capas

Se manejan dos conceptos a nivel de arquitectura:

- Capas lógicas. Son capas de software que se ocupan de la distribución lógica de los componentes y la funcionalidad.
- Capas físicas. Se ocupan de la distribución de componentes y funcionalidad en distintos servidores.

A su vez, las capas lógicas se despliegan sobre las capas físicas como se puede observar en la siguiente imagen (ver Figura 4.2).

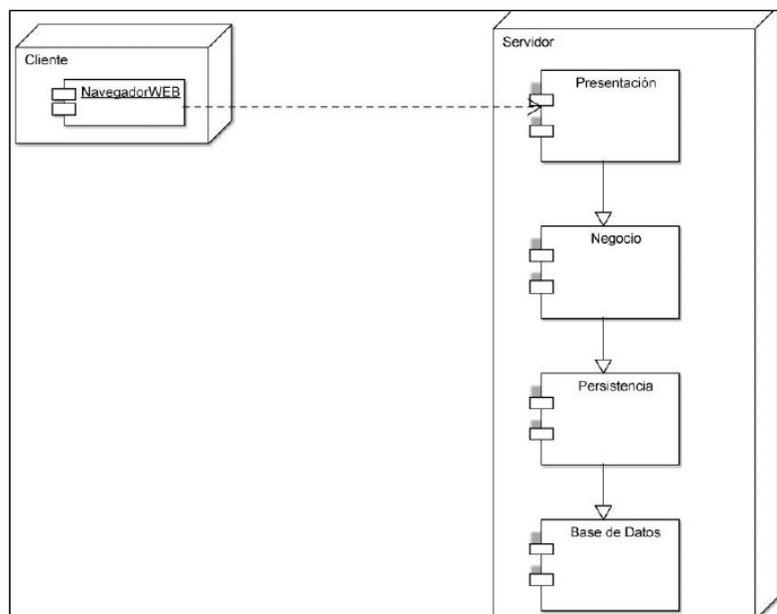


Figura 4.2. Modelo de 3 capas lógicas y 2 capas físicas

A continuación, se explican con mayor detalle las capas lógicas que componen la arquitectura:

1. Capa de presentación: Se apoya sobre una capa de servicios bien definida. Esto significa que la capa de presentación deberá ser fina y no deberá contener lógica de negocio, sino el código que sea específico de la presentación como aquel para manejar las interacciones con la web. En esta capa se presentan las páginas web, implementadas con JavaServer Faces en conjunto con la librería de componentes Primefaces (Ultimate UI Framework for Java EE).

2. Capa de negocio:

Esta capa provee las siguientes funcionalidades:

- a. Lógica de negocio que es específica de los requerimientos funcionales: mientras que los objetos de dominio contienen lógica de negocio, esta capa implementa lo que es específico de una Historia de Usuario.
- b. Puntos de entrada claramente definidos para las operaciones de negocio. La capa de servicios provee las interfaces usadas por la capa de presentación. Esta capa debe exponer interfaces no implementaciones.
- c. Administración de transacciones.
- d. Verificación de restricciones de seguridad.

En esta capa se usa Spring Framework para la inyección de dependencias y la integración de los distintos frameworks.

3. Objetos de dominios persistentes. Es el núcleo del modelo del dominio. Está formada por verdaderos objetos, que contienen comportamiento y encapsulan su estado. Es transversal a todas las capas.
4. Capa de persistencia: En esta capa se utiliza el framework Hibernate (Hibernate ORM) para el mapeo Objeto/Relacional diseñada con el patrón Data Access Objects (Objetos de Acceso a Datos) también llamado DAO's, encapsulan el acceso a los objetos persistentes del dominio, persisten los objetos transitorios y actualizan los objetos existentes. Se encarga de gestionar el acceso a la base de datos, que en este caso es una base MySQL.
5. Seguridad de aplicación: Se usa Spring Security para la autorización y autenticación de los usuarios, que es transversal a todas las capas.

En la Tabla 4.2 se muestran las distintas capas y las tecnologías utilizadas en cada una de ellas:

Capa de presentación	<ul style="list-style-type: none"> • Primefaces • JSF 2.0 	Spring Security
Capa de negocio	<ul style="list-style-type: none"> • Spring Framework • Opencv 	
Capa de persistencia	<ul style="list-style-type: none"> • Hibernate • MySQL 	

Tabla 4.2: Tecnologías utilizada en cada una de las capas

En la Figura 4.3 se puede ver una representación gráfica en forma resumida la estructura de la arquitectura.

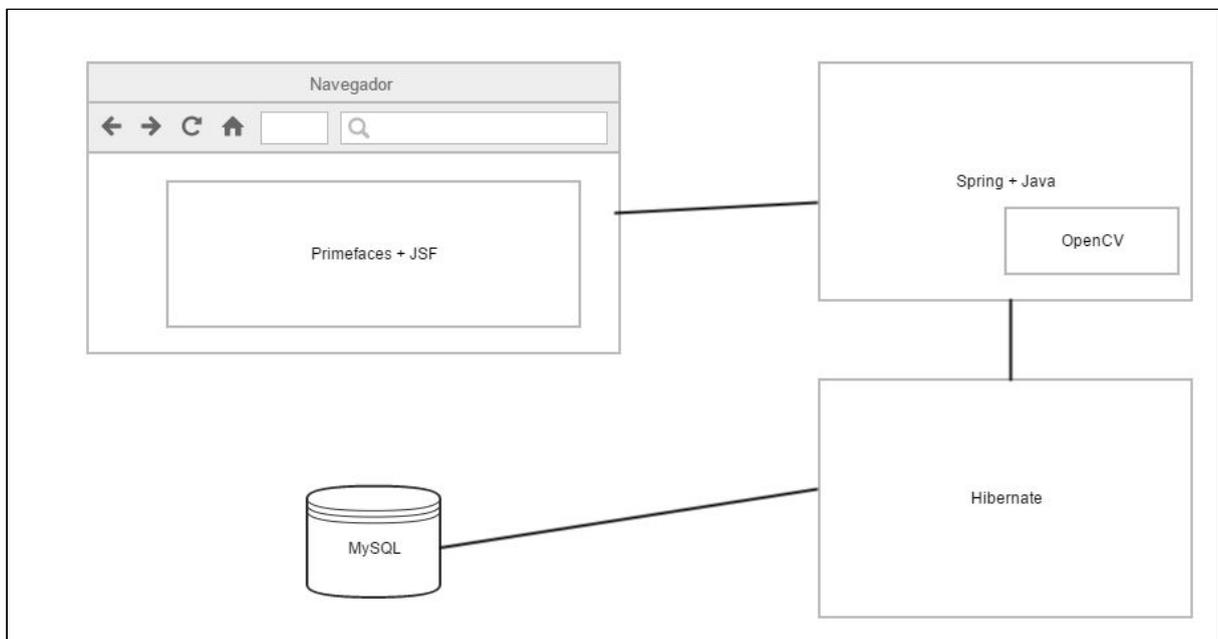


Figura 4.3. Estructura de la arquitectura

Todas las herramientas mencionadas se encuentran bajo licencia Open Source. La estructura de la aplicación se define utilizando el gestor de proyectos Maven, que se puede usar tanto dentro del sistema operativo como dentro del entorno de desarrollo integrado utilizado, que en este caso es IntelliJ. La publicación de la aplicación se realizará en el contenedor de servlets Apache Tomcat, que se encuentra bajo la licencia Apache License Versión 2. Un contenedor de servlets es una aplicación

capaz de recibir peticiones de páginas web y redireccionar estas peticiones a un servlet, que implementa las especificaciones de los servlets y de las JavaServer Faces(JSF).

JavaServer Faces (JSF) es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas.

También, se debe mencionar que el equipo cuenta con experiencia en la utilización de estas tecnologías lo que permite acelerar aún más los tiempos de desarrollo. De hecho parte del equipo participó en el proyecto de LEGISVOTO y en el desarrollo que llevó a cabo la implementación de SAUPI.

4.2.2 Librería de visión artificial a utilizar

Para el desarrollo de los algoritmos y el procesamiento de las imágenes se utiliza la librería de visión artificial OpenCV (Open Computer Vision) que se integra a la arquitectura elegida. Esta librería de código abierto fue desarrollada originalmente por los laboratorios Intel, y contiene más de 2500 algoritmos optimizados, desde los algoritmos clásicos hasta los más recientes en visión artificial que solo necesitan ser invocados sin la necesidad de implementarlos. Es adoptada en todo el mundo por grandes empresas (Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota) y centros de investigación, posee una activa comunidad de más de 47 mil usuarios, lanzada bajo licencia BSD y es libre para uso académico y comercial.

Se utiliza esta librería dado que su contenido representa gran conocimiento y experiencia adquirida garantizando una implementación eficiente. También proporciona una comunidad de desarrollo que puede asistir en la solución de inconvenientes y cuenta con completa documentación.

4.2.3 Lenguaje de programación a utilizar

El principal lenguaje de programación utilizado es Java. Por un lado la arquitectura escogida utiliza dicho lenguaje y por otro se tiene experiencia en la utilización de este lenguaje, tanto en el ámbito académico como laboral. Además, Java es una de los lenguajes de programación más populares, que cuenta con una gran comunidad y documentación.

Entre las principales características de Java se puede mencionar que es orientado a objetos, multiplataforma y de código abierto.

4.2.4 Otras tecnologías

Se utilizan otras tecnologías complementarias al desarrollo como son el repositorio de código, el sistema de control de versiones y se genera la documentación asociada al proyecto.

Para gestionar el proyecto se utiliza GitLab ya que provee funciones de gestión de proyectos con metodologías ágiles e integración con diferentes herramientas de desarrollo. A partir de la versión 8.11 incorporó un tablero llamado "Issue Boards" donde se pueden gestionar las tareas, cada tarea se puede priorizar según su posición en la lista, se puede asignar una fecha de finalización, un responsable y el Sprint al que pertenece. Dentro del Board se puede generar una lista que representa el Product Backlog; los Sprints pueden configurarse como Milestones que permiten organizar y agrupar tareas, configurando opcionalmente una fecha inicio y una fecha de vencimiento. En cada Milestone se puede ver el porcentaje de avance a medida que las tareas son concluidas. Además, esta plataforma ofrece un sistema de repositorio y sistema de control de versiones Git de código abierto y basado en la nube.

Otra de las ventajas de GitLab es que ayuda a configurar el entorno de manera sencilla. Se basa en buenas prácticas de gestión de la configuración de software ya

que se apoya en las técnicas de integración y desarrollo continuos. Esto permite llevar a cabo el proyecto en el menor tiempo posible orientado a los resultados.

Por lo tanto, participa en todo el ciclo de vida de desarrollo de software, lo que permite eliminar pasos innecesarios en el flujo de trabajo y se centra exclusivamente en la construcción de software de gran calidad.

4.2.5 Flujo de trabajo

Una buena práctica cuando se trabaja con Git es utilizar GitFlow, este es un flujo de trabajo avanzado de Git que establece la forma de gestionar las ramas. GitFlow fue definido por Vincent Driessen en el año 2010 ("A successful Git branching model," n.d.) con las siguientes ramas (ver Figura 4.4):

- Rama *main* hospeda la última versión estable del proyecto, es la rama principal o de producción.
- Rama *develop* contiene los últimos desarrollos realizados para el próximo lanzamiento; utilizada para integrar las diferentes *features*.
- Por cada característica a implementar se crea un *feature branch*, a partir de *develop*.
- La preparación del siguiente lanzamiento se realiza en una *release branch*. Utilizadas para integración y configuración de releases previas a su despliegue en producción. Se crean a partir de *develop* y se unen tanto a *master* como a *develop*.
- Un fallo o bug en producción se soluciona en una *hotfix branch*, se crean a partir de la rama *master*. Luego se realiza un merge con *master* y *develop* para aplicar el parche tanto en producción como en desarrollo.

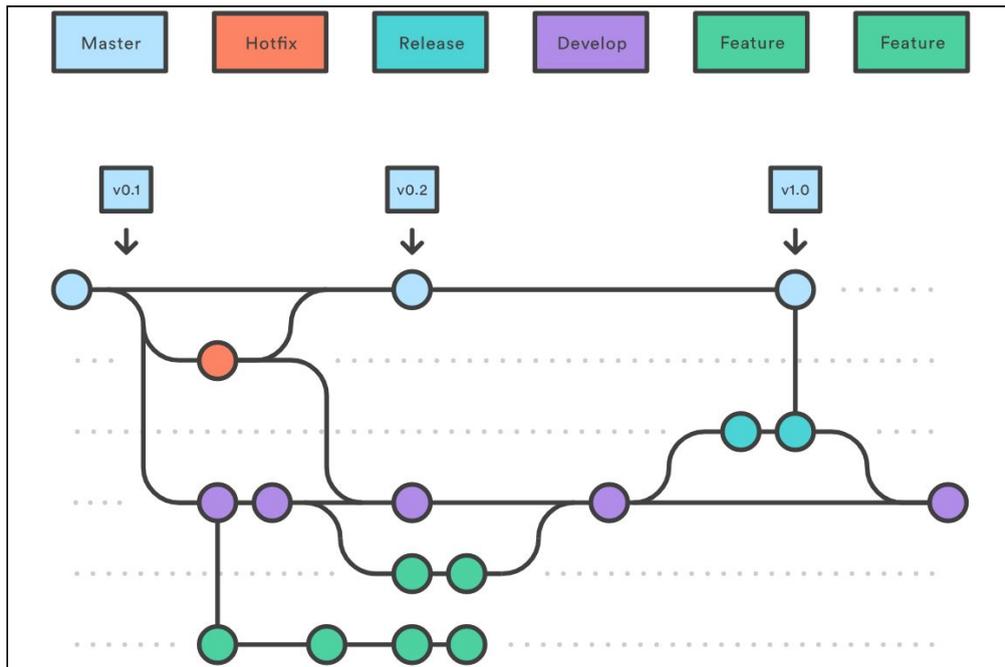


Figura 4.4 - Gitflow (Atlassian, n.d.)

4.2.6 Técnicas de programación

Al momento de escribir el código se sigue la filosofía de código limpio propuesta originalmente por Robert C. Martin también conocido como Unble Bob, que reúne un conjunto de ideas que intentan hacer el código fácil de leer, mantener, extender y menos propenso a errores, los doce principios son (Martin, 2008):

1. Usar nombres significativos.
2. Realizar unidades de código pequeñas.
3. Realizar unidades que hacen una única cosa.
4. Los métodos deben tener un número limitado de argumentos.
5. Aplicar el principio DRY: Don't Repeat Yourself.
6. Evitar usar comentarios siempre que sea posible.
7. Utilizar un formato único en el código.
8. Abstractar datos: no usar getters y setters de forma indiscriminada.
9. Ley de Demeter.
10. Lanzar excepciones en vez de códigos de retorno.
11. Establecer fronteras.
12. Escribir tests.

4.3 Planificación y ejecución del proyecto

De acuerdo con el análisis de los procesos y las funcionalidades a desarrollarse en el Sistema de “Detección Automática de Enfermedades en Abejas Melíferas”, se tienen los siguientes alcances:

- **Administrador de Aplicación Web:**
 - Permite crear, eliminar, modificar usuarios y ver la información de cada usuario.
- **Módulo de identificación y conteo de esporas de nosemas.**
 - Permite crear un apiario, con su información correspondiente y asociarle un conjunto de imágenes. Como resultado retorna el número de esporas de nosemas presentes en cada imagen.
- **Módulo de identificación y conteo de varroas**
 - Al igual que el módulo anterior permite crear un apiario y asociarle un número de imágenes. Como resultado retorna el número de ácaros de varroa presentes en cada imagen.

Administrador de la Aplicación Web

- **Usuarios**
 - Son las personas que utilizan el sistema para la detección de enfermedades en las abejas, ya sea de nosemosis o varroasis. El usuario puede ingresar al sistema a través de su cuenta de gmail y puede consultar los análisis realizados por el.

Módulo de identificación y conteo de esporas de nosemas

- **Creación de un apiario para un conjunto de imágenes:**
 - Permite crear un apiario y asignarle un nombre, además de otra información complementaria como la fecha, la ubicación y el número de colmenas.

- Agregar imágenes a un apiario:
 - Se pueden cargar imágenes a un apiario determinado. A cada imagen se le asigna un identificador automáticamente.
- Borrar, editar y ver un apiario y sus imágenes:
 - Es posible borrar, editar y ver la información de un apiario, así como las imágenes asociadas.
- Búsqueda de apiarios e imágenes:
 - Se pueden buscar los distintos apiarios creados según los criterios que se establezcan y dentro de cada uno buscar las distintas imágenes asociadas.
- Número de esporas por imagen y apiarios:
 - Se pueden visualizar los resultados del análisis para un apiario determinado o por cada imagen en particular. Este resultado es el número de esporas presentes en la imagen que luego se utiliza para definir si la enfermedad está presente.

Módulo de identificación y conteo de varroas

- Creación de un apiario para un conjunto de imágenes:
 - Permite crear un apiario y asignarle un nombre, además de otra información complementaria como la fecha, la ubicación y el número de colmenas.
- Agregar imágenes a un apiario:
 - Se pueden cargar imágenes a un apiario determinado. A cada imagen se le asigna un identificador automáticamente.
- Borrar, editar y ver un apiario y sus imágenes:
 - Es posible borrar, editar y ver la información de un apiario, así como las imágenes asociadas.
- Búsqueda de apiarios e imágenes:
 - Se pueden buscar los distintos apiarios creados según los criterios que se establezcan y dentro de cada uno buscar las distintas imágenes asociadas.

- Número de ácaros por imagen y apiarios:
 - Se pueden visualizar los resultados del análisis para un apiario determinado o por cada imagen en particular. Este resultado es el número de varroas presentes en la imagen que luego se utiliza para definir si la enfermedad está presente.

Equipo de trabajo

El equipo de trabajo considerado para llevar a cabo el desarrollo del Proyecto es el siguiente (ver Tabla 4.3).

Rol	Persona
Dueño del Producto (Product Owner)	En el caso de la nosemosis se considera al Dr. Sergio Damian Abate, profesor de la Sede Atlántica de Universidad, como principal fuente de conocimientos. En el caso de la varroasis es la Lic. Paola Pizzingrilli quien cumple este rol.
Scrum Master	La persona a la que le corresponde este rol es la Dra. Paola Britos, quien velará por el buen desenvolvimiento y trabajo del equipo.
Team	Este rol es asumido por el tesista Patricio Nicolas Castro y el co-director, el Ing. Mauro German Cambarieri.

Tabla 4.3. Definición del equipo de trabajo

Durante la etapa de planificación (Sprint 0) se conforma el equipo así y se definen los roles de cada miembro en la planificación de cada Sprint del proyecto.

Planificación del Sprint

Como resultado del análisis y planificación realizados en el Sprint 0, se planificó el desarrollo de la aplicación en 4 Sprints, cada Sprint tiene el objetivo de implementar las tareas como un conjunto. A continuación se presentan los Sprint definidos:

- Sprint Nro. 0: Definición de la Arquitectura.
- Sprint Nro. 1: Administrador de la Aplicación Web.

- Sprint Nro. 2: Módulo de identificación y conteo de esporas de nosemas.
- Sprint Nro. 3: Módulo de identificación y conteo de ácaros de varroa.

Las funcionalidades a implementar en cada Sprint fueron descritas utilizando historias de usuario priorizadas según la importancia que estas tenían para el Dueño del Producto.

Definición de la Lista de Producto (Product Backlog)

El Product Backlog contiene toda la funcionalidad que el producto final debería tener. Tal como lo dice la metodología, para el presente proyecto se ha elaborado el Backlog, identificando las funcionalidades, priorizando cada una de ellas y realizando una estimación del tiempo requerido para su implementación.

El Backlog para el presente proyecto se encuentra definido en la siguiente tabla (ver Tabla 4.4). Él mismo se creó dentro de la herramienta GitLab (ver Figura 4.5). Utilizando GitLab, cada Sprint se configuró como un Milestone (ver Figura 4.6), donde se asociaron todas las tareas correspondientes.

Nro. Sprint	Tarea	Tiempo/semanas
0	Definición de la Arquitectura	2 semanas
1	Administrador de la Aplicación Web	4 semanas
2	Módulo de identificación y conteo de esporas de nosemas	6 semanas
3	Módulo de identificación y conteo de varroas	6 semanas

Tabla 4.4. Definición del Product Backlog

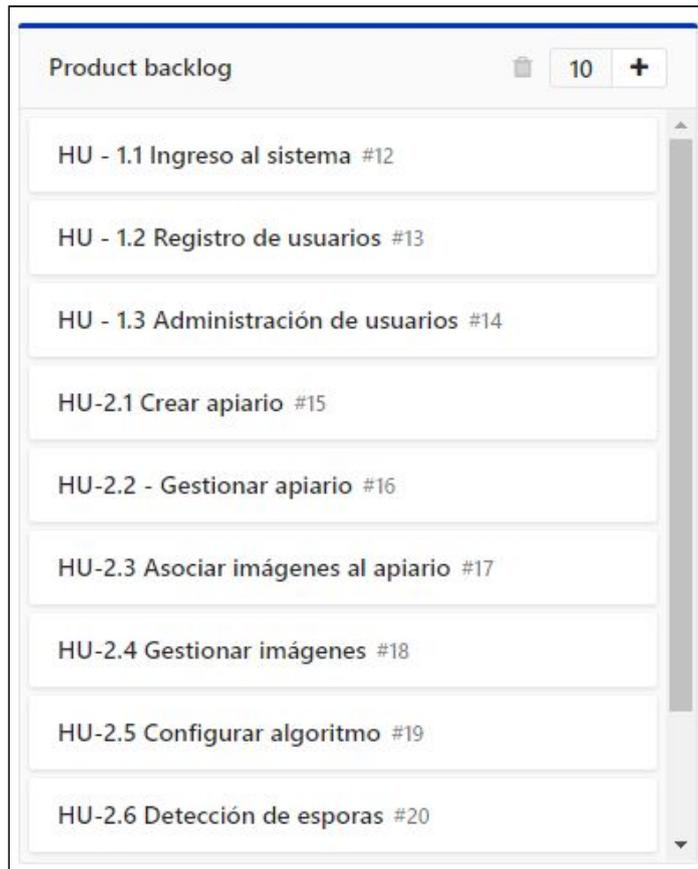


Figura 4.5. Configuración de la Lista de Producto en GitLab Board

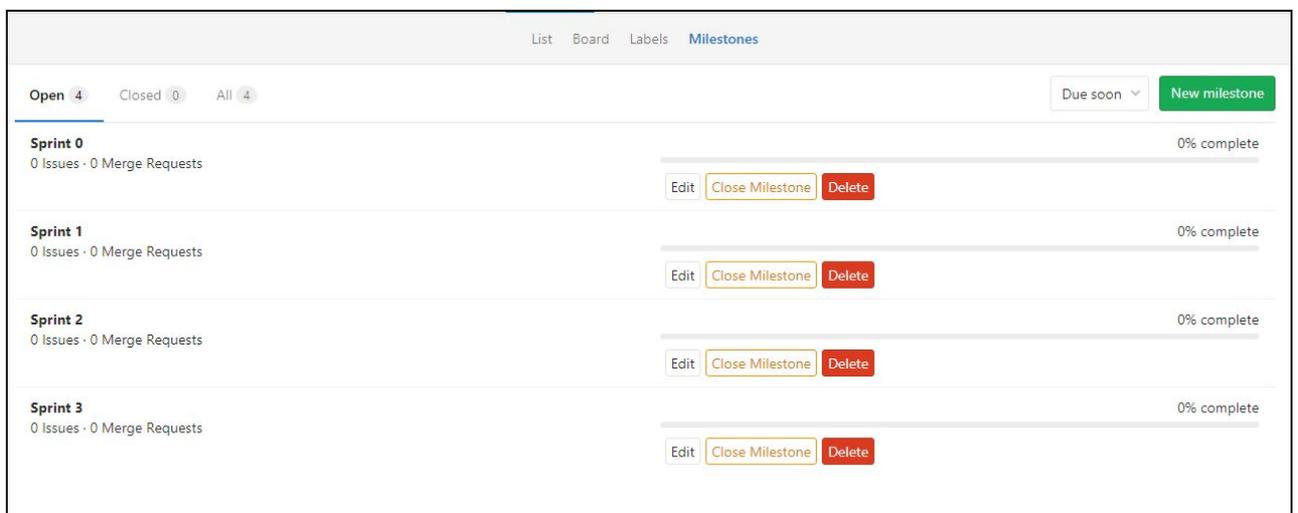


Figura 4.6. Listado de Milestones en Gitlab

4.3.1 Sprint Nro. 0, instalación del entorno de desarrollo, repositorio y servidor de la aplicación

Luego de haber realizado un análisis funcional y realizar un diseño base del modelo a utilizar se comienza con el Sprint 0. Como se mencionó la configuración de este se realiza dentro de GitLab y se establecen el conjunto de tareas pertenecientes al Sprint, que se puede ver a continuación (ver Figura 4.7 y Figura 4.8):

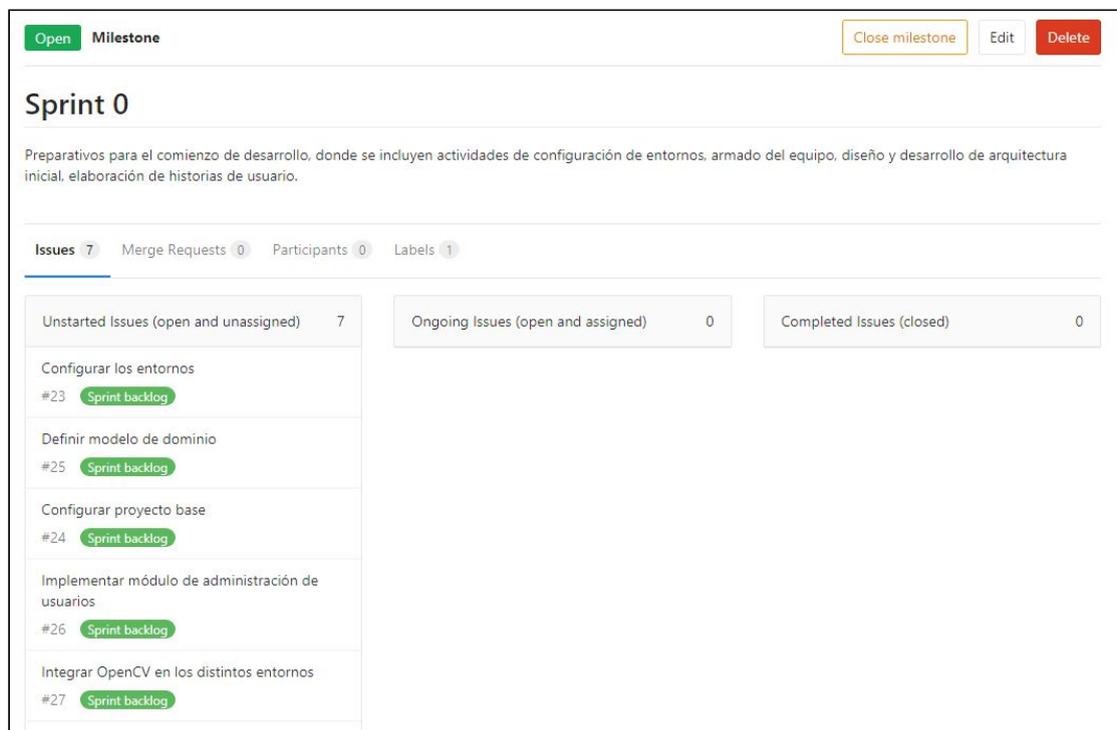


Figura 4.7. Configuración del Sprint Nro. 0 en GitLab

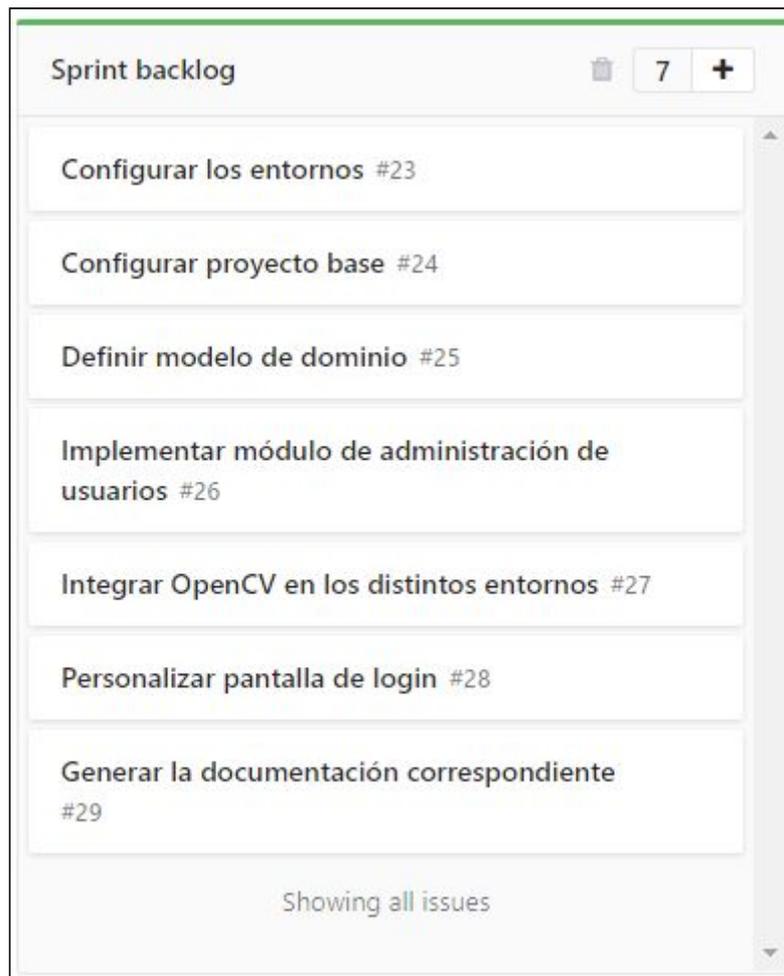


Figura 4.8. Sprint Backlog del Sprint Nro. 0

Los Sprints posteriores se elaboran cada uno en mayor detalle describiendo las historias de usuarios. Como se eligió una metodología ágil se estiman e incorporan tareas según se avance en el desarrollo.

Lo ideal para llevar a cabo el proyecto es tener varios entornos de desarrollo (Lugani Cátedra de “Sistemas y organizaciones,” 2015). Esto permite tener configurado distintos servidores donde probar la aplicación previamente a la salida a producción. En nuestro caso definiremos los siguientes entornos:

- **Desarrollo:** Es el entorno local donde cada desarrollador tiene el proyecto ejecutando en su máquina.
- **Preproducción:** Este entorno tiene características similares al de producción y se utiliza por el equipo para probar los cambios realizados.
- **Producción:** Es el entorno real donde se ejecuta la aplicación.

Existen herramientas que permiten publicar la aplicación en los diferentes entornos de forma automática. De esta forma, esta tarea se realiza fácilmente. Por lo tanto, vale la pena invertir horas de trabajo para automatizar este proceso ya que aporta un gran valor en el rendimiento del equipo. A continuación se detalla la configuración de cada uno de los entornos describiendo cada parte:

Entorno de desarrollo

- Entorno de Desarrollo Integrado: IntelliJ IDEA 2016.3.4. Build #IU-163.12024.16, built on January 31, 2017. Licensed to Patricio Nicolas Castro. Subscription is active until February 17, 2018. For educational use only.
- JRE: 1.8.0_111-b14 amd64.
- JVM: Java HotSpot(TM) 64-Bit Server VM by Oracle Corporation.
- Administrador de MySQL: phpMyAdmin
- Servidor: Apache Tomcat 7.0.69
- Esquema de base de datos: daeam_dev
- Repositorio de código en GitLab:
<http://pcastro@git.lia.unrn.edu.ar/pcastro/daeam.git>
- Configuración de OpenCV desde el entorno (ver Figura 4.9)

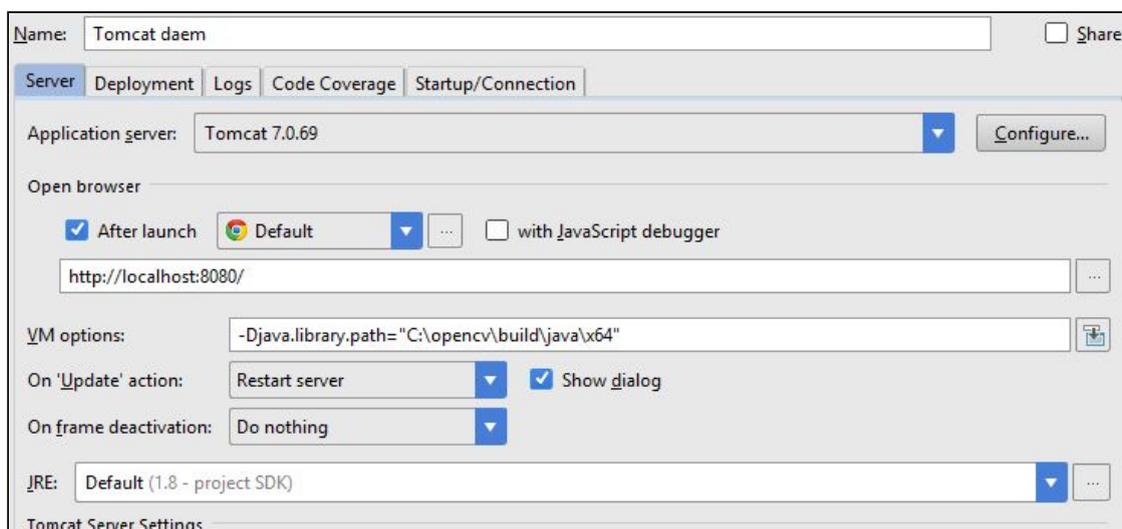


Figura 4.9. Configuración de OpenCV en entorno de desarrollo

Entorno de pre-producción

- Se encuentra en: <http://dev.lia.unrn.edu.ar/>
- Sistema operativo: Ubuntu 16.04.2 LTS x86_64
- Servidor: Apache Tomcat 7.0.69, <http://dev.lia.unrn.edu.ar:8080/manager/html>
- Administrador de base de datos: phpMyAdmin - <http://dev.lia.unrn.edu.ar/phpmyadmin/>
- Esquema de base de datos: daeam_dev
- Configuración de OpenCV del lado del servidor: Se realiza igual que en el entorno de producción (Ver Entorno de producción)

Entorno de producción

- Se encuentra en: <http://tesis.lia.unrn.edu.ar/>
- Sistema operativo: Ubuntu 16.04.2 LTS x86_64
- Servidor: Apache Tomcat 7.0.69 - <http://tesis.lia.unrn.edu.ar:8080/manager/html>
- Administrador de base de datos: phpMyAdmin - <http://tesis.lia.unrn.edu.ar/phpmyadmin/>
- Esquema de base de datos: daeam
- Configuración de OpenCV del lado del servidor:

```
sudo apt-get -y install libopencv-dev build-essential cmake git  
libgtk2.0-dev pkg-config python-dev python-numpy libdc1394-22  
libdc1394-22-dev libjpeg-dev libpng12-dev libjasper-dev  
libavcodec-dev libavformat-dev libswscale-dev libgstreamer0.10-dev  
libgstreamer-plugins-base0.10-dev libv4l-dev libtbb-dev libqt4-dev  
libfaac-dev libmp3lame-dev libopencore-amrnb-dev  
libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev  
x264 v4l-utils unzip
```

```
mkdir opencv
```

```
cd opencv
```

```
wget https://github.com/Itseez/opencv/archive/3.0.0.zip -O
opencv-3.0.0.zip

unzip opencv-3.0.0.zip

cd opencv-3.0.0

mkdir build

cd build

cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local
-D WITH_TBB=ON -D WITH_V4L=ON -D WITH_QT=ON -D WITH_OPENGL=ON ..

make -j $(nproc)

sudo make install

sudo /bin/bash -c 'echo "/usr/local/lib" >
/etc/ld.so.conf.d/opencv.conf'
```

4.3.1.1 Publicación automática

Como ya se mencionó, GitLab permite configurar la publicación automática en los diferentes entornos dado que incluye las prácticas de integración continua/desarrollo continuo (CI / CD).

Los entornos de preproducción y producción se encuentran en servidores de la UNRN y en estos se publica la aplicación. La publicación automática se realiza en el entorno de preproducción, sin embargo, la publicación en el entorno de producción es realizada con la confirmación por parte del equipo de desarrollo, de manera manual.

Ambos entornos se basan en las buenas prácticas de gestión de la configuración del software y contiene las siguientes características:

1. Integración continua,
2. Despliegue continuo,
3. Política de control de versiones,
4. Sistema de control de versiones,
5. Espacios de trabajos privados,
6. Pruebas automatizadas,
7. Construcción automatizada,
8. Despliegue automatizado, y
9. Utilización de la metodología ágil Scrum.

A continuación se presenta un diagrama del entorno (Figura 4.10):

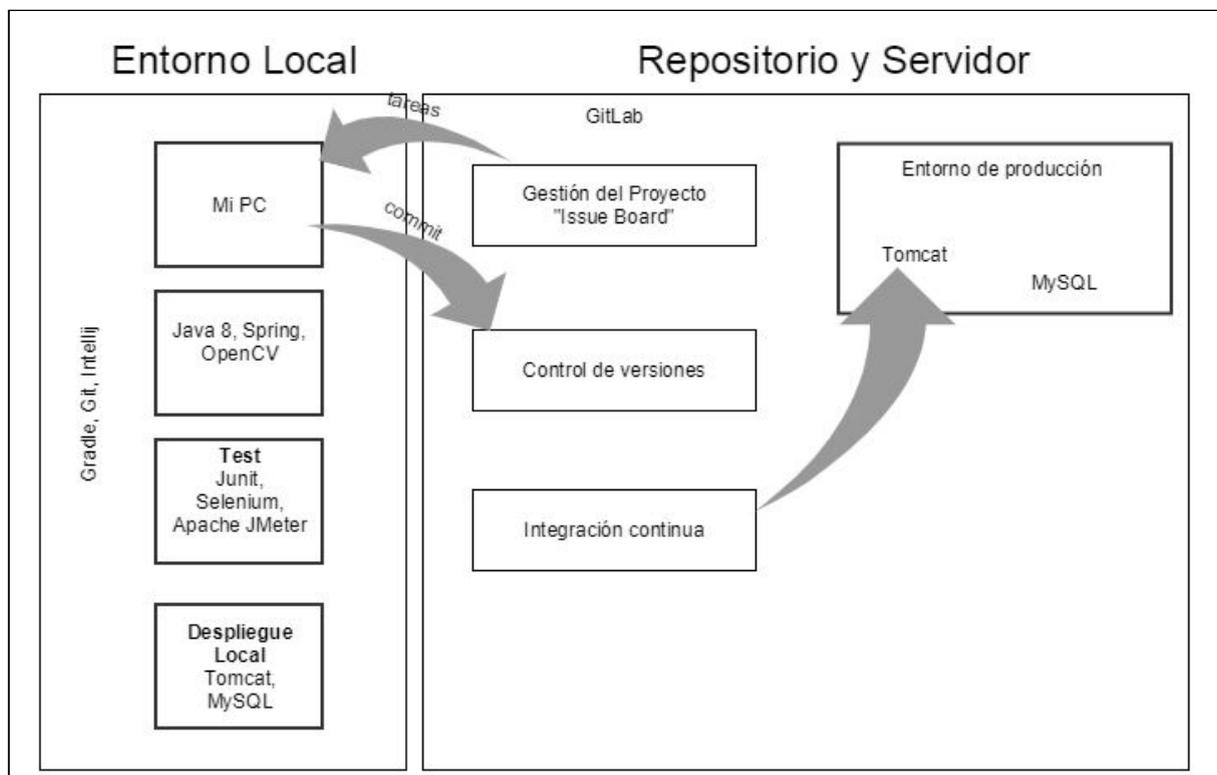


Figura 4.10. Entorno de trabajo

4.3.1.2 Estructura del proyecto

El proyecto DAEAM consiste en los siguientes módulos:

- daeam: Un proyecto general que contiene un archivo maven padre de tipo POM y dos proyectos hijos.

- daeam-business: que es un proyecto que genera un archivo binario de tipo jar.
- daeam-web: es un proyecto que genera un archivo binario de tipo war que tiene como dependencia a daeam-business.

A continuación se presenta la estructura de paquetes del proyecto, es decir cómo está organizado el mismo:

daeam

```

--- pom.xml
--- daeam-business
    --- ar.edu.unrn.lia.dao
        --- ar.edu.unrn.lia.dao.impl
    --- ar.edu.unrn.lia.dbconf
    --- ar.edu.unrn.lia.exception
    --- ar.edu.unrn.lia.generic
    --- ar.edu.unrn.lia.logger
    --- ar.edu.unrn.lia.model
    --- ar.edu.unrn.lia.seguridad
    --- ar.edu.unrn.lia.service
        --- ar.edu.unrn.lia.service.impl
    --- ar.edu.unrn.lia.util
--- daeam-web
    --- ar.edu.unrn.lia.bean
    --- ar.edu.unrn.lia.exception
    --- ar.edu.unrn.lia.export
    --- ar.edu.unrn.lia.security
    --- resources
    --- webapp
        --- error
        --- pages
        --- resources
        --- templates
        --- WEB-INF
        --- index.jsp

```

--- login.xhtml

Descripción del objetivo de los paquetes más importantes:

- Paquete model - `ar.edu.unrn.lia.model`
Define las entidades o el dominio de la aplicación, son los objetos reales persistentes de negocio.
- Paquete DAO - `ar.edu.unrn.lia.dao`
Data Access Object: Interfaz de entrada a los datos, se encarga del acceso a la base de datos.
- Paquete DAOImpl - `ar.edu.unrn.lia.dao.impl`
Se encarga de implementar la interfaz DAO y envía la consulta directamente a la base de datos.
- Paquete service - `ar.edu.unrn.lia.service`
Interfaz de entrada a la lógica de negocio, escrita desde la perspectiva del usuario. Servicios que son accedidos desde objetos de negocio y/u otros servicios.
- Paquete serviceImpl - `ar.edu.unrn.lia.service.impl`
Se encarga de la implementación de la lógica de negocio.
- Paquete bean - `ar.edu.unrn.lia.bean`
Punto de entrada desde la capa de presentación, sin lógica de negocio que actúa como controlador.

Una vez instalado y configurado el proyecto, la estructura de paquetes vista desde el entorno de desarrollo queda de la siguiente forma (ver Figura 4.11, Figura 4.12 y Figura 4.13).

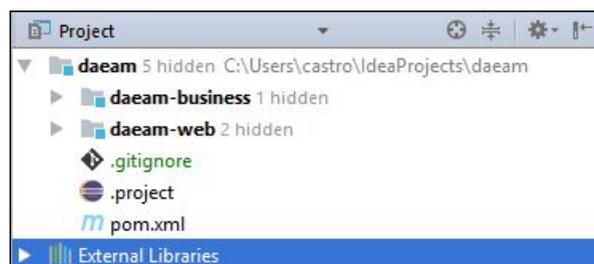


Figura 4.11. Proyecto daeam con los subproyectos

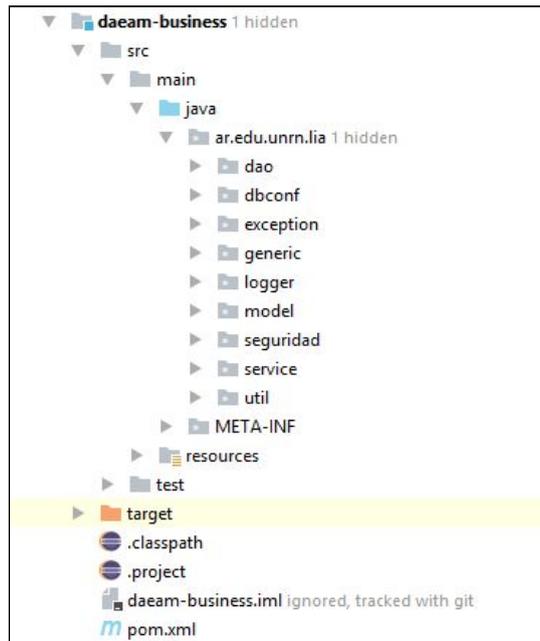


Figura 4.12. Proyecto daeam-business

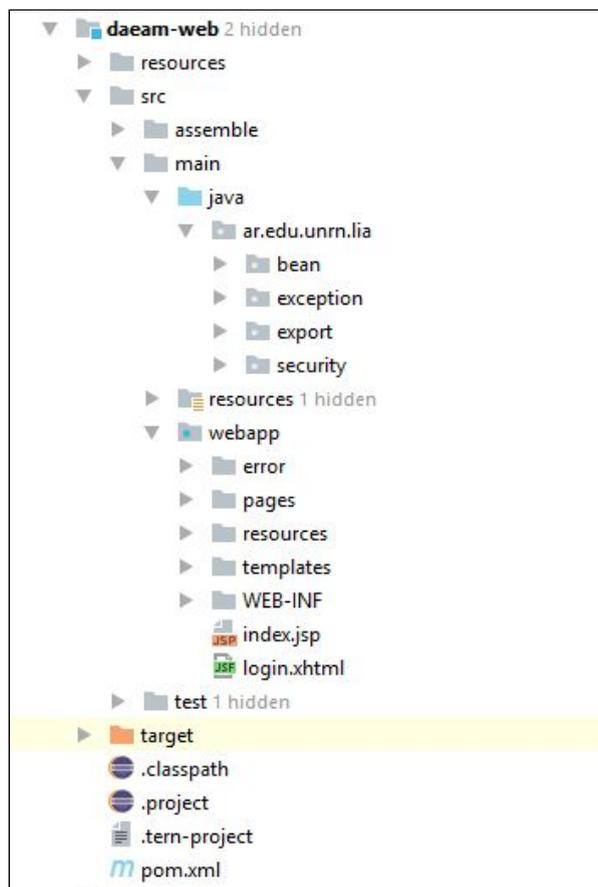


Figura 4.13. Proyecto daeam - web

4.3.1.3 Interfaces del proyecto base

Las pantallas de inicio de sesión y bienvenida del proyecto base son las siguientes (ver Figura 4.14 a Figura 4.20).

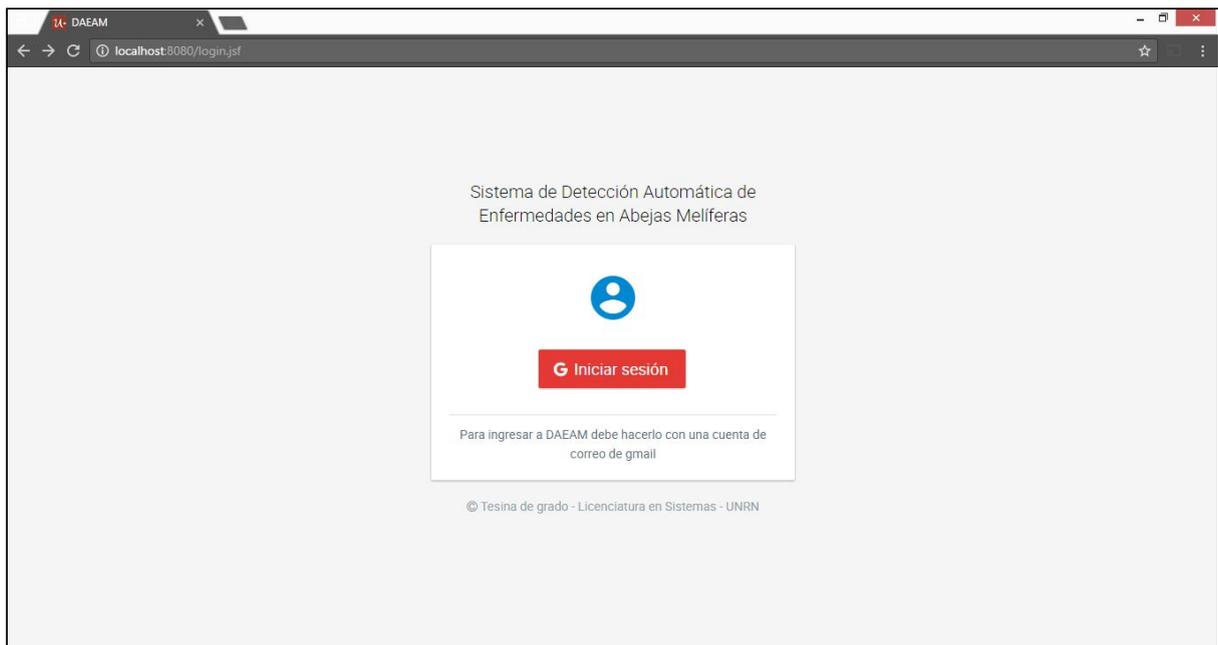


Figura 4.14. Pantalla de inicio de sesión

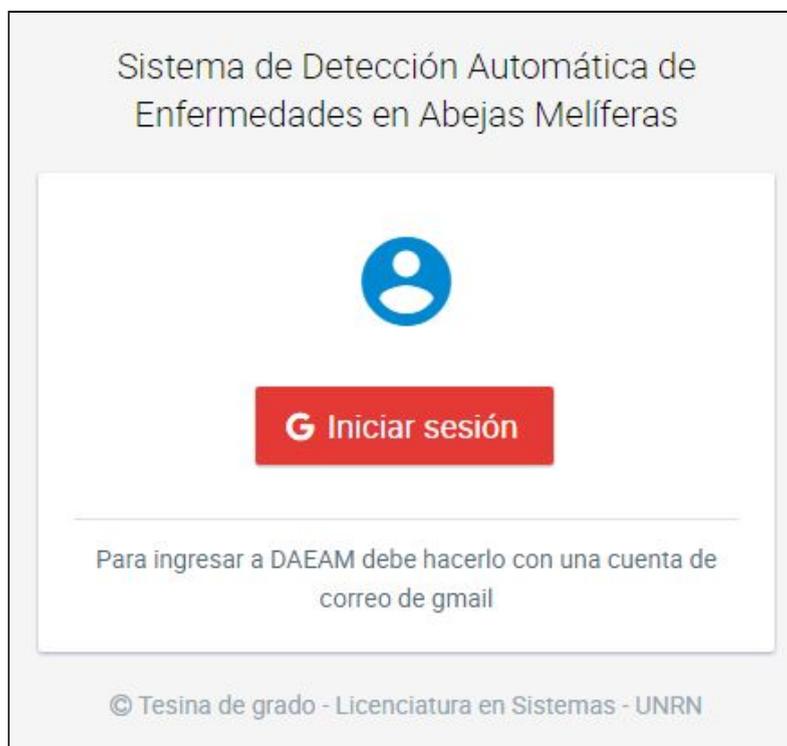


Figura 4.15. Detalle pantalla de inicio de sesión

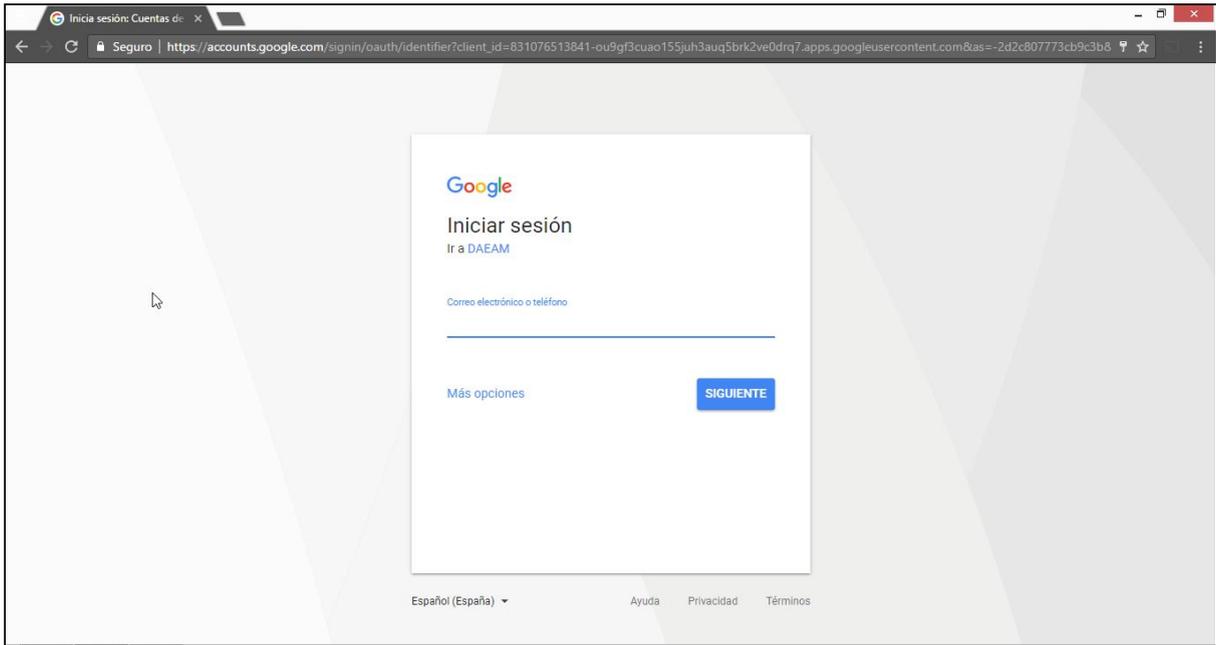


Figura 4.16. Inicio de sesión con correo electrónico

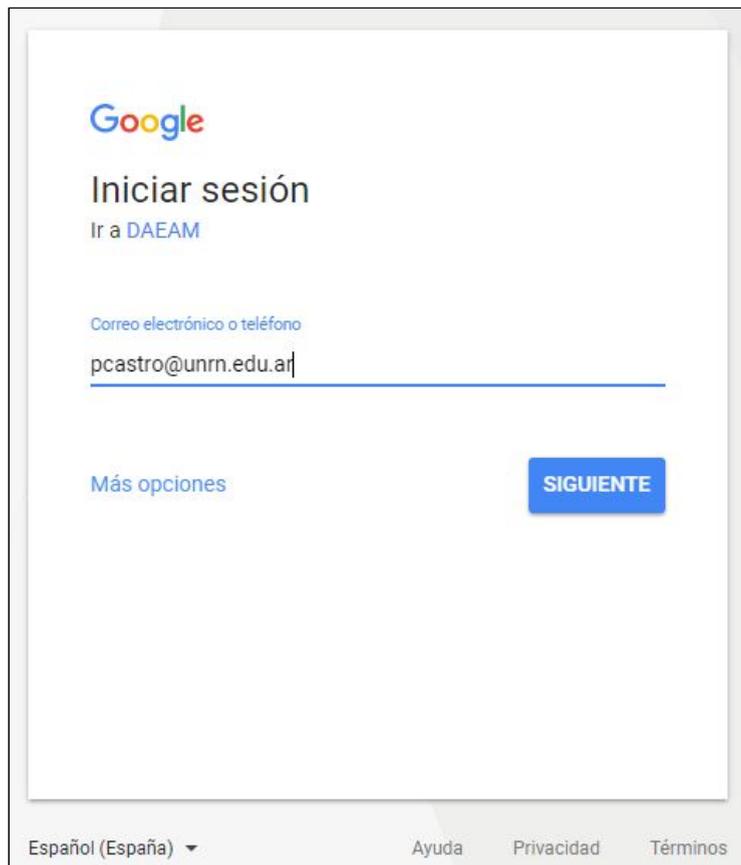


Figura 4.17. Ingreso de correo electrónico

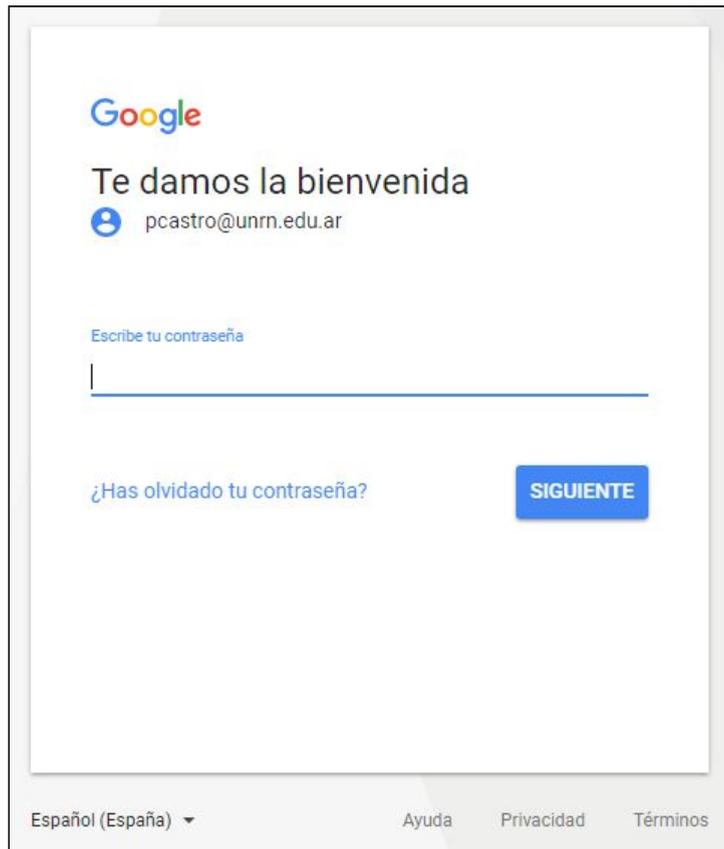


Figura 4.18. Ingreso de contraseña del correo electrónico

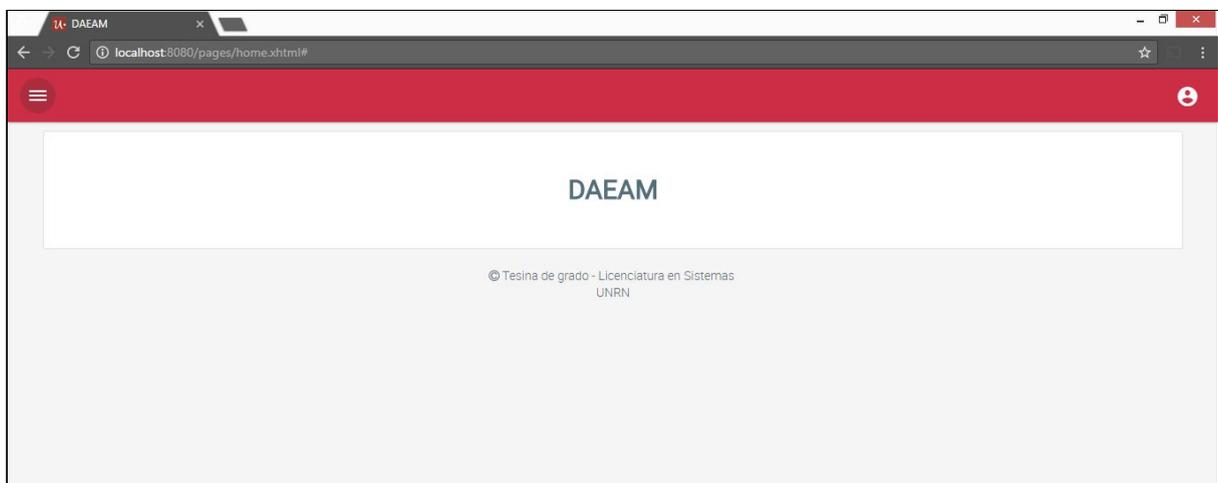


Figura 4.19. Pantalla de bienvenida

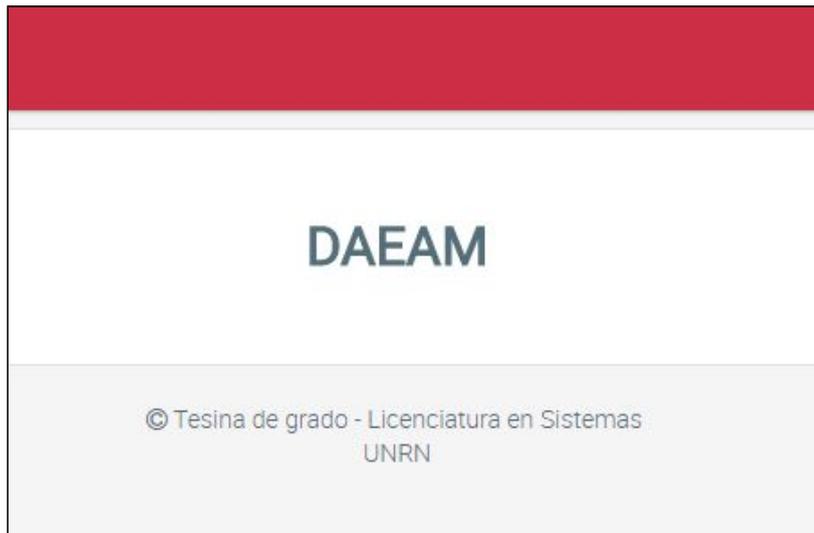


Figura 4.20. Detalle de la pantalla de bienvenida

4.3.2 Sprint Nro. 1 Administración del Sistema Web

Este Sprint tiene como objetivo implementar las funcionalidades requeridas para la administración de los usuarios del sistema: crear, eliminar, y modificar usuarios, además de ver el detalle de cada uno de los usuarios. La configuración del Sprint se realiza en GitLab y se establece el conjunto de tareas pertenecientes a este, tal como se muestra en las siguientes imágenes (ver Figura 4.21 y Figura 4.22):

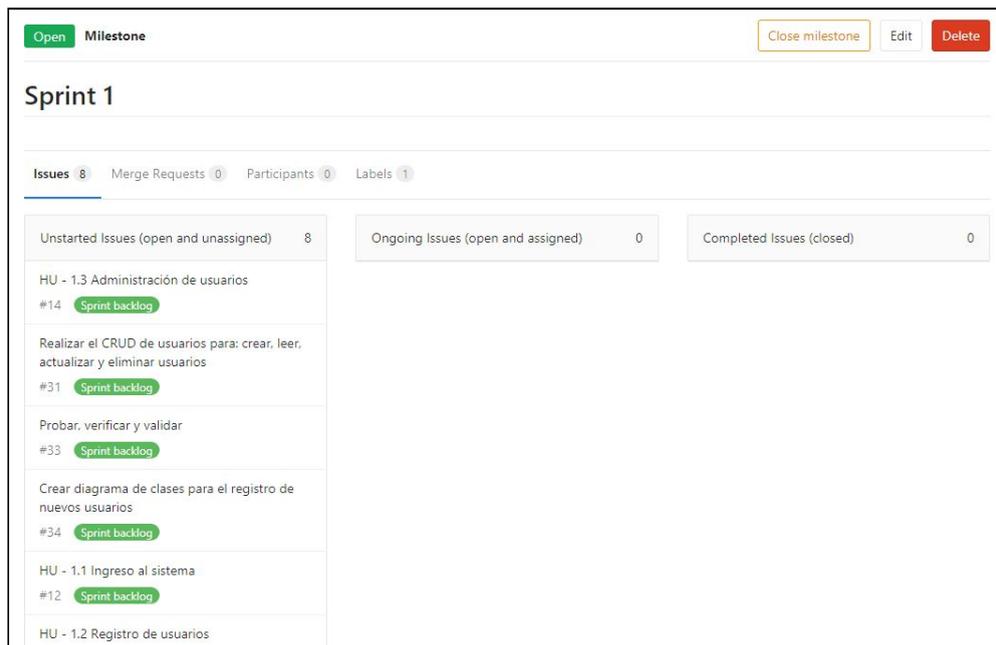


Figura 4.21. Configuración del Sprint Nro. 1 en GitLab

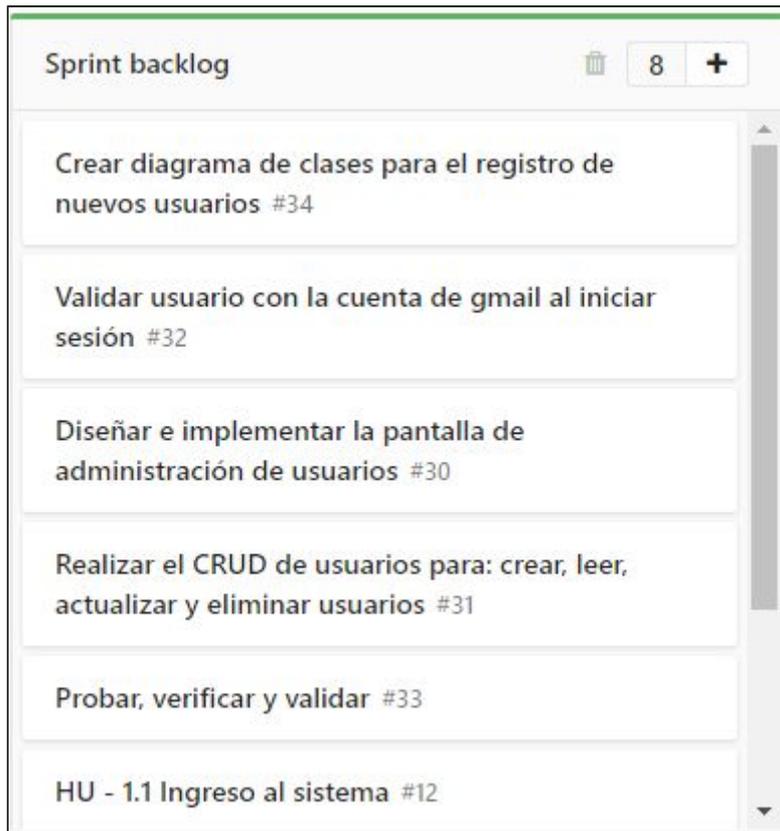


Figura 4.22. Sprint Backlog del Sprint Nro. 1

Requisitos funcionales

- El sistema permitirá dar de alta, a los usuarios.
- Permitirá editar y dar de baja a los usuarios.
- Los usuarios se podrán inhabilitar.
- Los usuarios accederán a la aplicación a través de su cuenta de gmail.
- El sistema debe manejar roles para diferentes tipos de usuarios.
- Permitirá consultar datos de los usuarios.

Requisitos no funcionales

- El sistema brindará seguridad para ser utilizado por usuarios registrados del sistema que cuenten con una cuenta de gmail donde solo podrán ver y realizar las funcionalidades correspondientes a su rol.
- El sistema tendrá una interfaz amigable y de fácil uso.

Historias de usuario Sprint Nro. 1

Las historias de usuarios consideradas en el Sprint Nro. 1 se describen a continuación.

ID: HU-1.1	Nombre: Ingreso al sistema	Usuario: Administrador, Laboratorista
Descripción: Iniciar sesión con mi cuenta de correo de gmail para poder realizar las funcionalidades disponibles para mi rol.		
Criterios de aceptación: Un usuario no puede operar sino se encuentra registrado en el sistema o no inicio sesión. Se podrá acceder a la aplicación con el login de Google+. El usuario solo puede realizar la funcionalidad disponible para su rol.		

ID: HU-1.2	Nombre: Registro de usuarios	Usuario: Administrador
Descripción: Poder completar el formulario que me permite dar de alta a usuarios.		
Criterios de aceptación: Un usuario no puede registrarse con la misma dirección de mail que otro existente. Se validan que se completen los campos obligatorios. Se envía un mensaje de "Éxito" luego de completar la información del formulario.		

ID: HU-1.3	Nombre: Administración de usuarios	Usuario: Administrador
Descripción: Encargarme de la administración de los usuarios, editar su información relacionada y eliminar los usuarios que se encuentran registrados.		
Criterios de aceptación: Se actualiza la información del usuario correspondiente. Se elimina el usuario deseado. Se edita la información de un usuario en particular.		

Definición del equipo de trabajo Sprint Nro. 1

El equipo de trabajo considerado para llevar a cabo el Sprint Nro. 1, está conformado de la misma forma en el Sprint 0, descrito en la Tabla 4.3

4.3.3 Sprint Nro. 2 Módulo de identificación y conteo de esporas de nosemas

Este Sprint tiene como objetivo implementar las funcionalidades requeridas para la identificación y conteo de esporas de nosemas en un apiario o una imagen en particular. La configuración del mismo se realiza en GitLab y se establece el conjunto de tareas pertenecientes al Sprint, tal como se muestra en las siguientes imágenes (ver Figura 4.23 y Figura 4.24):

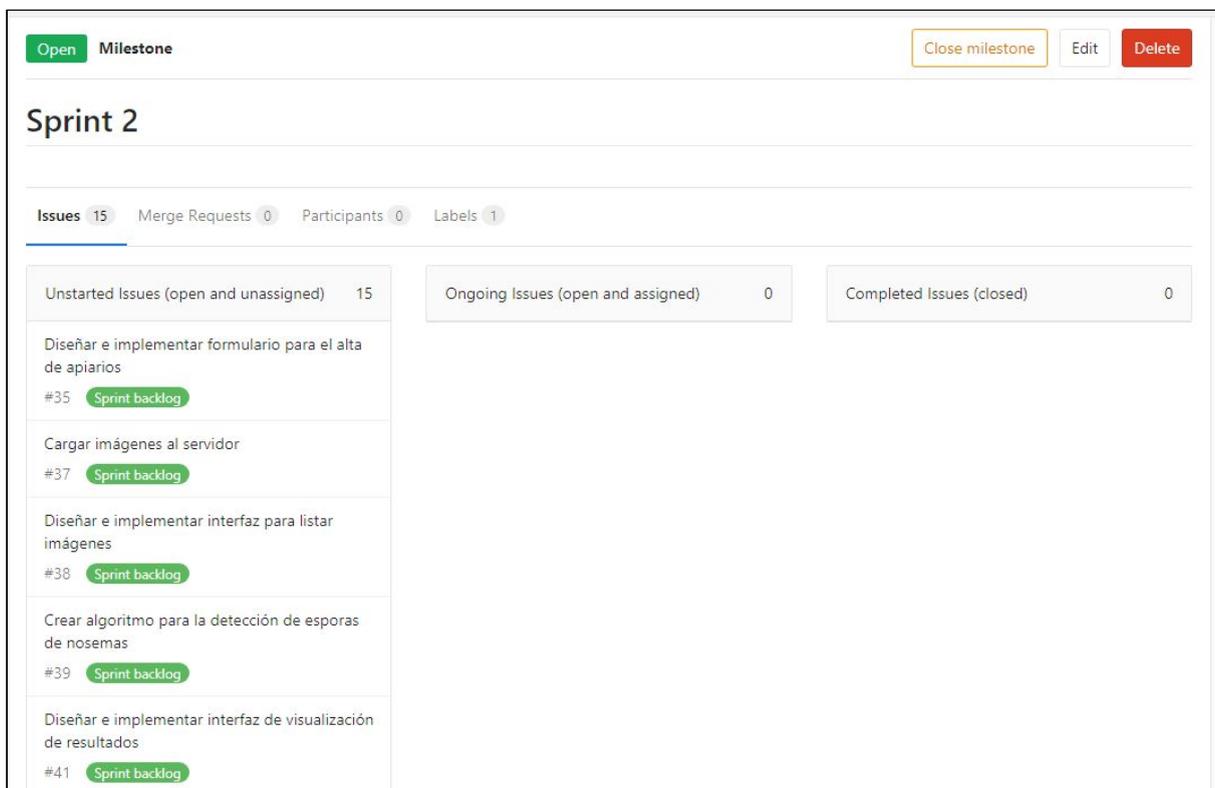


Figura 4.23. Configuración del Sprint Nro. 2 en GitLab

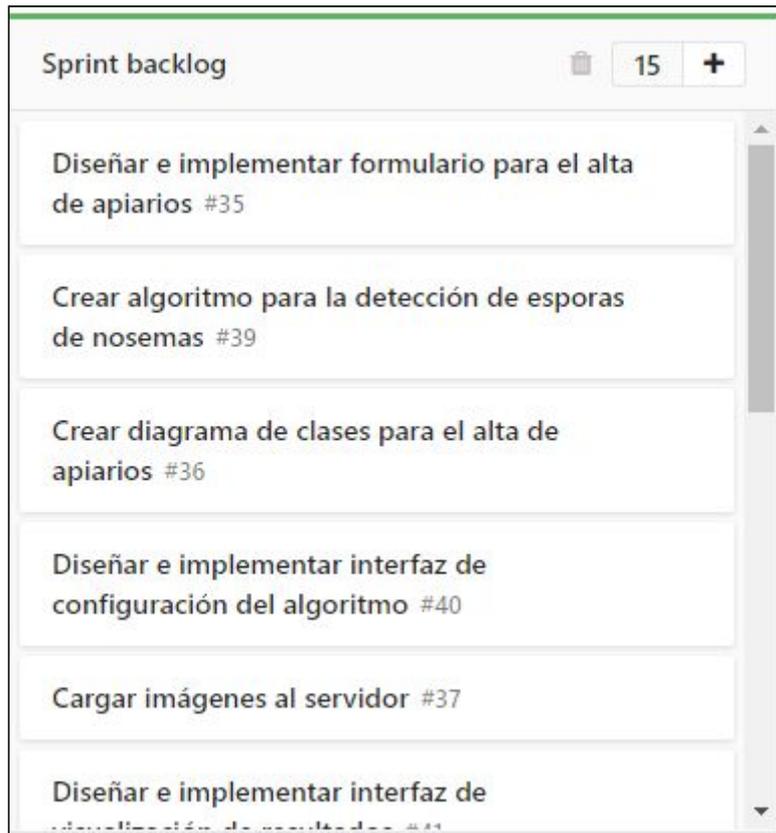


Figura 4.24. Sprint Backlog del Sprint Nro. 2

Requisitos funcionales

- La aplicación permitirá crear apiarios, editar y eliminar.
- La aplicación permitirá agregar imágenes a un apiario en particular.
- La aplicación permitirá buscar, seleccionar una imagen o apiario.
- La aplicación permitirá detectar y contar el número de esporas de noseemas presentes en cada imagen.
- Permitirá ver en número de esporas presentes en cada imagen o apiario.

Requisitos no funcionales

- La aplicación brindará seguridad para ser utilizada por usuarios registrados del sistema con diferentes roles
- La aplicación tendrá una interfaz amigable y de fácil uso.

Historias de usuario Sprint Nro. 2

A continuación se definen las historias de usuario de este Sprint.

ID: HU-2.1	Nombre: Crear apiario	Usuario: Laboratorista
Descripción: Añadir un nuevo apiario con un nombre, una localización, nombre del productor, nombre del análisis, la cantidad de colmenas y la fecha en la que se realizó el análisis.		
Criterios de aceptación: Se valida que se completen los campos obligatorios. Se envía un mensaje de “Éxito” luego de completar la información del formulario. Se registra un nuevo apiario con la información relacionada.		

ID: HU-2.2	Nombre: Gestionar apiarios	Usuario: Laboratorista
Descripción: Listar todos los apiarios existentes, editar la información de uno existente, ver la información relacionada a un apiario determinado o eliminar uno existente con toda su información relacionada.		
Criterios de aceptación: Se listan los apiarios creados por el usuario. Se actualiza la información del apiario correspondiente. Se elimina el apiario deseado.		

ID: HU-2.3	Nombre: Asociar imágenes al apiario	Usuario: Laboratorista
Descripción: Cargar imágenes en un apiario determinado.		
Criterios de aceptación: Deben existir nuevas imágenes subidas al servidor. Se envía un mensaje de “Éxito” al realizar esta operación.		

ID: HU-2.4	Nombre: Gestionar imágenes	Usuario: Laboratorista
<p>Descripción: Listar todos las imágenes de un apiario, ampliar la imagen al hacer click sobre la misma o eliminar una o más imágenes de un apiario con toda la información relacionada.</p>		
<p>Criterios de aceptación: Se visualiza la información de la imagen correspondiente. Se amplía la imagen al hacer un click sobre la misma. Se elimina el imagen deseada.</p>		

ID: HU-2.5	Nombre: Configurar algoritmo	Usuario: Laboratorista
<p>Descripción: A partir del conjunto de imágenes cargadas seleccionar una y configurar los parámetros disponibles en la aplicación para realizar la detección de esporas de noseemas que se encuentran en la imagen.</p>		
<p>Criterios de aceptación: A medida que se actualizan los parámetros de la imagen original, visualizar el resultado de la misma en ese mismo momento. La parámetros de la configuración serán los que se apliquen para el análisis del conjunto de imágenes.</p>		

ID: HU-2.6	Nombre: Detección de esporas de noseemas	Usuario: Laboratorista
<p>Descripción: La aplicación debe permitir realizar la detección automática de esporas de noseemas a partir de los parámetros definidos en la configuración de la misma o a través de una configuración por defecto.</p>		
<p>Criterios de aceptación: Según el conjunto de imágenes definidos y la configuración establecida ejecutar un análisis que detecte automáticamente el número de esporas presentes en cada una de las imágenes de un apiario en particular.</p>		

ID: HU-2.7	Nombre: Visualizar resultados	Usuario: Laboratorista
<p>Descripción:</p> <p>La aplicación me debe mostrar los resultados del análisis de imágenes asociadas a un apiario determinado, permitiendo listar todos estos resultados y visualizar la información relacionada de cada uno.</p>		
<p>Criterios de aceptación:</p> <p>Listar el número de esporas de nosemas en cada imagen.</p> <p>Mostrar las imágenes resultantes donde se marquen cada espora identificada, etiquetada con un número de manera secuencial.</p> <p>Mostrar el total de esporas detectadas en el apiario.</p>		

Seguidamente, se describe el procedimiento para el reconocimiento de cada parásito:

- ***Detección y conteo automático de esporas de nosemosis***

Uno de los principales problemas al desarrollar un sistema de visión por computador es poder elegir las técnicas de procesamiento adecuadas para conseguir extraer la información necesaria (Gonzalez & Woods, 2008) y lograr el objetivo planteado. En consecuencia, se probaron distintas técnicas basadas en diferentes sistemas automatizados desarrollados precisamente en esta misma área (Alvarez-Ramos, 2010; Alvarez-Ramos, 2013). De ahí que se realizan diferentes etapas para poder elaborar el trabajo entre las cuales se encuentran la adquisición de la imagen, el pre-procesamiento, la segmentación, la búsqueda de contornos, la detección de elipses y, por último, el conteo de estas elipses que es la forma que tienen las esporas de nosemas.

Adquisición de la imagen

La adquisición de la imagen se realizó a través de una cámara digital para microscopio marca Mikoba modelo DCM900 (USB2.0) de 9 Megapíxeles, con chip CMOS, montada a un microscopio óptico marca Leica modelo DM500 (ver Figura 4.24). Las imágenes se obtuvieron con objetivo de microscopio seco de 40X, adquiriendo 400 aumentos totales, y se digitalizaron utilizando el software propio de la cámara fotográfica.



Figura 4.24. Elementos de adquisición de la imagen

Se tomó una muestra de abejas pertenecientes a un apiario de la localidad de Carmen de Patagones, que se conservaron con alcohol etílico hasta su procesamiento conforme método estandarizado (OIE, 2013). Con el fin de conocer aquellos parámetros dependientes de métodos de microscopía óptica que permiten obtener imágenes con menor ruido de fondo, algunas imágenes se toman directamente y otras luego de incluir diferentes modificaciones a las técnicas microscópicas básicas, como el agregado de colorantes (colorante azul número 2 del kit T15, Biopur) y el uso de microscopía de contraste de fase. Este trabajo fue realizado en el laboratorio de la Fundación Barrera Zoofitosanitaria Patagónica - FunBAPA, laboratorio LR: 0016 (Red Nacional de Laboratorios del SENASA).

Pre-procesamiento

En esta etapa se aplican diversas técnicas de pre-procesamiento de imágenes microscópicas, tales como la conversión a escala de grises, filtro de desenfoque gaussiano, umbral adaptable y transformaciones morfológicas. Se pueden visualizar los resultados de estas técnicas en las Figura 4.25 a la Figura 4.28.

El objetivo principal de emplear estas técnicas es mejorar la calidad de la imagen. En principio se convierte la imagen de la muestra a escala de grises, con el fin de eliminar los niveles de tono y saturación, manteniendo la luminancia. Después de eso, se aplica el filtro de desenfoque gaussiano para hacer la imagen más suave usando un núcleo de convolución obtenido mediante experimentación. Así, al estar más suavizadas se detectan menos los bordes del fondo y se destacan los bordes de las esporas.

La técnica de umbral adaptable se utiliza para transformar las imágenes resultantes en imágenes binarias y eliminar la mayor parte del fondo de la misma. Las operaciones morfológicas, como la dilatación y la erosión se aplican a la imagen binaria para aumentar el relieve de los bordes y eliminar en mayor medida aquellos objetos que no se desean detectar para lograr una resultante más satisfactoria.

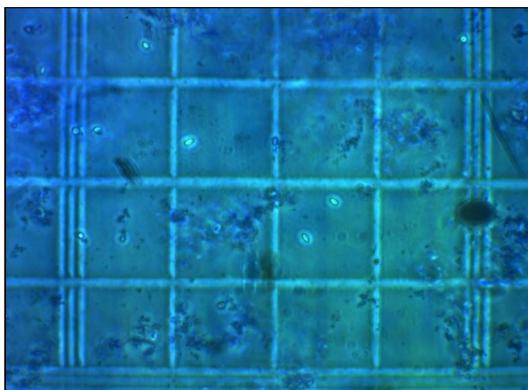


Figura 4.25. Imagen Original

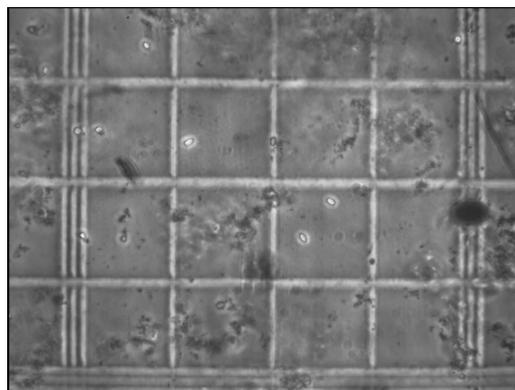


Figura 4.26 – Imagen pre-procesada en escala de grises

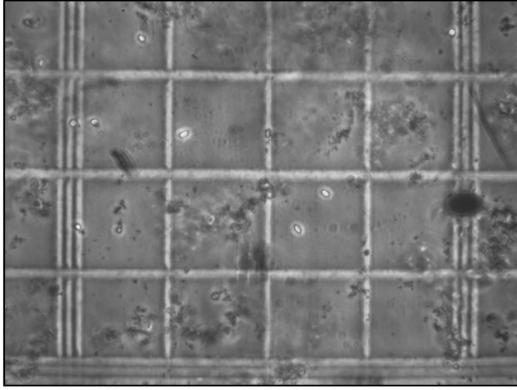


Figura 4.27. Imagen pre-procesada con el filtro Gaussian Blur

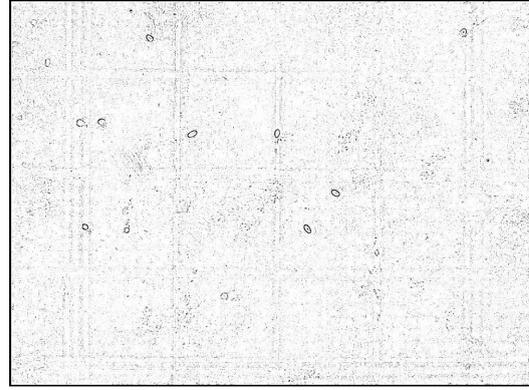


Figura 4.28. Imagen binarizada utilizada para encontrar los contornos

Segmentación:

- **Búsqueda de contornos:** Para llevar a cabo la tarea de encontrar los contornos de cada espora, se emplea la función de la biblioteca OpenCV llamada `findContours` (“Structural Analysis and Shape Descriptors — OpenCV 2.4.13.0 documentation,” n.d.-b) que utiliza el algoritmo Suzuki (Suzuki, 1985) en su implementación. Esta función es capaz de extraer todos los contornos de los píxeles presentes en la imagen con valor diferente a cero. Aquí sólo se utilizan los bordes exteriores de las esporas que se encuentran en las imágenes, ya que esa información es necesaria para poder contabilizarlas. En la Figura 4.29 se puede observar el resultado de los contornos obtenidos a partir de la imagen binarizada.



Figura 4.29. Búsqueda de los contornos

- Detección de elipses: Como las esporas de nosemas pueden ser consideradas elipses debido a su forma, en esta parte de la etapa de segmentación se aplica la función fitEllipse (“Structural Analysis and Shape Descriptors — OpenCV 2.4.13.0 documentation,” n.d. -a) proporcionada por la librería OpenCV, que implementa el algoritmo de detección de elipses propuesto por Fitzgibbon (Fitzgibbon, 1995). Dado un conjunto de puntos retorna la elipse que mejor se adapta a estos puntos, en un término de mínimos cuadrados. La imagen resultante devuelve una elipse para cada contorno, motivo por el cual en la siguiente etapa se filtra dicha imagen para descartar aquellas elipses que no tengan el tamaño deseado.

Reconocimiento o clasificación

- Conteo de esporas: Por lo general, las esporas de nosemas tienen una forma elíptica más alargada que circular y con un determinado tamaño que varía en un rango de 5-7 μ m x 3-4 μ m aproximadamente. Considerando estas medidas, se asume que toda figura preseleccionada de forma elíptica con medidas de ancho y longitud de la elipse dentro del rango mencionado, corresponde a una espora de Nosema spp. En caso contrario, la parametrización descarta del conteo las elipses al no pertenecer al criterio de selección. Los resultados del análisis se pueden ver en el siguiente capítulo.

4.3.4 Sprint Nro. 3 Módulo de identificación y conteo de ácaros de varroa

Este Sprint tiene como objetivo implementar las funcionalidades requeridas para la identificación y conteo de ácaros de varroas para un apiario en particular. La configuración del Sprint Nro. 3, como en los casos anteriores, se realiza en GitLab donde se establecen el conjunto de tareas pertenecientes al mismo, tal como se muestra en las siguientes figuras (ver Figura 4.30 y Figura 4.31):

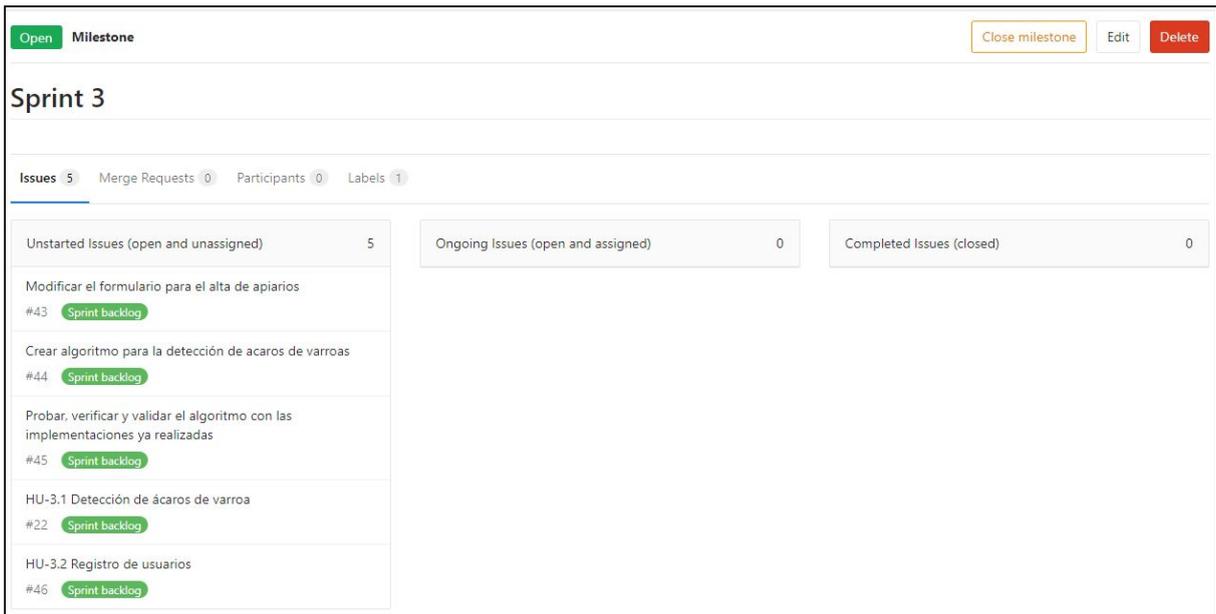


Figura 4.30. Configuración del Sprint Nro. 3 en GitLab

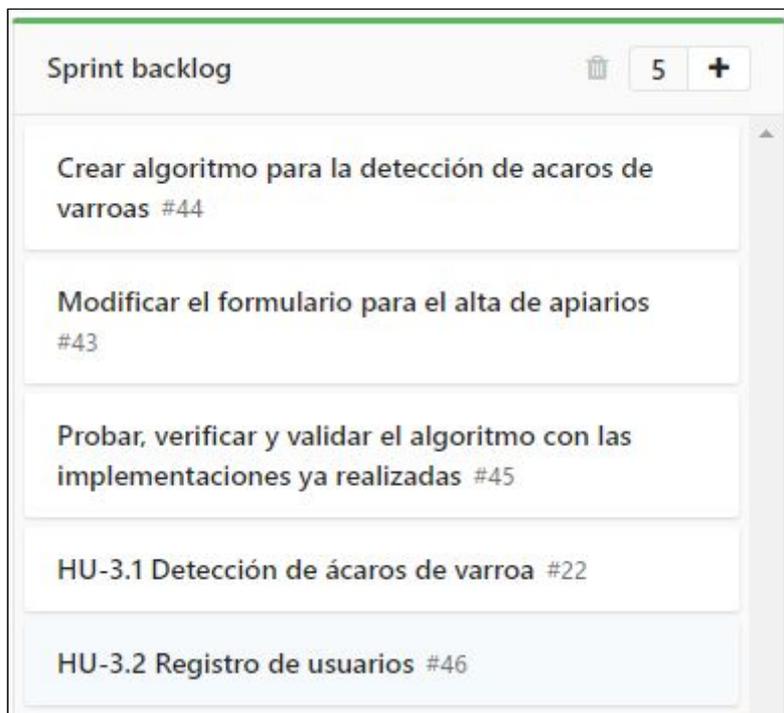


Figura 4.31. Sprint Backlog del Sprint Nro. 3

Requisitos funcionales

- La aplicación permitirá detectar y contar el número de ácaros de varroas presentes en cada imagen.
- La aplicación permitirá la registración de usuarios.

Requisitos no funcionales

- La aplicación brindará seguridad para ser utilizada por usuarios registrados del sistema con diferentes roles.
- La aplicación tendrá una interfaz amigable y de fácil uso.

Historias de usuario Sprint Nro. 3

En principio en este Sprint se define una única historia de usuario dado que se utilizan las funcionalidades ya implementadas. Se agrega una nueva historia de usuario referida al registro de usuarios que en el backlog del principio no estaba definida. Ambas se detallan a continuación:

ID: HU-3.1	Nombre: Detección de ácaros de varroa	Usuario: Laboratorista
Descripción: Al igual que con la detección automática de esporas de nosemas, a partir de los parámetros definidos en la configuración de la aplicación, se debe poder detectar automáticamente los ácaros de varroa tanto en las imágenes tomadas desde los pisos trampa como aquellas tomadas desde el microscopio. Para esto también quiero utilizar las funcionalidades que ya existen para la gestión de apiarios e imágenes.		
Criterios de aceptación: Según el conjunto de imágenes definidos y la configuración establecida ejecutar un análisis que permita detectar automáticamente el número de ácaros de varroas presentes en cada una de las imágenes y en todo el apiario.		

ID: HU-3.2	Nombre: Registro de usuario	Usuario: Laboratorista/ Apicultor
<p>Descripción:</p> <p>Registrarme en la aplicación completando un formulario con mi información como correo electrónico, nombre y lugar de trabajo. Una vez registrado el usuario quiero recibir un mail que confirme mi acción y me permita confirmar el registro en la aplicación.</p>		
<p>Criterios de aceptación:</p> <p>Se registra el usuario con la información correspondiente.</p> <p>Se recibe un mail con el link de confirmación de registro en la aplicación.</p> <p>Una vez confirmado el registro puede iniciar sesión, caso contrario no puede iniciar sesión.</p>		

Aquí se describe el procedimiento para el reconocimiento de los ácaros de varroa:

- ***Detección y conteo automático de ácaros de varroas***

Se trabajó en un algoritmo de detección y conteo de ácaros de varroas. El propósito del mismo es detectar cada uno de los ácaros presentes en cada una de las imágenes de un apiario para lograr un análisis de este último.

Los algoritmos de detección generalmente se basan en técnicas de segmentación. Se denomina segmentación al proceso que divide la imagen en segmentos que resaltan los objetos que tienen significado, en este caso las varroas. El enfoque utilizado para resolver este problema es el método de umbral adaptativo.

El planteamiento de la solución consiste en un procesamiento de las imágenes en etapas o pasos. Se realizaron diferentes etapas para poder desarrollarlo, entre las que se encuentran la adquisición de la imagen, el pre-procesamiento, la segmentación, la búsqueda de contornos, la detección de elipses y, por último, el conteo de estas elipses que en este contexto representan los ácaros de varroas.

Adquisición de la imagen

Para poder ejecutar el análisis de forma adecuada las imágenes utilizadas para un apiario se toman bajo las mismas condiciones y deben ser del mismo tamaño, ya que sino lo que el algoritmo segmenta para una imagen no es significativo para otra distinta.

Las fotografías se toman desde una cámara de un celular LG 50 con una resolución de 5 megapíxeles y un aumento de 10 x 20, es decir, 200 de aumento. La muestra pertenece a un apiario del INTA ubicado en el Campo Forestal General San Martín que está localizado en el paraje Las Golondrinas, Chubut.

Pre-procesamiento

Aquí se aplican diversas técnicas de pre-procesamiento de imágenes. En caso que el fondo de la imagen sea blanco se vuelve a negro, luego se obtiene una imagen más nítida, se aplica un filtro de laplace, se realiza la conversión de la imagen a escala de grises y se aplica un filtro de desenfoque gaussiano. Se pueden visualizar los resultados de estas técnicas en las Figura 4.32 a Figura 4.36.

Como se indicó en el párrafo anterior, en principio dada una imagen si el fondo es blanco se transforma a negro. Esto ayuda a discriminar los objetos de primer plano más fácilmente cuando se aplica la transformación de distancia. En la imagen siguiente (ver Figura 4.32) se muestra el resultado de aplicar esta etapa, dado que el fondo no es blanco la misma no se modifica.



Figura 4.32. Imagen de varroas desde el microscopio

En la siguiente etapa se crea un kernel que es utilizado para afinar la imagen, esto sirve para adelgazar los bordes de los objetos de primer plano (ver Figura 4.33). Luego, se aplica un filtro laplaciano (ver Figura 4.34).



Figura 4.33. Nueva imagen más nítida



Figura 4.34. Imagen con filtro de Laplace

Ahora se transforma la nueva imagen en una escala de grises y luego se aplica el filtro de desenfoque gaussiano para suavizar la imagen y realzar los bordes de los ácaros y detectar menos los bordes del fondo. (ver Figura 4.35 y Figura 4.36).

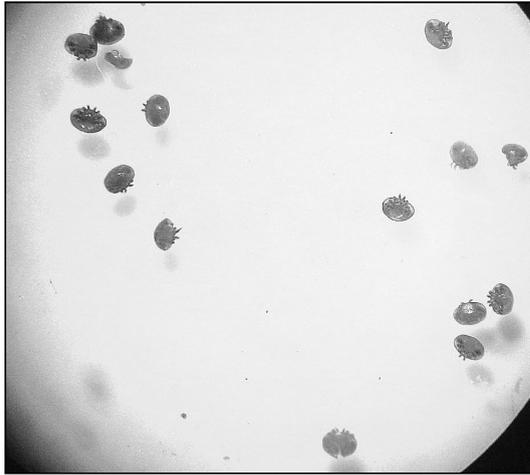


Figura 4.35. Escala de grises

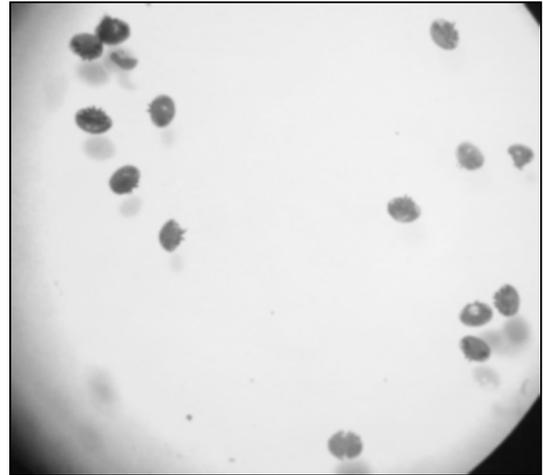


Figura 4.36. Imagen con filtro de desenfoque gaussiano

Segmentación:

Un método para la realizar la segmentación es comparar cada píxel respecto a un umbral de manera que los píxeles queden etiquetados para formar grupos de elementos reconocibles. Debido a su simplicidad, la técnica del umbral (Thresholding) es el método más comúnmente usado pero también es uno de los más propensos a errores.

Sin embargo, resulta muy satisfactorio siempre que:

- Las varroas estén lo suficientemente separadas entre sí.
- El umbral pueda distinguir lo suficiente las varroas respecto del fondo.

El último objetivo de esta técnica es determinar los contornos que definen los objetos. En resumen, lo primero que se aplica en esta etapa es la técnica de umbral adaptable o adaptativo, utilizada para transformar las imágenes resultantes en imágenes binarias y eliminar la mayor parte del fondo de la misma (ver Figura 4.37). Este algoritmo en lugar de utilizar un valor global de umbral para toda la imagen, utiliza un umbral diferente para cada punto de la imagen. Así se obtienen diferentes umbrales para las diferentes regiones de la imagen, consiguiendo mejores resultados en imágenes con diferencias de iluminación.

Cabe destacar tres parámetros importantes:

- `adaptiveMethod`: decide cómo se calcula el valor umbral. Existen dos métodos diferentes, `ADAPTIVE_THRESH_MEAN_C` donde se consigue aislar los píxeles de la imagen cuyos valores disten de la media global un cierto valor `C` constante, y `ADAPTIVE_THRESH_GAUSSIAN_C` donde, a diferencia del método anterior, el valor umbral se establece mediante la suma ponderada de los píxeles vecinos.
- `Tamaño de bloque`: decide el tamaño del área de vecindad.
- `C`: determina la constante que indica cuánto debe desviarse el valor del píxel de la media para ser aislado.

A continuación se realiza la transformación de distancia en la imagen binaria anterior (ver Figura 4.38). Además se normaliza la imagen de salida para poder ajustar el resultado (ver Figura 4.39).

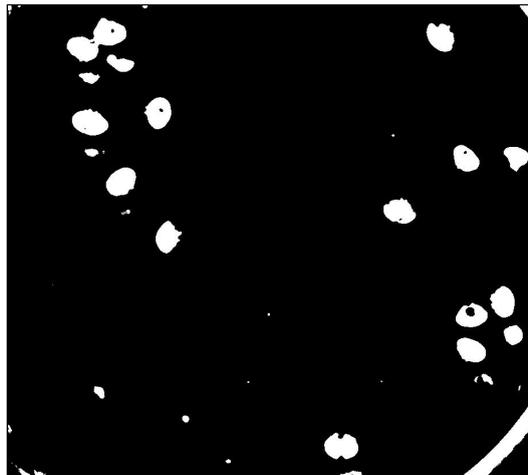


Figura 4.37. Imagen binarizada

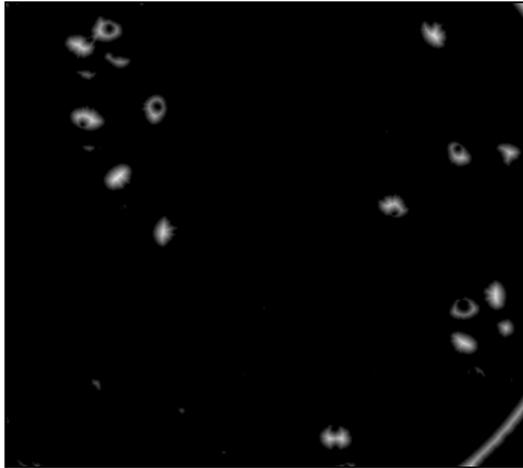


Figura 4.38. Transformación de distancia

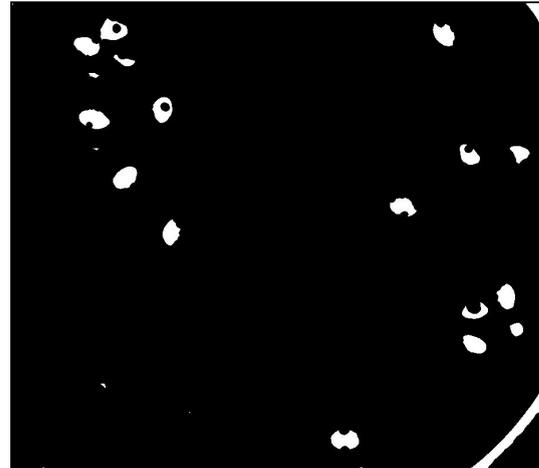


Figura 4.39. Image con objetos binarios

Luego se realiza la operación morfológica de dilatación para agrandar y poder extraer los marcadores de los objetos en primer plano (ver Figura 4.40) y se aplica la operación `findContours()` para encontrar todos esos marcadores (ver Figura 4.41).



Figura 4.40. Imagen dilatada



Figura 4.41. Dibujar contornos

Reconocimiento o clasificación

La última etapa de segmentación que se aplica es la detección de elipses. Por su forma las varroas pueden considerarse como elipses, en consecuencia, en este caso se aplica el método `fitEllipse()` de OpenCV. El objetivo de la última etapa del proceso es detectar y contar los ácaros de varroas. Para detectar cada varroa es necesario identificar qué segmentos son varroas.

Se puede ver en la siguiente imagen el resultado obtenido para esta prueba (ver Figura 4.42) en la que se marcan las varroas y cada una se etiqueta con un número.

Capítulo 5: Validación de los resultados

En este capítulo se muestran los resultados alcanzados en base a la prueba del sistema con un usuario administrador. Se presentan tablas e imágenes según las pruebas realizadas con los algoritmos de detección propuestos, se analizan los resultados obtenidos en cada caso y se comparan con el método actual.

5.1 Casos de prueba

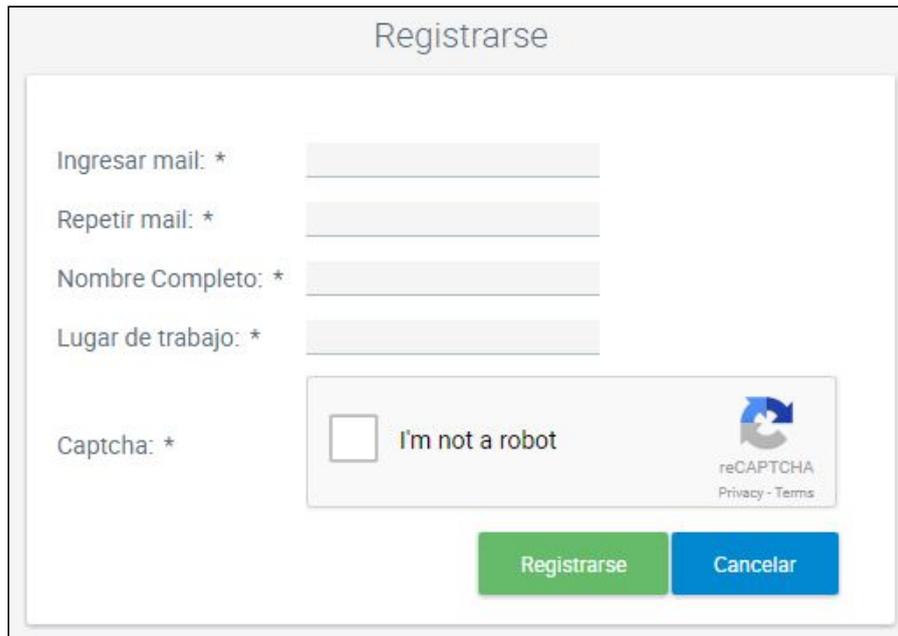
Se inició sesión con el usuario pcastro@unrn.edu.ar que tiene rol de administrador (ver Figura 5.1). A la aplicación se puede acceder desde:

<http://tesis.lia.unrn.edu.ar:8080/daeam>.



Figura 5.1. Inicio de sesión

Para realizar el registro de usuarios se agregó un botón de registro llamado “Registrarse” debajo del botón de inicio de sesión (Figura 5.1). Para probarla, una vez seleccionada esta opción, se registró un usuario completando el formulario de registro (Figura 5.2).



The image shows a registration form titled "Registrarse". It contains the following fields and elements:

- Form title: Registrarse
- Field: Ingresar mail: * (with an asterisk indicating it is required)
- Field: Repetir mail: * (with an asterisk indicating it is required)
- Field: Nombre Completo: * (with an asterisk indicating it is required)
- Field: Lugar de trabajo: * (with an asterisk indicating it is required)
- Field: Captcha: * (with an asterisk indicating it is required)
- Captcha widget: Includes a checkbox, the text "I'm not a robot", and the reCAPTCHA logo with links for "Privacy" and "Terms".
- Buttons: A green "Registrarse" button and a blue "Cancelar" button.

Figura 5.2. Formulario de registro

El formulario se completó con la siguiente información (Figura 5.3):

- Mail: castropnicolas@gmail.com.
- Nombre completo: Castro Patricio Nicolas.
- Lugar de trabajo: LIA.

Luego al hacer click en el botón “Registrarse” se muestra un mensaje que indica que la acción fue realizada de manera exitosa (Figura 5.4) y además se envía un correo electrónico al mail ingresado (Figura 5.5). Para finalizar el proceso de registración el último paso es ir al link recibido que confirma el registro.

Figura 5.3. Formulario de registro



Figura 5.4. Formulario de registro



Figura 5.5. Mail de confirmación

Al iniciar sesión de forma exitosa lo primero que se ve es la pantalla de bienvenida, con todas las opciones habilitadas (ver Figura 5.6). Como es un usuario administrador tiene acceso a todas las opciones disponibles en el sistema.



Figura 5.6. Página de bienvenida

Caso de prueba: C.P. 1 - Creación de usuario

En esta primera parte se muestran los resultados de la aplicación para el módulo de administración de usuarios. Se creó el usuario con la información de Sergio Damián Abate.

Para crear un usuario se selecciona la opción “Gestionar usuarios” en la interfaz de bienvenida (ver Figura 5.7). En la siguiente pantalla se observa el listado de los usuarios existentes y en la parte superior derecha se puede ver la opción “Nuevo” (ver Figura 5.8). Se selecciona esta opción y aparece el formulario para completar con los datos del usuario correspondiente (ver Figura 5.9).

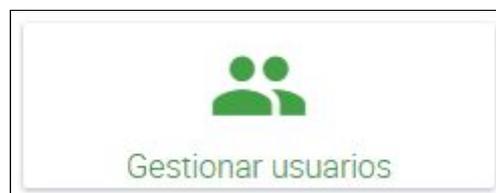


Figura 5.7. Gestionar usuarios

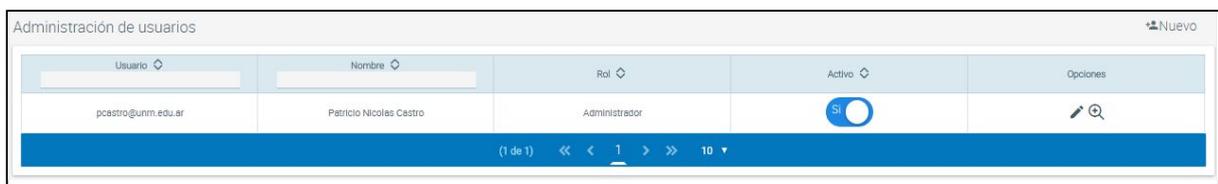


Figura 5.8. Administración de usuarios

The screenshot shows a web form titled "Crear usuario" with a "Listar" link in the top right corner. The form contains four input fields, each with an asterisk indicating it is required: "Nombre de usuario: *", "Nombre completo: *", "Lugar de trabajo: *", and "Rol: *". The "Rol" field is a dropdown menu with the text "Seleccionar Rol" and a downward arrow. At the bottom right of the form are two buttons: "Guardar" (green) and "Cancelar" (blue).

Figura 5.9. Formulario para crear usuario

Se completa el formulario con los datos del usuario. Para el caso antes mencionado (Figura 5.10):

- Nombre de usuario: sabate@unrn.edu.ar.
- Nombre completo: Sergio Damián Abate.
- Lugar de trabajo: FunBAPA.
- Rol: Laboratorista.

Al dar click en el botón “Guardar” se crea el usuario exitosamente. Una vez realizada dicha acción el sistema se redirige a la lista de usuarios actualizada con la información del nuevo usuario y se muestra el mensaje “Operación realizada con éxito” (Figura 5.11).

The screenshot shows the same "Crear usuario" form, but now the input fields are filled with data: "Nombre de usuario: *" contains "sabate@unrn.edu.ar", "Nombre completo: *" contains "Sergio Damián Abate", "Lugar de trabajo: *" contains "Fundación Barrera Zoonosanitaria Patagónica Funbapa", and the "Rol: *" dropdown menu is set to "Laboratorista". The "Guardar" and "Cancelar" buttons remain at the bottom right.

Figura 5.10. Completar formulario con los datos del usuario

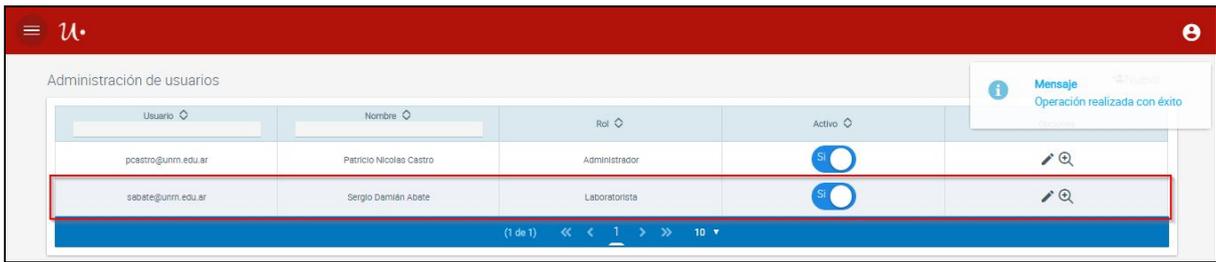


Figura 5.11. Usuario creado con éxito

Caso de prueba: C.P. 2 - Creación de apiario

En esta parte se muestran los resultados de la aplicación para el módulo de administración de apiarios. Se creó un apiario completando el formulario con la información correspondiente.

Para realizar esta tarea, se selecciona la opción “Gestionar apiarios” en la interfaz de bienvenida (ver Figura 5.12). En la siguiente pantalla se observa el listado de los apiarios existentes y en la parte superior derecha se puede ver la opción “Nuevo” (ver Figura 5.13). Se selecciona esta opción y se muestra el formulario para completar con los datos del apiario correspondiente (ver Figura 5.14).



Figura 5.12. Gestionar apiarios



Figura 5.13. Listado de apiarios

Figura 5.14. Formulario para crear apiario

Después se completa el formulario. En el caso realizado se completó con los datos del apiario (ver Figura 5.15):

- Nombre del apiario: Campo Inta - San Martín. Microscopio.
- Nombre del productor: Paola Andrea Pizzingrilli.
- Nombre del análisis: Caso de prueba varroas en el microscopio.
- Coordenadas (latitud, longitud): -41,96 / -71,53.
- Fecha del análisis: 30/08/2017.
- Cantidad de colmenas: 30.
- Tipo de análisis: Varroasis.

Luego, al dar click en el botón “Guardar” se crea el apiario de manera exitosa. Una vez realizada dicha acción el sistema se redirige a la lista de apiarios actualizada con la información del nuevo apiario y mostrando el mensaje “Operación realizada con éxito” (ver Figuras 5.16 y 5.17).

Mapa Satellite

Camino de la Cruz

Camino del Valle

Golondrinas

de los Pañiles

Google

Datos de mapas ©2017 Google Términos de uso Informar de un error de Maps

Nombre del apiario: *

Campo Inta - San Martín. Microscopio.

Nombre del productor: *

Paola Andrea Pizzingrilli.

Nombre del análisis: *

Caso de prueba varroas en el microscopio

Localización del apiario (Coordenadas satelitales):

Latitud: * -41,96 Longitud: * 71,53

Fecha del análisis: *

30/11/17

Cantidad de colmenas: *

30

Tipo de análisis: *

Nosemosis Varroasis

Guardar Cancelar

Figura 5.15. Completar formulario para crear apiario



Figura 5.16. Mensaje al crear un apiario con éxito

Administración de apiarios Nuevo

Nombre del apiario	Nombre del productor	Localización del apiario	Cantidad de colmenas	Tipo de análisis	Opciones
Campo Inta - El Bolson	Paola Andrea Pizzingrilli	-42.0 -72.0	30	Varroasis	
Varroas 2017	Paola Andrea Pizzingrilli	-42.0 -71.0	50	Varroasis	
Varroas 2017.2	Paola Andrea Pizzingrilli	-42.0 -71.0	30	Varroasis	

(1 de 1) << < 1 > >> 10 ▼

Figura 5.17. Listado con el nuevo apiario

Caso de prueba: C.P. 3 - Cargar imágenes en un apiario

En este caso se muestran los resultados de la aplicación para el módulo de administración de imágenes. Se cargaron las diferentes imágenes en el apiario anteriormente creado.

Para llevar a cabo esta funcionalidad, primeramente se selecciona la opción “Gestionar imágenes” en la interfaz de bienvenida (ver Figura 5.18). En caso de no tener un apiario seleccionado en la siguiente pantalla se observa el listado de los apiarios existentes y se selecciona en cual de estos se desea cargar las imágenes

(ver Figura 5.19). Si se tiene un apiario seleccionado o al seleccionar uno de estos aparece el listado de las imágenes cargadas en el mismo y en la parte superior derecha se puede ver la opción “Nuevo”, que se utiliza para cargar las imágenes (ver Figura 5.20).



Figura 5.18. Gestionar imágenes

Seleccionar un apiario

Nombre del apiario	Nombre del productor	Localización del apiario	Cantidad de colmenas	Tipo de análisis	Opción
Campo Inta - El Bolson	Paola Andrea Pizzingrilli	-42.0, -72.0	30	Varroasis	⊕
Varroas 2017	Paola Andrea Pizzingrilli	-42.0, -71.0	50	Varroasis	⊕
Varroas 2017.2	Paola Andrea Pizzingrilli	-42.0, -71.0	30	Varroasis	⊕

(1 de 1) << < 1 > >> 10 ▾

Figura 5.19. Listado de apiarios disponibles

Administración de imágenes para el apiario Campo Inta - El Bolson + Nuevo

Imagen	Archivo	Opciones
No se encontraron imágenes cargadas		

(1 de 1) << < > >> 10 ▾

Figura 5.20. Apiario sin imágenes

Para el caso de prueba, el listado de imágenes estaba vacío dado que aún no se habían cargado imágenes en el mismo. Luego de seleccionar la opción “Nuevo” aparece la pantalla de carga de imagen (ver Figura 5.21).

Alta de imagen Listar

+ Seleccionar imagen

Guardar
Cancelar

© Tesina de grado - Licenciatura en Sistemas
UNRN

Figura 5.21. Agregar imagen al apiario

A continuación se da click en el botón “Seleccionar imagen” y se visualiza una ventana que permite seleccionar desde la propia computadora que imagen se desea cargar (ver Figura 5.22). Se selecciona una imagen en particular, luego se presiona el botón “Abrir” y la imagen se carga de forma exitosa y aparece el mensaje lo confirma (ver Figura 5.23).

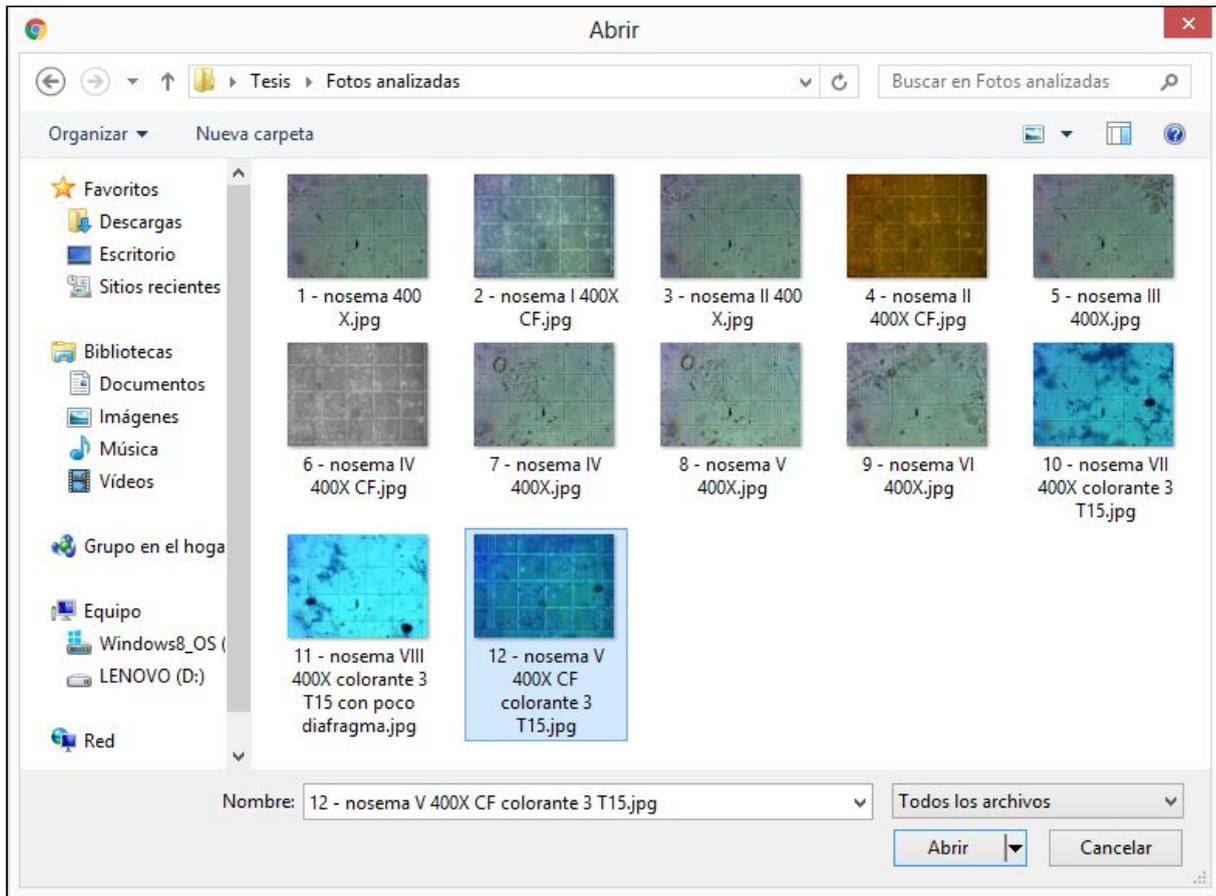


Figura 5.22. Seleccionar imagen desde la computadora



Figura 5.23. Imagen subida con éxito

Para la prueba se repitieron los pasos anteriormente mencionados y se cargaron todas las imágenes correspondientes en el apiario. El resultado de las imágenes cargadas se puede ver en la lista que se muestra a continuación (ver Figura 5.24).

Imagen	Archivo	Opciones
	1508500109081_12_-_nosema_V_400X_CF_colorante_3_T15.jpg	 
	1508500060018_6_-_nosema_IV_400X_CF.jpg	 
	1508500101982_11_-_nosema_VIII_400X_colorante_3_T15_con_poco_diafragma.jpg	 
	150850006337_1_-_nosema_400_X.jpg	 
	1508500045130_4_-_nosema_II_400X_CF.jpg	 
	1508500052191_5_-_nosema_III_400X.jpg	 
	1508500087178_9_-_nosema_VI_400X.jpg	 
	1508500068751_7_-_nosema_IV_400X.jpg	 
	1508500094843_10_-_nosema_VII_400X_colorante_3_T15.jpg	 

Figura 5.24. Listado de imágenes de un apiario para el análisis de nosemosis

Caso de prueba: C.P. 4 - Configuración del algoritmo

En este caso se muestran los resultados de la aplicación al momento de configurar el algoritmo. Se probó la configuración para la detección de esporas de nosemas en las imágenes de un apiario. Cabe aclarar que la configuración se realiza de la misma forma para la detección de nosemas y varroas; más adelante se muestra un caso de detección de configuración para la detección de varroas.

Para llevar adelante la configuración, en primer lugar se selecciona la opción “Configurar algoritmo” en la interfaz de bienvenida (ver Figura 5.25). En el caso de no tener ningún apiario seleccionado en la siguiente pantalla se observa el listado de los apiarios existentes y se debe seleccionar en cual de estos se desea configurar el algoritmo (ver Figura 5.26). Si se tiene un apiario seleccionado o al seleccionar uno de estos aparece el listado de las imágenes cargadas en el mismo y en este momento se selecciona una de las imágenes con las que se desea realizar la configuración (ver Figura 5.27).



Figura 5.25. Configurar algoritmo

Seleccionar un apiario

Nombre del apiario	Nombre del productor	Localización del apiario	Cantidad de colmenas	Tipo de análisis	Opción
Uno	Sergio Damian Abatte	Carmen de Patagonas	100	Nosemosis	<input type="radio"/>
Campo Inta - El Bolson	Paola Andrea Pizzingrilli		30	Varroasis	<input type="radio"/>
Varrosas 2017	Paola Andrea Pizzingrilli	El Bolson	50	Varroasis	<input type="radio"/>
Aplario UNRN - Campo INTA Nosema	Paola Andrea Pizzingrilli	El Bolson	1	Nosemosis	<input type="radio"/>
Pisos Trampa	Paola Andrea Pizzingrilli	El Bolson	20	Nosemosis	<input type="radio"/>
Varrosas 2017.2	Paola Andrea Pizzingrilli	El bolson	30	Varroasis	<input type="radio"/>
Aplario UNRN - Campo INTA	Paola Andrea Pizzingrilli	El Bolson	1	Varroasis	<input type="radio"/>

(1 de 1) << < 1 > >>

Figura 5.26. Listado de apiarios a seleccionar

Seleccionar imagen

Imagen	Archivo	Opción
	1508500109081_12_-_nosema_V_400X_CF_colorante_3_T15.jpg	<input type="radio"/>
	1508500060018_5_-_nosema_IV_400X_CF.jpg	<input type="radio"/>
	1508500101982_11_-_nosema_VIII_400X_colorante_3_T15_con_poco_diafragma.jpg	<input type="radio"/>
	150850006337_1_-_nosema_400_X.jpg	<input type="radio"/>
	1508500045130_4_-_nosema_II_400X_CF.jpg	<input type="radio"/>
	1508500052191_5_-_nosema_III_400X.jpg	<input type="radio"/>
	1508500087178_9_-_nosema_VI_400X.jpg	<input type="radio"/>
	1508500068751_7_-_nosema_IV_400X.jpg	<input type="radio"/>
	1508500094843_10_-_nosema_VII_400X_colorante_3_T15.jpg	<input type="radio"/>
	1508500078048_8_-_nosema_V_400X.jpg	<input type="radio"/>

(1 de 2) << < 1 2 > >> 10 ▼

Figura 5.27. Listado de imágenes del apiario seleccionado

A continuación se muestra la pantalla de “Configuración para la detección de nosemosis” en la parte superior de la misma aparecen los parámetros del algoritmo, y en la parte inferior la imagen original seleccionada -izquierda- y la imagen resultante al aplicar dichos parámetros donde se identifican y etiquetan cada una de las esporas -derecha- (Figura 5.28).

Para el caso de prueba que se incluye, los parámetros y la configuración elegida fue:

- Tamaño del bloque: 43.
- Constante de desvío: 9.
- Área del objeto: 30.
- Largo mínimo del objeto: 50.
- Largo máximo del objeto: 100.

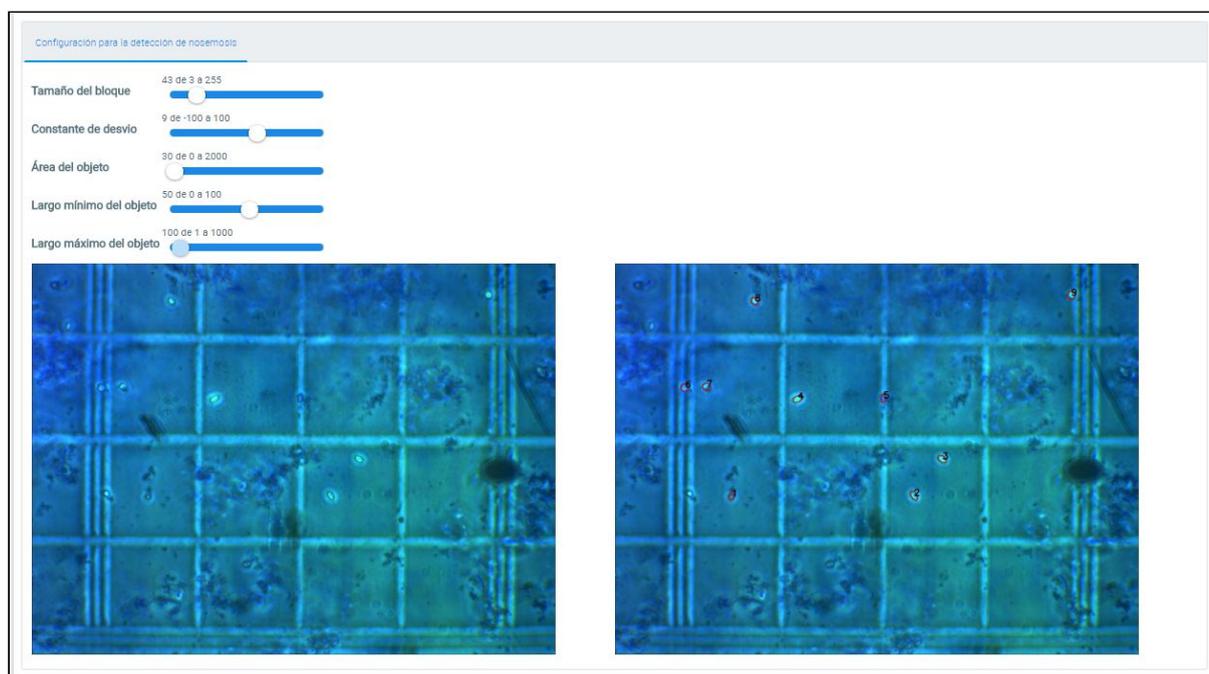


Figura 5.28. Configuración del algoritmo para la detección de esporas de nosemas

Caso de prueba: C.P. 5 - Detección y conteo de nosemas

Este caso de prueba se realizó en conjunto con el experto Sergio Damian Abate, quien se encargó de realizar el conteo manual y corroborar los resultados. Después de crear el apiario, cargar las imágenes y configurar el algoritmo para la detección de nosemas se procesaron las imágenes. Para esta prueba se cargaron 12 imágenes con diversa calidad y cantidad de nosemas. Las mismas fueron obtenidas en una sesión de captura por el laboratorio FunBAPA. Los resultados y validación de dicho procesamiento se muestran a continuación junto a la comparación del tiempo para el análisis de nosemosis con el método manual y el automático.

Se completó cada uno de los campos del formulario con los datos del apiario de la siguiente forma (Figura 5.29):

- Nombre del apiario: FUNBAPA - Carmen de Patagones.
- Nombre del productor: Sergio Damian Abatte.
- Nombre del análisis: Primer control de nosemosis.
- Coordenadas (latitud, longitud): -40,78 / -62,96.
- Fecha del análisis: 01/08/2017.
- Cantidad de colmenas: 100.
- Tipo de análisis: Nosemosis.

Nombre del apiario: *
Carmen de Patagones

Nombre del productor: *
Sergio Damian Abatte

Nombre del análisis: *
Control de nosemosis

Localización del apiario (Coordenadas satelitales):
Latitud: * -41,00 Longitud: * -63,00

Fecha del análisis: *
05/09/17

Cantidad de colmenas: *
100

Tipo de análisis: *
 Nosemosis Varroasis

Guardar Cancelar

Figura 5.29. Crear apiario para la detección de nosemas

El procesamiento de las imágenes se realiza desde la opción “Gestionar imágenes”. En la parte inferior del listado de imágenes aparece el botón “Procesar imágenes para detectar nosemas” (ver Figuras 5.30 y 5.31).



Figura 5.30. Botón para procesar imágenes para detectar nosemas

	150850006337_1_-_nosema_400_X.jpg	 
	1508500045130_4_-_nosema_II_400X_CF.jpg	 
	1508500052191_5_-_nosema_III_400X.jpg	 
	1508500087178_9_-_nosema_VI_400X.jpg	 
	1508500068751_7_-_nosema_IV_400X.jpg	 
	1508500094843_10_-_nosema_VII_400X_colorante_3_T15.jpg	 
	1508500078048_8_-_nosema_V_400X.jpg	 

(1 de 2) << < 1 2 >> >> 10 ▾

Procesar imágenes para detectar nosemas

Figura 5.31. Listado de imágenes a procesar para la detección de nosemas

Al realizar esta acción aparece una imagen que muestra que se está realizando el procesamiento (ver Figura 5.32). Al finalizar desaparece dicha imagen y aparece el mensaje de “Operación realizada con éxito”.



Figura 5.32. Procesando imágenes

Los resultados se ven desde la opción “Gestionar resultados” en la pantalla de bienvenida (ver Figura 5.33). Al seleccionar dicha opción en la siguiente pantalla se observa el listado de las imágenes resultantes. Para cada imagen se indica el número de esporas detectadas y para cada espora detectada se muestra información como la etiqueta, el alto, el ancho, el área y el perímetro (ver Figura 5.34). Al final del listado se puede observar el número total de esporas detectado para todo el apiario, es decir, la suma de lo detectado en cada imagen que este caso resulto 552 (ver Figura 5.35).



Figura 5.33. Gestionar resultados

Etiqueta	Alto	Ancho	Área	Perímetro
1	56,33	37,63	1378,5	167,81
4	76,45	41,59	2484,0	189,99
6	54,82	51,73	40,0	653,30
3	67,07	43,37	2280,5	175,92
5	61,49	39,02	1847,5	165,95
8	59,62	44,27	2048,5	166,10
7	55,96	48,38	941,0	265,59
2	70,88	41,66	2311,0	183,17
9	59,19	49,94	927,5	277,04

Cantidad: 9

(1 de 1) << < 1 > >> 10 ▾

Figura 5.34. Resultados de la detección de nosomes para imagen seleccionada

Total encontrado: 552

Figura 5.35. Total de esporas encontradas

Comparación de la eficiencia para el análisis de nosis entre ambos métodos

Una vez que se llevaron a cabo todas las etapas, se validó la eficiencia del método propuesto, comparando los resultados del conteo de forma manual (método actualmente utilizado que se consideró como prueba de referencia o “gold standard”) con el conteo automático.

A continuación, se muestran los resultados obtenidos con uno de los casos estudiados (caso número 12), caracterizado por una diferencia mínima entre el número de esporas identificadas aplicando el método automático (Figura 5.36) y el método manual (Figura 5.37).

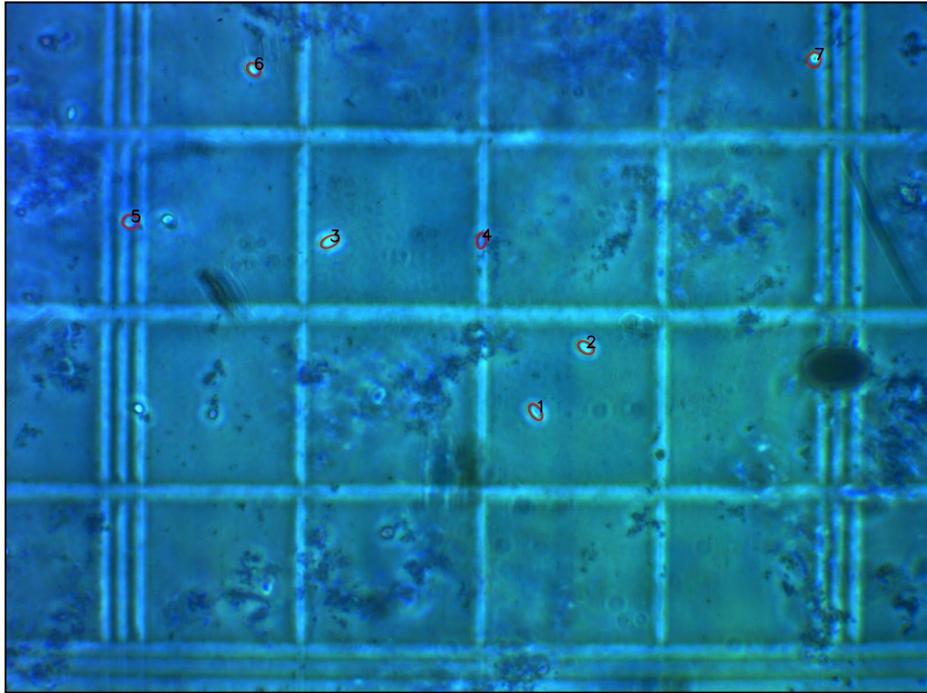


Figura 5.36. Imagen con las esporas etiquetadas aplicando el método automático

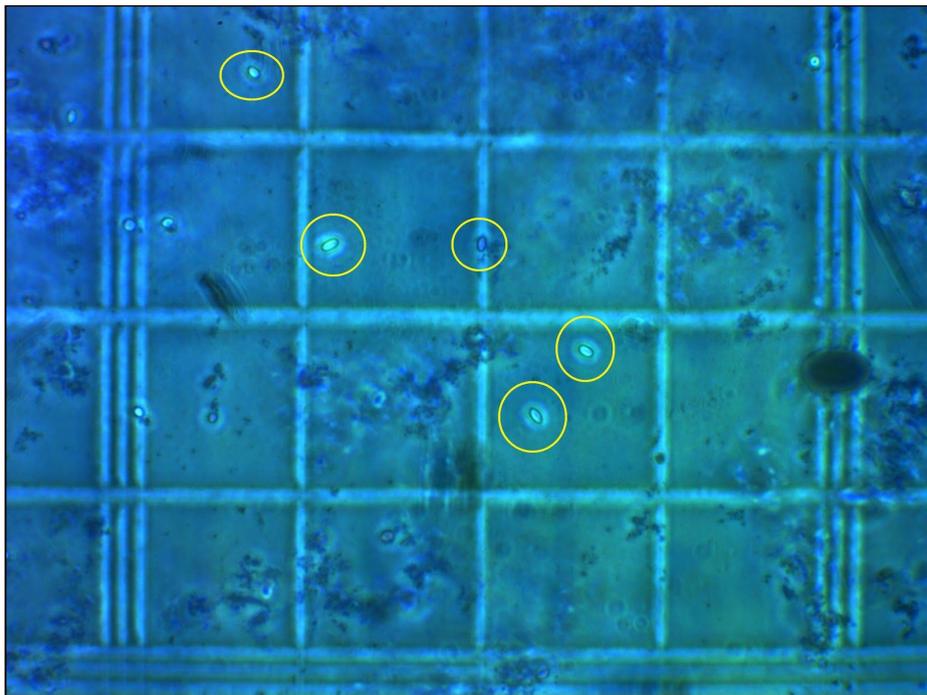


Figura 5.37. Imagen con las esporas etiquetadas aplicando el método manual

Del mismo modo, se muestran los resultados obtenidos con otro de los casos (caso número 8), en el que se observó la mayor diferencia entre el número de esporas

identificadas aplicando el método automático (ver Figura 5.38) y el método manual (ver Figura 5.39).

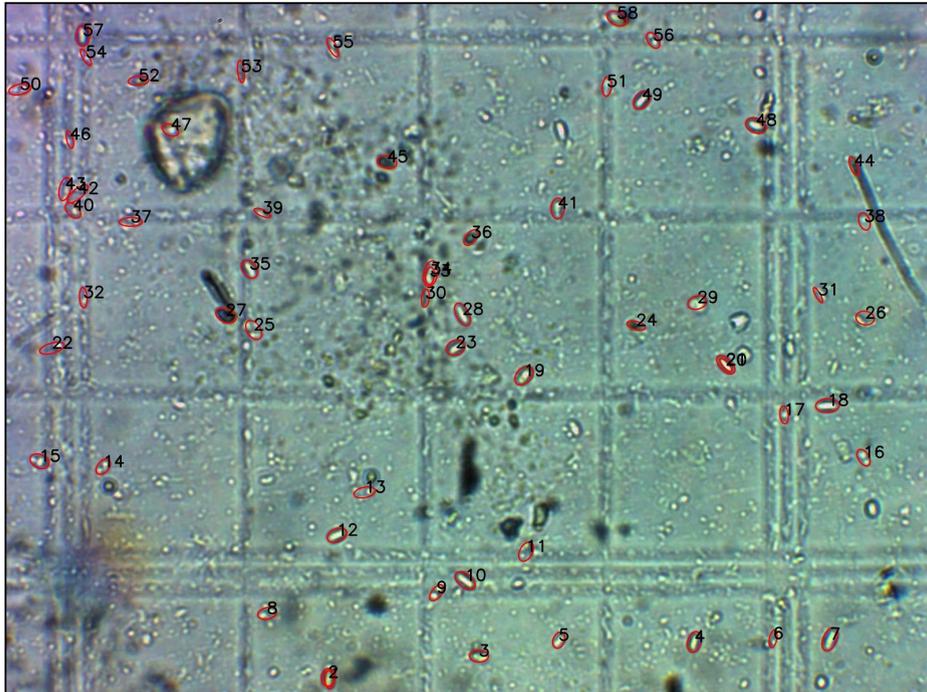


Figura 5.38. Imagen con las esporas etiquetadas aplicando el método automático

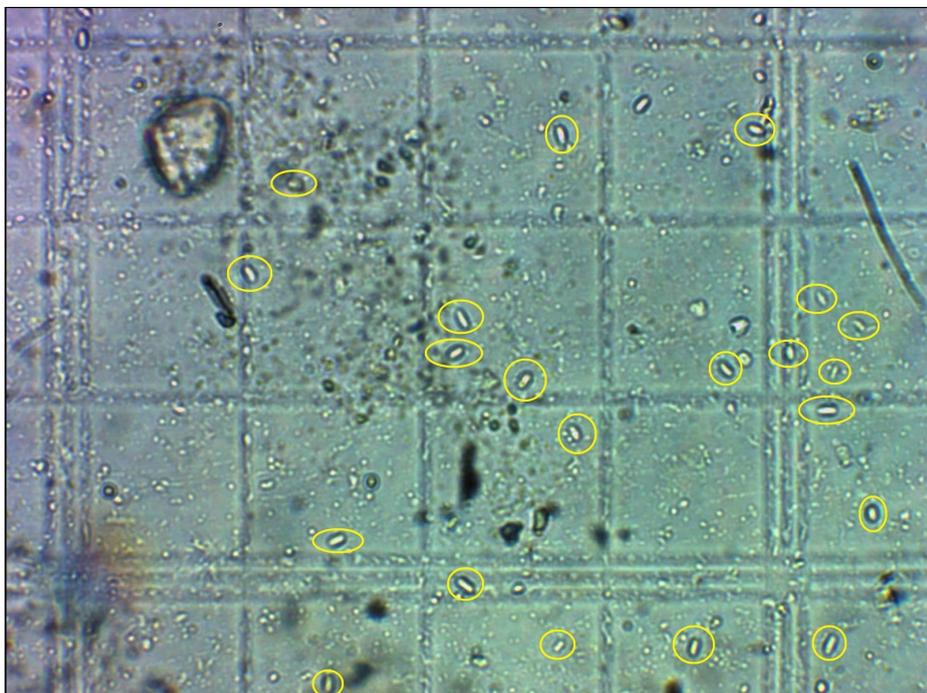


Figura 5.39. Imagen con las esporas etiquetadas aplicando el método manual

Los resultados de los 12 casos tratados (imágenes), comparando identificación y recuento por ambos métodos, se muestran en la Figura 5.40.

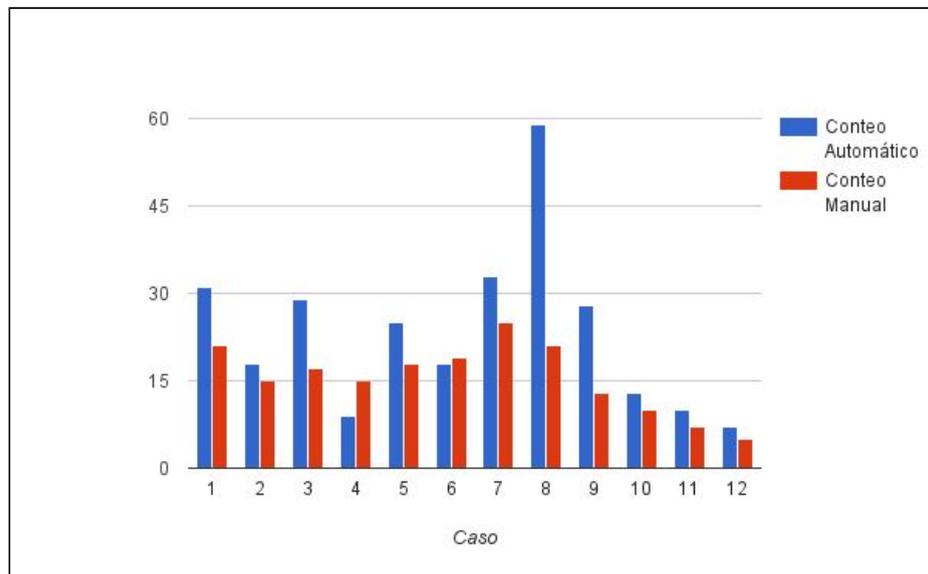


Fig. 5.40. Comparación de recuentos de esporas de Nosema spp por método automático y manual

Comparación del tiempo para el análisis de nosemosis entre ambos métodos

Para realizar esta comparación, se midió el tiempo empleado para identificar y contar las esporas de nosemas en cada una de las 12 muestras, utilizando el minuto como unidad de tiempo. Los resultados se muestran en la Tabla 5.1.

Caso	Tiempo con método automático	Tiempo con método manual
1	0,1	3,8
2	0,1	4
3	0,1	4,3
4	0,1	3,5
5	0,1	4,4
6	0,1	4,7
7	0,1	4,8
8	0,1	5
9	0,1	5,1
10	0,1	4,3
11	0,1	3,9
12	0,1	3,5
Tiempo Total	1,2	51,3

Tabla 5.1. Comparación del tiempo insumido entre el método manual y el automático para la detección de nosemas

Caso de prueba: C.P. 6 - Detección y conteo de varroas

Imágenes tomadas al microscopio

Para este caso se creó un apiario para la detección de varroas en el que se cargaron ocho imágenes tomadas desde el microscopio (ver Figura 5.41). Se eligió una de estas para configurar el algoritmo (ver Figura 5.42) y se configuró el algoritmo con los siguientes parámetros (ver Figura 5.43):

- Tamaño del bloque: 111.
- Constante de desvío: 8.
- Área del objeto: 410.
- Largo mínimo del objeto: 30.
- Largo máximo del objeto: 110.

Imagen	Archivo	Opciones
	1507692159935_20170914_191248.jpg	 
	1507692123021_20170914_191006.jpg	 
	1507692150043_20170914_191227.jpg	 
	1507692140969_20170914_191208.jpg	 
	1507692100048_20170914_190926.jpg	 
	1507692177527_20170914_191324.jpg	 
	1507692108525_20170914_190940.jpg	 
	1507692166723_20170914_191253.jpg	 

(1 de 1) << < 1 > >>

Procesar imágenes para detectar varroas

Figura 5.41. Listado de imágenes a procesar para la detección de varroas

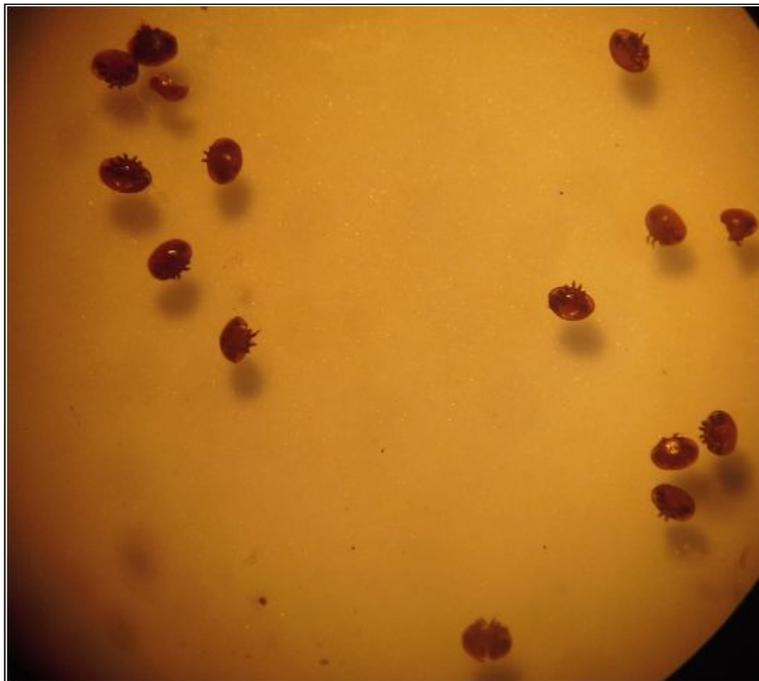


Figura 5.42. Imagen elegida para configurar el algoritmo

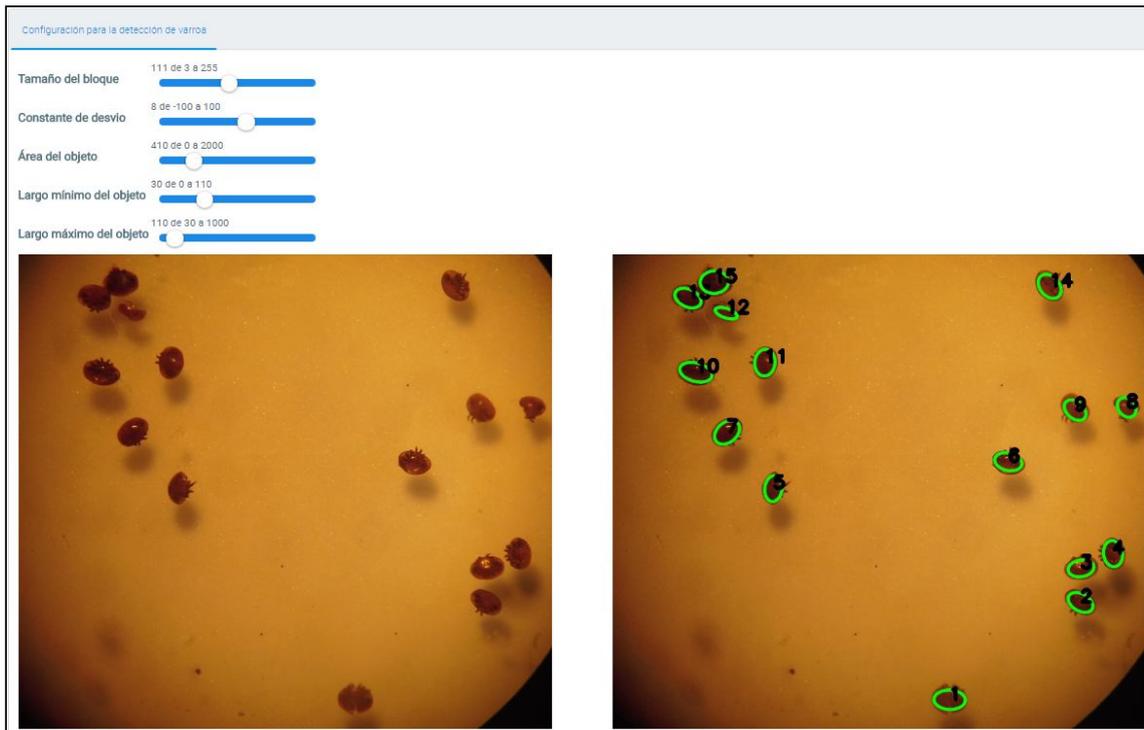


Figura 5.43. Configuración del algoritmo para la detección de varroas

En la Figura 5.44 se puede ver la imagen resultante de la configuración. Como para la detección de noseemas el procesamiento de las imágenes para la detección de varroas se realiza desde la opción “Gestionar imágenes”. En la parte inferior del listado de imágenes aparece el botón “Procesar imágenes para detectar varroas” (ver Figura 5.45).

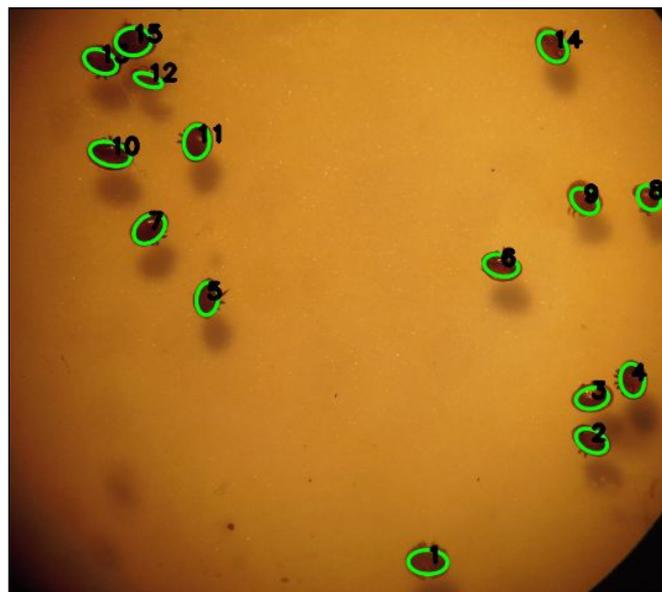


Figura 5.44. Detalle de la imagen resultante



Figura 5.45. Botón para procesar imágenes para detectar varroas

Al finalizar el procesamiento de manera exitosa los resultados se ven desde la opción "Gestionar resultados" en la pantalla de bienvenida. Al seleccionar esta opción se observa el listado de las imágenes resultantes, en cada imagen se indica el número de esporas detectadas y al final el número de esporas para todo el apiario. A su vez de cada imagen se muestra información como la etiqueta, el alto, el ancho, el área y el perímetro de cada espora de nosema detectada (ver Figura 5.46). Al final del listado se puede observar el número total detectado, la suma de todas las imágenes, que en este arrojo 117 (ver Figura 5.47).

Etiqueta	Alto	Ancho	Área	Perímetro
2	46,40	29,52	1057.0	123,57
4	45,50	32,41	1130.5	129,64
10	56,72	31,89	1371.0	162,68
8	36,86	29,03	720.5	96,98
1	53,46	33,96	1303.0	159,34
3	44,73	28,82	711.0	149,54
6	47,53	30,40	1067.0	143,71
9	42,56	28,72	847.5	133,64
11	46,54	35,01	1272.5	136,12
7	48,66	34,22	1295.0	132,05

Figura 5.46. Resultados de la detección de varroas para la imagen seleccionada



Figura 5.47. Total de varroas encontradas

Comparación de la eficiencia para el análisis de varroasis entre ambos métodos

Aquí también se validó la eficiencia del método propuesto, comparando los resultados del conteo de forma manual con el conteo automático.

A continuación, se muestran los resultados obtenidos con las imágenes del caso probado. Los resultados de las 8 imágenes, comparando identificación y recuento por ambos métodos se pueden ver en forma resumida en la siguiente tabla (ver Tabla 5.2) y gráficamente en la imagen Figura 5.48.

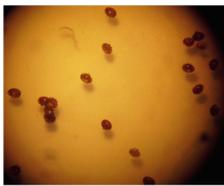
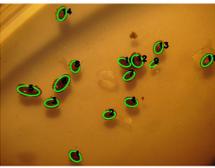
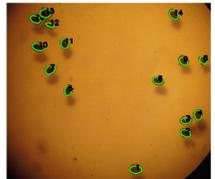
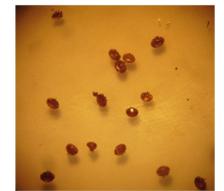
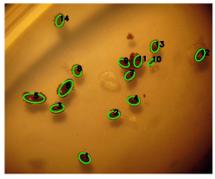
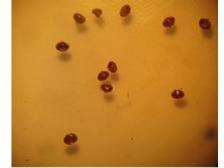
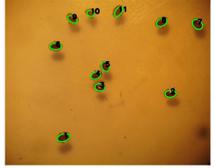
Caso	Original	Resultado	Conteo automático	Conteo manual
1			17	19
2			14	16
3			15	15
4			15	15
5			14	16
6			11	11
7			14	16
8			17	19
Total			117	127

Tabla 5.2. Cuadro comparativo entre el método de detección manual y automático para la detección de varroas

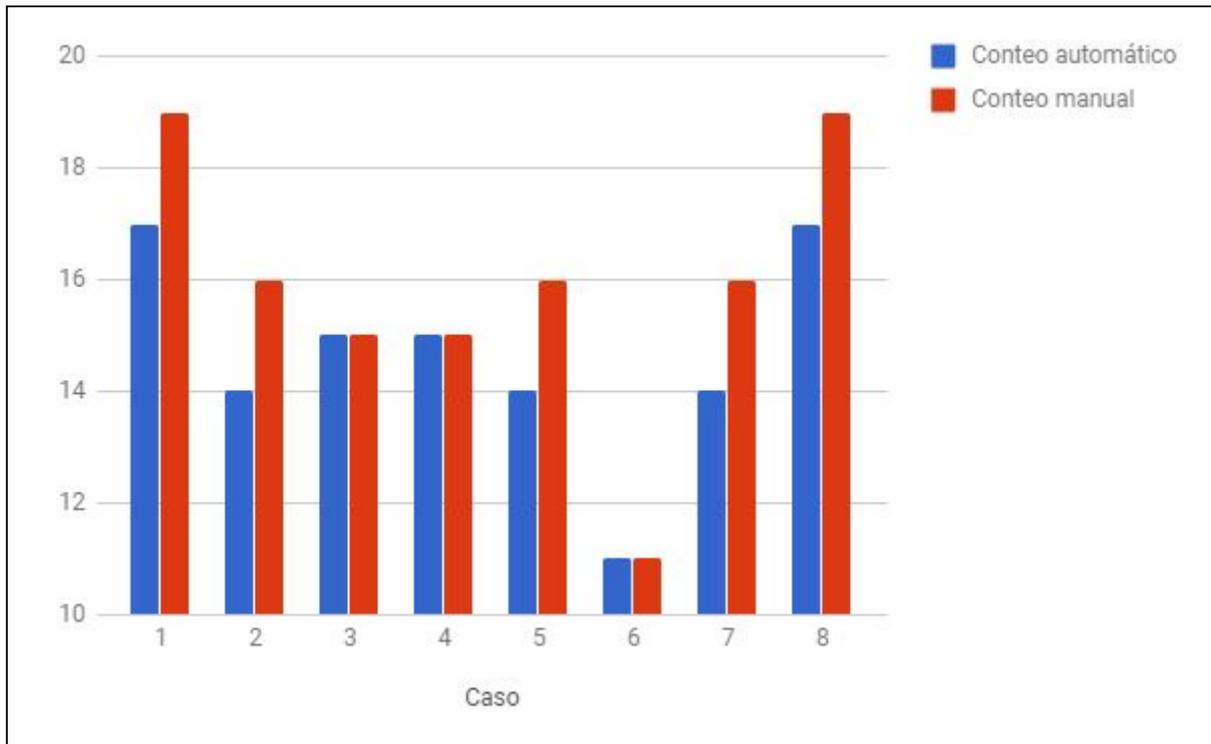


Figura 5.48. Gráfica comparativa de recuentos de ácaros de varroas entre método automático y manual

Comparación del tiempo para la detección de varroas entre ambos métodos

Para esta comparación se midió el tiempo empleado para identificar y contar los ácaros de varroas en cada una de las 8 imágenes, utilizando el minuto como unidad de tiempo. Los resultados se muestran en la Tabla 5.3.

Caso	Tiempo con método automático	Tiempo con método manual
1	0,05	0,3
2	0,05	0,3
3	0,05	0,3
4	0,05	0,3
5	0,05	0,3
6	0,05	0,3
7	0,05	0,3
8	0,05	0,3
Tiempo Total	0,4	2,4

Tabla 5.3. Comparación del tiempo insumido entre el método manual y el automático para la detección de varroas

Imágenes tomadas a los pisos trampa

Para este caso se realizó una prueba de detección de varroas en los pisos trampas en lugar del microscopio. Se tomaron imágenes de un apiario ubicado en el Campo Forestal General San Martín que está localizado en el paraje Las Golondrinas en la provincia de Chubut. Estos apiarios son administrados por INTA Bariloche.

Se creó el apiario dentro de la aplicación con la información correspondiente y se le cargaron las imágenes (ver Figuras 5.49 y 5.50). Se completó cada uno de los campos del formulario con los datos del apiario de la siguiente manera:

- Nombre del apiario: Campo Inta - San Martín.
- Nombre del productor: Paola Andrea Pizzingrilli.
- Nombre del análisis: Caso de prueba varroasis.
- Coordenadas (latitud, longitud): -41,96 / -71,53.
- Fecha del análisis: 30/08/2017.
- Cantidad de colmenas: 30.
- Tipo de análisis: Varroasis.

The screenshot shows a web application interface for creating a hive record. On the left is a map showing the location of the hive in Golondrinas, Chubut. On the right is a form with the following fields:

- Nombre del apiario: * Campo INTA - San Martín
- Nombre del productor: * Paola Andrea Pizzingrilli
- Nombre del análisis: * Caso de prueba varroasis
- Localización del apiario (Coordenadas satelitales):
Latitud: * -42,00 Longitud: * -71,00
- Fecha del análisis: * 04/10/17
- Cantidad de colmenas: * 30
- Tipo de análisis: *
 Nosemosis Varroasis

At the bottom right of the form are two buttons: 'Guardar' (green) and 'Cancelar' (blue).

Figura 5.49. Crear apiarios para detectar varroas en los pisos trampa

Imagen	Archivo	Opciones
	1508626203947_IMG-20170917-WA0026.jpg	 
	1508626224073_IMG-20170917-WA0028.jpg	 
	1508626110356_IMG-20170914-WA0062.jpg	 
	1508626129717_IMG-20170914-WA0066.jpg	 
	1508626142010_IMG-20170917-WA0000.jpg	 
	1508626173710_IMG-20170917-WA0025.jpg	 
	1508626119358_IMG-20170914-WA0063.jpg	 
	1508626213999_IMG-20170917-WA0027.jpg	 
	1508626163503_IMG-20170917-WA0023.jpg	 
	1508626151268_IMG-20170917-WA0021.jpg	 

(1 de 1) << < 1 >> >>

Figura 5.50. Listado de imágenes de los pisos trampa

Luego de elegir una de las imágenes para configurar los parámetros del algoritmo se realizó el procesamiento para la detección de varroas. La configuración utilizada fue (Figura 5.51):

- Tamaño del bloque: 111.
- Constante de desvío: 40.
- Área del objeto: 410.
- Largo mínimo del objeto: 80.
- Largo máximo del objeto: 110.

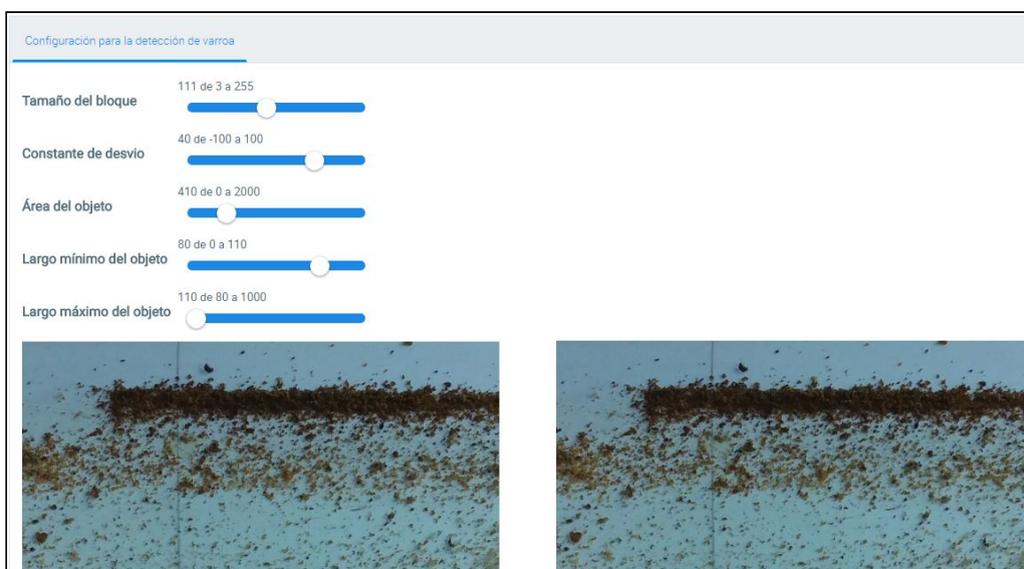


Figura 5.51. Crear apiarios para detectar varroas en los pisos trampa

Los resultados arrojaron que no se detectó ningún ácaro en el conjunto de imágenes de prueba (ver Figura 5.52 y Figura 5.53). Este resultado es válido y la experta Paola Pizzingrilli constató que no habían ácaros en las imágenes.

Imagen	Objetos detectados				
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					

Figura 5.52. Detección de varroas en los pisos trampa

Imagen	Objetos detectados				
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < 1 > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					
	Etiqueta <input type="text"/>	Alto	Ancho	Área <input type="text"/>	Perímetro: <input type="text"/>
Cantidad: 0	No se detectaron objetos en esta imagen				
(1 de 1) << < > >> 10 ▼					

Figura 5.53. Listado de resultados

En este caso no se muestra la tabla comparativa entre el conteo de los métodos de detección manual y automático, sin embargo, se muestra la tabla comparativa respecto al tiempo (ver Tabla 5.4).

Caso	Tiempo con método automático	Tiempo con método manual
1	0,06	0,3
2	0,06	0,3
3	0,06	0,3
4	0,06	0,3
5	0,06	0,3
6	0,06	0,3
7	0,06	0,3
8	0,06	0,3
9	0,06	0,3
10	0,06	0,3
Tiempo Total	0,6	2,4

Tabla 5.4. Comparación del tiempo insumido entre el método manual y el automático para la detección de varroas

Capítulo 6: Conclusiones y líneas futuras

Este trabajo se centró en la detección automática de dos enfermedades que se encuentran en las abejas melíferas, que son la varroosis y la nosemosis utilizando técnicas de visión artificial. Por un lado se presentó un método para la identificación y conteo de esporas de nosemas de forma automática en imágenes microscópicas basado en su forma y tamaño. Por otro lado un método para realizar la misma tarea pero para identificar automáticamente ácaros de varroas en una imagen basados en las mismas características morfológicas.

Para la detección de nosemas en 11/12 casos estudiados (aproximadamente el 92%), los resultados obtenidos al comparar ambos métodos fueron similares (ver Figura 5.37). En 1/12 casos estudiados (aproximadamente el 8 %) se observó una diferencia significativa entre los resultados obtenidos por ambos métodos (ver caso número 8 en la Figura 5.38). En este caso, el método automático identificó 59 esporas, de las cuales 38 fueron falsos positivos al comparar con el método manual que identificó sólo 21 esporas. Se asumió que el método automático identificó falsos positivos debido a que el método manual fue considerado como método de referencia. Este criterio se confirmó aplicando el método manual a la imagen del caso 8 procesada por el método automático, y confirmando que el mismo contabilizó como esporas a objetos diferentes.

La imagen del caso número 8 se obtuvo luego de realizar un procesamiento estándar y mínimo de la muestra: mortereado de abejas, suspensión en agua usando un volumen en mililitros igual al número de abejas mortereadas. Esta suspensión se homogeneizó y se sometió a observación microscópica, para obtener la imagen digital final sin aplicar procesamientos adicionales tendientes a eliminar cualquier ruido de fondo sobre la imagen de la muestra, conforme recomendaciones estandarizadas por la Organización Mundial de Sanidad Animal (OIE, 2013). En este tipo de imágenes, el ojo humano debe realizar un gran esfuerzo para discernir, reconocer y contar las esporas de nosemas. Evidentemente, en este tipo de

situaciones, el cerebro humano pone en juego simultáneamente muchos otros criterios además del reconocimiento de tamaño y forma, para discernir entre un caso positivo (espora) de un caso negativo (no espora), como el brillo, el aspecto de la superficie de la espora, la homogeneidad de la refringencia, etc. Dado que hasta el momento el diagnóstico de nosemosis se realiza por observación microscópica mediante visión humana, se comprende que los procedimientos de procesamiento de muestras de abejas no hayan reparado en la necesidad de mejorar los protocolos de procesamiento para obtener imágenes con menos factores de confusión, con el fondo más claro y libre de objetos que no sean esporas de nosemas, situación que con el uso de la visión artificial se ha identificado como un problema. En el resto de los casos, antes de obtener las imágenes digitales se realizaron procesamientos adicionales a las recomendaciones de la OIE, que mejoraron la calidad de las mismas. Podemos concluir que el método automatizado que se describe en este trabajo, conforme la parametrización aplicada, es de utilidad sólo para imágenes con bajo nivel de ruido de fondo, y es aplicable para muestras de abejas procesadas con pasos adicionales a los sugeridos por la OIE.

El método automático se mostró significativamente más veloz que el método manual: para la observación de 12 muestras, consumió 50 veces menos tiempo (ver Tabla 5.1). Asumiendo que en 1,2 minutos el método pudo procesar 12 muestras, en una jornada de 8hs de trabajo el método sería capaz de procesar al menos 4.800 muestras, mientras que el método manual solo podría procesar 90. Considerando apiarios promedio de 100 colmenas, en un día de trabajo el método automático sería capaz de procesar las muestras de todas las colmenas de 48 apiarios, mientras que el método manual no alcanzaría a procesar la totalidad de las colmenas de un solo apiario. Si este análisis se traduce en costo económico del análisis de nosemosis, se puede pensar en que el método automatizado podría disminuirlo en 50 veces la fracción del costo debida a gastos por recursos humanos vinculados al análisis (horas de trabajo). Estos dos aspectos (mayor rapidez y menor costo) hacen del método automatizado una herramienta aplicable para estudios epidemiológicos a escala provincial o nacional.

La detección de varroas se probó tomando imágenes tanto del microscopio como de los pisos trampa. Para el primer caso el porcentaje de acierto es mayor al 92%. En aquellos casos en que los ácaros de varroa se encontraban superpuestos no se detectaron de manera correcta haciendo que éstos se contabilicen en forma conjunta en lugar de separados que hubiese sido lo adecuado. Para el segundo se comprobó que los apiarios no se encontraban infestados y se mostró la diferencia entre el tiempo insumido entre ambos métodos. De todos modos, faltaría probar aquellos casos en que los pisos trampa contienen ácaros, en esta época del año no fue posible obtener imágenes con varroas en los apiarios estudiados.

En ambos casos este trabajo se realizó con un laboratorista para cada caso. Debiera tenerse en cuenta que el método manual es susceptible de subjetividad del laboratorista, por esta razón es esperable que se presenten variaciones en los resultados para la misma muestra, cuando ésta es analizada por diferentes personas. Al respecto, se prevé realizar una validación de método incluyendo la participación de diversos laboratorios con muestras de las mismas colmenas, para confirmar la fortaleza del método automático con respecto a la repetibilidad de resultados y unificación de criterio de discriminación. Como trabajos futuros, además, se tiene previsto la optimización del método automático mejorando cada una de las etapas, en especial la segmentación y los filtros en la detección de contornos y conteo de elipses.

Se deberá identificar además a qué colmena pertenece cada imagen y a partir de ello calcular estadísticamente el nivel de incidencia de cada enfermedad por colmena, la necesidad de tratamiento antimicrobiano, y la eficiencia de dicho tratamiento realizando análisis posteriores. El método manual (actualmente utilizado) no cuenta con un registro inequívoco de resultados al que se pueda recurrir en aquellos casos de conflicto entre el propietario del apiario y los laboratorios de diagnóstico, cuando los resultados de análisis no correspondan con el criterio del propietario del apiario en casos de compra - venta, así como movimiento de colmenas en el marco de la trashumancia, cada vez más frecuente ante el cambio del ecosistema agrícola en respuesta a la fuerte sojización y pérdida

de zonas con floración aptas para apicultura. El resguardo de las imágenes ofrece una solución a este problema, sumando trazabilidad al diagnóstico de ambas enfermedades, de aplicación para los controles sanitarios de organismos oficiales.

Para contar las varroas en los pisos trampa sería mejor realizar la detección en el mismo momento que se visitan los apiarios, por lo tanto en un futuro sería mejor contar con una aplicación móvil que haga dicha tarea. En cuanto al método se buscará generalizarlo para la detección de otras enfermedades similares, en abejas u otras especies, basados en las características de forma y tamaño del agente etiológico.

Finalmente se destaca, que durante el desarrollo de esta tesina se realizó la siguiente publicación: Castro, P.; Cambarieri, M.; Abate S.; Britos, P.; Vivas, H. 2016. *Identificación automática de nosemosis en imágenes microscópicas*. 8° Congreso de AgroInformática (CAI 2016), pág. 94-105, correspondiente a la 45° Jornadas Argentinas de Informática (JAIIO 2016). Buenos Aires. Argentina. ISSN: 2525- 0949. Web: <http://45jaiio.sadio.org.ar/node/86> (Vigente a Noviembre 2017).

Capítulo 7: Bibliografía

- Alvarez-Ramos, C. M., Nino, E., & Santos, M. (2013). *Automatic Classification of Nosema Pathogenic Agents through Machine Vision techniques and Kernel-based Vector Machine*. In 2013 8th Computing Colombian Conference (8CCC).
- Alvarez-Ramos, C. M., Peñas, M. S., & Meana, A. (2010). *Sistema de reconocimiento y clasificación de agentes patógenos de Nosemosis*. Trabajo de Fin de Máster. Universidad Complutense de Madrid
- Atlassian. (n.d.). Gitflow Workflow | Atlassian Git Tutorial. Consultado el 22 octubre del 2017, desde <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- Baggio, D. L. (2015). *OpenCV 3.0 Computer Vision with Java*. Packt Publishing Ltd.
- Cardozo, M. M. (2014). *Manejo do apiário localizado no complexo da cidade das abelhas—Florianópolis—SC*.
- CONASA. Recomendación para el control de varroasis (2010). Consultado el 30 de agosto del 2017, desde http://www.senasa.gob.ar/sites/default/files/ARBOL_SENASA/ANIMAL/ABEJAS/PROD_PRIMARIA/SANID_APICOLA/EES/VAROOSIS/recomendaciones_para_el_control_de_la_varroosis_20101.pdf
- D. T. OpenCV. Structural Analysis and Shape Descriptors. (n.d. -b). Consultado el 15 de abril de 2016 desde http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#fitzgibbon95. OpenCV 2.4.13.0 documentation.
- D. T. OpenCV. Structural Analysis and Shape Descriptors. (n.d. -b). Consultado el 15 de abril de 2016 desde http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#suzuki85. OpenCV 2.4.13.0 documentation.
- Driessen, V. (2010). A successful Git branching model. Consultado el 3 Junio de 2017, desde <http://nvie.com/posts/a-successful-git-branching-model/>

- Fitzgibbon, A., & Fisher, R. (1995). *A Buyer's Guide to Conic Fitting. Proc. of the British Machine Vision Conference 1995*. 513-22
- Fries, I. (2010, 01). *Nosema ceranae in European honey bees (Apis mellifera)*. *Journal of Invertebrate Pathology*, 103. S73–S79. doi:10.1016/j.jip.2009.06.017
- González, A. (2013). *Sistema de gestión de calidad en un establecimiento de extracción y fraccionado de miel* (Bachelor's thesis).
- Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing 3rd edition*. Prentice-Hall
- Hervias, D; Mogni, F (2005) El auge de la apicultura en la Argentina: breve introducción al tema. *Apuntes Agroeconómicos* (Facultad de Agronomía, Universidad de Buenos Aires). Consultado el 1 de Mayo del 2017, desde https://www.agro.uba.ar/apuntes/no_3/apicultura.htm
- Hibernate. Hibernate ORM. (n.d.). Consultado el 21 de abril de 2017, desde <http://hibernate.org/orm/>
- Lugani, C., Cátedra de “Sistemas y organizaciones,” E. (2015). Varios entornos de desarrollo: desarrollo, integración, preproducción y producción. Consultado el 3 de Junio de 2017 desde <http://hdl.handle.net/20.500.12049/171>
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson Education.
- Meana, A.; Martín, R.; Higes, M. (2009) *La silenciosa pandemia de las abejas*. M. Investigación y ciencia. Edición Española de Scientific American, págs 8-10.
- OIE - *World Organisation for Animal Health. Manual de la OIE* (2013). Capítulo 2.2.4. Nosemosis de las abejas melíferas. Consultado el 10 de mayo de 2016 desde http://www.oie.int/fileadmin/Home/esp/Health_standards/tahm/2.02.04_NOSE_MOSIS_FINAL.pdf
- Oracle, Java (n.d.). Consultado el 5 de mayo del 2017, desde <http://java.com/es/download>
- Pivotal Software, Inc. Spring.io. (n.d.). Consultado el 21 de abril del 2017, desde <https://spring.io/>
- PrimeTek. Ultimate UI Framework for Java EE. (n.d.). Consultado el 21 de abril de 2017, desde <https://www.primefaces.org/>

- Schwaber, K., & Sutherland, J. (2016). La guía de scrum. La guía definitiva de scrum, las reglas del juego. Consultado el 2 de Abril del 2017, desde <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>
- Scrum.org. What is Scrum? (2015). Consultado el 8 de Junio del 2017 desde, <https://www.scrum.org/resources/what-is-scrum>
- Suzuki, S., & Abe, K. (1985). *Topological structural analysis of digitized binary images by border following*. Computer Vision, Graphics, and Image Processing, 32-46.
- Szeliski, R. (2011). *Computer vision: Algorithms and applications*. London: Springer Science & Business Media.
- Vivas, L. Cambarieri, M.;, García Martínez, N.; Petroff, M.; Muñoz Abbate, H.. (2013) *Un Marco de Trabajo para la Integración de Arquitecturas de Software con Metodologías Ágiles de Desarrollo*. XVIII Congreso Argentino de Ciencias de la Computación.