



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Ingeniería de Sistemas y Automática

Desarrollo de un sistema de visión artificial basado en
redes convolucionales para el conteo automatizado de
microcápsulas en tejidos

Trabajo Fin de Máster

Máster Universitario en Automática e Informática Industrial

AUTOR/A: Martin Sanchís, Ana

Tutor/a: Naranjo Ornedo, Valeriana

Cotutor/a externo: PERIS TORRENT, MARIA

CURSO ACADÉMICO: 2022/2023

Resumen

El presente trabajo se enfoca en el desarrollo de un sistema de visión artificial para el conteo de microcápsulas insertadas en tejidos mediante análisis de imagen microscópica de alta resolución. El objetivo principal es diseñar y desarrollar un sistema que permita determinar la cantidad de microcápsulas presentes en diferentes secciones de los tejidos estudiados. Esta información resulta útil para evaluar la eficacia de los procesos de incorporación de las microcápsulas, así como para determinar la cantidad de producto necesario y la duración de las microcápsulas adheridas a los tejidos.

En este proyecto, se han utilizado redes neuronales convolucionales como base para el diseño de arquitecturas específicas. Estas arquitecturas cuentan con rutas de compresión y expansión, y han sido entrenadas y validadas utilizando una base de datos anotada, que también es parte del trabajo realizado. Adicionalmente, se ha desarrollado una aplicación web para mejorar la usabilidad de este proyecto.

La aplicación web permite a los usuarios cargar imágenes microscópicas de los tejidos y obtener el conteo automatizado de las microcápsulas presentes en dichas imágenes. Esto facilita el análisis y la obtención de resultados de manera más rápida y eficiente.

Palabras clave Visión artificial; Redes convolucionales; Microcápsulas; Tejidos; Análisis de imagen; Conteo automatizado; Aplicación web.

Resum

El present treball s'enfoca en el desenvolupament d'un sistema de visió artificial per al comptatge de microcàpsules inserides en teixits mitjançant anàlisi d'imatge microscòpica d'alta resolució. L'objectiu principal és dissenyar i desenvolupar un sistema que permeti determinar la quantitat de microcàpsules presents en diferents seccions dels teixits estudiats. Aquesta informació resulta útil per a avaluar l'eficàcia dels processos d'incorporació de les microcàpsules, així com per a determinar la quantitat de producte necessari i la duració de les microcàpsules adherides als teixits.

En aquest projecte, s'han utilitzat xarxes neuronals convolucionals com a base per al disseny d'arquitectures específiques. Aquestes arquitectures compten amb rutes de compressió i expansió, i han sigut entrenades i validades utilitzant una base de dades anotada, que també és part del treball realitzat. Addicionalment, s'ha desenvolupat una aplicació web per a millorar la usabilitat d'aquest projecte.

L'aplicació web permet als usuaris carregar imatges microscòpiques dels teixits i obtenir el comptatge automatitzat de les microcàpsules presents en aquestes imatges. Això facilita l'anàlisi i l'obtenció de resultats de manera més ràpida i eficient.

Paraules clau: Visió artificial; Xarxes convolucionals; microcàpsules; Teixits; Anàlisi d'imatge; Comptatge automatitzat; Aplicació web.

Abstract

The present work focuses on the development of an artificial vision system for counting microcapsules inserted in tissues by means of high resolution microscopic image analysis. The main objective is to design and develop a system to determine the number of microcapsules present in different sections of the studied tissues. This information is useful to evaluate the efficiency of the microcapsule incorporation processes, as well as to determine the amount of product needed and the duration of the microcapsules adhered to the tissues.

In this project, convolutional neural networks have been used as the basis for the design of specific architectures. These architectures have compression and expansion paths, and have been trained and validated using an annotated database, which is also part of the work done. Additionally, a web application has been developed to improve the usability of this project.

The web application allows users to upload microscopic images of tissues and obtain automated counts of the microcapsules present in these images. This facilitates faster and more efficient analysis and results.

Keywords: Computer vision; Convolutional networks; Microcapsules; Tissues; Image analysis; Automated counting; Web application.

En primer lugar, quiero expresar mi profundo agradecimiento a mis tutoras, Maria y Valery. Su ayuda y conocimientos han sido fundamentales en mi Trabajo de Fin de Máster. Estoy muy agradecida por la oportunidad que me han ofrecido y especialmente a Valery, quien es parte de mi familia y ha buscado siempre lo mejor para mí. Gracias por todo.

A mis amigos, quienes han estado a mi lado en cada paso de este emocionante viaje. Vuestra presencia ha hecho que esta experiencia sea inolvidable. Gracias por escuchar mis preocupaciones y compartir las risas que han aligerado la carga. Estoy agradecida por teneros en mi vida.

A Fran, por todo lo que hemos compartido juntos y por mantenerme motivada y acompañada en cada decisión. Tu amor incondicional y constante apoyo han sido un verdadero regalo en mi vida.

Gracias por quererme día a día y por ser mi fuente de inspiración.

Por último, quiero expresar mi profundo agradecimiento a mi familia, por estar siempre ahí y por aguantarme día a día. Vuestro apoyo han sido fundamentales en mi camino. Gracias por creer en mí, por darme fuerza y por ser sostenerme en cada paso que doy. Estoy eternamente agradecida por tener una familia tan maravillosa como la mía.

Índice general

I Memoria

1. Introducción	1
1.1. Estado del arte	1
1.2. Motivación	3
1.3. Justificación	3
1.4. Objetivos del proyecto	5
1.5. Estructura del documento	5
2. Marco teórico	7
2.1. Microcápsulas	7
2.1.1. Composición de las microcápsulas	8
2.1.2. Procesos de microencapsulación	10
2.1.3. Adhesión de las microcápsulas	12
2.1.4. Microcápsulas en la industria textil	13
2.2. Inteligencia Artificial	14
2.2.1. Redes neuronales artificiales	15
2.2.2. Perceptrón multicapa	17
2.2.3. Red neuronal convolucional	21
3. Materiales	27
3.1. Base de datos	27
3.2. Software	28
3.3. Hardware	30
4. Sistema de detección	31
4.1. Metodología	31
4.1.1. Preprocesado de imágenes	32
4.1.2. Partición de imágenes	33
4.1.3. U-Net	36
4.1.4. Postprocesado de imágenes	40
4.2. Resultados	43
5. Aplicación web	51
5.1. Estructura y comunicaciones	52
5.2. <i>Frontend</i>	52
5.2.1. Componentes	53
5.3. <i>Backend</i>	55
5.3.1. Protocolo de comunicación HTTP	56

5.3.2. Base de datos en MongoDB	58
5.3.3. Implementación del Modelo de segmentación	60
5.3.4. <i>Docker</i>	61
5.4. Traza de ejecución	62
6. Conclusiones y propuestas de futuro	65
Bibliografía	67
II Presupuesto	
1. Introducción	75
1.1. Alcance y descripción general	75
2. Costes de desarrollo	77
2.1. Personal	77
2.2. Herramientas y licencias	79
3. Total del presupuesto	81
4. Conclusión	83
Bibliografía	85
III Anexos	
A. Manual de usuario	89

Índice de figuras

2.1. Fibras textiles con las microcápsulas adheridas	8
2.2. Estructura de las microcápsulas	8
2.3. Tipos de microcápsulas según su morfología	10
2.4. Fases principales del método de coacervación [26]	11
2.5. Método de secado por aspersión [27]	11
2.6. Liberación del principio activo [30]	13
2.7. Disciplinas: IA, ML y DL [43]	15
2.8. Comparación de la estructura entre redes neuronales biológicas y artificiales [45]	16
2.9. Componentes del perceptrón	17
2.10. Principales funciones de activación del perceptrón	18
2.11. Método del descenso por gradiente	19
2.12. Componentes del perceptrón multicapa [48]	20
2.13. Proceso de retropropagación: forward-backward [50]	21
2.14. Operaciones de la capa de convolución [51]	22
2.15. Ejemplo de capas de <i>pooling</i> [52]	22
2.16. Red neuronal convolucional [53]	23
2.17. Arquitectura LeNet-5 [55]	24
2.18. Arquitectura AlexNet [58]	24
2.19. Arquitectura VGG16 [60]	25
2.20. Arquitectura GoogLeNet [62]	25
2.21. Arquitectura ResNet [64]	26
3.1. Phenom Microscope de la compañía FEI [65].	27
3.2. Ejemplo de la base de datos, imagen microscópica de un tejido con microcápsulas	28
4.1. Diagrama de bloques de la metodología del sistema de detección	31
4.2. Imagen de la base de datos procesada	32
4.3. Ejemplo de <i>ground truth</i>	33
4.4. División de imágenes en el proyecto	34
4.5. Técnica de volteo horizontal	35
4.6. Técnica de volteo vertical	35
4.7. Técnica de cambio de intensidad	36
4.8. Arquitectura U-Net	37
4.9. Principios de escalabilidad compuesta de EfficientNet-B7 [85]	38
4.10. Arquitectura de EfficientNet-B7 [86]	39
4.11. Proceso de postprocesado de la imagen i023	41
4.12. Proceso del postprocesado de la imagen i034	41
4.13. Postprocesado de la imagen i023	42

4.14. Postprocesado de la imagen i034	42
4.15. Estructura de una matriz de confusión binaria	44
4.16. Matriz de confusión en validación	45
4.17. Matriz de confusión en test	46
4.18. Resultado de la imagen i008	48
4.19. Resultado de la imagen i017	48
4.20. Resultado de la imagen i034	48
4.21. Resultado de la imagen i061	49
5.1. Estructura general de comunicaciones	52
5.2. Vista de la página de inicio	54
5.3. Vista de la página de los proyectos	54
5.4. Vista de la página principal de cada proyecto	55
5.5. Vista de la página de subir imágenes	55
5.6. Arquitectura del <i>framework</i> Node.js [88]	56
5.7. Arquitectura del protocolo HTTP	57
5.8. Funcionamiento del patrón de diseño MVC [93]	58
5.9. Visualización de la información de una imagen almacenada en mongoDB	59
5.10. Visualización de la información de los proyectos almacenada en mongoDB	59
5.11. Interconexión entre el <i>frontend</i> y la base de datos	60
5.12. Relación de los componentes en <i>docker</i> [98]	61
5.13. Levantamiento del servicio Flask	63
5.14. Proceso esquemático del uso de la aplicación	63
A.1. Página inicial de la aplicación web	89
A.2. Pulsar el botón de “Iniciar”	90
A.3. Rellenar el campo de “Nombre de proyecto”	90
A.4. Pulsar el botón de “Crear”	91
A.5. Acceder al proyecto	91
A.6. Pulsar el botón para subir imágenes	92
A.7. Subir imágenes nuevas	92
A.8. Imágenes subidas	93
A.9. Seleccionar la imagen que desea	93
A.10. Parámetros de la imagen	94
A.11. Pulsar el botón de “Número de microcápsulas”	94
A.12. Resultado final	95
A.13. Pulsar el botón de “Guardar resultados”	95
A.14. Ejemplo del archivo Excel	96

Índice de tablas

4.1. Métricas en las tres etapas del desarrollo para el modelo propuesto	43
4.2. Predicciones del conjunto de test	46
4.3. Resultados de validación y test	47
4.4. Resultados del conteo	49
2.1. Desglose de los costes de personal	78
2.2. Desglose de las tareas realizadas	79
2.3. Desglose de los costes del software	79
2.4. Desglose de los costes del hardware	80
3.1. Desglose de los costes totales del proyecto	81

Listado de siglas empleadas

Adam Adaptive Moment Estimation.

ANN Artificial Neural Networks.

API Application Programming Interface.

CNN Convolutional Neural Network.

CORS Cross-Origin Resource Sharing.

CVBLab Computer Vision and Behaviour Analysis Laboratory.

DL Deep Learning.

DNN Deep Neural Network.

FN False Negative.

FP False Positive.

GPS General Problem Solver.

GPU Graphics Processing Units.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

IA Inteligencia artificial.

ILSVRC ImageNet Large Scale Visual Recognition Challenge.

JC Intersection over Union.

MAE Mean Absolute Error.

MBConv Mobile Inverted Bottleneck Convolution.

ML Machine Learning.

MLP Multilayer Perceptron.

MSE Mean Squared Error.

MVC Model-View-Controller.

NAS Network Attached Storage.

NoSQL No Structured Query Language.

PPV Positive Predictive Value.

RAM Random Access Memory.

ReLU Rectified Linear Unit.

RPM Revoluciones Por Minuto.

SGD Stochastic Gradient Descen.

SPC Specificity.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

UI User Interface.

UPV Universitat Politècnica de València.

VGG Visual Geometry Group.

Parte I

Memoria

Capítulo 1

Introducción

Durante años, los tejidos han desempeñado un papel fundamental en nuestra sociedad, ya sea como vestimenta para protegernos de las condiciones atmosféricas adversas o como material para mantenernos abrigados en climas fríos. Sin embargo, con los avances tecnológicos, se ha logrado llevar los tejidos a un nivel superior al incorporar microcápsulas y dotarlos de propiedades innovadoras.

Estas microcápsulas son diminutas esferas que contienen sustancias activas, como aceites esenciales, fragancias, agentes hidratantes o incluso medicamentos. Al aplicar estas microcápsulas en los tejidos, se logra una interacción dinámica entre las sustancias encapsuladas y el entorno, lo que abre un amplio abanico de posibilidades en términos de funcionalidad y beneficios para el usuario final.

Por ejemplo, en la industria textil, las microcápsulas se utilizan para crear tejidos que pueden ofrecer propiedades refrescantes, liberando agentes refrescantes en respuesta al calor corporal. También se pueden incorporar microcápsulas con propiedades antimicrobianas para evitar la proliferación de bacterias y olores desagradables en la ropa deportiva. Además, las microcápsulas pueden contener ingredientes hidratantes que se liberan gradualmente para mantener la piel suave y húmeda durante todo el día.

Estos avances en la tecnología de microcápsulas han revolucionado la industria textil y han abierto un mundo de posibilidades para la creación de “tejidos inteligentes” y funcionales. Los tejidos ya no se limitan a ser simples prendas de vestir, sino que se convierten en aliados que mejoran nuestra comodidad, bienestar y calidad de vida en diferentes situaciones.

En este contexto, el presente trabajo se centra en la aplicación de los conocimientos adquiridos en el Máster Universitario en Automática e Informática Industrial para el desarrollo de un sistema de visión artificial que permita el conteo de microcápsulas insertadas en tejidos. Este proyecto ha sido propuesto por el grupo de investigación CVBLab en colaboración con el Departamento de Ingeniería Textil y Papelera, ambos de la Universitat Politècnica de València (UPV).

1.1. Estado del arte

En la industria textil, el uso de microcápsulas es de gran relevancia debido a sus diversas aplicaciones y beneficios en la mejora de las propiedades y funcionalidades de los tejidos. El presente estado del arte resume la situación actual en la investigación sobre las microcápsulas textiles y des-

taca la arquitectura de red neuronal convolucional U-Net como herramienta adecuada para abordar diversos desafíos relacionados con las microcápsulas en este sector.

En la investigación actual sobre la aplicación de microcápsulas en la industria textil, destacan varias líneas de estudio que concentran un considerable interés. Estas áreas de enfoque abordan diversos aspectos relacionados con las microcápsulas y su integración en los tejidos textiles. Entre las líneas de investigación más destacadas se encuentran:

- **Microcápsulas con retardantes de llama:** esta línea de investigación se centra en recubrir diferentes tipos de tejidos con microcápsulas de hidrogenofosfato diamónico, cuya cubierta de resina de poliuretano confiere propiedades ignífugas y de resistencia a la abrasión. Un ejemplo de este enfoque es el trabajo realizado en el laboratorio de investigación Gemtex de la Universidad de Lille, donde se desarrollan nuevas estrategias para la creación de textiles ignífugos [1].
- **Microcápsulas con insecticidas:** este área de investigación es especialmente relevante en regiones donde los insectos pueden transmitir enfermedades graves. En países como la India, donde las picaduras de mosquitos pueden llevar a enfermedades como el dengue y la malaria, se están desarrollando microcápsulas con insecticidas para proteger a la población. La Universitat Politècnica de València ha llevado a cabo investigaciones en este campo, explorando formas de combatir las enfermedades transmitidas por insectos a través de textiles tratados [2].
- **Microcápsulas con fines medicinales:** en esta línea de investigación, se busca incorporar sustancias medicinales en microcápsulas para aplicaciones textiles. Por ejemplo, extracto de Aloe Vera en microcápsulas ya que puede proporcionar propiedades beneficiosas para la piel, como cicatrización, protección y efectos antiinflamatorios [3].
- **Microcápsulas para textiles con propiedades antimicrobianas:** en respuesta a la creciente preocupación por la propagación de enfermedades infecciosas, se investiga el desarrollo de microcápsulas que contengan agentes antimicrobianos. Estos agentes podrían ser liberados de manera controlada para conferir propiedades antibacterianas y antivirales a los tejidos. Esta línea de investigación busca mejorar la higiene y seguridad en entornos médicos como hospitales o en ropa deportiva [4].
- **Microcápsulas para textiles inteligentes:** esta línea de investigación se centra en el desarrollo de microcápsulas capaces de responder a estímulos externos, como cambios de temperatura, humedad o pH. Estas microcápsulas podrían contener compuestos que se liberen en respuesta a ciertas condiciones, lo que permitiría la creación de textiles inteligentes con funcionalidades adaptativas. Estos textiles podrían ser utilizados en diversas aplicaciones, desde ropa deportiva que regula la temperatura corporal hasta textiles para entornos extremos [5].
- **Microcápsulas para liberación controlada de fragancias:** esta línea se enfoca en desarrollar microcápsulas capaces de liberar fragancias de manera controlada y prolongada. Estas microcápsulas podrían integrarse en tejidos para crear textiles perfumados que mantengan su aroma durante períodos más largos de tiempo. Esto podría tener aplicaciones en la industria de la moda, la lencería y el sector de productos para el hogar [6].

Adicionalmente, es importante destacar que la aplicación de la arquitectura U-Net para el conteo automatizado de elementos en imágenes textiles es una línea de investigación aún en desarrollo,

aunque se están observando avances prometedores en este área. Existen trabajos recientes que indican el potencial de utilizar redes neuronales convolucionales como U-Net para resolver este tipo de problemas específicos. Un ejemplo ilustrativo es el trabajo realizado por Weidi Xie, J. Alison Noble y Andrew Zisserman de la Universidad de Oxford [7], quienes aplicaron una arquitectura U-Net para la detección y el conteo de células en imágenes microscópicas.

Además de este trabajo, se han realizado otros esfuerzos en la dirección de aplicar arquitecturas de redes neuronales para la detección y conteo de elementos en imágenes textiles. Un ejemplo destacado es el estudio llevado a cabo por M. Ganz, et al. [8], donde se empleó una combinación de técnicas de aprendizaje profundo para identificar y cuantificar hilos en tejidos compuestos.

Asimismo, la investigación realizada por R. Sawant y R. Deshmukh [9] explora la aplicación de redes convolucionales para la detección y clasificación automatizada de defectos en telas.

Estos trabajos resaltan la tendencia hacia la utilización de técnicas de aprendizaje profundo para resolver desafíos relacionados con imágenes de textiles. Aunque este campo de investigación aún está en sus primeras etapas en relación con su aplicación en la industria textil, estas iniciativas sientan las bases para futuras investigaciones que busquen utilizar arquitecturas de aprendizaje profundo para tareas de conteo y análisis de microcápsulas en imágenes textiles.

1.2. Motivación

El desarrollo de una aplicación web para la identificación y el conteo automático de microcápsulas en imágenes microscópicas de tejidos se motiva por varias razones. Por una parte, la necesidad de evaluar la eficacia de los procesos de incorporación de microcápsulas. Realmente existe la necesidad de contar de manera precisa las microcápsulas presentes en los tejidos para evaluar la eficacia de los procesos de incorporación. Gracias a esto, se permitirá determinar la cantidad de producto necesaria y la duración de las microcápsulas adheridas, lo que contribuirá a mejorar los procesos y optimizar los recursos utilizados en la industria textil.

Por otra parte, el avance y aplicación práctica de la tecnología de microencapsulación en la industria textil. Es decir, la implementación de una aplicación web para la identificación y el conteo automático de microcápsulas permitirá utilizar tecnologías avanzadas, como las redes neuronales convolucionales, en la industria textil. Esto abrirá nuevas posibilidades para mejorar la funcionalidad y las propiedades de los tejidos, y facilitará la adopción de esta tecnología en la práctica industrial.

1.3. Justificación

La implementación de un sistema de visión artificial basado en redes convolucionales permitirá un conteo automatizado y preciso de las microcápsulas presentes en las imágenes microscópicas de los tejidos. Esto facilitará la evaluación de la eficacia de los procesos de incorporación de las microcápsulas, así como la determinación de la cantidad de producto necesario y la duración de las microcápsulas adheridas a los tejidos.

La aplicación permitirá en los laboratorios controlar la cantidad de microcápsulas aplicadas en cada muestra de tejido. Esto es especialmente importante, ya que el tamaño de la microcápsula y

su concentración influyen en su efectividad, por ejemplo, en aspectos como el olor o la repelencia a insectos. Al conocer con precisión la cantidad de microcápsulas presentes, se podrá ajustar la aplicación para lograr el comportamiento deseado y asegurar la calidad del producto final.

Con el conteo preciso de las microcápsulas en cada muestra, será posible optimizar las concentraciones de los principios activos aplicados en los tejidos. Actualmente, algunas empresas aplican exceso de microcápsulas para asegurar su efectividad, lo que resulta en costes adicionales. Mediante la optimización de las concentraciones, se podrán reducir los costes de producción y aumentar la competitividad de la empresa.

Al optimizar las concentraciones de microcápsulas, se reducirá la cantidad de vertidos de residuos en las aguas residuales, lo que contribuirá a una producción más sostenible y respetuosa con el medio ambiente. Al controlar adecuadamente la cantidad de microcápsulas aplicadas, se minimizará el impacto ambiental y se promoverá la responsabilidad corporativa en términos de sostenibilidad.

La implementación de una aplicación web permitirá a los usuarios cargar fácilmente imágenes microscópicas de los tejidos y obtener el conteo automatizado de las microcápsulas presentes en cada imagen. Esto agilizará el proceso de análisis y obtención de resultados, lo que resultará en una mayor eficiencia y productividad en la evaluación de las muestras.

Este proyecto se encuentra estrechamente relacionado con varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas:

1. ODS 9: Industria, Innovación e Infraestructura: La implementación de un sistema de visión artificial basado en redes convolucionales representa un avance tecnológico significativo en el campo de la industria, al mejorar la eficiencia y precisión en el conteo de microcápsulas en tejidos. Además, esta innovación contribuirá a optimizar los procesos de producción y control de calidad en la fabricación de productos basados en microcápsulas.
2. ODS 12: Producción y Consumo Responsables: La optimización de las concentraciones de microcápsulas mediante el conteo preciso de las mismas permitirá reducir el exceso de aplicación y minimizar el impacto ambiental, al disminuir la cantidad de vertidos de residuos en las aguas residuales. Esto se traduce en una producción más responsable y sostenible, promoviendo prácticas que respeten el medio ambiente.
3. ODS 13: Acción por el Clima: Al contribuir a la reducción de vertidos de residuos y promover prácticas de producción más sostenibles, este proyecto apoya la lucha contra el cambio climático y fomenta la adopción de acciones que mitiguen el impacto ambiental en la industria.

En conclusión, este proyecto no solo busca mejorar la eficiencia y precisión en el conteo de microcápsulas en tejidos, sino que también tiene un impacto positivo en el desarrollo sostenible al alinearse con varios Objetivos de Desarrollo Sostenible. La implementación de esta tecnología puede contribuir a una producción más responsable, sostenible y respetuosa con el medio ambiente, lo que promueve el bienestar social y ambiental a nivel global.

1.4. Objetivos del proyecto

El objetivo principal de este proyecto es diseñar una aplicación web para la identificación y el conteo automático de microcápsulas en imágenes microscópicas de tejidos mediante la aplicación de redes neuronales convolucionales. En concreto, se va a utilizar una arquitectura de redes neuronales convolucionales denominada U-Net.

Esta aplicación permitirá evaluar la eficacia de los procesos de incorporación de las microcápsulas, determinar la cantidad de producto necesaria y la duración de las microcápsulas adheridas a los tejidos. De esta manera, se contribuirá al avance y la aplicación práctica de esta fascinante tecnología en la industria textil y se abrirán nuevas posibilidades para mejorar aún más los tejidos del futuro.

La colaboración con el Departamento Ingeniería Textil y Papelera de la Universitat Politècnica de València tiene como objetivo principal facilitarles métodos de conteo automático de microcápsulas, lo que contribuirá a impulsar sus líneas de investigación en el campo de la microencapsulación en la industria textil.

Para cumplir con los objetivos generales de este trabajo será necesario cumplir los siguientes objetivos específicos:

- Realizar un estudio exhaustivo para procesar adecuadamente las imágenes microscópicas utilizadas en el análisis de las microcápsulas en tejidos. Esto implicará investigar y aplicar técnicas de preprocesado de imágenes, como la segmentación de objetos de interés.
- Construir una base de datos anotada con una amplia variedad de imágenes microscópicas de tejidos que contengan microcápsulas, junto con anotaciones precisas que indiquen la ubicación y cantidad de microcápsulas en cada imagen.
- Adquirir conocimientos sobre el funcionamiento de las redes neuronales convolucionales para diseñar, entrenar, ajustar y evaluar un modelo que pueda contar las microcápsulas en los tejidos de manera precisa y eficiente. Esto permitirá obtener resultados confiables y contribuirá al éxito general del proyecto.
- Adquirir conocimientos sobre el desarrollo de aplicaciones web para crear una interfaz de usuario intuitiva y funcional. Esto implica comprender los lenguajes de programación web, así como el uso de *frameworks* y bibliotecas populares.

1.5. Estructura del documento

Con el fin de garantizar la mejor comprensión de este proyecto, se procede a especificar la estructura de capítulos de esta memoria:

En el Capítulo 1, se desarrolla una breve introducción de la motivación del proyecto. Además, se realiza un breve estudio de las investigaciones previas relacionadas con estos campos. Por último, se indica cuál es el objetivo del trabajo y su estructura.

En el Capítulo 2, se resume la investigación del marco teórico necesario para comprender y contextualizar el proyecto. Se abordan diferentes aspectos relacionados con la microencapsulación en la industria textil, la inteligencia artificial IA y las redes neuronales convolucionales.

En el Capítulo 3, se describen la base de datos utilizada para el conteo automático de microcápsulas en imágenes microscópicas de tejidos, así como los programas y librerías empleados para el diseño y desarrollo de los modelos y de la aplicación web.

En el Capítulo 4, se detalla el método propuesto para el desarrollo de este proyecto, incluyendo el preprocesamiento de las imágenes, la selección de la arquitectura U-Net, las técnicas de optimización aplicadas y el postprocesado de las imágenes. Además, se describen las métricas de evaluación utilizadas para medir el desempeño de los modelos desarrollados, se presentan los experimentos de ablación realizados para evaluar la importancia de las diferentes etapas del proceso, y se comparan los resultados obtenidos.

En el Capítulo 5, se presenta el desarrollo de la aplicación web, centrándose tanto en el *frontend* como en el *backend*. Se describen la estructura de comunicaciones y se muestra la traza de ejecución para poner en marcha el proyecto. Es decir, se detallan los pasos necesarios y las acciones realizadas desde el inicio hasta la implementación y funcionamiento completo de la aplicación web.

En el Capítulo 6, se concluye el trabajo y se presentan propuestas para futuras investigaciones en el campo del conteo automático de microcápsulas en tejidos. Se destacan las contribuciones y las limitaciones del proyecto, así como las posibles direcciones para ampliar y mejorar los métodos y la aplicación desarrollada.

Capítulo 2

Marco teórico

En este capítulo, se presentará el marco teórico necesario para contextualizar el problema. Se abordarán dos aspectos fundamentales. Por un lado, se explicará qué son las microcápsulas y se profundizará en los elementos que las componen. También se explorarán los diferentes procesos de microencapsulación disponibles en la actualidad, destacando su aplicación en la industria textil. Por otro lado, una breve explicación sobre el concepto de inteligencia artificial IA, enfatizando en las redes neuronales artificiales y su capacidad para aprender y realizar tareas complejas. Se hará especial énfasis en las redes neuronales convolucionales que son empleadas en este proyecto.

2.1. Microcápsulas

Las microcápsulas son diminutas partículas con un tamaño en el rango de micrómetros, que contienen compuestos encapsulados en su interior. En el contexto textil, estas microcápsulas se utilizan para transmitir propiedades funcionales a los tejidos. Su naturaleza microscópica permite la aplicación discreta en las fibras y superficies de los textiles, ofreciendo una serie de ventajas y beneficios únicos.

Las microcápsulas se aplican en los tejidos a través de técnicas de acabado y recubrimiento. Los compuestos encapsulados pueden variar según la aplicación y pueden incluir fragancias, agentes antimicrobianos, repelentes de insectos, sustancias hidratantes para la piel, entre otros. La liberación controlada de estos compuestos se produce en respuesta a estímulos específicos, como el movimiento, la humedad, la temperatura o el pH, lo que brinda una funcionalidad inteligente y adaptativa a los textiles.

La adherencia de estas microcápsulas a los tejidos es un proceso crítico para garantizar su efectividad y durabilidad en la aplicación textil. Este paso se logra a menudo mediante el uso de agentes adhesivos o pegamentos diseñados específicamente para unir las microcápsulas a las fibras textiles de manera efectiva. En la figura 2.1 se puede observar una imagen con microcápsulas adheridas a las fibras textiles en la que se diferencia el pegamento de la microcápsula.

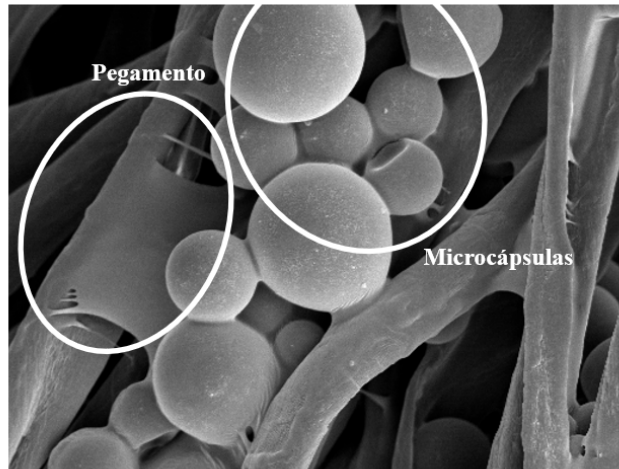


Figura 2.1: Fibras textiles con las microcápsulas adheridas

2.1.1. Composición de las microcápsulas

La microencapsulación es una tecnología de protección para encapsular materiales sólidos, líquidos o gaseosos dentro de una cubierta protectora, lo que da lugar a la formación de microcápsulas. Las microcápsulas tienen forma esférica o irregular y se dividen en dos partes, las cuales se pueden observar en la figura 2.2: núcleo y membrana.

Por una parte, el núcleo es la parte intrínseca y contiene el material central también referido como ingrediente activo o material de carga [10]. Por otra parte, la membrana es la parte extrínseca que protege el núcleo de manera permanente o temporal del ambiente externo [11].

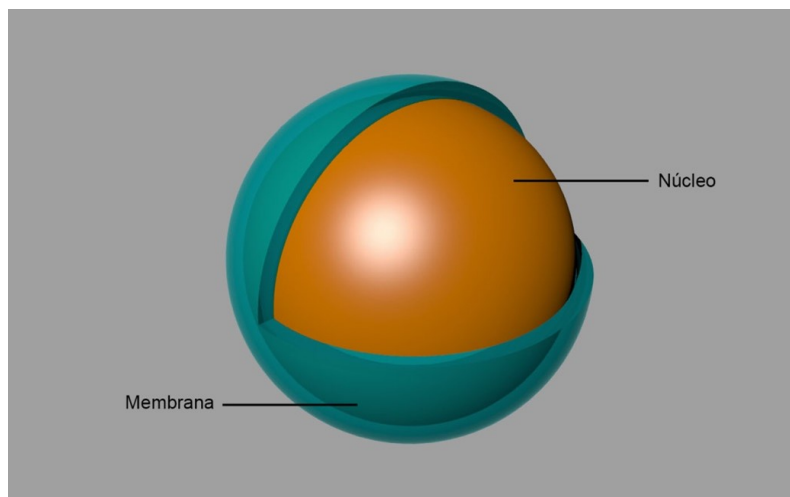


Figura 2.2: Estructura de las microcápsulas

El revestimiento de la microcápsula también puede ser denominado además de membrana, envoltura, pared o cáscara. Las membranas son típicamente compuestas de matrices de polímeros sintéticos o naturales, utilizados por su capacidad para formar una película. Es importante que es-

tos materiales no reaccionen con los materiales utilizados en el núcleo [12] [13]. Por lo tanto, es de gran importancia elegir el material de la membrana en función de la naturaleza del material del núcleo [14].

La elección del material de la membrana es un proceso crucial y estratégico, ya que debe cumplir con una serie de requisitos específicos. Uno de los objetivos principales es asegurar que la membrana proporcione la adecuada protección física y química al material del núcleo, evitando su exposición a factores externos como la humedad, la luz y el oxígeno, que podrían comprometer su funcionalidad y estabilidad a lo largo del tiempo [15] [16].

Además, los materiales de la membrana deben ser compatibles con los componentes internos de la microcápsula y con los procesos de encapsulación utilizados. Esto incluye aspectos como la adhesión a la superficie del núcleo, la resistencia a la degradación y la capacidad de formar una barrera efectiva. Diferentes aplicaciones pueden requerir diferentes propiedades de la membrana, por lo que la selección del material debe ser adaptada específicamente a las necesidades del producto final.

En la búsqueda de materiales adecuados para la membrana de microcápsulas, se han investigado diversas opciones, desde polímeros naturales hasta polímeros sintéticos, cada uno con sus propias ventajas y desafíos. Ejemplos de polímeros comúnmente utilizados incluyen alginato, quitosano, gelatina, polietileno, polipropileno, poliuretano, entre otros. Estos materiales han sido estudiados en términos de sus propiedades mecánicas, de barrera, de degradación y de afinidad con los componentes del núcleo, lo que permite adaptar la elección del material a las características específicas de la aplicación deseada [15] [16] [17].

La composición del núcleo activo es de amplia diversidad y es esta la cualidad que define la aplicación específica de las microcápsulas. Algunos ejemplos de los componentes presentes en el núcleo de las microcápsulas abarcan desde perfumes, fármacos y fertilizantes hasta sabores volátiles, probióticos y semillas de frutas [18]. Además, la compatibilidad del material del núcleo y el de la cubierta emerge como un factor crítico para potenciar la eficiencia de la microencapsulación. Es una práctica extendida llevar a cabo un pretratamiento en el material del núcleo con el propósito de mejorar la compatibilidad entre ambos componentes.

De igual modo, la morfología de las microcápsulas (tamaño y forma) depende del material del núcleo, ya que es resultado de la deposición del material de la membrana sobre el núcleo. Además, el método utilizado para sintetizar las microcápsulas también influye en la morfología de las microcápsulas. Teniendo en cuenta esto, existen diferentes tipos de microcápsulas que podemos observar en la figura 2.3. En la cual hace referencia, (A) simple o mononuclear; (B) regular, paredes múltiples, mononuclear; (C) regular, multinuclear; (D) regular, núcleo disperso en la matriz; (E) irregular, mononuclear; y (F) irregular, multinuclear [15] [19] [20] [21].

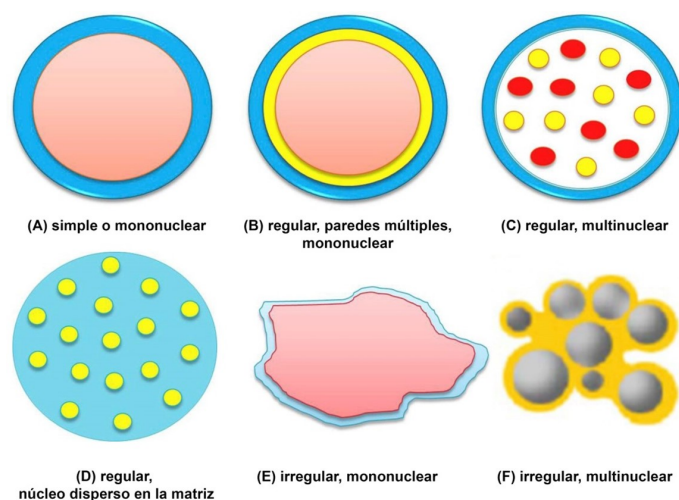


Figura 2.3: Tipos de microcápsulas según su morfología

2.1.2. Procesos de microencapsulación

Tal como se ha mencionado, las microcápsulas desempeñan un papel fundamental en la protección de materiales inestables y la liberación controlada de sustancias [10]. Existe una amplia variedad de técnicas de microencapsulación desarrolladas para lograr la liberación controlada de ingredientes activos, enfocándose en la velocidad regulada [12] [16] [22].

Según lo señalado por Salaün (2016), se han desarrollado más de 200 métodos de microencapsulación, y la mayoría de ellos comparten tres etapas básicas: aislamiento de los componentes del núcleo, formación de micropartículas y endurecimiento de la membrana [23].

La elección del método de síntesis apropiado debe considerar diversos factores, tales como las propiedades físicas y químicas deseadas tanto del núcleo como de los materiales de recubrimiento, la aplicación prevista del producto final, el costo del proceso de síntesis, el impacto del uso de solventes orgánicos en la salud y las consideraciones medioambientales [23] [24]. La distinción entre los métodos radica en cómo se encapsula el agente activo durante la formación de las microcápsulas, es decir, en el proceso de endurecimiento de la membrana. Esto puede ocurrir de manera mecánica/física, química o físico-química [24] [23].

En el contexto de aplicaciones textiles, es crucial que las microcápsulas posean ciertas características específicas. Por ejemplo, su tamaño debe oscilar entre 1 y 40 μm , deben mantener una estabilidad térmica de hasta 150 $^{\circ}\text{C}$ y presentar robustez mecánica para soportar las condiciones de deposición en textiles y resistir los ciclos de lavado. Por esta razón, se dan preferencia a los métodos químicos y físico-químicos, ya que los procesos mecánicos pueden generar partículas con diámetros promedio demasiado grandes para su uso en textiles [25].

En este contexto, se exploran diferentes métodos de microencapsulación, cada uno de los siguientes métodos presenta ventajas y desafíos específicos, lo que los hace adecuados para diferentes aplicaciones y requerimientos de encapsulación:

- **Coacervación:** es un método basado en la separación de fases que utiliza interacciones electrostáticas o de hidrofobicidad para formar una fase líquida rica en polímeros alrededor del

núcleo. Esta fase coacervada se solidifica posteriormente para formar la membrana de la microcápsula [24]. Este proceso se puede apreciar en la figura 2.4.



Figura 2.4: Fases principales del método de coacervación [26]

- Extrusión simple: es un proceso donde la fase de núcleo y la fase de recubrimiento se mezclan y se extruyen a través de una boquilla. Las microcápsulas se forman a medida que el material de recubrimiento rodea el núcleo durante el proceso de extrusión [24].
- Extrusión en doble emulsión: en esta, se forma una emulsión primaria en la que el núcleo está rodeado por el material de recubrimiento. Luego, esta emulsión primaria se introduce en una segunda fase de recubrimiento, formando una emulsión secundaria. Las microcápsulas se forman cuando la emulsión secundaria se solidifica [24].
- Secado por aspersión: este método implica atomizar una solución que contiene tanto el núcleo como el material de recubrimiento en una corriente de aire caliente. La evaporación del solvente resulta en la formación de microcápsulas sólidas con el material de recubrimiento alrededor del núcleo [24]. La figura 2.5 representa las partes del proceso y su evolución correspondiente.

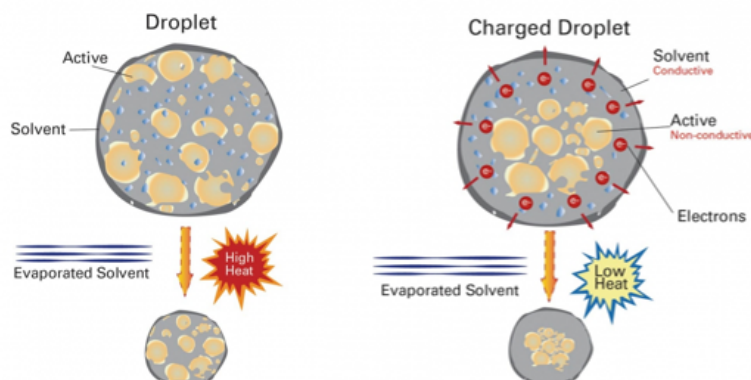


Figura 2.5: Método de secado por aspersión [27]

2.1.3. Adhesión de las microcápsulas

La adhesión efectiva de las microcápsulas a los sustratos textiles es un factor clave para garantizar la durabilidad y la funcionalidad de los productos textiles microencapsulados. Los procesos de adhesión involucran la unión química o física entre las microcápsulas y la superficie del tejido, asegurando que las microcápsulas permanezcan en su lugar durante el uso y el lavado.

Uno de los enfoques comunes para lograr una adhesión eficiente es la elección adecuada de adhesivos. Los adhesivos son sustancias utilizadas para unir dos superficies de diferentes materiales, y en el contexto de las microcápsulas textiles, ayudan a fijar las microcápsulas al tejido de manera segura y duradera.

Los tipos de adhesivos utilizados en la microencapsulación de textiles pueden ser clasificados en varias categorías [25]:

- Adhesivos químicos: estos adhesivos involucran reacciones químicas entre los componentes del adhesivo y las microcápsulas, así como la superficie del tejido. Los adhesivos químicos pueden ofrecer una adhesión sólida y permanente, pero su selección debe tener en cuenta la compatibilidad química con los materiales y procesos de fabricación involucrados.
- Adhesivos físicos: los adhesivos físicos se basan en interacciones físicas, como fuerzas de van der Waals y enlaces de hidrógeno, para unir las microcápsulas al tejido. Aunque pueden ser menos permanentes que los adhesivos químicos, los adhesivos físicos a menudo son más flexibles y permiten una mayor movilidad de las microcápsulas en la superficie del tejido.
- Adhesivos termofusibles: estos adhesivos se aplican en estado fundido y se solidifican cuando se enfrían, formando una unión sólida entre las microcápsulas y el tejido. Son eficaces para fijar microcápsulas en áreas específicas del tejido y pueden ser activados mediante calor.
- Adhesivos biocompatibles: estos adhesivos están diseñados para ser seguros para la piel y el medio ambiente. Son esenciales en aplicaciones textiles que requieren contacto directo con la piel, como textiles médicos y prendas deportivas.

Por una parte, la elección del tipo de adhesivo dependerá de factores como el tipo de microcápsula, el tejido base, las condiciones de uso y el proceso de fabricación. Es fundamental realizar pruebas de adhesión y evaluar la durabilidad de la unión en condiciones reales antes de su implementación en productos finales.

Por otra parte, la adhesión de las microcápsulas a su sustrato es otro de los aspectos esenciales. Esta determina la durabilidad y la funcionalidad del producto final. Los procesos de adhesión están influenciados por múltiples factores, incluidos los materiales utilizados, la morfología de las microcápsulas y las condiciones ambientales en las que se encuentran [28].

El tipo de adhesión puede variar según la interacción entre las microcápsulas y el sustrato. Algunas interacciones comunes incluyen:

- Adhesión mecánica: en este tipo de adhesión, las microcápsulas quedan atrapadas físicamente en la superficie del sustrato debido a fuerzas mecánicas, como la rugosidad de la superficie o las propiedades adhesivas de los materiales involucrados.

- Adhesión química: en algunos casos, las microcápsulas pueden formar enlaces químicos con la superficie del sustrato. Esto puede aumentar significativamente la adhesión y la durabilidad en condiciones adversas.
- Adhesión por capilaridad: ocurre cuando las microcápsulas son absorbidas por los poros o las cavidades de la superficie del sustrato debido a las fuerzas capilares.
- Adhesión electrostática: en algunos casos, las cargas eléctricas en la superficie de las microcápsulas y el sustrato pueden generar atracción electrostática, lo que resulta en una adhesión eficaz.

Es importante destacar que el proceso de adhesión puede influir en la liberación controlada de los ingredientes encapsulados. Una adhesión más fuerte puede restringir la liberación, mientras que una adhesión más débil puede permitir una liberación más rápida [28].

2.1.4. Microcápsulas en la industria textil

Las microcápsulas han demostrado ser una innovación valiosa en la industria textil, permitiendo la incorporación de diversas funcionalidades en los tejidos. Estas microcápsulas pueden contener una variedad de ingredientes activos que se liberan de manera controlada en respuesta a diferentes estímulos, mejorando así las propiedades y características de los textiles.

Los microencapsulados destinados a aplicaciones textiles generalmente no requieren membranas solubles [29], excepto en casos excepcionales. Esto se debe a que los principios activos contenidos en el núcleo de las microcápsulas se liberan mediante la rotura de la cápsula o mediante la permeabilidad de las membranas como representa la figura 2.6. La insolubilidad de estas membranas es beneficiosa ya que contribuye a preservar la integridad de las microcápsulas durante los ciclos de lavado. Este aspecto es particularmente relevante en productos que, debido a su uso final, están sujetos a lavados continuos.

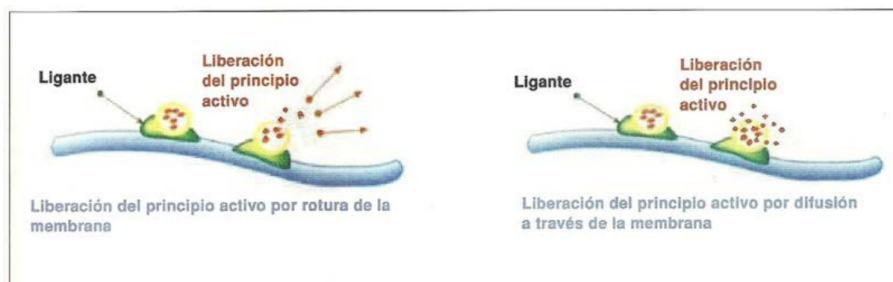


Figura 2.6: Liberación del principio activo [30]

Uno de los usos más comunes de las microcápsulas en textiles es la liberación de fragancias o aromas. La incorporación de microcápsulas con fragancias en los tejidos puede proporcionar un efecto duradero de perfume, que se libera gradualmente con el movimiento o el contacto con la piel [31].

Además de los aromas, las microcápsulas también se utilizan para impartir propiedades funcionales a los textiles. Por ejemplo, se pueden incorporar microcápsulas con agentes repelentes de insectos

para crear textiles resistentes a los mosquitos y otros insectos [25]. También se han desarrollado textiles con microcápsulas que contienen agentes antimicrobianos para mejorar la higiene y la durabilidad [32].

Otra aplicación prometedora es la creación de textiles termorreguladores. Las microcápsulas pueden contener materiales de cambio de fase que absorben o liberan calor en función de la temperatura ambiente, permitiendo que los textiles regulen la temperatura corporal y aumenten la comodidad del usuario [33].

Además, las microcápsulas pueden utilizarse para la liberación de ingredientes cosméticos, como cremas hidratantes o vitaminas, directamente en la piel a través del contacto con los textiles. Esto ofrece beneficios adicionales para la piel y la comodidad del usuario [31].

En la industria deportiva, las microcápsulas también se han utilizado para la liberación controlada de fragancias energizantes o refrescantes en tejidos deportivos, mejorando así la experiencia del usuario durante la actividad física [25].

En resumen, las microcápsulas han abierto un mundo de posibilidades en la industria textil, permitiendo la creación de textiles con propiedades y características únicas que van más allá de la simple función de vestimenta

2.2. Inteligencia Artificial

En las décadas de mediados del siglo XX, los cimientos de la Inteligencia Artificial (IA) comenzaron a establecerse a través de una serie de eventos y desarrollos clave. Uno de los momentos más significativos tuvo lugar en 1956 con el Dartmouth Workshop, un evento que reunió a diversos científicos para explorar la viabilidad de crear máquinas capaces de emular la capacidad de pensamiento humano. Durante este taller, se acuñó oficialmente el término “Inteligencia Artificial”, marcando el inicio formal de la investigación en este campo [34].

En la misma época, investigadores como Allen Newell y Herbert A. Simon avanzaron en la programación de la lógica simbólica. Su trabajo resultó en la creación del “Logic Theorist”, un programa informático capaz de demostrar teoremas de lógica matemática, lo que sentó las bases para la programación de sistemas basados en reglas y la representación del conocimiento [35].

Otro hito importante fue el desarrollo del “Geometry Theorem Prover” por John McCarthy en 1959. Este programa demostraba teoremas geométricos utilizando reglas lógicas, y McCarthy también acuñó el término “Inteligencia Artificial” [36].

Alan Turing introdujo el “Test de Turing” en 1950, una prueba para evaluar la inteligencia de las máquinas. Según este test, una máquina podría considerarse inteligente si sus respuestas fueran indistinguibles de las de un humano en una conversación [37].

En este periodo, se desarrollaron los primeros programas de juego, como el “Checker-playing Program” de Arthur Samuel, que aprendió a jugar damas a través del ajuste y la retroalimentación de parámetros [38]. También, Newell y Simon crearon el “General Problem Solver” (GPS), un programa que resolvía problemas mediante la aplicación de reglas lógicas generales. Con este programa se demostró la capacidad de las computadoras para razonar y resolver problemas [39].

De esta manera, la IA busca desarrollar sistemas y programas capaces de realizar tareas que requieren inteligencia humana. Estas tareas abarcan desde el procesamiento y análisis de datos hasta la

toma de decisiones, el aprendizaje autónomo y la resolución de problemas complejos. En esencia, la IA persigue dotar a las máquinas de la capacidad de comprender, razonar, aprender y adaptarse, permitiendo que emulen y, en algunos casos, superen las capacidades cognitivas humanas en contextos específicos.

Dentro del ámbito de la IA, se delinean diversas disciplinas, cada una con su enfoque particular. Por un lado, la IA abarca un campo más amplio. Como se ha mencionado anteriormente, engloba la creación de sistemas inteligentes y racionales capaces de resolver problemas complejos, tomando decisiones y realizando tareas diversas que normalmente requerirían intervención humana [40].

Por otro lado, una de las ramas más relevantes dentro de este campo es el aprendizaje automático conocido como *machine learning* (ML), el cual se centra en la construcción de algoritmos y modelos que permiten a las máquinas aprender patrones y relaciones a partir de datos, con el fin de tomar decisiones y realizar predicciones [41]. Con esta disciplina, nace el concepto de aprendizaje profundo, conocido como *deep learning*, (DL) como una subcategoría del *machine learning*.

El *deep learning* se centra en la creación de redes neuronales profundas o deep neural networks (DNN), que se caracterizan por tener múltiples capas ocultas capaces de aprender representaciones jerárquicas de los datos. Esta subcategoría ha llevado a avances notables en la interpretación y el análisis de imágenes, el procesamiento del lenguaje natural y otros campos [42].

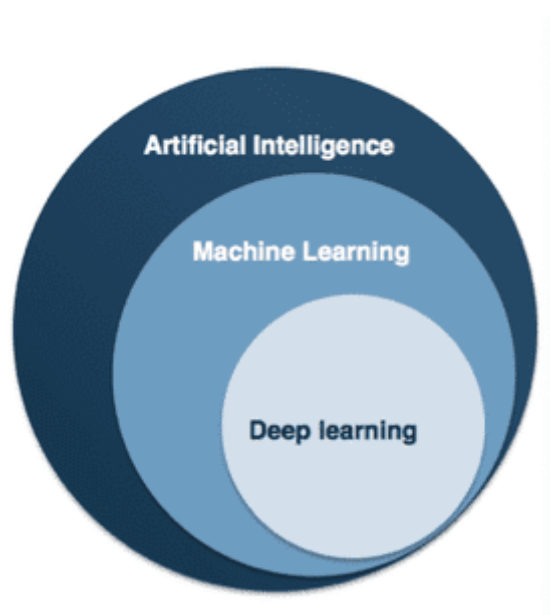


Figura 2.7: Disciplinas: IA, ML y DL [43]

Por lo tanto, mientras que la IA es un campo amplio y conceptual, el ML y el DL son enfoques más específicos que han revolucionado la capacidad de las máquinas para aprender y resolver problemas de manera autónoma. Esta diferenciación se aprecia con claridad en la figura 2.7.

2.2.1. Redes neuronales artificiales

Las redes neuronales artificiales o *artificial neural networks* (ANN) son un componente esencial en el campo de la inteligencia artificial. Estas redes están diseñadas para simular el funcionamiento

de las redes neuronales biológicas del cerebro humano.

El concepto de redes neuronales artificiales tiene sus raíces en la década de 1940, pero fue en la década de 1980 cuando experimentaron un renacimiento significativo debido a avances en el campo y al aumento de la capacidad de cómputo [44]. Desde entonces, las ANN han evolucionado constantemente, y en particular, las (DNN) han demostrado un gran éxito en tareas de *deep learning*, como el reconocimiento de imágenes y el procesamiento del lenguaje natural.

De forma similar a las redes neuronales biológicas, las ANN están compuestas por un conjunto de nodos interconectados, conocidos como neuronas artificiales. Cada neurona artificial recibe entradas, realiza operaciones matemáticas simples en esas entradas y produce una salida que luego se transmite a otras neuronas en la red. Estas conexiones entre neuronas se ajustan mediante de los pesos durante el proceso de entrenamiento de la red. Estos pesos son valores numéricos que determinan la fuerza y la dirección de la influencia que una neurona tiene sobre otra en la propagación de la información a través de la red. A través del ajuste de estos pesos, la red neuronal puede mejorar su capacidad para realizar tareas específicas, como el reconocimiento de patrones, la clasificación de datos y la toma de decisiones. En este sentido, las ANN aprovechan la capacidad de las neuronas artificiales para procesar información y aprender de los datos para realizar tareas complejas de procesamiento y análisis de información.

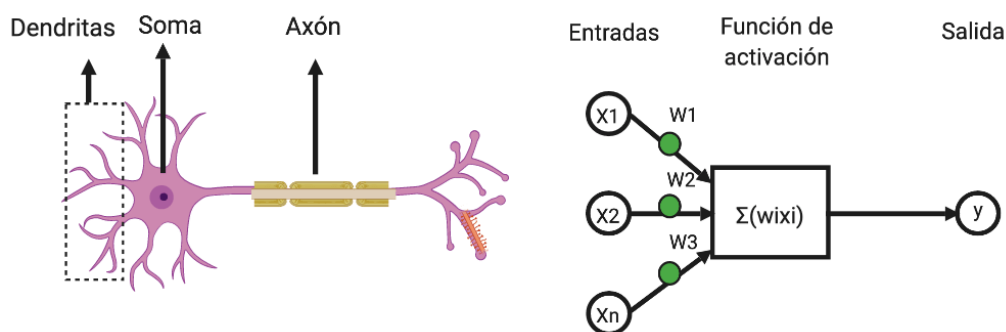


Figura 2.8: Comparación de la estructura entre redes neuronales biológicas y artificiales [45]

En la figura 2.8 se puede observar los componentes y estructura mencionados anteriormente, además de la analogía que existe entre ambas redes. Se podría decir que las dendritas son el equivalente a la entrada de las ANNs, el soma o cuerpo celular el equivalente a la función de activación y el axón a la salida de la red neuronal artificial.

A pesar de su similitud a las redes neuronales biológicas, existen diferencias notables entre ambas. Mientras que las neuronas biológicas utilizan impulsos eléctricos y químicos para comunicarse, las neuronas artificiales utilizan funciones matemáticas para modelar esta comunicación [46]. Las ANN pueden variar en su forma y estructura, con arquitecturas como las redes neuronales *feedforward*, recurrentes y convolucionales, cada una adaptada a diferentes tipos de tareas y datos.

Las ANN han demostrado ser particularmente poderosas en la resolución de problemas complejos que incluyen reconocimiento de patrones, clasificación, regresión, procesamiento de lenguaje natural, etc.

2.2.2. Perceptrón multicapa

En 1957, en el laboratorio aeronáutico de Cornell, Frank Rosenblatt presentó el perceptrón como una unidad central en el campo de las redes neuronales, planteando la “regla de aprendizaje del perceptrón” [47]. El modelo del perceptrón tomó forma gracias a la creación de hardware personalizado llamado “Mark 1 Perceptron”. Fue diseñado para imitar la capacidad humana de reconocer patrones en imágenes.

Los componentes principales del perceptrón son las entradas, los pesos, los nodos y las salidas. En cuanto a su estructura, las neuronas efectúan sumas ponderadas de sus entradas en cada uno de los nodos y aplican funciones de activación para establecer umbrales, permitiendo así la interpretación de entradas sensoriales como luz, sonido o temperatura. Mediante esta umbralización, el perceptrón puede operar como un modelo de clasificación binaria. Este umbral establece una línea de decisión lineal que busca minimizar la distancia entre los puntos y dicha línea. Las partes comentadas anteriormente se puede observar en el esquema de la figura 2.9.

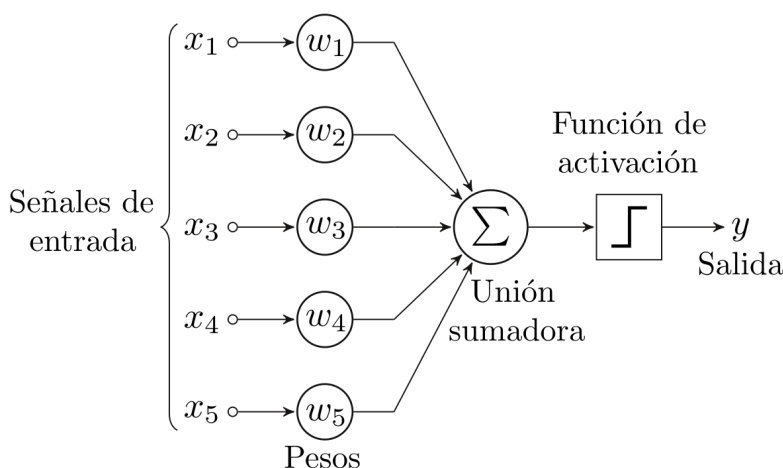


Figura 2.9: Componentes del perceptrón

En el contexto del diseño y funcionamiento de los perceptrones, existen dos aspectos cruciales: las funciones de activación y el método de optimización. Las funciones de activación influyen en la capacidad de la red para aprender y adaptarse a los datos, mientras que el proceso de optimización asegura que la red ajuste sus parámetros de manera eficiente para realizar tareas específicas.

Dentro del ámbito de las funciones de activación para los perceptrones, se encuentra una amplia gama de opciones entre las cuales se pueden elegir. Estas funciones juegan un papel crucial en la capacidad de la red para aprender y adaptarse a los datos. Entre las opciones más notables se encuentran la función sigmoide, la tangente hiperbólica y la ReLu (Rectified Linear Unit).

La función sigmoide es una de las más utilizadas, dicha función transforma los valores de entrada de entrada se transformen en una escala entre 0 y 1, lo que facilita la interpretación de las salidas como probabilidades o niveles de activación. En la figura 2.10 aparece la fórmula que permite esta forma y la representación de la misma. Por otro lado, la tangente hiperbólica también tiene una forma similar, pero su rango se extiende de -1 a 1. Esto la hace más simétrica y centrada en torno al origen, lo que puede ser beneficioso en ciertos casos.

La ReLu, ha ganado popularidad en los últimos años debido a su eficiencia computacional y su

capacidad para mitigar el problema del desvanecimiento de gradientes. En esta función, los valores negativos se convierten en cero, mientras que los valores positivos se mantienen sin cambios. En la figura 2.10 aparece la fórmula que permite estos valores y la representación de la función. Esto resulta en una activación no lineal que permite que las señales relevantes pasen sin atenuación.

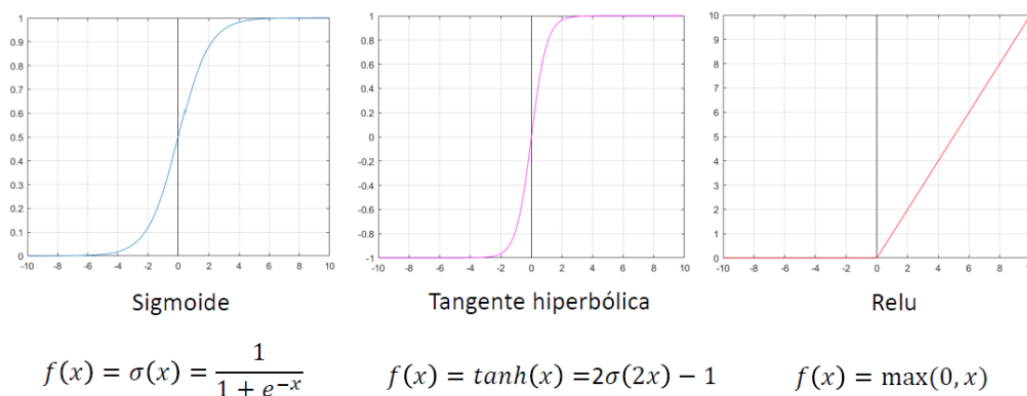


Figura 2.10: Principales funciones de activación del perceptrón

Una vez comprendidas las funciones de activación, es esencial considerar cómo se integran en el proceso de optimización. Durante el entrenamiento de una red neuronal, el objetivo es ajustar los pesos y bias para minimizar la función de pérdida y, por lo tanto, mejorar el rendimiento de la red. Esta función de pérdida cuantifica la diferencia entre las predicciones del modelo y las etiquetas reales. Para reducir el número de predicciones incorrectas, se utiliza un método de optimización que busca encontrar los valores de los parámetros del modelo que minimizan esta función de pérdida.

El bias, también conocido como sesgo, es un parámetro adicional en cada neurona que permite ajustar su nivel de activación y su respuesta ante diferentes entradas. Es un componente fundamental para controlar la linealidad de una neurona y personalizar sus respuestas según el contexto y las preferencias aprendidas durante el entrenamiento.

Uno de los métodos de optimización más utilizados es el Descenso por Gradiente Estocástico (SGD), nos permite automatizar de forma más eficiente el ir probando coeficientes de los modelos. En este método, en cada iteración del proceso de entrenamiento, se calcula el gradiente de la función de pérdida con respecto a los parámetros del modelo.

Como se ve en la figura 2.11, el gradiente representa la pendiente en el punto que nos encontremos de la función de coste e indica la dirección en la que la función de pérdida aumenta más rápidamente, por lo que el objetivo es moverse en la dirección opuesta para disminuir la pérdida.

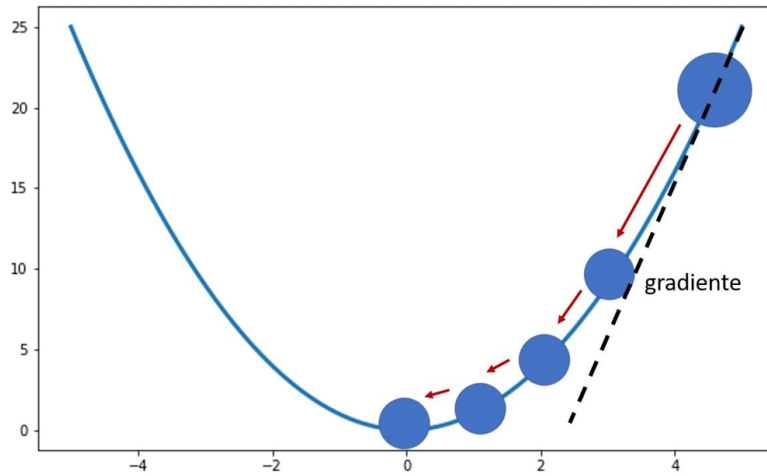


Figura 2.11: Método del descenso por gradiente

La actualización de los pesos se realiza restando una fracción del gradiente de la función de pérdida a los pesos actuales. La tasa de aprendizaje, también conocida como *learning rate*, controla el tamaño de este paso. La ecuación general para la actualización de los pesos en el Descenso por Gradiente Estocástico es:

$$w_{n+1} = w_n - \gamma \cdot \nabla L(w_n) \quad (2.1)$$

Donde w_{n+1} es el nuevo valor de los pesos, w_n es el valor actual de los pesos, γ es la tasa de aprendizaje, y $\nabla L(w_n)$ es el gradiente de la función de pérdida con respecto a los pesos actuales.

Sumando a esto, la tasa de aprendizaje determina cuánto impacto tiene la actualización de los pesos en cada iteración. Si el *learning rate* es muy pequeño, el proceso de optimización puede ser demasiado lento y puede quedarse atrapado en mínimos locales. Por otro lado, si el *learning rate* es demasiado grande, el proceso puede ser inestable y oscilar alrededor del mínimo global. En consecuencia, encontrar el valor adecuado de la tasa de aprendizaje es una tarea crítica en el diseño y entrenamiento de redes neuronales.

En cuanto a las funciones de pérdida, también conocidas como funciones de costo o funciones de error, como se ha comentado anteriormente, desempeñan un papel crucial en el proceso de entrenamiento. Estas funciones son las que proporcionan una medida cuantitativa de qué tan bien está funcionando el modelo. Algunas de las funciones de pérdida más utilizadas son:

- Error cuadrático medio (MSE): calcula el promedio de los cuadrados de las diferencias entre las predicciones y las etiquetas reales. Es especialmente útil en problemas de regresión.
- Entropía cruzada (*Cross-Entropy*): ampliamente utilizada en problemas de clasificación, mide la discrepancia entre las distribuciones de probabilidad real y pronosticada por el modelo.
- Error absoluto medio (MAE): Calcula el promedio de las diferencias absolutas entre las predicciones y las etiquetas reales. Otra opción para problemas de regresión.
- Pérdida de visagra (*hinge loss*): Empleado en clasificación binaria, penaliza las predicciones incorrectas y busca maximizar el margen entre clases.

- Pérdida de log-probabilidad (*log-likelihood loss*): Similar a la entropía cruzada, se utiliza en problemas de clasificación y mide la diferencia entre las distribuciones de probabilidad.

El progreso del perceptrón condujo al desarrollo del Perceptrón Multicapa, también conocido como Multilayer Perceptron (MLP). A medida que los investigadores profundizaban en la capacidad de los perceptrones individuales para resolver problemas de clasificación binaria, se hizo evidente que las limitaciones de un solo perceptrón eran insuficientes para abordar problemas más complejos y de mayor dimensionalidad.

El Perceptrón Multicapa, al introducir múltiples capas de neuronas interconectadas, permite a la red aprender relaciones más complejas y representar funciones no lineales. Estas capas intermedias, también llamadas capas ocultas, proporcionan una mayor capacidad de aprendizaje y generalización.

Cada capa en el MLP consiste en múltiples neuronas artificiales, y las conexiones entre estas neuronas están ponderadas por valores que se ajustan durante el proceso de entrenamiento. La entrada se propaga hacia adelante a través de la red, y cada neurona realiza una combinación lineal de sus entradas ponderadas, seguida de una función de activación no lineal. Esta estructura permite al MLP capturar relaciones no lineales en los datos, lo que lo convierte en un modelo muy flexible y poderoso.

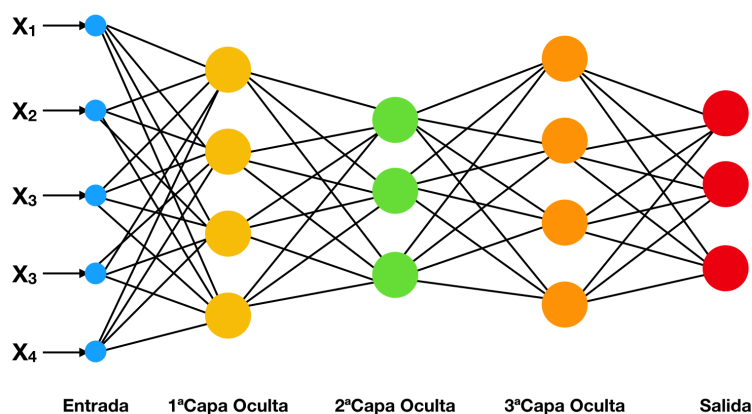


Figura 2.12: Componentes del perceptrón multicapa [48]

La optimización y entrenamiento de un Perceptrón Multicapa involucra procesos fundamentales como la propagación hacia adelante (*forward propagation*) y la propagación hacia atrás (*backward propagation*), que forman parte del algoritmo de retropropagación. Este algoritmo fue desarrollado en 1986 por David Rumelhart, Geoffrey Hinton y Ronald Williams con la publicación del artículo “Learning representations by back-propagating errors” [49]. Esta técnica marcó un antes y un después al permitir el entrenamiento eficiente de redes neuronales con múltiples capas.

En la etapa de *forward propagation*, los datos de entrada son procesados a través de las capas de neuronas, y la salida de la red se obtiene aplicando la función de activación adecuada a la combinación lineal de las entradas ponderadas por los valores ajustados de los pesos.

La función de activación en la capa de salida puede variar según el tipo de problema a resolver. Para la clasificación de múltiples clases, una elección común es la función Softmax. Esta función toma

un vector de valores y los convierte en una distribución de probabilidad, lo que permite interpretar la salida como las probabilidades de pertenencia a cada clase. Por otro lado, en problemas de clasificación binaria, la función Sigmoide es comúnmente utilizada en la capa de salida. La función Sigmoide también se aplica a la combinación lineal de entradas ponderadas y produce una salida en el rango entre 0 y 1, que puede interpretarse como la probabilidad de pertenencia a la clase positiva.

Después de la etapa de forward propagation, se realiza la retropropagación, backward propagation, que calcula los gradientes de la función de pérdida con respecto a los pesos de la red. Estos gradientes indican cómo deben ajustarse los pesos para minimizar la función de pérdida. Utilizando el algoritmo de Descenso por Gradiente Estocástico mencionado anteriormente, los pesos se actualizan en la dirección opuesta al gradiente, escalados por la tasa de aprendizaje.

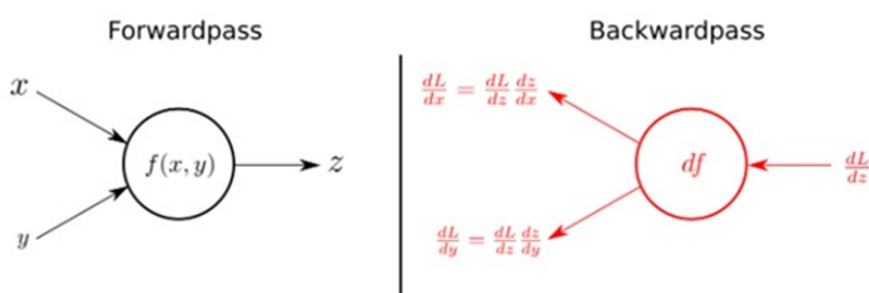


Figura 2.13: Proceso de retropropagación: forward-backward [50]

Este proceso iterativo de forward propagation y backward propagation se repite a lo largo de múltiples iteraciones permitiendo que el Perceptrón Multicapa ajuste sus pesos y bias para mejorar gradualmente su rendimiento en la tarea deseada.

2.2.3. Red neuronal convolucional

Las Redes Neuronales Convolucionales o ConvNets (CNN) han revolucionado el campo del procesamiento de imágenes y la visión por computadora gracias a su capacidad para capturar características y patrones visuales mediante filtros optimizados por entrenamiento. Las CNN son una subclase de las redes neuronales artificiales, diseñadas específicamente para procesar datos en forma de cuadrículas o matrices, como imágenes y vídeos.

El funcionamiento de las CNN se basa en la convolución, que simula la operación de detección de características visuales en el cerebro humano. En lugar de tratar cada píxel por separado, las CNN aplican filtros convolucionales, también conocidos como kernels, a la imagen de entrada. Estos kernels son matrices de valores que se deslizan sobre la imagen para detectar patrones específicos, como bordes, texturas y formas, en diferentes regiones de la imagen. Con la aplicación de diversos filtros, se generan mapas de características que destacan diferentes aspectos de la imagen. Estos mapas son combinados a través de capas adicionales de la red para llevar a cabo tareas como clasificación, detección de objetos o segmentación. Los tipos de capas más comunes en una arquitectura convolucional son los siguientes:

- Capas de convolución: compuestas por bloques de *kernels*, como se ha mencionado, estas capas tienen la función de extraer la información relevante de los datos. Las capas iniciales se dedican a extraer características de bajo nivel, mientras que las capas posteriores se enfocan en características de alto nivel. Llevan a cabo la convolución entre la matriz de entrada y cada kernel individual en múltiples iteraciones.

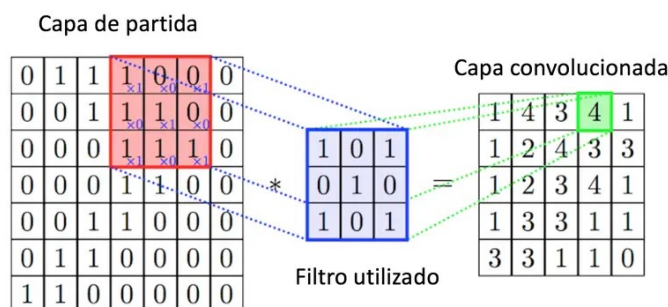


Figura 2.14: Operaciones de la capa de convolución [51]

- Capas de activación: Después de aplicar la convolución, es común aplicar una función matemática con una transformación no lineal para introducir la convergencia del modelo.
- Capas de diezmado (*pooling*): Estas capas reducen la dimensionalidad espacial de las características, lo que disminuye la cantidad de parámetros en la red y hace que sea más eficiente. El *pooling* generalmente se realiza tomando el máximo valor (*max pooling*) o el promedio (*average pooling*) de un grupo de valores en una región. Esto permite conservar las características más importantes mientras reduce la sensibilidad a pequeñas variaciones.

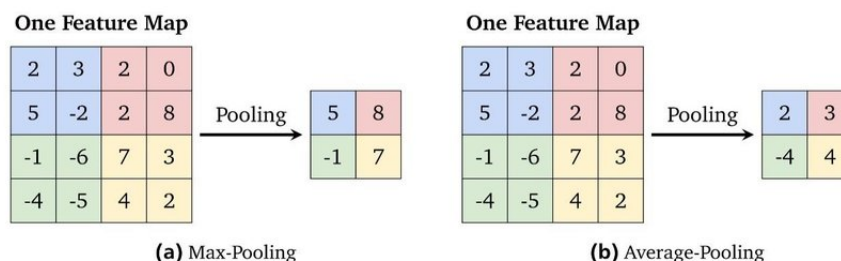


Figura 2.15: Ejemplo de capas de *pooling* [52]

- Capas de normalización y regularización: Estas capas son empleadas para mejorar la generalización y el rendimiento de la red. La capa de *batch normalization* normaliza las activaciones en cada capa, lo que acelera el entrenamiento y reduce el riesgo de sobreajuste. La capa de *dropout* es una técnica de regularización que desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para evitar la dependencia excesiva entre ellas.
- Capas totalmente conectadas (*fully-connected*): Tras estas capas, se puede agregar una o varias capas *fully-connected*. Estas capas son similares a las de una red neuronal tradicional, donde cada neurona está conectada a todas las neuronas de la capa anterior.

Una práctica común es utilizar un MLP en esta etapa, que consta de capas totalmente conectadas. Después de que la CNN haya extraído características significativas de los datos de entrada, el MLP se utiliza para realizar la clasificación final o tareas específicas de regresión.

Esta combinación de una CNN seguida de un MLP aprovecha las capacidades de las CNN para capturar patrones locales en datos de alta dimensionalidad, como imágenes, y luego utiliza un MLP para realizar tareas de alto nivel, como la clasificación de objetos en una imagen o la estimación de valores numéricos. Es una arquitectura ampliamente utilizada en aplicaciones de visión por computadora y procesamiento de imágenes.

- **Capa de aplanamiento (flatten):** La capa de Flatten transforma las matrices de características en un vector unidimensional. Esto es necesario para que la información pueda ser alimentada a las capas totalmente conectadas que siguen en la arquitectura de la red. La capa Flatten es un paso crucial para la transición de las características extraídas en las etapas anteriores hacia las decisiones finales de clasificación o regresión.

La secuencial combinación de estas capas determina la arquitectura de las CNN, que puede variar en complejidad y profundidad dependiendo del número de capas. La figura 2.16 ofrece un ejemplo visual de cómo estas capas se unen y forman una arquitectura de CNN.

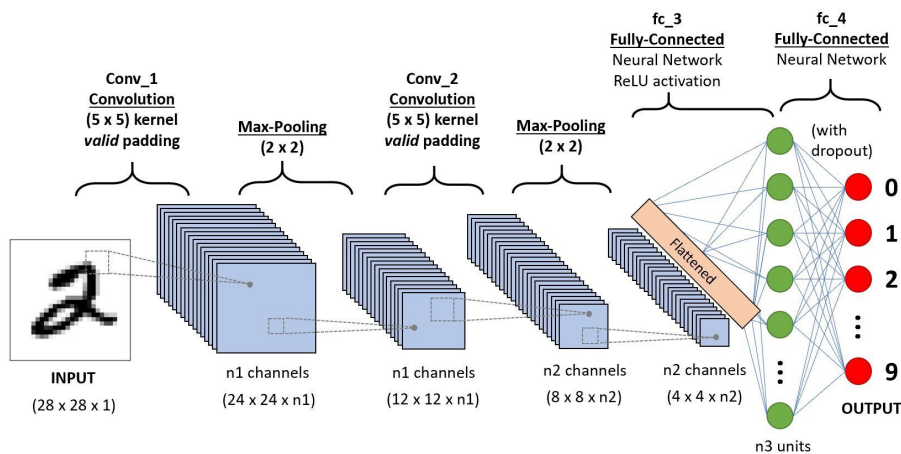


Figura 2.16: Red neuronal convolucional [53]

En las últimas décadas, varias arquitecturas de CNN han surgido, cada una diseñada para abordar desafíos específicos. La arquitectura LeNet-5, propuesta por LeCun et al. en 1998 [54], fue una de las primeras redes convolucionales utilizadas para el reconocimiento de dígitos escritos a mano. LeNet-5 consta de siete capas, incluyendo capas de convolución, capas de submuestreo y capas totalmente conectadas. Las capas de convolución son responsables de extraer características locales de las imágenes, mientras que las capas de submuestreo reducen la dimensionalidad y la complejidad de los datos. Estas características, junto con la introducción de la función sigmoide como función de activación, permitieron a LeNet-5 lograr resultados notables en la clasificación de dígitos manuales.

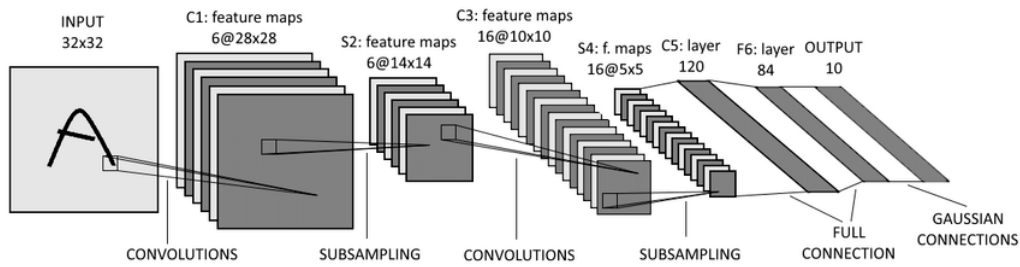


Figura 2.17: Arquitectura LeNet-5 [55]

Posteriormente, la arquitectura AlexNet, presentada por Krizhevsky et al. en 2012 [56], marcó un avance significativo en la clasificación de imágenes a gran escala y dio inicio al auge de las CNN. AlexNet fue diseñada para el desafío de ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [57] y consta de ocho capas, incluyendo cinco capas de convolución y tres capas totalmente conectadas. Una característica clave de AlexNet es el uso de la función de activación ReLU en lugar de la función sigmoide, lo que ayudó a mitigar el problema de la desvanecimiento por gradientes y acelerar el proceso de entrenamiento. Además, AlexNet introdujo la técnica de regularización conocida como “dropout”, que consiste en apagar aleatoriamente algunas neuronas durante el entrenamiento para evitar el sobreajuste.

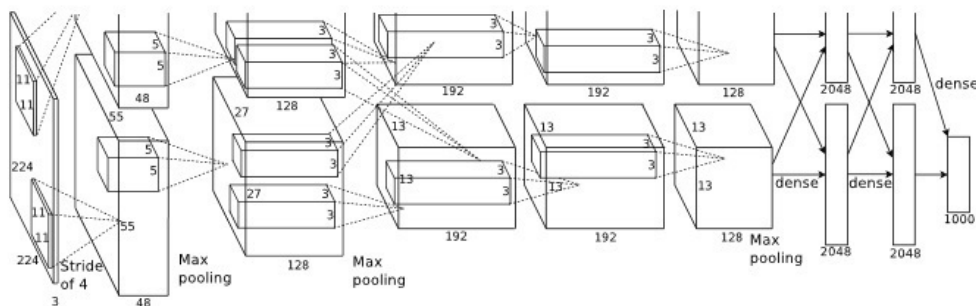


Figura 2.18: Arquitectura AlexNet [58]

Otra arquitectura influyente es VGG (Visual Geometry Group), propuesta por Simonyan y Zisserman en 2014 [59]. VGG se caracteriza por su profundidad, utilizando capas convolucionales de tamaño reducido (3x3) en todas sus capas, lo que permitió aprender características más locales y específicas. VGG16 y VGG19 son las variantes más conocidas, con 16 y 19 capas respectivamente. A pesar de su simplicidad en la arquitectura, VGG logró un rendimiento impresionante en la clasificación de imágenes debido a su profundidad y enfoque en la calidad de las representaciones.

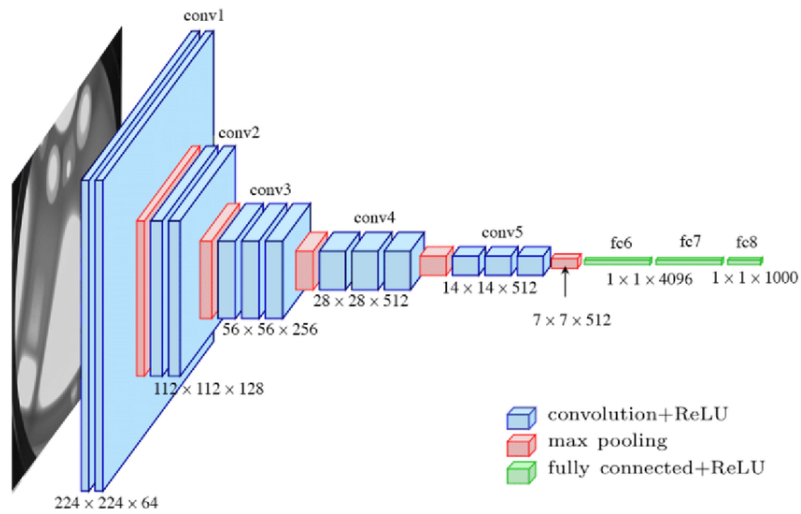


Figura 2.19: Arquitectura VGG16 [60]

La arquitectura GoogLeNet, también conocida como Inception, desarrollada por Szegedy et al. en 2014 [61], introdujo módulos de Inception que permiten a la red aprender características a diferentes escalas mediante la combinación de filtros de distintos tamaños en paralelo. Esto mejora la capacidad para capturar información tanto a nivel local como global en una imagen, lo que contribuye a una representación más rica y eficaz. A pesar de constar de 22 capas convolucionales, esta arquitectura demuestra una eficiente utilización de la memoria.

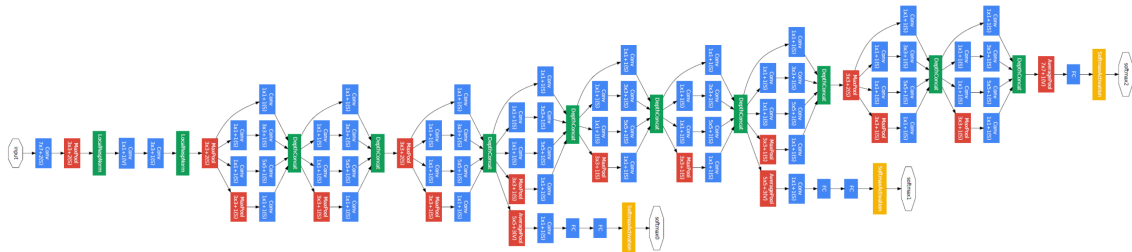


Figura 2.20: Arquitectura GoogLeNet [62]

Por último, la arquitectura ResNet, presentada por He et al. en 2015 [63], marcó un hito en la capacidad de entrenar redes neuronales extremadamente profundas. La principal innovación de ResNet radica en la introducción de bloques residuales que permiten entrenar redes aún más profundas y superar el problema de la desvanecimiento por gradientes.

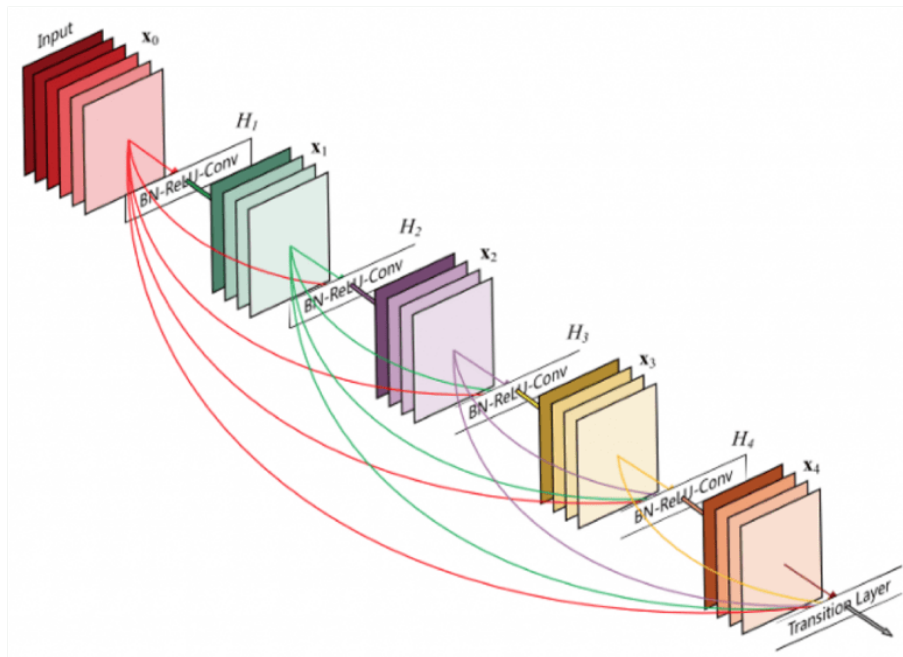


Figura 2.21: Arquitectura ResNet [64]

En resumen, las redes neuronales convolucionales son una potente herramienta en el procesamiento de imágenes, capaces de aprender y extraer características de manera automatizada. Sus arquitecturas especializadas y su capacidad para capturar patrones visuales complejos las hacen esenciales en una amplia gama de aplicaciones, desde la clasificación de imágenes hasta la detección de objetos y la segmentación.

Capítulo 3

Materiales

En este capítulo, se proporciona información sobre los recursos empleados en este proyecto. Se detalla el origen y las principales características de la base de datos utilizada, así como el software y hardware necesarios para desarrollar los algoritmos, entrenar las redes neuronales convolucionales y evaluar los resultados obtenidos.

3.1. Base de datos

La base de datos utilizada para el desarrollo de este proyecto ha sido proporcionado por el Departamento de Ingeniería Textil y Papelera de la Universitat Politècnica de València. Esta base de datos consta de un conjunto de 77 imágenes de tejidos con microcápsulas adheridas mediante técnicas de pulverización e impregnación.

La obtención de estas imágenes ha sido llevada a cabo mediante una meticulosa observación de los tejidos bajo un microscopio de última generación, el Phenom Microscope de la compañía FEI, con sede en Hillsboro, Oregón, Estados Unidos.



Figura 3.1: Phenom Microscope de la compañía FEI [65].

El procedimiento para obtener estas imágenes ha sido minucioso y preciso. Para ello, se colocaron

las muestras de tejido sobre un portamuestras estándar y se recubrieron con una mezcla especial de oro y platino, garantizando así una excelente calidad en los resultados visuales. Posteriormente, estas muestras de tejido tratadas se sometieron a un análisis bajo el microscopio con una aceleración de voltaje de 10kV, junto con un aumento óptimo, permitiendo así capturar cada detalle de las microcápsulas presentes.

La captura de estas imágenes se ha llevado a cabo con el fin de comprender a fondo la distribución de estas microcápsulas en los tejidos. Pero con la falta de herramientas lo suficientemente avanzadas para analizar en profundidad dicha distribución no se ha podido concluir este objetivo. Aquí radica el propósito fundamental de este proyecto.

Las imágenes son de vital importancia para lograr el conteo automatizado de microcápsulas, ya que son una parte fundamental del sistema desarrollado.



Figura 3.2: Ejemplo de la base de datos, imagen microscópica de un tejido con microcápsulas

En el capítulo 4 se describe en detalle la preparación de estas imágenes (preprocesado) para el entrenamiento del modelo de IA desarrollado.

3.2. Software

En el desarrollo de este proyecto, el software es empleado para el proceso de análisis de imágenes y la implementación de técnicas de IA. La programación y depuración de los modelos han sido realizadas en Python, un lenguaje de programación versátil y ampliamente utilizado en el campo de la IA.

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito, conocido por su legibilidad y simplicidad de sintaxis. Fue creado a finales de la década de 1980 por Guido van Rossum y desde entonces ha ganado una amplia popularidad en diversas áreas de desarrollo de software y ciencia de datos. Python es reconocido por su amplia gama de bibliotecas y herramientas especializadas [66]. Entre las librerías imprescindibles para las aplicaciones relacionadas con la IA, se han empleado las siguientes:

- Numpy: Una librería de uso libre para el cálculo numérico y operaciones con matrices, fundamental en el procesamiento de datos y el análisis de imágenes [67].
- Pandas: Una librería también de uso libre que se utiliza para el manejo y análisis de datos estructurados, lo que resulta útil en la organización y manipulación de conjuntos de datos para el proyecto [68].
- PyTorch: Un popular *framework* de aprendizaje profundo que permite la construcción y entrenamiento de redes neuronales de forma eficiente, facilitando la implementación de algoritmos de IA [69].
- Segmentation models: esta librería se enfoca específicamente en la segmentación de imágenes, proporcionando modelos preentrenados y herramientas para abordar tareas de segmentación [70].
- TiffFile: una herramienta útil para leer y escribir imágenes en formato TIFF, común en el campo de la investigación científica y biomédica. En el cual, se encuentran las imágenes de la base de datos [71].
- OpenCV: una librería de código abierto, para el procesamiento de imágenes esencial en la manipulación y análisis de las imágenes utilizadas en el proyecto [72].
- Albumentations: una biblioteca ampliamente reconocida y utilizada en el ámbito del aumento de datos, permitiendo realizar transformaciones avanzadas y personalizadas en conjuntos de imágenes [73].

En cuanto al etiquetado de las imágenes, se han utilizado Roboflow, una herramienta versátil para el etiquetado de imágenes y creación de conjuntos de datos. Gracias a la API de anotación automática ha sido posible etiquetar los grandes volúmenes de microcápsulas, ya que etiqueta automáticamente.

Para el preprocesado de las imágenes, se ha empleado MATLAB, un entorno de programación y software ampliamente reconocido en el campo del procesamiento de imágenes y la visión por computadora. MATLAB ofrece una gama completa de herramientas y funciones que facilitan la manipulación, mejora y análisis de imágenes [74].

Además, para el entrenamiento de las redes neuronales convolucionales en servidores proporcionados por CVBLab, se ha empleado MobaXterm, una herramienta que permite la gestión remota y el acceso a servidores Linux de manera sencilla y eficiente [75].

El desarrollo de la aplicación web ha sido posible gracias al uso del *framework* Vue.js. Vue.js es un framework de JavaScript utilizado para el desarrollo del frontend de aplicaciones web. Este framework permite crear interfaces de usuario interactivas y dinámicas de manera eficiente y modular. Se centra en la creación de componentes reutilizables y ofrece una estructura organizada para gestionar la lógica de la interfaz, el enrutamiento entre diferentes vistas y la administración del estado de la aplicación [76]. Entre las librerías para aplicaciones de este tipo algunas de las empleadas han sido:

- Vuetify: es una biblioteca de componentes de interfaz de usuario para Vue.js. Proporciona componentes de alta calidad y completamente personalizables, como botones, barras de navegación, etc. Estos componentes facilitan la creación de una interfaz de usuario moderna

y atractiva. Vuetify también ofrece un conjunto de utilidades que simplifican el diseño y la interacción de elementos [77].

- Axios: es una biblioteca JavaScript que se utiliza para realizar solicitudes HTTP desde un navegador. Proporciona una interfaz simple y fácil de usar para interactuar con servicios web y API, lo que facilita la obtención y el envío de datos desde y hacia un servidor [78].
- Exceljs: es una biblioteca de JavaScript que permite crear, leer y manipular archivos de hojas de cálculo de Excel en formato XLSX en el navegador. Te permite generar hojas de cálculo dinámicas y personalizadas con contenido estructurado, fórmulas y formato [79].
- Openseadragon: es de código abierto y se utiliza para crear visores de imágenes de alto rendimiento y capacidad de zoom. Está especialmente diseñada para mostrar imágenes grandes, en este caso las imágenes textiles de alta resolución, y permite a los usuarios explorar y ampliar regiones específicas de la imagen de manera fluida [80].
- Body-parser: es un *middleware* utilizado para analizar y procesar los cuerpos de las solicitudes HTTP. Facilita la extracción de datos enviados desde formularios y solicitudes POST [81].

Por último, para el desarrollo del servicio de IA se ha utilizado Flask, un *microframework* de desarrollo web para Python que se centra en la simplicidad y la flexibilidad. Proporciona las herramientas esenciales para construir aplicaciones web de manera rápida y eficiente [82].

3.3. Hardware

Para la realización de este proyecto, se ha requerido la utilización de diversas infraestructuras de computación de altas prestaciones informáticas proporcionadas por el grupo de investigación en el que se ha realizado el trabajo, CVB Lab.

CVB Lab dispone de seis servidores i7 @4.20GHz con 32GB de RAM que albergan potentes GPUs. En concreto, (1x) NVIDIA A100 Tensor Core GPU de 40GB, (4x) tarjetas gráficas NVIDIA Titan XP y (4x) tarjetas gráficas NVIDIA Titan V. El grupo también dispone de dos servidores NAS (Synology DS416 y Synology DS918+) en los que se alojan los datos y la documentación de los diferentes proyectos que se desarrollan con 32 y 16 TB de capacidad, respectivamente. Para realizar la fase de entrenamiento de este proyecto, sólo ha sido necesario hacer uso de uno de los servidores i7 @4.20GHz con 32 GB de RAM, su respectiva GPU NVIDIA Titan V y los servidores NAS.

Capítulo 4

Sistema de detección

En este capítulo se desglosará el proceso del sistema de conteo de microcápsulas. En primer lugar, el tratamiento aplicado a las imágenes de la base de datos, preprocesado y transformación de imágenes. En segundo lugar, la elección de la arquitectura de red neuronal convolucional específica que ha sido adaptada para abordar este problema así como la elección de hiperparámetros óptimos. Por último, la etapa de postprocesamiento, donde se aplicarán transformaciones a los resultados de la red.

4.1. Metodología

La metodología empleada en este proyecto se basa en una serie de pasos diseñados para la detección y segmentación precisa de microcápsulas en imágenes microscópicas de tejidos textiles. Este proceso se lleva a cabo mediante el uso de técnicas de procesamiento de imágenes y aprendizaje profundo a través de redes neuronales convolucionales. En la figura 4.1 se presenta un diagrama de bloques general del proceso.

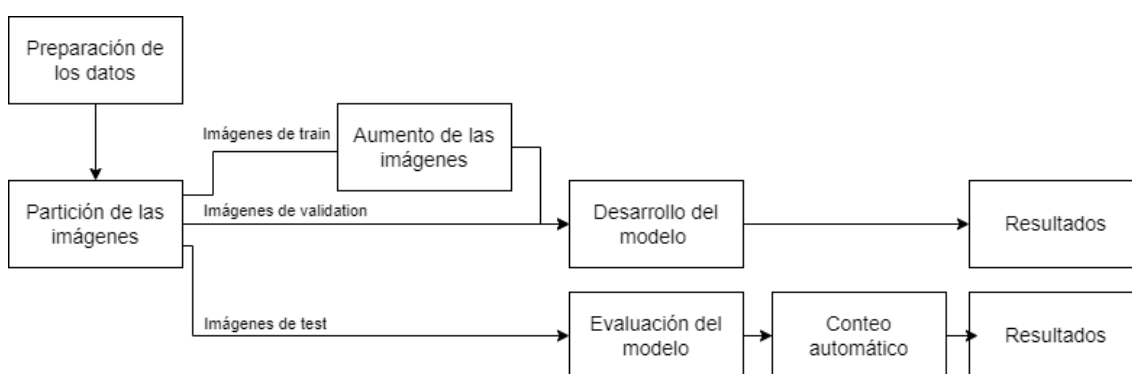


Figura 4.1: Diagrama de bloques de la metodología del sistema de detección

4.1.1. Preprocesado de imágenes

Para lograr la adecuación de las imágenes de la base de datos para su utilización en el proyecto, se han recortado las imágenes mediante el empleo de funciones de Matlab, el resultado final se aprecia en la figura 4.2. Esta acción puntual consiste en la eliminación de la porción inferior de las imágenes, que contiene detalles relacionados con ciertos parámetros de captación de la imagen que está siendo analizada. Con esta medida (512x384), se persigue optimizar las imágenes para su posterior procesamiento y análisis.

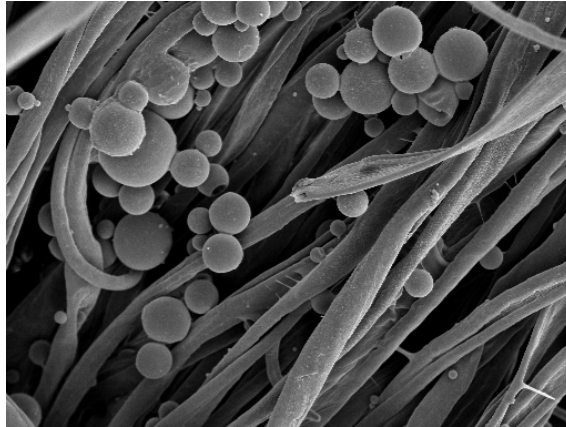


Figura 4.2: Imagen de la base de datos procesada

Una vez las imágenes microscópicas de alta resolución están recortadas, se inicia el proceso de generación del *ground truth* de las imágenes. Este proceso es vital para el entrenamiento de la red diseñada.

El *ground truth*, en este contexto, refiere a una segmentación manual que se emplea como punto de referencia, y frente a la cual se contrastan los resultados obtenidos por un método automático. En este sentido, para capacitar adecuadamente a esta red neuronal convolucional (CNN), es crucial proporcionarle en primera instancia información detallada sobre la localización de las microcápsulas. Esta información le permitirá aprender los patrones inherentes a estas microestructuras y, en consecuencia, habilitará su capacidad para identificarlas en futuras instancias.

La creación del *ground truth* ha involucrado dos etapas esenciales:

- Generación de archivos XML: se ha hecho uso de la herramienta Roboflow [83] para generar los archivos XML que contienen la información sobre la localización, el tamaño y la forma de cada una de las microcápsulas.
- Generación de máscaras: a partir de esta información, se ha desarrollado un código que extrae las características de los archivos XML y genera máscaras correspondientes a cada una de las microcápsulas, estableciendo así una representación 2D de su ubicación y forma en las imágenes, como se puede observar en la figura 4.3.

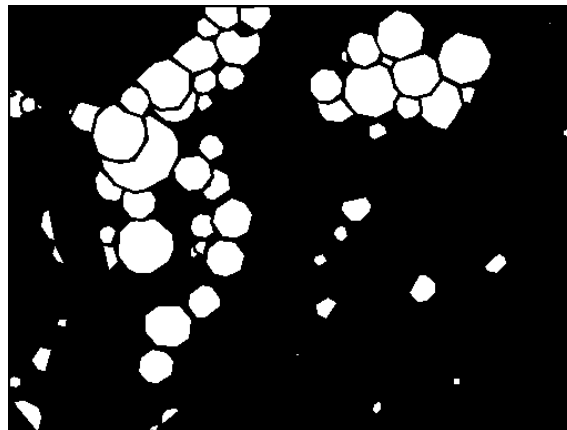


Figura 4.3: Ejemplo de *ground truth*

4.1.2. Partición de imágenes

Seguidamente de completar el preprocesamiento de las imágenes y llevar a cabo la creación del *ground truth*, es indispensable realizar una división de la base de datos utilizada en tres subconjuntos esenciales para el adecuado entrenamiento y validación de una red neuronal. Estos subconjuntos de imágenes se emplean en diversas fases del proceso y son conocidos como conjuntos de entrenamiento, validación y testeo.

En el conjunto de entrenamiento, las imágenes son empleadas para instruir al modelo y permitirle interiorizar las características y patrones inherentes en los datos. Con cada iteración de entrenamiento, las mismas imágenes son presentadas a la red neuronal, permitiendo que el modelo ajuste sus parámetros y comprenda las características sutiles.

El conjunto de validación, independiente del conjunto de entrenamiento, se utiliza para evaluar el rendimiento del modelo durante el proceso de entrenamiento. Esto proporciona información valiosa para ajustar los hiperparámetros y las configuraciones del modelo. Mientras el modelo se entrena con el conjunto de entrenamiento, la evaluación se realiza en el conjunto de validación, ayudando a prevenir el sobreajuste del modelo.

Finalmente, el conjunto de prueba entra en juego una vez finalizado el proceso de entrenamiento. Este conjunto independiente de datos se utiliza para evaluar el rendimiento final del modelo. Proporciona una medida imparcial de la precisión, exactitud y otros indicadores de rendimiento del modelo.

Para la red diseñada en este proyecto, se ha decidido dividir la base de datos en tres subconjuntos siguiendo una asignación aleatoria. El 80 % de las imágenes se destinan al conjunto de entrenamiento y validación. Dentro de este conjunto, el 80 % se utiliza para entrenamiento, mientras que el 20 % se reserva para validación. El 20 % restante se asigna al conjunto de test. Como resultado, el conjunto de datos original, compuesto por 77 imágenes, se divide el número de imágenes como se muestra en la figura 4.4.

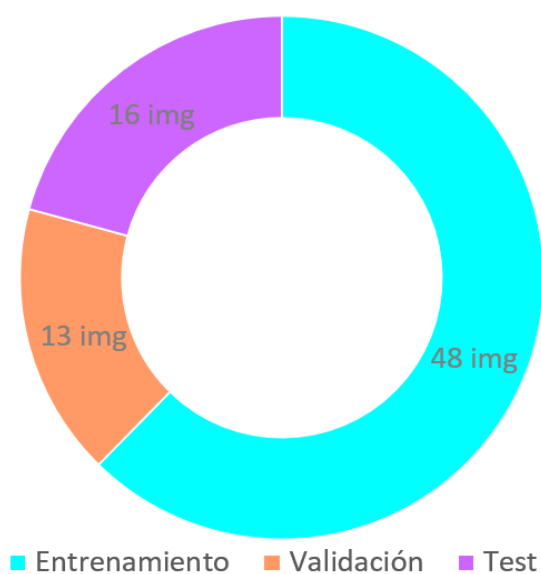


Figura 4.4: División de imágenes en el proyecto

Técnicas de aumento de datos

El aumento de datos o *data augmentation*, una técnica ampliamente utilizada en el campo de la IA, desempeña un papel esencial para aumentar la cantidad de datos disponibles para el entrenamiento del modelo mediante la generación de nuevas muestras a partir de las existentes.

En el contexto de este proyecto, se han aplicado diversas técnicas de *data augmentation* para mejorar la calidad y la diversidad del conjunto de entrenamiento. Esto no solo amplía la cantidad de datos disponibles, sino que también ayuda a abordar desafíos como el desequilibrio en las clases de objetos. La biblioteca de transformación que se ha utilizado es Albumentations, una herramienta con gran variedad de aumentos de datos.

Entre las técnicas de *data augmentation* implementadas se encuentran la horizontal y vertical *flip* (volteo horizontal y vertical) con una probabilidad del 50 % cada una. Estas transformaciones generan versiones reflejadas de las imágenes, lo que añade variabilidad al conjunto de datos y enriquece el aprendizaje de la red neuronal.

La técnica de horizontal *flip* se puede observar en la figura 4.5, esta técnica consiste en reflejar la imagen a lo largo de un eje vertical, de modo que los objetos que originalmente estaban a la izquierda ahora se encuentran a la derecha y viceversa. De igual manera, la técnica de vertical *flip* consiste en reflejar la imagen a lo largo del eje horizontal como se puede observar en la figura 4.6. Esto se realiza aplicando una transformación a nivel de píxel a la imagen.

Este proceso implica seleccionar aleatoriamente si se aplicará o no la transformación. La probabilidad de aplicación se especifica y en este caso es un 50 %. Si se decide aplicar el volteo, se invierten las posiciones de los píxeles de la imagen a lo largo del eje correspondiente.

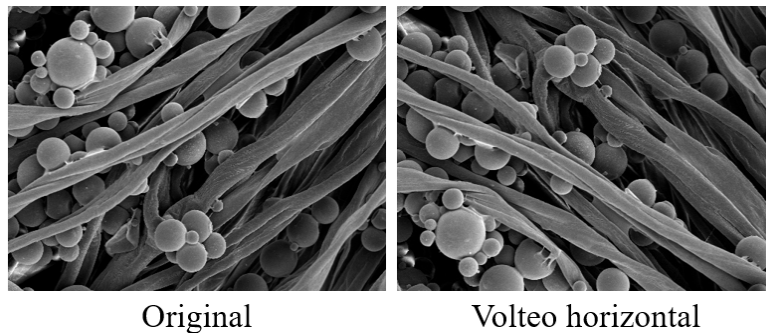


Figura 4.5: Técnica de volteo horizontal

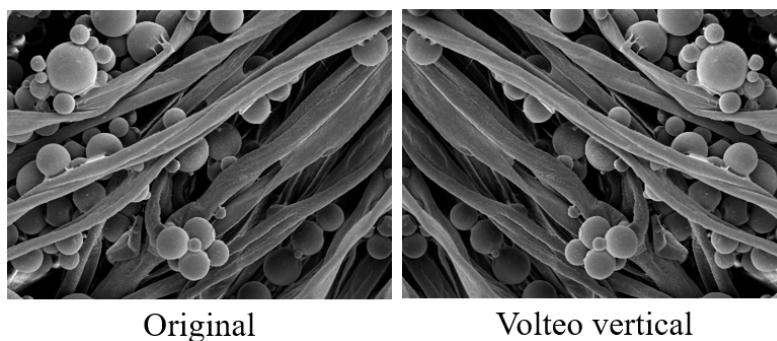


Figura 4.6: Técnica de volteo vertical

Otra de las técnicas empleadas es el cambio de intensidad, un proceso que implica ajustar el brillo y el contraste de la imagen de forma aleatoria. Esta transformación se aplica a nivel de píxel y se ajusta tanto el brillo como el contraste dentro de rangos especificados de manera aleatoria. En la figura 4.7 se puede observar dos ejemplos de esta técnica

Para el brillo, un valor aleatorio se suma a cada píxel de la imagen. Este valor puede estar en el rango del 70% al 130%. Esto permite que la imagen sea más brillante o más oscura en cada aplicación de la transformación, lo que simula condiciones de iluminación variables.

Para el contraste, un valor aleatorio se multiplica por cada píxel de la imagen. Al igual que con el brillo, este valor puede estar en el rango del 70% al 130%. Esto afecta la diferencia entre los niveles de color en la imagen, lo que se traduce en una mayor o menor saturación de los colores y en un cambio en la percepción visual.

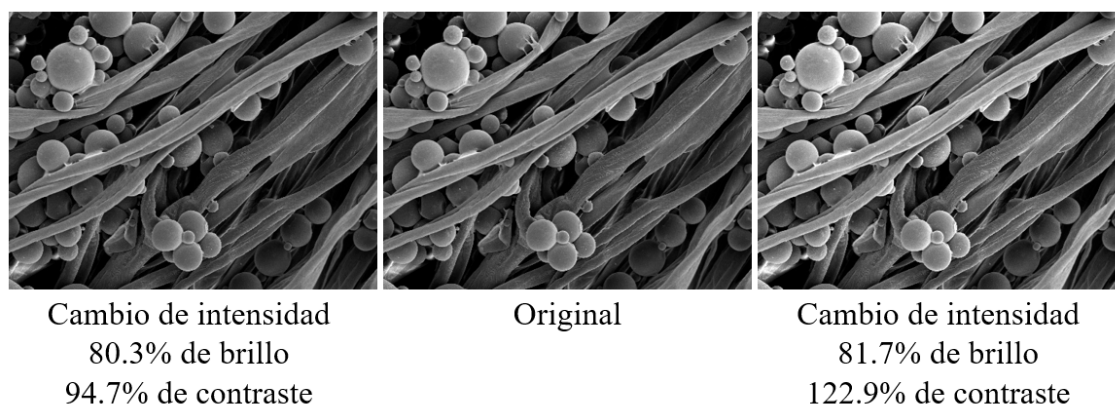


Figura 4.7: Técnica de cambio de intensidad

Al adoptar estas estrategias de *data augmentation*, se logra un conjunto de entrenamiento más completo, diversificado y balanceado, lo que finalmente contribuye a la mejora del rendimiento y la capacidad de generalización de la red neuronal en la tarea de segmentación semántica de microcápsulas en imágenes microscópicas.

4.1.3. U-Net

La U-Net es una arquitectura prominente en el ámbito de las redes neuronales convolucionales y fue propuesta en 2015 por Olaf Ronneberger, Thomas Box y Philipp Fischer [84]. Su diseño revolucionario nació con el propósito específico de abordar la segmentación de imágenes biomédicas, una tarea fundamental en el ámbito médico y de investigación. A través de su diseño, la U-Net ofrece una solución para la segmentación semántica. Esta técnica avanzada tiene como objetivo asignar una etiqueta semántica a cada píxel de una imagen. En esencia, busca dividir la imagen en segmentos y asignar a cada uno de estos segmentos una etiqueta que indique la categoría o clase a la que pertenecen los objetos en esa región. Esto permite una identificación precisa y rápida de las diversas estructuras y objetos presentes en la imagen.

La primera fase, llamada contractiva o codificador de la U-Net, sigue la estructura de una red neuronal convolucional convencional. En esta fase se encuentran los bloques compuestos por una capa convolucional, una capa de activación lineal y una capa de submuestro, conocidas como capas de *pooling*. Por lo general, la capa convolucional de cada bloque debe tener el doble de filtros que la del bloque anterior, mientras que las capas de submuestreo suelen reducir las dimensiones a la mitad, con un factor de redimensionado de $1/2$. Este proceso también se conoce como *downsampling*. Una vez que la imagen ha pasado por esta fase contractiva, se reduce a un vector de características, que se denomina vector de estado latente con toda la información extraída de la imagen de entrada.

La segunda fase llamada expansiva o decodificador adopta una estructura inversa en comparación a la fase anterior. En esta etapa, en lugar de usar capas de *pooling*, se implementan capas de interpolación, conocidas como capas *upsampling*. Estas capas de interpolación tienen la tarea de revertir las reducciones aplicadas en la fase inicial de la red, es decir, aumentan las dimensiones de la matriz de entrada en un factor de 2. Posteriormente, la salida del bloque correspondiente se concatena con la entrada del bloque actual y se somete a un proceso similar al realizado durante el *downsampling*.

Las dos fases de la U-Net trabajan en conjunto para extraer las características esenciales de la imagen de entrada y luego reconstruirla en sus dimensiones originales. La unión de estas dos fases da lugar a una forma de “U”, como se observa en la figura 4.8, permitiendo una segmentación eficiente al mismo tiempo que mantiene la integridad de las características relevantes en la imagen.

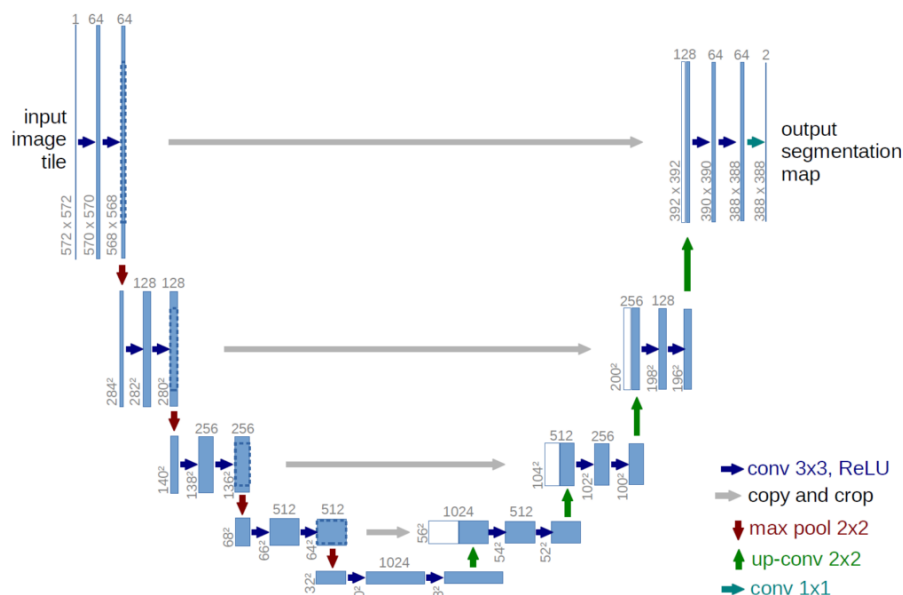


Figura 4.8: Arquitectura U-Net

El modelo U-Net diseñado en este proyecto se construye utilizando el codificador “efficientnet-b7”, un modelo pre-entrenado con pesos obtenidos del conjunto de datos “Imagenet”. El modelo tiene una entrada de 1 canal, imágenes en escala de grises, y una salida de 2 clases, donde cada clase corresponde a una región de la imagen, en este caso, microcápsula o fondo.

La estructura general de la combinación U-Net con “efficientnet-b7” puede describirse de la siguiente manera:

- **Codificador EfficientNet-B7:** El codificador (*encoder*) de la U-Net se toma de EfficientNet-B7. Esto significa que la parte del modelo que se encarga de extraer características de alto nivel de la imagen se toma del EfficientNet-B7 pre-entrenado, aprovechando su capacidad para capturar características complejas y generales.
- **Decodificador U-Net:** El decodificador (*decoder*) de la U-Net se utiliza para procesar las características extraídas por el codificador EfficientNet-B7. El decodificador consta de capas de muestreo ascendente (*up-sampling*) y concatena las características de diferentes niveles de resolución espacial. Esto permite que el modelo aprenda a realizar una segmentación precisa y restaure la resolución de la imagen a medida que se desplaza hacia la capa de salida.

EfficientNet-B7 es uno de los modelos más grandes de la familia EfficientNet, conocida por su eficiencia y rendimiento en tareas de visión por computadora. A medida que aumenta el número de versión de EfficientNet (de B0 a B7), generalmente aumenta el tamaño y la capacidad de representación de la red, lo que a su vez implica un aumento en la cantidad de parámetros. Su arquitectura se basa en principios de escalabilidad compuesta, lo que significa que se puede ajustar

simultáneamente el ancho, la profundidad y la resolución de la entrada a medida que avanza en capas. En la figura 4.9 se observa ejemplos de distintos escalados de EfficientNet-B7.

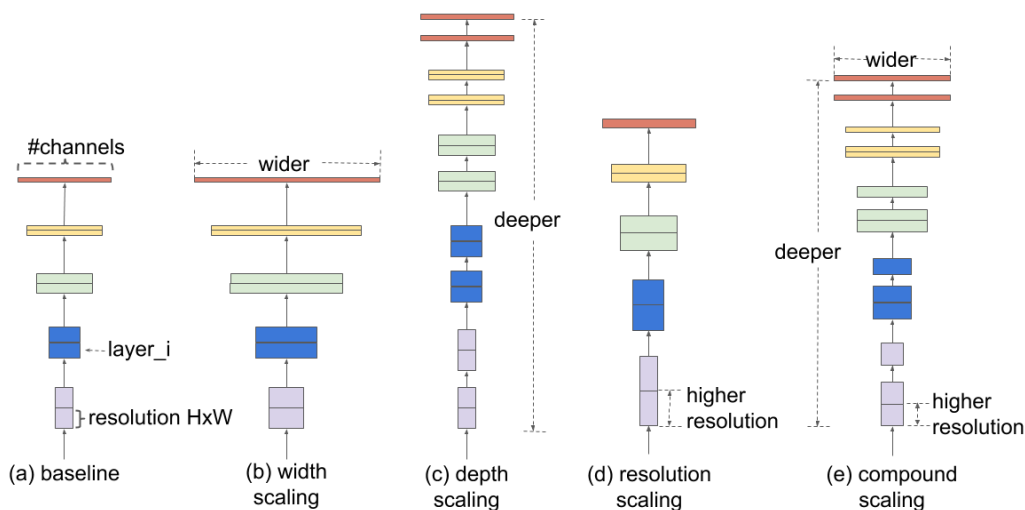


Figura 4.9: Principios de escalabilidad compuesta de EfficientNet-B7 [85]

Para comprender mejor este modelo, es importante explorar sus bloques principales:

- **Bloque inverso móvil (MBConv)**: los bloques MBConv, son los componentes fundamentales de EfficientNet-B7. Cada bloque MBConv consta de tres partes: una convolución 1x1 para reducir la dimensionalidad, una convolución 3x3 que realiza la convolución principal y otra convolución 1x1 para expandir la dimensionalidad nuevamente. Esta estructura reduce drásticamente la cantidad de cálculos necesarios mientras permite la captura de características complejas en imágenes.
- **Escala compuesta**: la arquitectura de EfficientNet-B7 se compone de varias etapas, cada una con su propia configuración de bloques MBConv. A medida que avanzamos en estas etapas, la arquitectura ajusta la escala aumentando tanto el ancho como la profundidad de la red. Esto permite la captura de características de diferentes escalas y complejidades.
- **Atención y conexiones residuales**: EfficientNet-B7 utiliza mecanismos de atención y conexiones residuales para mejorar la captura y propagación de características importantes a lo largo de la red. Las conexiones residuales ayudan a evitar problemas de desvanecimiento de gradientes, permitiendo un entrenamiento más estable y profundo.
- **Regularización**: para evitar el sobreajuste durante el entrenamiento, EfficientNet-B7 utiliza técnicas de regularización, como la regularización L2 y la eliminación estocástica (*dropout*).
- **Función de activación Swish**: En lugar de utilizar la función de activación ReLU estándar, EfficientNet-B7 utiliza la función Swish, que ha demostrado ser más eficaz en la mayoría de los casos. Swish es una función de activación suave que ayuda a que la red aprenda características más ricas.
- **Convoluciones direccionales**: EfficientNet-B7 incorpora convoluciones direccionales para mejorar la representación de características en direcciones específicas, lo que puede ser útil en la detección de patrones y orientaciones en imágenes.

Estos bloques forman la arquitectura de este modelo, la cual se puede observar en la figura 4.10.

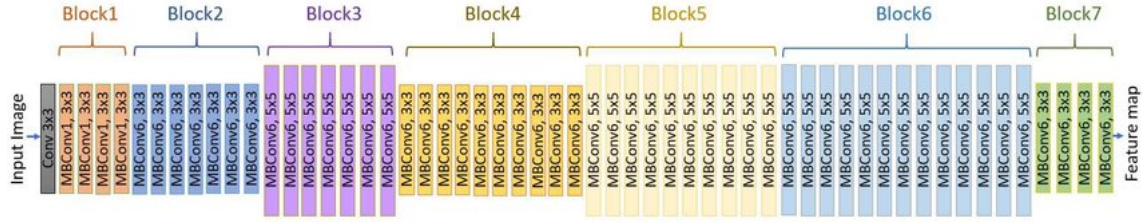


Figura 4.10: Arquitectura de EfficientNet-B7 [86]

Este modelo ha sido pre-entrenado en el conjunto de datos ImageNet, que contiene una amplia variedad de imágenes etiquetadas que abarcan miles de categorías. Esta fase de pre-entrenamiento permite a EfficientNet-B7 aprender representaciones visuales útiles y generalizables de una amplia gama de objetos y patrones visuales. Como resultado, el modelo está equipado para detectar bordes, texturas, formas y características visuales en imágenes textiles, lo que lo convierte en una elección sólida para la tarea de segmentación de este proyecto.

La configuración de los parámetros de entrenamiento desempeña un papel esencial en el rendimiento y la eficacia del modelo. Cada uno de estos parámetros se selecciona y se ajusta de acuerdo con las necesidades específicas, con el objetivo de lograr resultados precisos y una convergencia eficiente. A continuación, se detalla los parámetros clave utilizados en el proyecto:

- Tasa de aprendizaje (*learning rate*): la tasa de aprendizaje se establece en 0.00005 para controlar la velocidad a la que el modelo se ajusta a los datos durante el entrenamiento.
- Número de épocas (*epochs*): se ha entrenado la red durante 40 épocas. En otras palabras, el modelo iteró sobre todo el conjunto de datos de entrenamiento 40 veces. Este número de épocas es fundamental para permitir que el modelo aprenda las características de las micro-cápsulas en los textiles a medida que se ajusta a los datos.
- Decaimiento exponencial de la tasa de aprendizaje (*LR Exponential Decay*): se implementa un decaimiento exponencial de la tasa de aprendizaje después de la época 31. Esta estrategia es útil para afinar el modelo y mejorar la convergencia a medida que el entrenamiento avanza. El decaimiento exponencial gradualmente reduce la tasa de aprendizaje a lo largo del tiempo, lo que puede ser beneficioso para evitar oscilaciones en el entrenamiento.
- Función de pérdida (*loss function*): se utiliza la función de pérdida de entropía cruzada (*CrossEntropyLoss*) como la métrica principal para medir la diferencia entre las predicciones del modelo y las máscaras verdaderas de segmentación. Para cada píxel en la imagen de entrenamiento, se calcula la pérdida mediante la siguiente fórmula:

$$CELoss = - \sum_i [y_{true}^{(i)} \cdot \log(y_{pred}^{(i)}) + (1 - y_{true}^{(i)}) \cdot \log(1 - y_{pred}^{(i)})] \quad (4.1)$$

En esta fórmula:

$y_{true}^{(i)}$ representa la etiqueta verdadera para el píxel i , $y_{pred}^{(i)}$ representa la predicción del modelo para el píxel i . Y el sumatorio se realiza sobre todos los píxeles en la imagen de entrenamiento para calcular la pérdida total de entropía cruzada.

Durante el entrenamiento de modelos de segmentación, se minimiza la entropía cruzada ajustando los parámetros del modelo mediante algoritmos de optimización. El objetivo es encontrar los parámetros que minimizan esta función de pérdida para que el modelo haga predicciones precisas en la tarea de segmentación binaria.

Esta función de pérdidas se caracteriza por penalizar fuertemente las predicciones incorrectas, especialmente cuando se alejan mucho del valor real. Esto significa que cuanto más se acerque la máscara de segmentación predicha a la máscara real, menor será el valor de la entropía cruzada, lo que indica un mejor rendimiento del modelo en la tarea de segmentación binaria.

- Optimizador: se ha escogido el optimizador Adam (*Adaptive Moment Estimation*) para ajustar los pesos del modelo durante el entrenamiento. El optimizador Adam es una extensión del optimizador de SGD que combina las ventajas del método de gradiente descendente con momento y el método de gradiente descendente con adaptación de la tasa de aprendizaje. Esto lo hace muy eficaz en la optimización de modelos de redes neuronales.

Al igual que el SGD con momento (*momentum*), Adam utiliza un término de momento que ayuda a acelerar la convergencia. Esto permite que el optimizador “recuerde” las direcciones anteriores y se mueva más suavemente hacia el mínimo global.

Por una parte, Adam ajusta automáticamente la tasa de aprendizaje para cada parámetro de la red, lo que significa que la tasa de aprendizaje es adaptativa y se puede ajustar por separado para cada peso en la red. Esto es útil para evitar que la optimización se estanque en mínimos locales o diverja debido a tasas de aprendizaje demasiado grandes.

Por otra parte, Adam utiliza estimaciones de primer y segundo orden de los momentos (la media y la varianza no centrada de los gradientes). Esto permite una adaptación más precisa de la tasa de aprendizaje.

Por último, Adam también incluye una forma de regularización llamada “descenso de peso” que puede ayudar a evitar el sobreajuste.

En conjunto, esta combinación de arquitecturas y parámetros ofrece una solución sólida y eficaz para la detección de microcápsulas en textiles. Aprovechando la capacidad de EfficientNet-B7 para la extracción de características y la precisión de U-Net en la segmentación, hemos construido una herramienta correcta para la aplicación.

4.1.4. Postprocesado de imágenes

Tras la fase de procesamiento a través de la red neuronal, se procede a realizar un conjunto de operaciones posteriores con el propósito de llevar a cabo el conteo de microcápsulas presentes en la imagen. La idea de este postprocesado es contar las microcápsulas incluyendo aquellas que por errores de segmentación están unidas en las máscaras. Este postprocesado se basa en las máscaras que han sido previamente predichas mediante la U-Net y se utilizan funciones de la biblioteca OpenCV. Las etapas de este proceso son las siguientes:

Se inicia con un refinamiento de las máscaras para eliminar cualquier tipo de ruido o imperfección que pueda haber surgido durante la fase de predicción. Para lograrlo, se aplican operaciones morfológicas, que son transformaciones geométricas en la imagen que mejoran la calidad de las máscaras. Estas operaciones se realizan mediante el uso de un kernel, en este caso, un kernel de

5x5, cuya elección está relacionada con el tamaño de los detalles que se desean modificar en las máscaras. Las operaciones morfológicas, como la dilatación y la erosión, permiten eliminar detalles no deseados y resaltar las características de las microcápsulas en la imagen.

Después de esta etapa, se lleva a cabo el cálculo de las áreas seguras correspondientes tanto al fondo como al primer plano de las microcápsulas. Para lograr esto, se calcula la transformada de distancia de una imagen, en este caso, se utiliza la distancia de Chebyshev, para determinar la distancia desde cada píxel en la imagen hasta el objeto más cercano.

Con la transformada de distancia calculada, se procede a identificar las regiones que no están claramente definidas como parte del fondo o el primer plano de las microcápsulas. Estas áreas se consideran como “regiones desconocidas”. Esta distinción entre regiones seguras y desconocidas se logra mediante la operación de resta entre las áreas de fondo y primer plano seguras en la transformada de distancia. Las regiones donde no se puede determinar claramente si pertenecen al fondo o al primer plano son marcadas en esta etapa para su posterior procesamiento en el algoritmo de cuencas hidrográficas (*watershed*).

La siguiente fase implica la asignación de etiquetas a los marcadores que destacan regiones específicas en las máscaras. Esta asignación de etiquetas es esencial para identificar de manera efectiva las áreas correspondientes al primer plano en las máscaras. Los marcadores juegan un papel fundamental en el proceso de segmentación siguiente. En las figuras 4.11 y 4.12 se ve reflejado estas etapas.

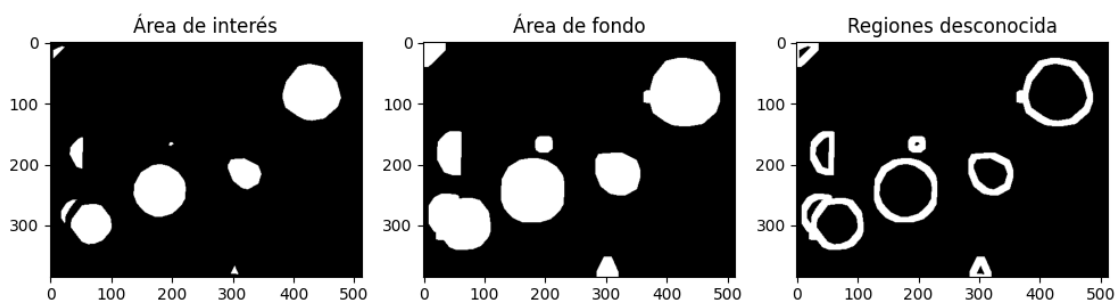


Figura 4.11: Proceso de postprocesado de la imagen i023

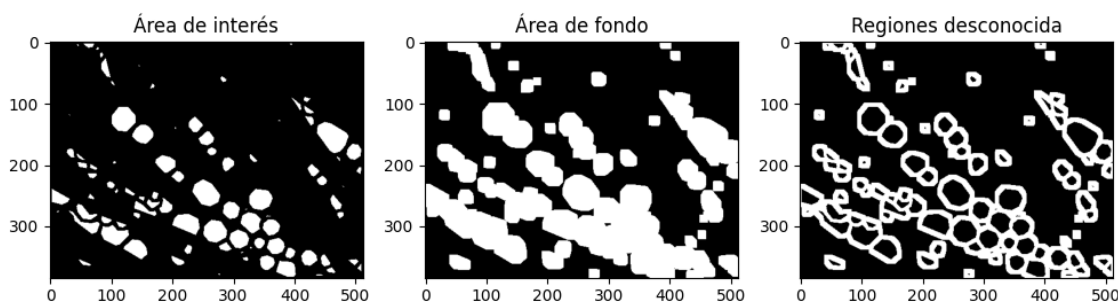


Figura 4.12: Proceso del postprocesado de la imagen i034

A continuación, se aplica el algoritmo de segmentación de cuencas hidrográficas a las máscaras. Este algoritmo es una técnica de procesamiento de imágenes que se inspira en la forma en que se forman las cuencas hidrográficas en un paisaje montañoso. El objetivo principal es dividir la ima-

gen en segmentos o cuencas discretas, donde cada segmento se asocia directamente a una entidad o región de interés, en este caso, microcápsulas.

Este algoritmo se basa en la idea de considerar la intensidad de los píxeles en la imagen como altitudes en un paisaje imaginario. Los puntos más bajos corresponden a valles y los puntos más altos a cimas. El algoritmo simula el llenado gradual de estos valles con agua, comenzando desde las cimas. A medida que el agua se llena en los valles, eventualmente se encuentran y forman bordes, marcando las divisiones naturales entre regiones.

El resultado final de aplicar el algoritmo de *watershed* es una partición clara de la imagen en regiones individuales, donde cada región se corresponde con una microcápsula. Estas regiones están delimitadas por límites precisos que siguen las características de la imagen, lo que facilita enormemente la identificación y el análisis de microcápsulas individuales en la imagen.

Finalmente, se completa el proceso con el conteo de microcápsulas. Esto se logra al determinar el número de segmentos o cuencas únicas que han sido obtenidas en el proceso de segmentación. Cada uno de estos segmentos representa una microcápsula detectada y cuenta como parte integral de la fase de análisis de imágenes. En las figuras 4.13 y 4.14 podemos observar una máscara respecto al resultado del algoritmo.

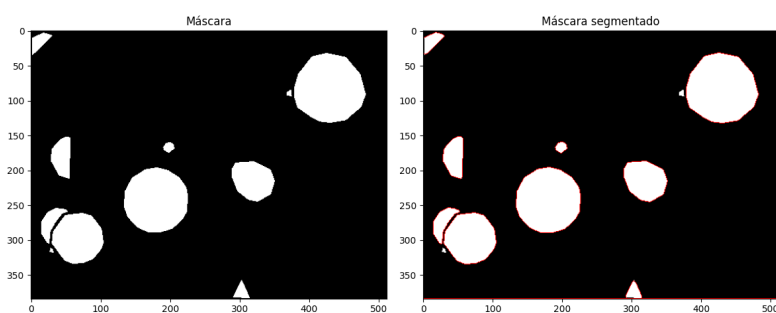


Figura 4.13: Postprocesado de la imagen i023

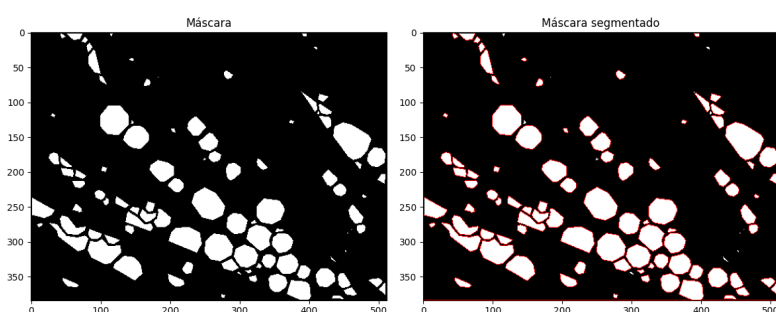


Figura 4.14: Postprocesado de la imagen i034

Como se ha explicado, el proceso de postprocesado después de la fase de la U-Net permite una evaluación cuantitativa del número de microcápsulas presentes en la imagen. A través de una serie de etapas, desde el refinamiento de las máscaras predichas hasta la segmentación y el conteo final, se logra una aproximación precisa al número de microcápsulas detectadas.

4.2. Resultados

Se procede a presentar los resultados del modelo de segmentación de microcápsulas en textiles mediante el uso de una red neuronal U-Net con EfficientNet-B7 como arquitectura base. Durante el proceso de desarrollo y entrenamiento de nuestro modelo, se llevaron a cabo evaluaciones para medir su desempeño en la tarea de segmentación de microcápsulas. Finalmente, se emplearán matrices de confusión y se calcularán parámetros como la precisión, la sensibilidad y la especificidad del modelo para la evaluación del mismo.

Generalmente en las tres etapas de desarrollo del modelo (entrenamiento, validación y test) hay una serie de métricas que se utilizan para evaluar y medir la calidad y eficacia de un modelo de segmentación en tareas de procesamiento de imágenes. En la tabla 4.1 podemos ver el resultado para nuestro modelo de las métricas que van a ser comentadas a continuación:

- *Accuracy* (Exactitud): la exactitud mide la proporción de predicciones correctas en el conjunto de prueba. En este caso, el modelo tiene una exactitud del 98 % en la etapa de entrenamiento, lo que indica que el 98 % de las predicciones en el conjunto de entrenamiento son correctas. Del mismo modo el modelo tiene un 97 % en el conjunto de validación y un 98 % para el conjunto de test.
- *Loss* (Pérdida): la pérdida en los datos de entrenamiento es baja (0.0464), este valor indica un buen ajuste del modelo a estas imágenes de entrada. En la validación, la pérdida es de 0.0871, con una diferencia de menos del 10 % en comparación con el entrenamiento. Estos datos sugieren que el modelo generaliza bien con nuevas imágenes y que el modelo no está sobreajustando y puede hacer predicciones precisas en nuevos ejemplos.
- *JC Score* (coeficiente de similitud de Jaccard): durante el entrenamiento, el modelo alcanza un *IoU score* de 0.81, lo que muestra una alta concordancia entre las máscaras predichas y las máscaras reales en el conjunto de entrenamiento. Para el conjunto de validación un 0.78 y para el del test un 0.69.
- *Recall* (exhaustividad): el modelo tiene una exhaustividad del 86 % en el conjunto de entrenamiento, un 76 % en el conjunto de validación y un 77 % en el conjunto de test, lo que muestra su capacidad para encontrar la mayoría de las microcápsulas en todos los conjuntos.
- *F1 Score*: en la etapa de entrenamiento el *F1 Score* tiene un valor de 0.8983. Al igual un 0.8744 y 0.7696 en el conjunto de validación y test respectivamente. Estos valores indican un buen equilibrio entre la capacidad del modelo para identificar las microcápsulas (precisión) y su capacidad para no perder microcápsulas (exhaustividad).

	Entrenamiento	Validación	Test
Accuracy	0.9815	0.9735	0.9815
Loss	0.0464	0.0871	–
IoU score	0.8153	0.7769	0.6944
Recall	0.8585	0.7559	0.7736
F1 Score	0.8983	0.8744	0.7696

Tabla 4.1: Métricas en las tres etapas del desarrollo para el modelo propuesto

Por otra parte, la matriz de confusión es una herramienta ampliamente utilizada para evaluar el rendimiento de una red en problemas de clasificación. Estas matrices son aplicables tanto en clasificación binaria como en clasificación multiclase. En la Figura 4.15 se presenta la estructura típica de una matriz de confusión en el contexto de la clasificación binaria.

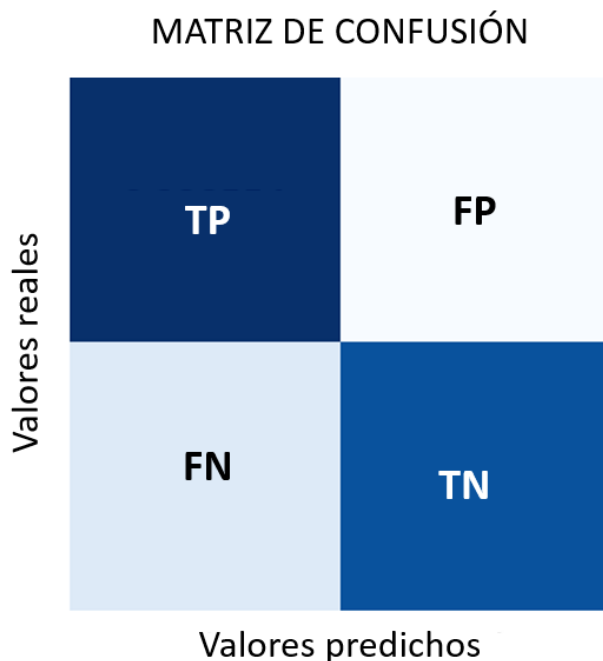


Figura 4.15: Estructura de una matriz de confusión binaria

En una matriz de confusión binaria, los componentes clave son:

- Verdaderos Positivos (TP): Representa los casos en los que el modelo clasificó correctamente las instancias positivas como positivas, en este caso, fondo como fondo.
- Falsos Positivos (FP): Indica las instancias negativas que el modelo clasificó incorrectamente como positivas, es decir, clasificó el fondo como microcápsula.
- Verdaderos Negativos (TN): Representa los casos en los que el modelo clasificó correctamente las instancias negativas como negativas, microcápsula como microcápsula.
- Falsos Negativos (FN): Indica las instancias positivas que el modelo clasificó incorrectamente como negativas, era microcápsula y se clasificó como fondo.

En la figura 4.16 los resultados corresponden a la etapa de validación de nuestro modelo de segmentación de microcápsulas en textiles. Durante esta fase crítica de evaluación, analizamos el rendimiento de la red neuronal en un conjunto de datos independiente, lo que nos permite evaluar su capacidad para generalizar y segmentar con precisión las microcápsulas en situaciones del mundo real. A continuación, presentamos la matriz de confusión que resume los resultados de esta validación.

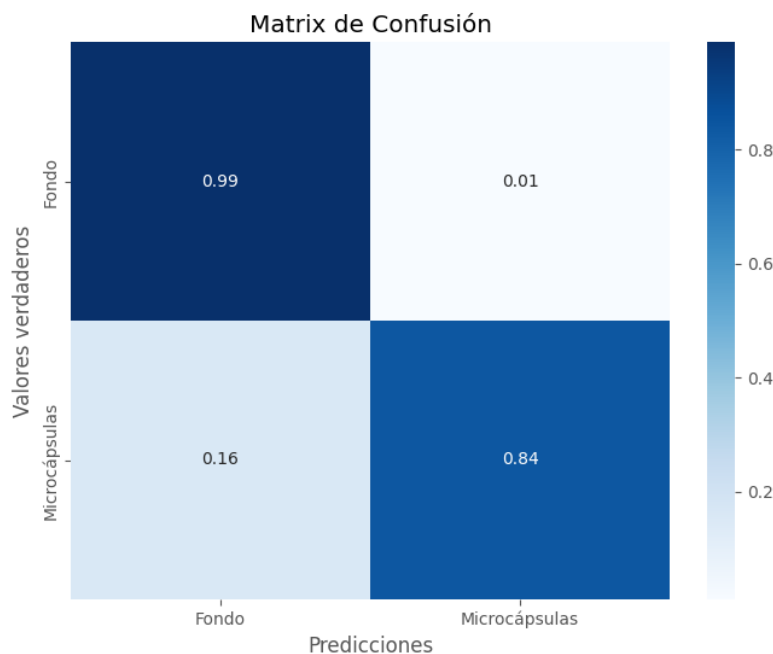


Figura 4.16: Matriz de confusión en validación

En el resultado de esta matriz de confusión se puede notar que el modelo ha logrado un alto número de verdaderos positivos (TP) con un valor de 0.99. Esto indica que casi el 99% de los píxeles pertenecientes a al fondo real han sido correctamente identificados como tales por el modelo.

Por otro lado, los falsos positivos (FP) son bastante bajos, representando solo un 0.01 del total de los píxeles del fondo real. Esto sugiere que el modelo ha perdido muy pocos píxeles de fondo real al clasificarlos incorrectamente como microcápsula.

Sin embargo, es importante notar que se han registrado algunos falsos negativos (FN) con un valor de 0.16, indicando que alrededor del 16% de los píxeles de microcápsulas se clasificaron incorrectamente como píxeles de fondo.

Por último, los verdaderos negativos (TN) representan las microcápsulas correctamente clasificadas como microcápsulas, con un valor de 0.84, lo que indica que el modelo ha acertado en la clasificación de microcápsulas en un 84% de las veces.

Estos resultados sugieren que el modelo tiene buen rendimiento en la detección de píxeles de microcápsulas, con una alta tasa de verdaderos positivos y un bajo número de falsos negativos. Tras la validación, se procede a la fase de pruebas finales. Durante esta etapa, se evalúa el rendimiento de la red neuronal en un conjunto de datos nuevo, lo que simula las condiciones del mundo real. En la tabla 4.2 se muestran los píxeles detectados de cada una de las imágenes de test.

Lo que se debe comparar es la cantidad de píxeles que son microcápsulas y ha detectado como microcápsulas comparado con el número de píxeles de microcápsulas que ha detectado como fondo, es decir, TN respecto a FN, y observamos que la TN siempre es más grande que la FN, lo que indica que la mayoría de microcápsulas las identifica correctamente.

Imagen	TP	FP	FN	TN
i003	189614.00	730.00	433.00	5831.00
i008	170718.00	2104.00	2791.00	209995.00
i017	183454.00	540.00	814.00	11700.00
i023	170201.00	1804.00	1460.00	23143.00
i031	184161.00	1163.00	2030.00	9254.00
i034	162755.00	3733.00	4702.00	25418.00
i037	182027.00	2198.00	1072.00	11311.00
i043	171197.00	2678.00	3211.00	19522.00
i048	196374.00	234.00	0.00	0.00
i052	175976.00	2259.00	2060.00	16313.00
i053	184263.00	810.00	1475.00	10060.00
i060	196552.00	56.00	0.00	0.00
i066	179783.00	2308.00	2203.00	12314.00
i068	189458.00	832.00	934.00	5384.00
i073	164897.00	1951.00	3560.00	26200.00
i074	165908.00	2675.00	5166.00	22859.00

Tabla 4.2: Predicciones del conjunto de test

A continuación, la figura 4.17 muestra la matriz de confusión de los resultados obtenidos durante la fase de test que demuestran la robustez y eficacia de nuestro enfoque en situaciones prácticas.

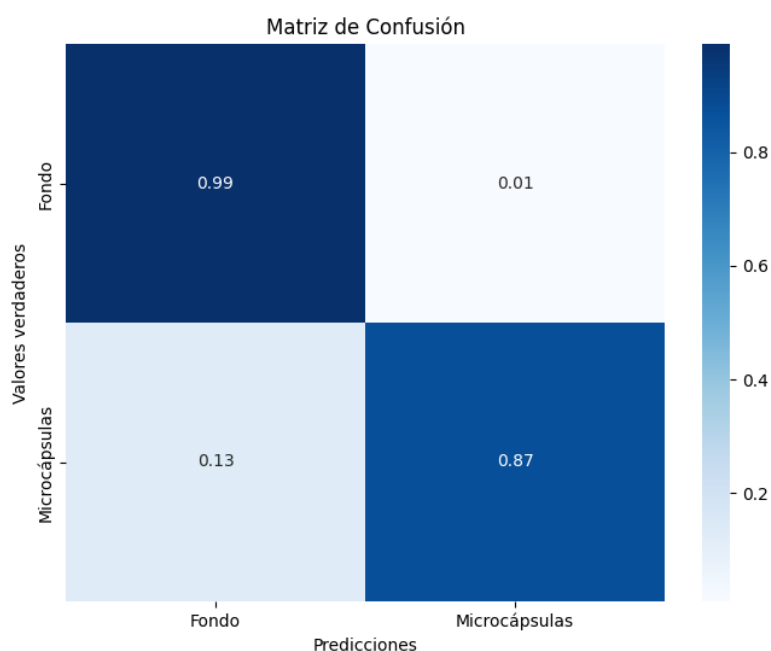


Figura 4.17: Matriz de confusión en test

Como se observa en las matrices, tanto de validación como de test, la mayoría de los valores se sitúan en la diagonal principal, lo que corresponde a verdaderos positivos y verdaderos negativos. Esto sugiere que la red está funcionando de manera efectiva. Sin embargo, para evaluar con mayor precisión el rendimiento de la U-Net que hemos diseñado, procederemos a explicar y calcular algunos parámetros críticos para ambos conjuntos.

La precisión (PPV), también conocida como valor predictivo positivo o tasa de verdaderos positivos, mide la proporción de predicciones positivas que son realmente verdaderas positivas. Este parámetro se calcula mediante la ecuación 4.2, es la razón entre los verdaderos positivos (TP) y el total de predicciones positivas (TP + FP).

$$PPV = \frac{TP}{TP + FP} \quad (4.2)$$

La sensibilidad (TPR) o recall mide la proporción de casos positivos reales que fueron correctamente identificados por el modelo mediante la ecuación 4.3. El recall se calcula como la razón entre los verdaderos positivos (TP) y el total de casos positivos reales (TP + FN).

$$TPR = \frac{TP}{TP + FN} \quad (4.3)$$

Por último, la especificidad (SPC), se basa en la tasa de verdaderos negativos, por lo que mide la proporción de casos negativos reales que fueron correctamente identificados como negativos por el modelo. Se calcula como la razón entre los verdaderos negativos (TN) y el total de casos negativos reales (TN + FP), este cálculo está reflejado en la ecuación 4.4.

$$SPC = \frac{TN}{TN + FP} \quad (4.4)$$

En la tabla de resultados 4.3 se presentan las métricas comentadas anteriormente con la que se puede evaluar el rendimiento del modelo diseñado:

	Precisión	Sensibilidad	Especificidad
Validación	0,990	0,860	0,989
Test	0,990	0,884	0,989

Tabla 4.3: Resultados de validación y test

Según estos resultados, tanto la precisión como la sensibilidad y la especificidad muestran valores altos. La alta precisión se considera a partir de un 80 %, esta refleja la capacidad de la red neuronal para identificar de manera precisa las microcápsulas, cometiendo errores en una fracción mínima de los casos. Por otro lado, la sensibilidad también muestra un rendimiento favorable, lo que indica que la red rara vez pasa por alto la detección de microcápsulas. Sin embargo, es importante destacar que en algunos casos, la red puede tener dificultades para identificar microcápsulas cuando estas se superponen, lo que explica la pequeña cantidad de falsos negativos. Finalmente, la especificidad se considera buena a partir de un 70 %, lo que sugiere que es altamente improbable que la red identifique erróneamente una microcápsula en áreas donde no existen. Este conjunto de resultados demuestra la eficacia y la robustez del modelo diseñado en la detección precisa de microcápsulas en diversas condiciones.

A continuación, se presentan algunos ejemplos de imágenes de microcápsulas procesadas por la red durante el proceso de test, figuras 4.18, 4.19, 4.20 y 4.21. Se incluye una comparación entre la imagen real y las máscaras predichas, que indican la ubicación de las microcápsulas detectadas. Además, se muestra una comparación entre el *ground truth* y las máscaras predichas. Esto proporciona una visualización de cómo el modelo se desempeña en la detección de microcápsulas en comparación con la ubicación real de las mismas.

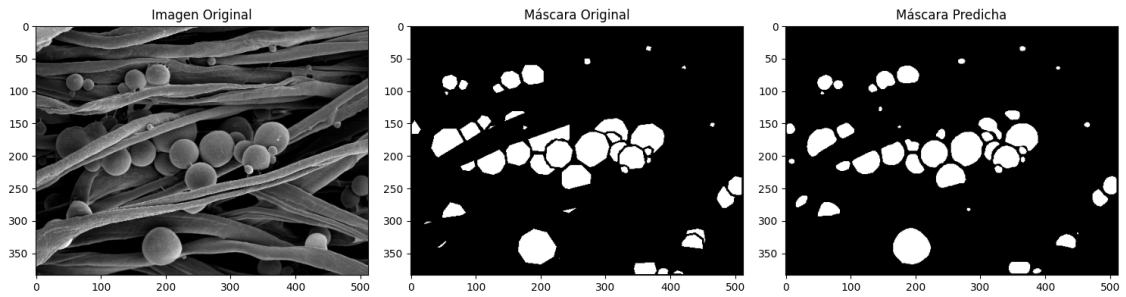


Figura 4.18: Resultado de la imagen i008

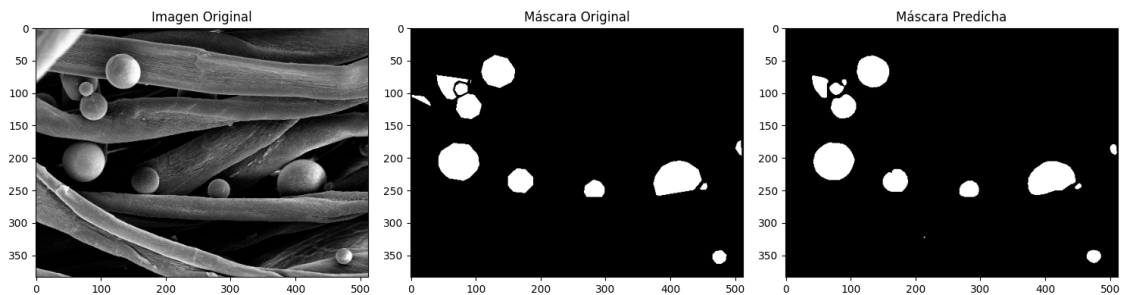


Figura 4.19: Resultado de la imagen i017

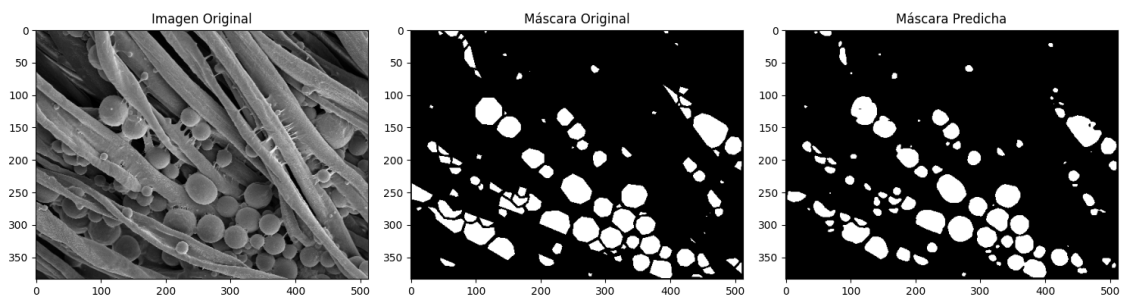


Figura 4.20: Resultado de la imagen i034

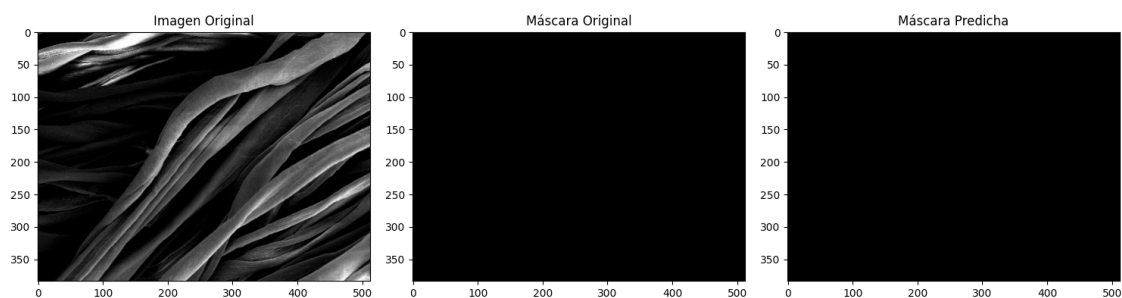


Figura 4.21: Resultado de la imagen i061

Una vez que se ha concluido que la red diseñada es válida y ha demostrado un buen rendimiento en la detección de microcápsulas, es el momento de proceder con los resultados del postprocesamiento y el conteo de las microcápsulas.

En la tabla 4.4, se presenta una comparación entre los valores obtenidos al contar regiones utilizando dos enfoques diferentes: a partir del archivo XML de referencia (microcápsulas reales) y a partir de una máscara generada por un algoritmo de segmentación. Estos dos métodos se utilizan para evaluar y validar la detección y segmentación de regiones de interés en una imagen.

Imagen	Nº de microcápsulas reales	Nº de microcápsulas detectadas
i003	18	13
i008	47	43
i017	12	13
i023	11	10
i031	25	26
i034	99	89
i037	32	32
i043	61	60
i048	0	1
i052	81	54
i053	49	39
i060	0	0
i068	25	25
i066	45	31
i073	63	61
i074	95	86
Media	41,4375	36,4375

Tabla 4.4: Resultados del conteo

La comparación de estos dos enfoques proporciona información importante sobre la precisión y el rendimiento del algoritmo de segmentación en relación con las anotaciones de referencia. Es común que el recuento basado en máscara pueda diferir del conteo basado en XML, ya que el enfoque automático puede detectar regiones adicionales o perder algunas regiones que están presentes en

las anotaciones manuales.

Finalmente, el modelo ha detectado aproximadamente el 87,93 % de las microcápsulas en el conjunto de test. Es importante destacar que en la evaluación de la detección y segmentación de microcápsulas en imágenes, el enfoque es evaluar la densidad y la distribución de las microcápsulas en la imagen y no un número exacto. Esto significa que no es grave que el recuento basado en máscara difiera ligeramente del conteo basado en el archivo XML de referencia.

Por lo tanto, en lugar de enfocarse exclusivamente en el número exacto de microcápsulas, es fundamental considerar la coherencia y la tendencia general de los resultados. Esta comprensión más completa del proceso de conteo asegura que se evalúe adecuadamente la capacidad del modelo para identificar regiones de interés en la imagen y su utilidad en la estimación de la densidad de microcápsulas en aplicaciones prácticas, teniendo en cuenta las limitaciones inherentes a la detección de objetos superpuestos.

Capítulo 5

Aplicación web

Se ha creado una aplicación web diseñada para el conteo automatizado de microcápsulas en imágenes textiles. Con el objetivo de llevar a cabo este proceso de manera eficiente y precisa, se han implementado varios componentes cruciales en el marco de esta aplicación.

Para aprovechar la potencia de la IA en la detección y conteo de microcápsulas, se ha integrado un servicio de Flask local, que actúa como motor de procesamiento para el modelo de IA. Esto permite que la aplicación realice análisis avanzados de imágenes y entregue resultados precisos a los usuarios.

En la parte frontal de la aplicación, se ha desarrollado la interfaz de usuario utilizando tecnologías Vue.js y HTML. Esta interfaz presenta cinco vistas interactivas, que permiten a los usuarios cargar imágenes, ingresar datos relevantes, recibir resultados precisos y almacenarlos en un informe visualmente comprensible.

El componente de *backend* de la aplicación se ha creado utilizando Node.js. Este *backend* se encarga de gestionar las interacciones entre la interfaz de usuario y la base de datos, así como con el servicio Flask que aloja el modelo de IA. El protocolo HTTP se ha utilizado para establecer conexiones tanto con la base de datos como con el servicio de IA.

En cuanto a la gestión de datos, se ha optado por MongoDB como base de datos para esta aplicación. La base de datos se emplea para almacenar y recuperar información relevante para cada proyecto y sus respectivas imágenes.

Para facilitar la implementación y la gestión de la aplicación en diferentes entornos, se ha utilizado *docker*. Mediante el uso de contenedores, cada componente de la aplicación, incluidos los servicios de Flask y Node.js, se puede empaquetar junto con sus dependencias, lo que garantiza la consistencia y la portabilidad en diversos sistemas.

A lo largo del capítulo, examinaremos en profundidad estas herramientas y tecnologías que se han empleado para lograr un producto final. La combinación de IA, desarrollo *frontend* y *backend*, protocolo HTTP, MongoDB y *docker* crea una solución eficiente con resultados precisos y útiles para los usuarios en el ámbito textil.

5.1. Estructura y comunicaciones

Este proyecto se compone de tres partes interrelacionadas que son fundamentales para el funcionamiento deseado:

- Aplicación web con interfaz de usuario: la aplicación web constituye la interfaz principal a través de la cual los usuarios interactúan con el sistema. Ofrece una experiencia intuitiva permitiendo a los usuarios cargar imágenes, introducir datos específicos de cada proyecto y visualizar los resultados generados. Es la encargada de generar peticiones HTTP tanto a la base de datos como al servicio con el modelo.
- Base de datos MongoDB: la base de datos MongoDB actúa como un almacén centralizado para todos los datos generados y utilizados en la aplicación. Permite el almacenamiento y recuperación eficiente de imágenes e información de proyectos. Recibe las peticiones HTTP de la aplicación web.
- Servicio Flask local con el modelo de IA: el servicio Flask actúa como el núcleo de procesamiento para el cálculo del número de microcápsulas en las imágenes cargadas. El servicio utiliza un modelo de IA previamente entrenado para realizar predicciones y procesamiento de imágenes. Los resultados se generan y se devuelven a la aplicación web mediante las peticiones HTTP.



Figura 5.1: Estructura general de comunicaciones

En conjunto, la aplicación web, la base de datos MongoDB y el servicio Flask con el modelo de IA forman una solución integral y de alto rendimiento para el conteo automatizado de microcápsulas en textiles. En la figura 5.1 se puede apreciar un esquema con las conexiones y los componentes del conjunto que forman la aplicación web.

5.2. Frontend

El *frontend* de una aplicación web es la parte visible e interactiva con la que los usuarios interactúan directamente. Consiste en la interfaz de usuario (UI) que permite a los usuarios realizar acciones, visualizar información y acceder a diferentes funciones de la aplicación. En esencia, el *frontend* es lo que los usuarios ven y con lo que interactúan en sus navegadores.

Para el desarrollo del *frontend* de la aplicación web se ha utilizado un *framework* que proporciona herramientas y estructuras predefinidas para crear una interfaz de usuario efectiva y dinámica. Un *framework* de *frontend* permite estructurar el código de manera organizada, manejar el enrutamiento entre diferentes vistas, administrar el estado de la aplicación y crear componentes reutilizables.

El *framework* empleado es Vue.js, un *framework* progresivo de JavaScript utilizado para la construcción de interfaces de usuario interactivas y dinámicas. Su enfoque se basa en la creación de componentes reutilizables y la capacidad de incorporar gradualmente funcionalidades más complejas según sea necesario. Vue.js es conocido por su flexibilidad y facilidad de integración con otros proyectos y bibliotecas [77].

En el contexto de una aplicación web desarrollada con Vue.js como *framework*, HTML sigue siendo fundamental. Aunque Vue.js facilita la creación de componentes y la gestión de la lógica de la interfaz, sigue siendo necesario utilizar HTML para estructurar y definir la jerarquía de elementos en la página.

En una aplicación Vue.js, los componentes están definidos en archivos separados y pueden contener tanto la lógica JavaScript como el marcado HTML que describe cómo se mostrarán esos componentes. Vue.js permite combinar HTML y JavaScript de una manera coherente y modular para crear interfaces de usuario reactivas.

Un componente puede ser conceptualmente comparado con un “bloque de construcción” que encapsula tanto el aspecto visual como la funcionalidad específica de una parte específica de una página web. Pueden representar elementos como botones, formularios, encabezados, pies de página e incluso elementos más complejos como gráficos interactivos o módulos de comentarios. Lo que distingue a los componentes es que encapsulan tanto la lógica como la presentación en una unidad autónoma, lo que permite su reutilización en diferentes partes del sitio web o en distintas aplicaciones.

Una de las características más potentes de Vue.js es su capacidad para definir componentes personalizados, lo que significa que puedes crear piezas de interfaz altamente modulares y luego ensamblarlas para formar una aplicación completa. En el siguiente apartado, se explorará en detalle los diferentes componentes que se han implementado en la aplicación web y cómo han contribuido a mejorar la experiencia del usuario y la eficiencia en el desarrollo.

Además de Vue.js, para potenciar el *frontend* de la aplicación web, se han incorporado dos bibliotecas esenciales: Vuetify y Exceljs. Vuetify ha sido fundamental para diseñar y estilizar todos los componentes visuales de la aplicación, desde botones y formularios hasta encabezados y pies de página. Su conjunto de componentes predefinidos y su facilidad de personalización han contribuido en gran medida a crear una interfaz de usuario atractiva.

Por otro lado, Exceljs ha desempeñado un papel crucial en la generación de informes dinámicos para los usuarios. Esta biblioteca permite crear archivos Excel personalizados con los resultados generados por la aplicación, con el fin de exportar datos de manera más sencilla.

5.2.1. Componentes

En el diseño de nuestra aplicación web, cada ruta se alinea con una vista específica, ofreciendo una experiencia de navegación coherente y enfocada. Estas vistas, a su vez, son representadas por componentes individuales que encapsulan tanto la lógica como la interfaz de usuario necesaria para

cada sección. A continuación, se presentan las vistas clave que los usuarios pueden explorar:

- **Página de inicio** (Start.vue, “/”), figura 5.2: esta página de bienvenida brinda a los usuarios información esencial sobre la aplicación y lo que pueden esperar de ella al acceder por primera vez. Les proporciona una introducción y una visión general de las funcionalidades disponibles. Además de la presentación, esta vista incluye un botón que permite a los usuarios iniciar la aplicación web y comenzar su exploración.

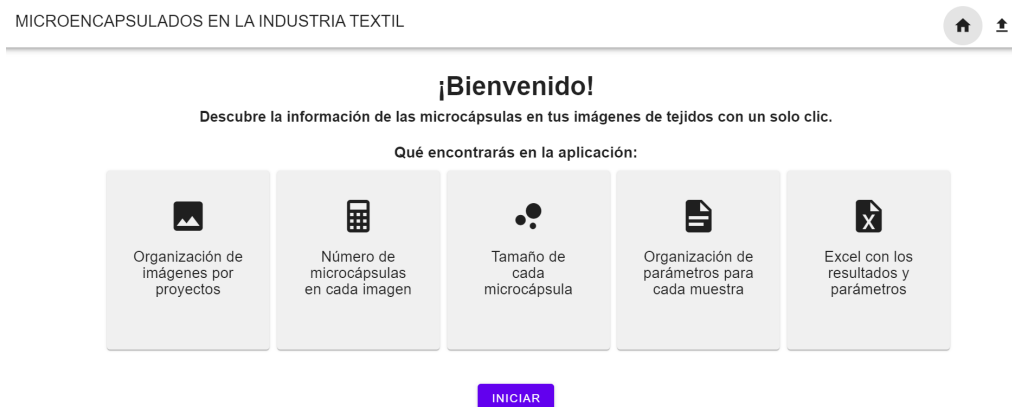


Figura 5.2: Vista de la página de inicio

- **Página de proyectos** (Projects.vue, “/projects”), figura 5.3: esta página muestra una lista de los proyectos creados y ofrece la posibilidad de crear nuevos proyectos. Aquí los usuarios pueden acceder rápidamente a los proyectos existentes y comenzar uno nuevo si es necesario.

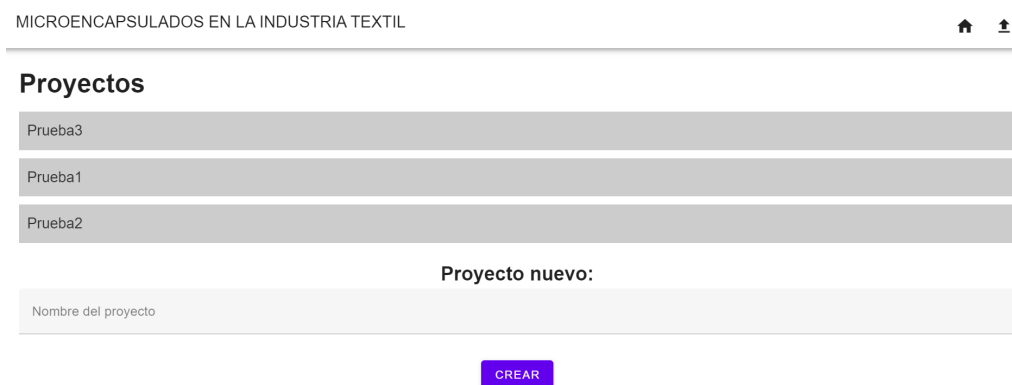


Figura 5.3: Vista de la página de los proyectos

- **Vista del proyecto** (Home.vue, “/detect/:projectName”), 5.4: la página principal de la aplicación está diseñada para un proyecto específico. Aquí los usuarios pueden cargar imágenes

relacionadas con el proyecto y verlas en su contexto. También pueden ingresar datos relevantes para cada imagen, como la escala, referencia, principio activo, membrana, concentración de microcápsulas y RPM si la microcápsula no es comercial. Los botones de cálculo del número y el tamaño de las microcápsulas brindan herramientas adicionales para el análisis.



Figura 5.4: Vista de la página principal de cada proyecto

- Subir imágenes (UploadFile.vue, “/upload”), figura 5.5: esta página permite a los usuarios cargar nuevas imágenes para su análisis. Proporciona información sobre el progreso de la carga de cada imagen y ofrece la capacidad de cancelar la carga en caso necesario.

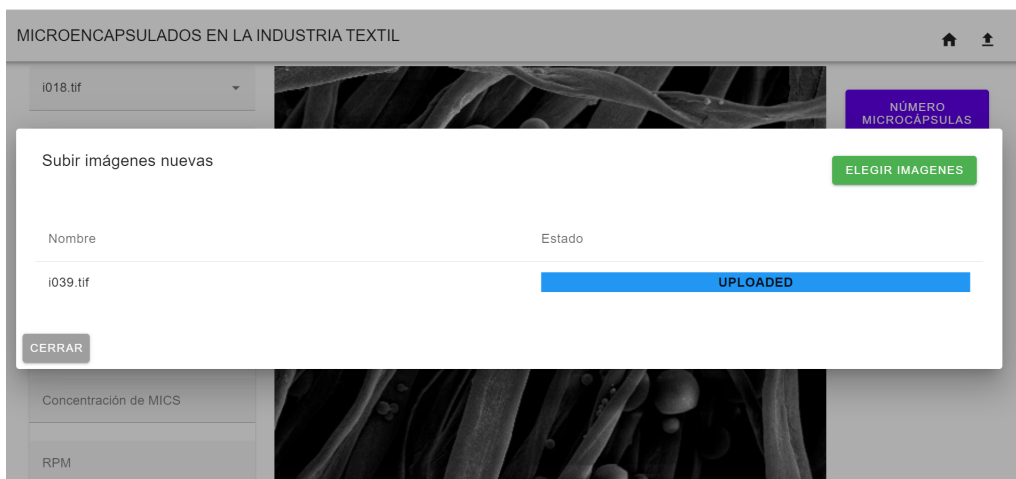


Figura 5.5: Vista de la página de subir imágenes

5.3. *Backend*

El *backend*, también conocido como el lado del servidor, es esencial para el funcionamiento de la aplicación web. Es la parte encargada de manejar la lógica operativa, el procesamiento de datos y la interacción con la base de datos.

Para desarrollar el *backend*, se ha empleado el *framework* de Node.js, una plataforma que permite construir aplicaciones de red eficientes y escalables. Este *framework* es muy versátil y es utilizado ampliamente en la industria para desarrollar aplicaciones web modernas y eficientes.

La arquitectura de Node.js se centra en el manejo eficiente de eventos y operaciones asíncronas. Utiliza un Event Loop para gestionar múltiples solicitudes al mismo tiempo sin bloquear el flujo. Emplea callbacks y funciones asíncronas para manejar operaciones de entrada/salida sin esperar a que terminen. Los módulos y Event Emitters organizan el código y gestionan eventos. Aunque es eficaz para aplicaciones en tiempo real y de alta concurrencia, no es adecuado para tareas de cálculo intensivo [87]. Esta arquitectura se ve reflejada en la figura 5.6.

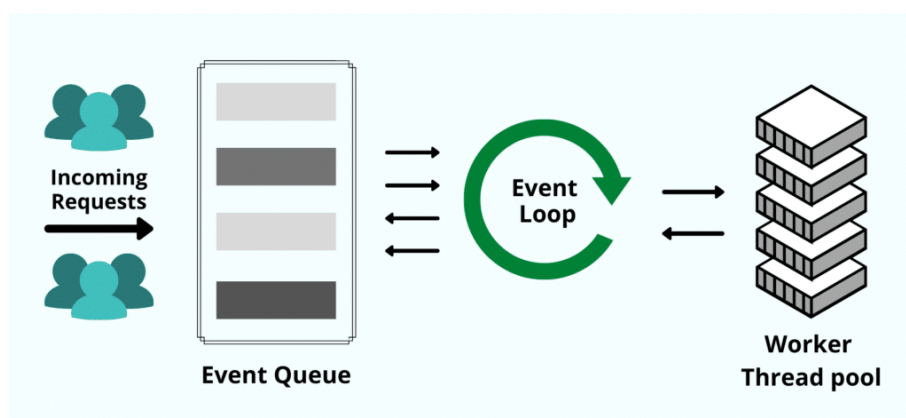


Figura 5.6: Arquitectura del *framework* Node.js [88]

Algunos de los principales módulos utilizados para el desarrollo del *backend* son:

- Cors: Este módulo proporciona middleware para habilitar CORS (Cross-Origin Resource Sharing), permitiendo que el *frontend* pueda realizar peticiones a diferentes dominios [89].
- Express: Express es un *framework* de aplicación web minimalista y flexible para Node.js, utilizado para definir rutas, manejar peticiones y respuestas, y construir APIs REST de manera sencilla y eficiente [90].
- HTTP: El módulo HTTP proporciona una funcionalidad de servidor HTTP básica, lo cual es esencial para la comunicación entre el *frontend* y el *backend*.
- Body-parser: Este módulo se utiliza para analizar el cuerpo (URL) de las solicitudes HTTP, permitiendo acceder a los datos enviados por el cliente [91].
- Path: El módulo path proporciona utilidades para trabajar con rutas de archivos y directorios.

En conjunto, estos componentes forman el núcleo del *backend* de la aplicación web, permitiendo las funcionalidades de la aplicación web como, la gestión de rutas, la comunicación con el *frontend* y la manipulación de datos en la base de datos.

5.3.1. Protocolo de comunicación HTTP

La comunicación HTTP (*Hypertext Transfer Protocol*) es un protocolo fundamental en la construcción y funcionamiento de aplicaciones web. En el contexto de nuestra aplicación, se utiliza

para establecer intercambios de datos entre el cliente y el servidor. La arquitectura cliente-servidor se basa en solicitudes y respuestas, donde el cliente realiza solicitudes al servidor y este responde con la información solicitada. Además, es un protocolo “sin estado”, es decir, no posee un registro de visitas. Esta estructura se puede ver reflejada en el figura 5.7.

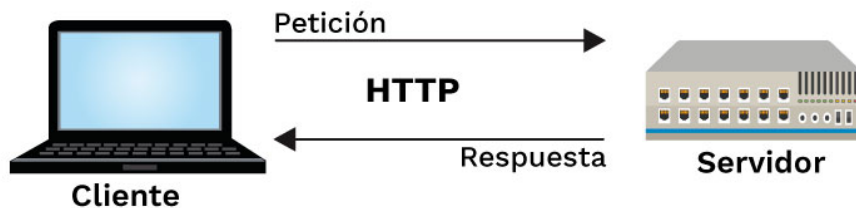


Figura 5.7: Arquitectura del protocolo HTTP

La comunicación HTTP se basa en un conjunto de métodos, como GET, POST, PUT y DELETE, que definen la acción que se realizará en el servidor en función de la solicitud del cliente. En el desarrollo de la aplicación, solo se emplean dos métodos del protocolo: GET y POST.

El método POST se utiliza para enviar datos al servidor, en este caso, para cargar imágenes y los nombres de proyectos en la base de datos. Al subir imágenes, el cliente envía los datos de la imagen al servidor, que luego los procesa y almacena en la base de datos. De manera similar, cuando se crea un nuevo proyecto, los detalles del proyecto se envían al servidor a través de una solicitud POST, lo que resulta en la inserción de la información en la base de datos.

El método GET se utiliza para recuperar información del servidor. En el contexto de la aplicación, se emplea para obtener la lista de imágenes y proyectos almacenados en la base de datos. Cuando se solicita la lista de imágenes, el servidor responde con los nombres de las imágenes disponibles en la base de datos. Del mismo modo, al solicitar la lista de proyectos, el servidor devuelve los nombres de los proyectos guardados. También se utiliza el método GET para obtener los resultados generados por la aplicación del modelo, permitiendo que el cliente acceda a la información relevante.

Para gestionar esta comunicación en la aplicación desarrollada, se emplea el módulo express de Node.js, el cual proporciona una plataforma para construir aplicaciones web y servicios API de manera rápida y sencilla.

Cada una de las solicitudes HTTP en nuestra aplicación está estructurada mediante la definición de rutas y sus correspondientes controladores. Esta estrategia permite que la aplicación responda de manera efectiva a las solicitudes entrantes, asegurando que cada acción se ejecute y proporcione la respuesta esperada.

Por un lado, cada ruta es representada por una URL específica, la cual se define en el archivo “api.routes.js”. Por ejemplo, la ruta “/file” está relacionada con la función “uploadFile” del controlador, esta es la encargada de manejar la subida de archivos. De manera similar, otras rutas como “/imageslist”, “/project”, “/projectslist” y “/segmentation-ana” también tienen sus propias funciones de controlador asignadas, crear un nuevo proyecto, cargar la lista de imágenes o de proyectos

y obtener los resultados del modelo de segmentación.

Por otro lado, los controladores contienen la lógica que se ejecuta cuando se accede a una ruta específica. Estos controladores procesan los datos de entrada, realizan operaciones según las necesidades de la solicitud y generan una respuesta adecuada para ser enviada de vuelta al cliente. Así, cada controlador se encarga de llevar a cabo cada una de las tareas, como interactuar con la base de datos, procesar la información, realizar cálculos y más.

Esta separación entre rutas y controladores sigue el patrón de diseño MVC, conocido como Modelo-Vista-Controlador, donde las rutas actúan como las “vistas” que el cliente accede, los controladores son el “controlador” que maneja la lógica y las operaciones, y el “modelo” generalmente corresponde a la interacción con la base de datos y los datos [92]. La figura 5.8 corresponde al funcionamiento del patrón MVC explicado anteriormente.

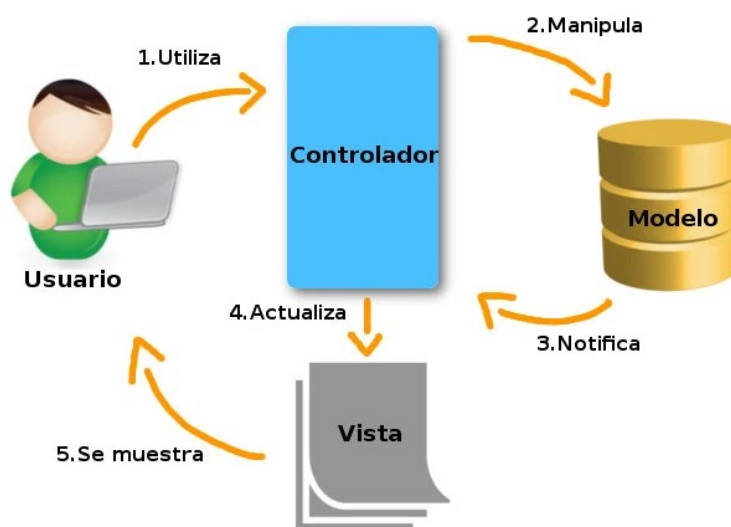


Figura 5.8: Funcionamiento del patrón de diseño MVC [93]

5.3.2. Base de datos en MongoDB

Una base de datos desempeña un papel crucial en el almacenamiento, organización y recuperación de los datos en una aplicación. En el contexto de la aplicación web desarrollada, se utiliza MongoDB como sistema de gestión de bases de datos. MongoDB es una base de datos NoSQL que se caracteriza por su flexibilidad y escalabilidad, lo que lo hace especialmente adecuado para almacenar grandes volúmenes de datos de manera eficiente [94].

En la aplicación desarrollada se usa la base de datos en MongoDB para almacenar las imágenes que suben al sistema los usuarios. Cada imagen se guarda en la base de datos junto con su información asociada, lo que permite a los usuarios acceder y gestionar sus datos de manera cómoda y eficiente. También se almacena el nombre de los proyectos creados por los usuarios. Almacenar las imágenes y los proyectos en la base de datos, permite a la aplicación cargarlos en una lista en cualquier momento mediante las peticiones get y post. La elección de MongoDB se basa en su capacidad para manejar datos no estructurados, como las imágenes, y su capacidad para adaptarse a los requisitos cambiantes de la aplicación.

Para entender en detalle la estructura y los contenidos de la base de datos, se ha utilizado la aplicación de escritorio “MongoDB Compass”. Esta herramienta ofrece una interfaz visual intuitiva que permite explorar la base de datos de una manera más accesible. A través de MongoDB Compass, se puede examinar las colecciones de datos, visualizar documentos individuales y como esta organizada la información almacenada en la base de datos de una manera más práctica y visual [95], se puede observar como es la visualización en las figuras 5.9 y 5.10.

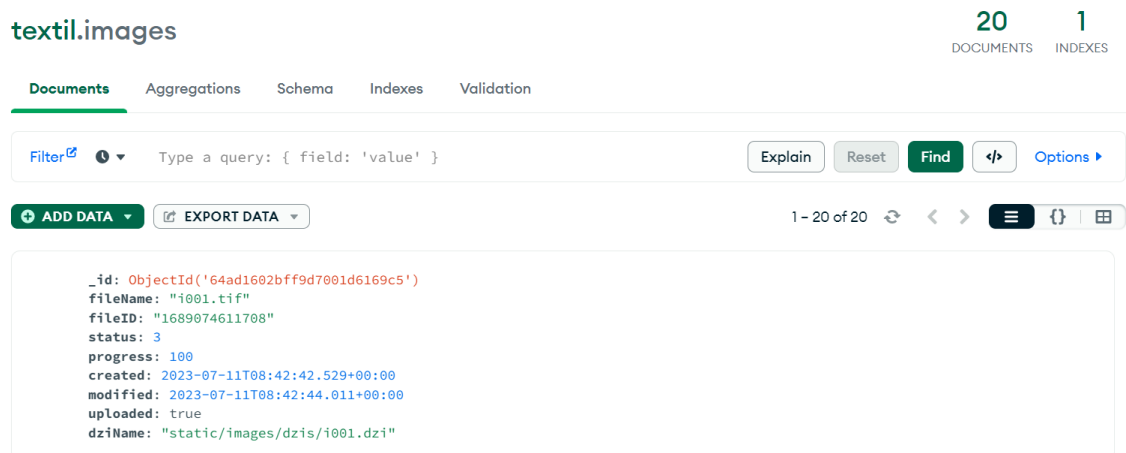


Figura 5.9: Visualización de la información de una imagen almacenada en mongoDB



Figura 5.10: Visualización de la información de los proyectos almacenada en mongoDB

Para establecer una la comunicación entre la aplicación desarrollada y la base de datos, se emplea la biblioteca “monk”, que actúa como intermediario. La conexión se establece utilizando los detalles de la dirección y el puerto del servidor MongoDB, permitiendo que la aplicación acceda y gestione los datos almacenados en la base de datos. Esta integración entre la aplicación y la base de datos es fundamental para garantizar que las operaciones de consulta y almacenamiento de información se realicen de manera eficiente en tiempo real.

Una vez comprendido el sistema de la API REST con Node.js, las llamadas GET y POST del protocolo HTTP, el *frontend* con Vue.js y la base de datos con MongoDB, es crucial entender cómo se interconectan estos componentes. En primer lugar, cuando el *frontend* necesita acceder a

la base de datos, realiza una llamada a Node.js, que a su vez realiza las solicitudes GET o POST a la base de datos. Posteriormente, la respuesta de la base de datos se comunica de nuevo al *frontend* a través de Node.js. La figura 5.11 ilustra de manera clara todas estas relaciones.

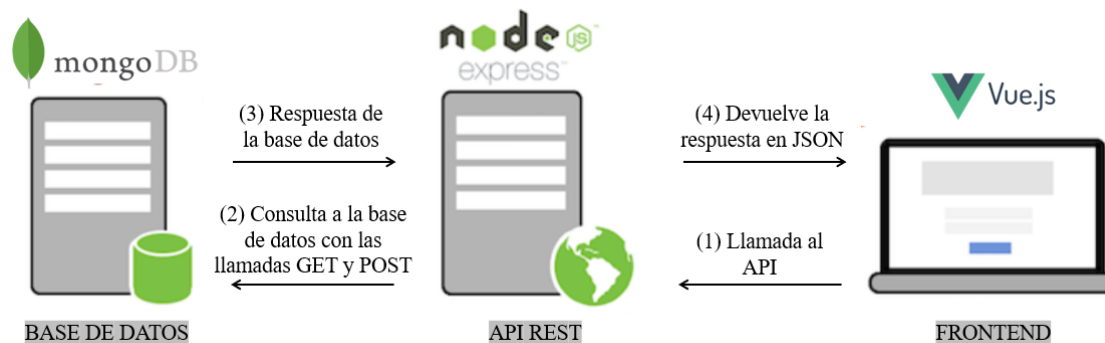


Figura 5.11: Interconexión entre el *frontend* y la base de datos

5.3.3. Implementación del Modelo de segmentación

Se ha diseñado un servicio local específico para la integración de la IA en la aplicación web. Este servicio se encarga de la implementación y ejecución del modelo de IA para llevar a cabo la segmentación y el análisis de las imágenes de microcápsulas. Esta tarea concluye en determinar el número de microcápsulas presentes en cada imagen, lo que ofrece información valiosa para los usuarios.

Este servicio se ha desarrollado utilizando el *framework* Flask en Python, aprovechando su capacidad para crear servicios web y responder a solicitudes HTTP de manera eficiente. Flask es un *framework* de desarrollo web ligero y versátil para Python. Su principal objetivo es facilitar la creación de aplicaciones web y servicios mediante la simplificación de tareas comunes, permitiendo a los desarrolladores concentrarse en la lógica de la aplicación en lugar de ocuparse de detalles técnicos complejos [96].

Debido a su naturaleza flexible y modular, Flask es especialmente popular entre los desarrolladores que desean construir aplicaciones web pequeñas o medianas con un enfoque personalizado y específico. En este caso Flask se utiliza para crear el servicio de segmentación y análisis de imágenes que interactúa con el modelo de IA. Proporciona una base sólida para manejar las solicitudes HTTP entrantes, procesar los datos enviados desde la aplicación y realizar las operaciones necesarias en el modelo para calcular el número de microcápsulas en cada imagen.

La ruta de este servicio se ha establecido como “/segmentation-ana” y se espera que reciba las solicitudes GET para devolver el número de microcápsulas. La información relevante para el procesamiento de imágenes, es decir, el nombre de la imagen que se desea procesar, se recibe en la URL. Un ejemplo de petición desde la aplicación web al servicio local es “/segmentation-ana/imageName=i005.tif”.

Cuando el servicio recibe una solicitud GET, extrae el nombre del proyecto asociado a la imagen de la URL. El modelo local tiene acceso a todas las imágenes y con la información extraída busca la imagen adecuada. Esta imagen actúa como entrada para el modelo de IA. En este punto, el modelo realiza la segmentación de la imagen y mediante el postprocesado se identifica y cuantifica el

número de microcápsulas presentes en la imagen.

Una vez que el modelo ha realizado su análisis y ha calculado el número de microcápsulas, el servicio responde a la solicitud con un código de estado “200”. Esta respuesta indica que el proceso se ha completado con éxito y que el número de microcápsulas ha sido calculado y está listo para ser utilizado en la aplicación.

Es importante resaltar que esta parte de la aplicación combina la potencia de la IA con la eficacia de las tecnologías web para ofrecer una solución integral. La capacidad de automatizar el proceso de conteo de microcápsulas mediante un servicio web agiliza y optimiza la experiencia del usuario, permitiendo una toma de decisiones más informada y precisa en el análisis de las imágenes.

5.3.4. Docker

La aplicación web desarrollada se ha integrado con *docker* para proporcionar un entorno de ejecución consistente y portátil. *docker* es una plataforma que permite empaquetar y distribuir aplicaciones junto con todas sus dependencias en contenedores aislados, lo que facilita la implementación y la ejecución en diferentes entornos sin preocuparse por las diferencias en la configuración del sistema [97].

Para comprender cómo funciona nuestra aplicación con *docker*, es importante destacar algunos aspectos clave sobre el funcionamiento de *docker*. Entre estos se encuentran los contenedores, las imágenes y los servicios. Gracias a la figura 5.12 se puede apreciar la relación entre estos componentes, que más adelante se explicarán en profundidad.

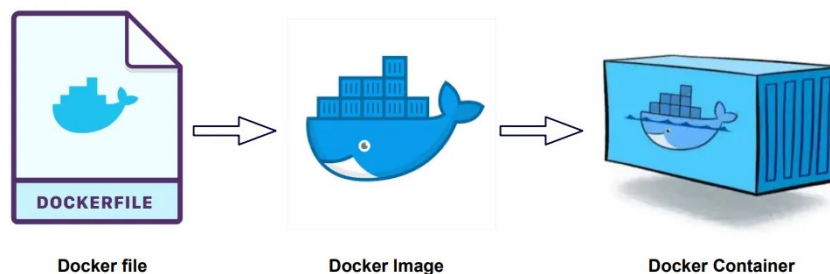


Figura 5.12: Relación de los componentes en *docker* [98]

Un contenedor *docker* es una unidad de software que contiene la aplicación, su código y las dependencias necesarias. Cada contenedor se ejecuta de manera aislada en su propio entorno, lo que garantiza que la aplicación funcione de manera coherente independientemente del sistema operativo o la infraestructura subyacente.

Una imagen de *docker* es un paquete ligero y autónomo que contiene todo lo necesario para ejecutar una aplicación, incluido el código, las bibliotecas y las configuraciones. Las imágenes se definen mediante el archivo “Dockerfile.dev”, que describe los pasos necesarios para construir la imagen. En este caso, este archivo está diseñado para crear un contenedor que ejecute una aplicación Node.js utilizando el entorno Alpine, que es una versión ligera de Linux. Describe los pasos necesarios para construirlo e instala sus dependencias. Inicia la aplicación utilizando el *script* dev definido en el archivo package.json.

La composición de servicios o *docker compose* es una herramienta que permite definir y ejecutar aplicaciones multi-contenedor. En nuestro caso, utilizamos *docker compose* para definir cómo se relacionan y se ejecutan los diferentes contenedores de nuestra aplicación, como el contenedor del *frontend*, el contenedor del *backend* y el contenedor de la base de datos.

En la sección mongo se establece la configuración para la base de datos MongoDB. Se define el nombre del contenedor como “textildb” y se utiliza la imagen de *docker* “mongo:6.0.6-jammy”, que es la versión específica de MongoDB que se utilizará. Además, se mapea el puerto 27017 del contenedor al mismo puerto en el *host*, lo que permite la comunicación con la base de datos desde fuera del contenedor.

La sección textil-back representa el *backend* de la aplicación. En esta sección se define el nombre de contenedor como textil-back y se configura la construcción de la imagen utilizando un archivo Dockerfile.dev en el directorio ./back. Esta imagen depende del servicio mongo, lo que significa que *docker compose* se asegurará de que el servicio de base de datos esté en funcionamiento antes de iniciar el *backend*. Los puertos y volúmenes se configuran para la comunicación y el montaje de archivos. Además, se definen variables de entorno importantes, como la URL de la base de datos MongoDB.

La sección textil-front representa el *frontend* de la aplicación. Al igual que el *backend*, se configura el nombre del contenedor, la construcción de la imagen y los puertos. Los volúmenes se utilizan para montar los archivos necesarios en el contenedor.

Gracias a *docker*, podemos garantizar que nuestra aplicación se ejecute de manera consistente en diferentes entornos, como desarrollo, prueba y producción. Además, la portabilidad de *docker* permite que el proceso de implementación y escalado sea mucho más eficiente y predecible.

De esta manera, al incorporar *docker*, se ha logrado encapsular cada uno de los componentes de la aplicación en contenedores aislados. Esto garantiza la uniformidad en su ejecución, su portabilidad, y su implementación en diversos contextos. Esta estrategia concede un control exhaustivo sobre las dependencias y la configuración de la aplicación, al mismo tiempo que simplifica el proceso de desarrollo.

5.4. Traza de ejecución

La traza de ejecución indica la secuencia de acciones que se crea al ejecutar cada parte de la aplicación, además del orden en que se inician y comunican estas partes del sistema. Esta parte desempeña un papel crítico en la comprensión de cómo interactúan las tres partes interrelacionadas que componen este proyecto: el servicio Flask con el modelo de IA, la aplicación web con interfaz de usuario y la base de datos MongoDB.

El proceso comienza con el levantamiento del servicio Flask local, que actúa como núcleo de procesamiento para el cálculo del número de microcápsulas en las imágenes cargadas. El servicio utiliza un modelo de IA previamente entrenado para realizar predicciones y procesamiento de imágenes. Este componente debe estar operativo antes de que la aplicación web pueda utilizarlo. Para comprobar que el servicio está levantado, debería aparecer algo parecido a la figura 5.13.

Una vez que el servicio Flask está en funcionamiento, se necesita arrancar el docker de la aplicación. Tras arrancarlo, se inicia la aplicación web desde cualquier navegador. Seguidamente, la base de datos MongoDB se activa automáticamente.

```

C:\Users\PC\Desktop\TFM\App\model_microcapsulas>py -3 tmp_MICS.py
* Serving Flask app 'tmp_MICS'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:6500
* Running on http://192.168.1.110:6500
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 106-806-208

```

Figura 5.13: Levantamiento del servicio Flask

Cuando el usuario cree un nuevo proyecto, se realizará una petición POST a la base de datos. Para cargar el nombre de los proyectos creados se realiza una peticiones GET. Seguidamente, al entrar en cualquiera de los proyectos creados, se procede a cargar las imágenes correspondientes al proyecto mediante peticiones POST a la base de datos continuada de una petición GET para cargar la lista de imágenes ya cargadas.

Tras este proceso, se selecciona la imagen corresponde y se rellenan los campos de información de la imagen. Cuando el usuario desee pulsar el botón para el calculo de número de microcápsulas, se realizará una petición POST al modelo Flask para indicarle el nombre de la imagen el cual devolverá el número calculado por el modelo de IA.

Por último, una vez obtenido el calculo, se pulsa el botón para guardar en los resultados en Excel. Se generará un archivo personalizado que contendrá los resultados organizados en filas y columnas la información calculada y los detalles de la imagen seleccionada. El archivo será nombrado automáticamente con el nombre de la imagen seleccionada y el proyecto perteneciente, y se descargará al dispositivo del usuario, lo que permitirá que este último lo almacene localmente.

Después de entender este procedimiento, en la figura 5.14, se presenta de manera esquemática lo que debe hacer el usuario y lo que realiza la aplicación desde la captura de las imágenes hasta la obtención del resultado final del conteo.

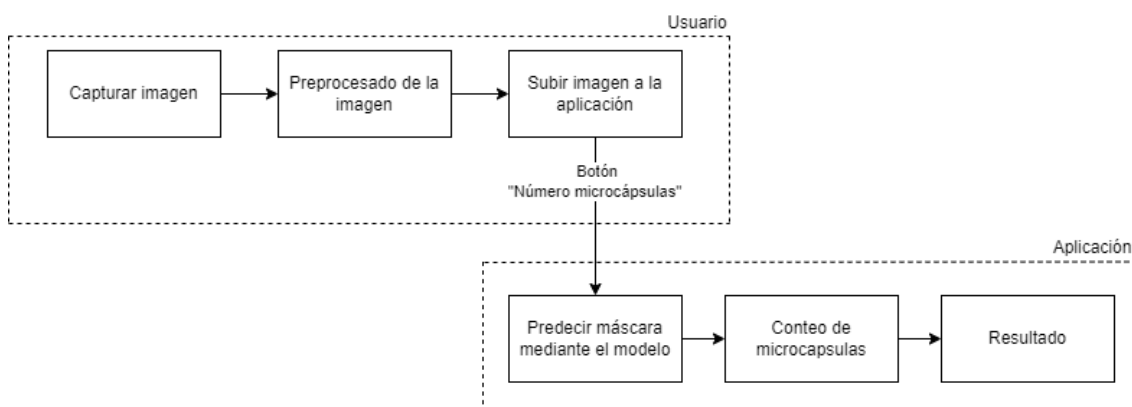


Figura 5.14: Proceso esquemático del uso de la aplicación

Capítulo 6

Conclusiones y propuestas de futuro

Este proyecto ha logrado su objetivo principal al diseñar una aplicación web capaz de identificar y contar automáticamente microcápsulas en imágenes microscópicas de tejidos. La implementación de la arquitectura de redes neuronales convolucionales U-Net ha demostrado ser efectiva en esta tarea, lo que representa un avance significativo en la automatización y precisión del análisis de microcápsulas en la industria textil.

La aplicación desarrollada tiene el potencial de contribuir de manera significativa al avance y la aplicación práctica de la tecnología de microencapsulación en la industria textil. Permite evaluar la eficacia de los procesos de incorporación de microcápsulas, determinar la cantidad de producto necesaria y estimar la duración de las microcápsulas adheridas a los tejidos. Estos resultados abren nuevas posibilidades para la mejora de tejidos del futuro.

Se llevarán a cabo pruebas de validación funcional y de usabilidad de la aplicación para garantizar su desempeño óptimo en diferentes entornos y por parte de los usuarios finales. Estos procesos de validación ayudarán a refinar aún más la aplicación, asegurando su fiabilidad y facilidad de uso. Además, se estudiará el aumento de las capacidades de la aplicación que puedan beneficiar aún más a la industria textil y a la tecnología de microencapsulación.

Una de las futuras mejoras planeadas para el sistema es la implementación de una funcionalidad en tiempo real. Esto permitiría a los usuarios realizar la segmentación y análisis de microcápsulas en tiempo real, lo que sería especialmente útil en aplicaciones que requieren monitoreo continuo y toma de decisiones instantáneas basadas en los resultados.

Para garantizar la integridad y la privacidad de los datos, se planea desarrollar una robusta estructura de usuarios que incluya autenticación y medidas de seguridad avanzadas. Esto asegurará que solo usuarios autorizados tengan acceso a la aplicación y sus funcionalidades, al tiempo que se protegen los datos.

Para simplificar y agilizar el proceso de segmentación, se prevé incorporar el preprocesado de imágenes directamente en la aplicación web. Esto incluiría funciones para el ajuste automático de brillo y contraste y el recorte o redimensión de las imágenes que facilitarían el análisis preciso de las microcápsulas.

Una mejora significativa será la incorporación de la capacidad de calcular automáticamente el tamaño de cada una de las microcápsulas detectadas en la imagen. Esto proporcionaría información valiosa sobre la distribución de tamaños y ayudaría en la caracterización detallada de las micro-

cápsulas en la muestra.

Actualmente, el servicio de IA se ejecuta localmente, pero en el futuro, se planea migrar a un servidor remoto o en la nube. Esto permitiría una mayor escalabilidad y accesibilidad, lo que es esencial para aplicaciones que requieren procesamiento intensivo de imágenes y acceso desde ubicaciones diversas.

Estas propuestas de futuro demuestran un enfoque proactivo en la mejora y expansión del sistema de segmentación de microcápsulas, lo que potenciará su utilidad y eficacia en una variedad de aplicaciones.

Bibliografía

- [1] S Solarski et al. “Synthesis of a polyurethane/clay nanocomposite used as coating: Interactions between the counterions of clay and the isocyanate and incidence on the nanocomposite structure”. En: *Journal of Applied Polymer Science* 91.1 (2004), págs. 238-244.
- [2] Maria Bonet-Aracil, Eva Bou-Belda y Jaime Gisbert-Payá. “In situ test: cotton sheets against mosquito bites in India”. En: *Cellulose* 26.10 (2019), págs. 4655-4663.
- [3] Fernando Marcela et al. “Microencapsulation of Essential Oils by Interfacial Polymerization Using Polyurea as a Wall Material”. En: *Journal of Encapsulation and Adsorption Sciences* 5.3 (2015), págs. 165-177.
- [4] Ankur Gupta, Ryan F. Landis y Vincent M. Rotello. “Nanoantibiotics: a new paradigm for treating bacterial infections”. En: *Drug Discovery Today* 26.5 (2021), págs. 1299-1305.
- [5] Jie Hu, Yuanyuan Liu y Lihua Dong. “Responsive textiles: a review of the progress in the development of smart clothing”. En: *Advanced Materials Interfaces* 7.3 (2020), pág. 1901731.
- [6] Vishnu Kumar Gupta, Amit Kumar y Arun Kumar. “Microencapsulation: An advanced technology for controlled fragrance release”. En: *Journal of Industrial and Engineering Chemistry* 97 (2021), págs. 62-74.
- [7] Weidi Xie, J Alison Noble y Andrew Zisserman. “Microscopy cell counting and detection with fully convolutional regression networks”. En: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 6.3 (2018), págs. 283-292.
- [8] Melanie Ganz et al. “Deep Learning for Counting Histological Features: A Comprehensive Review”. En: *Journal of Pathology Informatics* 12 (2021), pág. 28. DOI: 10.4103/jpi.jpi_73_20.
- [9] S. P. Sawant, S. Banerjee y S. Ghosh. “Automated cell counting using ImageJ plugin”. En: *Analytical Cellular Pathology* 2019 (2019), págs. 1-8. DOI: 10.1155/2019/7831379.
- [10] Nedra Azizi, Yves Chevalier y Mustapha Majdoub. “Isosorbide-based microcapsules for cosmeo-textiles”. En: *Industrial Crops and Products* 52 (2014), págs. 150-157.
- [11] Martin P Neubauer, Melanie Poehlmann y Andreas Fery. “Microcapsule mechanics: From stability to function”. En: *Advances in colloid and interface science* 207 (2014), págs. 65-80.
- [12] Jun-Seok Hwang et al. “Factors affecting the characteristics of melamine resin microcapsules containing fragrant oils”. En: *Biotechnology and Bioprocess Engineering* 11 (2006), págs. 391-395.
- [13] Jun-Seok Hwang et al. “Preparation and characterization of melamine-formaldehyde resin microcapsules containing fragrant oil”. En: *Biotechnology and Bioprocess Engineering* 11 (2006), págs. 332-336.

- [14] Md Saifullah et al. “Micro and nano encapsulation, retention and controlled release of flavor and aroma compounds: A critical review”. En: *Trends in Food Science & Technology* 86 (2019), págs. 230-251.
- [15] Ahmed M Bakry et al. “Review of encapsulation methods suitable for food antioxidants and its applications”. En: *International Journal of Food Science & Technology* 51.9 (2016), págs. 2087-2102.
- [16] Saifullah et al. Saifullah. “Preparation of Microcapsules Containing Phase Change Material (PCM) for Thermal Energy Storage: A Review”. En: *Energies* 12.12 (2019), pág. 2333.
- [17] L. Capablanca. “Desarrollo de Microcápsulas para la Liberación Controlada de Feromonas de Insectos Plaga”. Tesis doct. La Habana, Cuba: Universidad de La Habana, 2013.
- [18] Ricardo Adolfo Parra Huertas. “Revisión: microencapsulación de alimentos”. En: *Revista Facultad Nacional de Agronomía Medellín* 63.2 (2010), págs. 5669-5684.
- [19] Adrian Gibbs y Francis W Skinner. *Food Microstructures: Microscopy, Measurement and Modelling*. Aspen Publishers, 1999.
- [20] Laura A Capablanca. *Food packaging: Materiali, tecnologie e qualità degli alimenti confezionati*. Edizioni Agricole, 2013.
- [21] A. M. Saifullah et al. “Inclusion of core materials into microcapsules through emulsification: A comprehensive review”. En: *Processes* 7.4 (2019), pág. 185.
- [22] José A Teixeira, Gustavo A Silva y Robson C Evangelista. “Microencapsulation by spray drying of active compounds for food applications—a review”. En: *Drying technology* 30.13 (2012), págs. 1389-1414.
- [23] François Salaün y Bernard Bataille. “Microencapsulation methods: a review”. En: *Revue des ingénieurs textiles* 190.3 (2016), págs. 8-18.
- [24] M M D C Vila et al. “Microencapsulation of essential oils with biodegradable polymeric carriers for cosmetic applications”. En: *Journal of Cosmetic Dermatology* 14.3 (2015), págs. 236-247.
- [25] EM Soares-Latour et al. “Environmentally benign 100 % bio-based oligoamide microcapsules”. En: *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 524 (2017), págs. 193-203.
- [26] IPS Flow Actives. *Microencapsulación por coacervación*. 2023. URL: <https://ips-flowactives.com/es/tecnologia/microencapsulacion-por-coacervacion.html>.
- [27] Interempresas. *Secador por aspersión electrostático PoladDry™*. 2023. URL: <https://www.interempresas.net/Quimica/Articulos/167874-Secador-por-aspersacion-electrostatico-PoladDryTM.html>.
- [28] Guoqiang Han y Di Wu. “Microencapsulation and its applications in self-healing materials”. En: *Materials Today Communications* 9 (2016), págs. 22-31.
- [29] K Hong y S Park. “Melamine resin microcapsules containing fragrant oil: synthesis and characterization”. En: *Materials Chemistry and Physics* 58.2 (1999), págs. 128-131.
- [30] Lucía Capablanca Francés. “Evaluación de la Adhesión y Permanencia de Microcápsulas sobre Tejidos de Algodón”. En: (2011).

-
- [31] Yao Liu, A Kumar y Jinlian Hu. "Microencapsulation: A promising technique for textiles". En: *Journal of Cleaner Production* 228 (2019), págs. 1242-1258.
- [32] Jitendra Kumar Mishra y Tarun Kumar Rath. "A review on microencapsulation: A promising technique for textile application". En: *Advances in Polymer Technology* 2021 (2021).
- [33] Jing Chen, Jinlian Hu y Yao Liu. "Microencapsulation of phase change materials for textile applications: A review". En: *Textile Research Journal* 85.15 (2015), págs. 1615-1632.
- [34] John McCarthy et al. "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955". En: *AI magazine* 27.4 (2006), págs. 12-12.
- [35] Allen Newell y Herbert A Simon. "Computer science as empirical inquiry: Symbols and search". En: *ACM Turing award lectures*. 2007, pág. 1975.
- [36] John McCarthy. "Recursive functions of symbolic expressions and their computation by machine, part I". En: *Communications of the ACM* 3.4 (1960), págs. 184-195.
- [37] Alan M Turing. *Computing machinery and intelligence*. Springer, 2009.
- [38] Arthur L Samuel. "Some studies in machine learning using the game of checkers. II—Recent progress". En: *IBM Journal of research and development* 11.6 (1967), págs. 601-617.
- [39] Allen Newell, Herbert Alexander Simon et al. *Human problem solving*. Vol. 104. 9. Prentice-hall Englewood Cliffs, NJ, 1972.
- [40] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.
- [41] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.
- [42] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep learning*. MIT press, 2016.
- [43] SpotCloud. *La diferencia entre Inteligencia Artificial, Machine Learning y Deep Learning*. 2021. URL: <https://spotcloud.medium.com/la-diferencia-entre-inteligencia-artificial-machine-learning-y-deep-learning-cc415f20e63a> (visitado 26-07-2023).
- [44] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. "Deep learning". En: *nature* 521.7553 (2015), págs. 436-444.
- [45] FutureLab. *Introducción a Redes Neuronales - Parte 1*. Jun. de 2019. URL: <https://futurelab.mx/redes%20neuronales/inteligencia%20artificial/2019/06/25/intro-a-redes-neuronales-pt-1/> (visitado 26-07-2023).
- [46] Warren S McCulloch y Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". En: *The bulletin of mathematical biophysics* 5 (1943), págs. 115-133.
- [47] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." En: *Psychological review* 65.6 (1958), pág. 386.
- [48] Mariano A. Hernandez. *Redes Neuronales Artificiales*. URL: <https://www.linkedin.com/pulse/redes-neuronales-artificiales-mariano-a-hernandez/?originalSubdomain=es> (visitado 26-07-2023).
- [49] David E Rumelhart, Geoffrey E Hinton y Ronald J Williams. "Learning representations by back-propagating errors". En: *nature* 323.6088 (1986), págs. 533-536.
- [50] *What is Forward and Backward Propagation in Deep Learning?* URL: <https://www.nomidl.com/deep-learning/what-is-forward-and-backward-propagation-in-deep-learning/> (visitado 26-07-2023).
-

- [51] Diego Calvo. *Red Neuronal Convolutacional: ¿Qué es y cómo funciona?* URL: <https://www.diegocalvo.es/red-neuronal-convolutacional/> (visitado 26-07-2023).
- [52] Alla Eddine Guissous. “Skin lesion classification using deep neural network”. En: *arXiv preprint arXiv:1911.07817* (2019).
- [53] *Arquitectura de red neuronal convolutacional (CNN)*. Datapeaker. URL: <https://datapeaker.com/big-data/arquitectura-de-red-neuronal-convolutacional-arquitectura-cnn/> (visitado 26-07-2023).
- [54] Yann LeCun et al. “Gradient-based learning applied to document recognition”. En: *Proceedings of the IEEE* 86.11 (1998), págs. 2278-2324.
- [55] Vladimir Golovko et al. “A Shallow Convolutional Neural Network for Accurate Handwritten Digits Classification”. En: vol. 673. Feb. de 2017, págs. 77-85. ISBN: 978-3-319-54219-5. DOI: 10.1007/978-3-319-54220-1_8.
- [56] Alex Krizhevsky, Ilya Sutskever y Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. En: *Advances in neural information processing systems* 25 (2012).
- [57] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge”. En: *International journal of computer vision* 115 (2015), págs. 211-252.
- [58] Francisco Rodríguez et al. “Estudio y caracterización de variedades de ciruelas utilizando análisis de imagen y técnicas de deep learning”. En: sep. de 2016.
- [59] Karen Simonyan y Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. En: *arXiv preprint arXiv:1409.1556* (2014).
- [60] Daniel Pérez-Aguilar, Redy Risco-Ramos y Luis Casaverde-Pacherrez. “Transfer learning en la clasificación binaria de imágenes térmicas transfer learning for binary classification of thermal images”. En: *Mach. Learn.* 550.26 (2021), pág. 4.
- [61] Christian Szegedy et al. “Going deeper with convolutions”. En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, págs. 1-9.
- [62] *Understanding GoogLeNet Model - CNN Architecture*. URL: <https://www.geeksforgeeks.org/understanding-googlenet-model-cnn-architecture/> (visitado 26-07-2023).
- [63] Kaiming He et al. “Deep residual learning for image recognition”. En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, págs. 770-778.
- [64] *Una visión general de ResNet y sus variantes*. URL: <https://datascience.eu/es/aprendizaje-automatico/una-vision-general-de-resnet-y-sus-variantes/> (visitado 26-07-2023).
- [65] *Phenom Pure Desktop Scanning Electron Microscope*. URL: <https://www.atascientific.com.au/products/phenom-pure-desktop-scanning-electron-microscope/> (visitado 26-07-2023).
- [66] Python Software Foundation. *Python en Español*. 2023. URL: <https://es.python.org/>.
- [67] *NumPy: The fundamental package for scientific computing with Python*. NumPy Developers. URL: <https://numpy.org/> (visitado 26-07-2023).
- [68] *pandas: Powerful data structures for data analysis, time series, and statistics*. pandas Development Team. URL: <https://pandas.pydata.org/> (visitado 26-07-2023).

-
- [69] *PyTorch*. PyTorch. URL: <https://pytorch.org/> (visitado 26-07-2023).
- [70] *SMP Documentation*. SMP. URL: <https://smp.readthedocs.io/en/latest/index.html> (visitado 26-07-2023).
- [71] *tiff file - PyPI*. Python Package Index (PyPI). URL: <https://pypi.org/project/tiff file/> (visitado 26-07-2023).
- [72] *OpenCV*. URL: <https://opencv.org/> (visitado 26-07-2023).
- [73] *Albumentations - Fast Image Augmentation Library*. URL: <https://albumentations.ai/>.
- [74] MathWorks. *MATLAB*. 2023. URL: <https://es.mathworks.com/products/matlab.html>.
- [75] Mobatek. *MobaXterm*. 2023. URL: <https://mobaxterm.mobatek.net/>.
- [76] Vue.js. *Vue.js - The Progressive JavaScript Framework*. 2023. URL: <https://vuejs.org/>.
- [77] Vuetify. *Vuetify*. 2023. URL: <https://vuetifyjs.com/en/>.
- [78] Axios. *Axios Documentation*. <https://axios-http.com/es/docs/intro>.
- [79] ExcelJS. *ExcelJS on npm*. <https://www.npmjs.com/package/exceljs>.
- [80] OpenSeadragon. *OpenSeadragon Documentation*. <https://openseadragon.github.io/>.
- [81] Body-parser. *Body-parser on npm*. <https://www.npmjs.com/package/body-parser>.
- [82] Flask. *Flask Documentation*. <https://flask.palletsprojects.com/en/2.3.x/>.
- [83] *Roboflow - Annotate Images Online*. 2023. URL: <https://roboflow.com/annotate>.
- [84] Olaf Ronneberger, Philipp Fischer y Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". En: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, págs. 234-241.
- [85] Mingxing Tan y Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". En: *International conference on machine learning*. PMLR. 2019, págs. 6105-6114.
- [86] Bhakti Baheti et al. "Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment". En: jun. de 2020, págs. 1473-1481. DOI: 10.1109/CVPRW50498.2020.00187.
- [87] Node.js Foundation. *What is Node.js?* 2023. URL: <https://nodejs.org/en/about/>.
- [88] kinsta. *Qué es Node.js y por qué debería usarlo*. 2023. URL: <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>.
- [89] npm. *cors*. 2023. URL: <https://www.npmjs.com/package/cors>.
- [90] Express.js. *Express.js - Node.js web application framework*. 2023. URL: <https://expressjs.com/>.
- [91] npm. *body-parser*. 2023. URL: <https://www.npmjs.com/package/body-parser>.
- [92] Trygve Reenskaug. "Models–views–controllers". En: *Journal of Object-Oriented Programming* 1.3 (1979), págs. 26-49.
-

- [93] Mauricio Costanzo. *¿Qué es el patrón MVC?* 2020. URL: <https://platzi.com/tutoriales/1248-pro-arquitectura/5466-que-es-el-patron-mvc/>.
- [94] *MongoDB*. URL: <https://www.mongodb.com/es> (visitado 02-08-2023).
- [95] *MongoDB Compass*. URL: <https://www.mongodb.com/es/products/compass%7D> (visitado 02-08-2023).
- [96] *Flask Documentation*. 2023. URL: <https://flask.palletsprojects.com/en/2.1.x/>.
- [97] *Docker*. 2023. URL: <https://www.docker.com/>.
- [98] Bikram. *Docker Objects*. 2021. URL: <https://bikramat.medium.com/docker-objects-e561f0ce3365>.
- [99] España. “Real Decreto 99/2011, de 28 de enero, por el que se regulan las enseñanzas oficiales de doctorado”. En: *Boletín Oficial del Estado* 31.5 (feb. de 2011), págs. 13676-13706.
- [100] España. “Ley Orgánica 4/2007, de 12 de abril, por la que se modifica la Ley Orgánica 6/2001, de 21 de diciembre, de Universidades”. En: *Boletín Oficial del Estado* 89.15 (abr. de 2007), págs. 15939-15946.
- [101] España. “Real Decreto 131/2007, de 1 de febrero, por el que se regula el régimen de los trabajadores extranjeros en España”. En: *Boletín Oficial del Estado* 29.5 (feb. de 2007), págs. 5425-5433.
- [102] *Servicio de gestión de la I+D+i de la UPV*. Universitat Politècnica de València. 2023. URL: <http://www.upv.es/entidades/SGI/info/706854normalc.html>.

Parte II

Presupuesto

Capítulo 1

Introducción

1.1. Alcance y descripción general

El alcance y la descripción general engloban los límites y las metas del proyecto del desarrollo de la aplicación de conteo automatizado de microcápsulas en imágenes textiles, así como una visión general de los componentes y procesos involucrados en el proyecto. Esta sección proporciona una comprensión clara de lo que el proyecto abarcará y cómo se llevarán a cabo las diversas actividades.

El proyecto tiene como objetivo principal desarrollar una solución completa y eficiente para el conteo automatizado de microcápsulas en imágenes textiles. A través de una interfaz web intuitiva, los usuarios podrán cargar sus imágenes y recibir resultados precisos en términos de conteo de microcápsulas. El servicio de IA basado en U-Net permitirá un análisis preciso y rápido de las imágenes, mejorando la eficiencia y la confiabilidad del proceso de conteo.

El sistema se basa en una arquitectura modular, donde cada componente cumple una función específica y se comunica con los demás a través de protocolos HTTP. El uso de contenedores Docker garantizará la portabilidad y la escalabilidad del sistema en diferentes entornos.

Por lo que el alcance del proyecto se define de la siguiente manera:

- **Aplicación web y plataforma interactiva:** El proyecto se centra en la creación de una aplicación web con una interfaz de usuario intuitiva. Esta aplicación permitirá a los usuarios cargar imágenes y recibir resultados precisos de conteo de microcápsulas. Se desarrollarán cinco vistas interactivas que ofrecerán una experiencia intuitiva al usuario.
- **Desarrollo e integración de IA:** Se implementará un servicio utilizando Flask que aprovechará una arquitectura de red neuronal convolucional U-Net para la detección automatizada de microcápsulas en las imágenes. Este servicio procesará las imágenes cargadas y proporcionará el número de microcápsulas detectadas.
- **Base de datos y almacenamiento:** Se establecerá una base de datos MongoDB para almacenar imágenes cargadas, nombres de los proyectos y otros datos relevantes. Esto permitirá un acceso rápido y seguro a la información almacenada.
- **Frontend y Backend:** Se utilizarán tecnologías como Vue.js y Node.js para desarrollar el frontend y el backend respectivamente. El frontend permitirá a los usuarios cargar imágenes

y visualizar los resultados, mientras que el backend gestionará las solicitudes, el procesamiento de imágenes y la interacción con la base de datos.

- **Comunicación HTTP:** Se utilizarán métodos HTTP, como GET y POST, para permitir la comunicación entre el frontend, el backend, la base de datos y el servicio de IA. Esto garantizará una transferencia eficiente y segura de datos entre los diferentes componentes.
- **Uso de Docker:** El proyecto contempla la implementación de Docker para crear contenedores aislados que contengan los diferentes componentes del sistema, como el frontend, el backend, la base de datos y el servicio de IA. Esto facilitará el despliegue y la escalabilidad del sistema en diferentes entornos.

Capítulo 2

Costes de desarrollo

2.1. Personal

En el desarrollo de un proyecto, el recurso humano desempeña un papel fundamental en la materialización de los objetivos y la realización de las metas propuestas. Los costes asociados al personal también reflejan el valor y la experiencia que cada miembro del equipo aporta al proyecto. Los participantes de este proyecto son los siguientes:

- Valery Naranjo Ornedo, Catedrática de Universidad y tutora: ha brindado orientación y supervisión al proyecto.
- Maria Peris Torrent y Leticia Mariana Launet, ingenieras y cotutoras: han aportado sus habilidades técnicas tanto en el campo del desarrollo web como en el campo de la IA.
- María Angeles Bonet Aracil, Catedrática de Universidad: ha suministrado las imágenes textiles necesarias para el desarrollo del proyecto.
- Ana Martín Sanchis, Estudiante del MUAI y autora del proyecto: ha desarrollado el proyecto con gran dedicación.

A continuación en la tabla 2.1, se desglosan los honorarios y costes asociados a cada uno de los miembros del equipo que ha participado en este proyecto, teniendo en cuenta las leyes y regulaciones vigentes. Cada contribución individual se refleja en la calidad y el alcance de nuestro proyecto, y es esencial comprender cómo estos costes se alinean con la visión global de la iniciativa que estamos llevando a cabo.

Los sueldos de los ingenieros que trabajan en universidades suelen estar regulados por las leyes laborales generales, los convenios colectivos y las normativas específicas de las universidades. Los colegios profesionales de ingenieros pueden ofrecer orientación y recursos sobre aspectos profesionales y éticos, pero no tienen autoridad para establecer salarios.

Algunas de las leyes que regulan las condiciones laborales de los profesores, catedráticos y ingenieros en las universidades españolas son:

- Estatuto del Personal Investigador en Formación (EPIF): Esta ley establece las condiciones laborales y académicas de los investigadores en formación, incluidos los doctorandos que trabajan en proyectos de investigación [99].
- Ley Orgánica de Universidades (LOU): Esta ley establece el marco general para la organización y el funcionamiento de las universidades en España. Contiene disposiciones sobre la contratación y promoción de profesores e investigadores [100].
- Estatuto del Personal Docente e Investigador Contratado Laboral (EPDICL): Esta ley regula las condiciones laborales y de carrera de los profesores e investigadores contratados laborales en las universidades [101].

Incorporando la información actualizada hasta 2023, en relación al servicio de Gestión de la I+D+i de la UPV [102] y su contribución en la elaboración de presupuestos y los costes promedio de los diversos perfiles del proyecto, los costes unitarios de personal se detallan de la siguiente manera:

- Coste medio catedrático: 58.54 €
- Coste máximo técnico superior: 24.99 €
- Coste mínimo técnico superior: 13.32 €

Por otro lado, mediante la ayuda de una beca formativa de colaboración del Instituto de Investigación e Innovación en Bioingeniería (i3B) en la UPV, en concreto con el grupo de investigación Computer Vision and Behaviour Analysis Lab (CVBLab) se aporta una cantidad mensual, atendiendo a las horas de trabajo de un TFM, para incentivar la investigación y el estudio en este campo.

A partir de esta información, la tabla con los costes personales es la siguiente:

Descripción	Uds	Cantidad	Precio unitario (€/h)	Coste imputable
Catedrática, Valery	h	20	58,54	1170,80
Catedrática, Maria Angeles	h	5	58,54	292,70
Técnica, Maria	h	20	24,99	499,80
Técnica, Leticia	h	10	24.99	249,90
Estudiante, Ana	h	375	13.32	4995,00
TOTAL				7208,20

Tabla 2.1: Desglose de los costes de personal

Cada miembro del equipo ha desempeñado un papel fundamental para lograr los objetivos propuestos. En la tabla 2.2 se presenta el desglose de tareas y las horas dedicadas a cada una de ellas. La suma total de horas reales frente a las estimadas nos proporciona una comprensión valiosa del esfuerzo y tiempo invertido en cada etapa del proyecto.

Descripción de la tarea	Uds	Horas estimadas	Horas reales
Planteamiento de objetivos	h	10	15
Selección y preparación de las imágenes textiles	h	20	40
Estudio y desarrollo de modelos de IA	h	140	160
Creación y diseño del sitio web	h	140	150
Verificación del correcto funcionamiento	h	15	15
Redacción de documentos técnicos	h	50	50
TOTAL		375	430

Tabla 2.2: Desglose de las tareas realizadas

Esta visión detallada de las tareas y el tiempo dedicado a ellas nos permite apreciar la complejidad y diversidad de las contribuciones individuales que han dado forma al proyecto final.

2.2. Herramientas y licencias

En la realización del proyecto se depende en gran medida de herramientas y recursos tecnológicos. Tanto el software como el hardware han sido elementos clave, permitiéndonos desarrollar, probar y presentar las ideas de manera efectiva.

Las siguientes tablas detallan los costes asociados con las herramientas y licencias que se han utilizado en el transcurso de este proyecto. Se entiende que todo este equipo será utilizado en sucesivos trabajos o investigaciones, con lo que el coste será parcial, y no total.

Descripción	Uds	Precio unitario sin IVA (€)	Cuota de adquisición (€)	Periodo de amortización (meses)	Cuota de amortización (meses)	Tiempo de uso (meses)	Amortización
Sistema operativo Windows 11 Pro	1	181,82	181,82	48	3,787	7	26,515
Overleaf	1	0	0	12	0	7	0
Python	1	0	0	12	0	7	0
MongoDB	1	0	0	12	0	7	0
MobaXtern	1	0	0	12	0	7	0
Flask	1	0	0	12	0	7	0
JavaScript	1	0	0	12	0	7	0
Librerías	1	0	0	12	0	7	0
Roboflow	1	0	0	12	0	7	0
TOTAL sin IVA							26,515
IVA (21 %)							5,3
TOTAL							31,818

Tabla 2.3: Desglose de los costes del software

CAPÍTULO 2. COSTES DE DESARROLLO

Descripción	Uds	Precio unitario sin IVA (€)	Cuota de adquisición (€)	Periodo de amortización (meses)	Cuota de amortización (meses)	Tiempo de uso (meses)	Amortización
Phenom microscope	1	26884,3	26884,3	180	149,357	0,03	4,978
DELL Vostro 5471 Intel(R) Core(TM) i5-8250U, @1.60GHz, 16GB RAM	1	715,2	715,2	72	9,933	7	69,533
Procesador Intel i7 @4.20GHz	1	344	344	48	7,166	7	50,166
Tarjeta gráfica NVIDIA Titan V	1	3300	3300	48	68,750	7	481,25
NAS Synology DS918	1	77	77	48	1,604	7	11,229
TOTAL sin IVA							617,157
IVA (21 %)							123,431
TOTAL							740,589

Tabla 2.4: Desglose de los costes del hardware

Capítulo 3

Total del presupuesto

En la tabla 3.1 se ha desarrollado el resumen de los costes con el total del proyecto. Este resultado comprende todos los aspectos esenciales, desde los recursos humanos hasta las herramientas tecnológicas necesarias para llevar a cabo nuestro proyecto de manera efectiva.

El presupuesto total calculado para el proyecto es de siete mil novecientos ochenta con seiscientos siete euros.

	Costes (€)
Personal	7208,200
Software	31,818
Hardware	740,589
TOTAL	7980,607

Tabla 3.1: Desglose de los costes totales del proyecto

Capítulo 4

Conclusión

La elaboración de este presupuesto para el proyecto "Desarrollo de un sistema de visión artificial basado en redes convolucionales para el conteo automatizado de microcapsulas en textiles" ha sido un ejercicio esencial para comprender y gestionar los recursos financieros involucrados en esta iniciativa. A través de esta evaluación detallada, hemos obtenido una visión clara de cómo se distribuyen los costes en diversas áreas, desde el personal y las horas dedicadas hasta las herramientas y licencias utilizadas.

Bibliografía

- [99] España. “Real Decreto 99/2011, de 28 de enero, por el que se regulan las enseñanzas oficiales de doctorado”. En: *Boletín Oficial del Estado* 31.5 (feb. de 2011), págs. 13676-13706.
- [100] España. “Ley Orgánica 4/2007, de 12 de abril, por la que se modifica la Ley Orgánica 6/2001, de 21 de diciembre, de Universidades”. En: *Boletín Oficial del Estado* 89.15 (abr. de 2007), págs. 15939-15946.
- [101] España. “Real Decreto 131/2007, de 1 de febrero, por el que se regula el régimen de los trabajadores extranjeros en España”. En: *Boletín Oficial del Estado* 29.5 (feb. de 2007), págs. 5425-5433.
- [102] *Servicio de gestión de la I+D+i de la UPV*. Universitat Politècnica de València. 2023. URL: <http://www.upv.es/entidades/SGI/info/706854normalc.html>.

Parte III

Anexos

Apéndice A

Manual de usuario

Para acceder a la aplicación, siga estos pasos:

1. Abra cualquier navegador que tenga instalado en su dispositivo y escriba la dirección web correspondiente en la barra de direcciones.
2. Una vez que haya ingresado la dirección mencionada anteriormente, se encontrará en la página de inicio de la aplicación, que se visualiza de la siguiente manera:

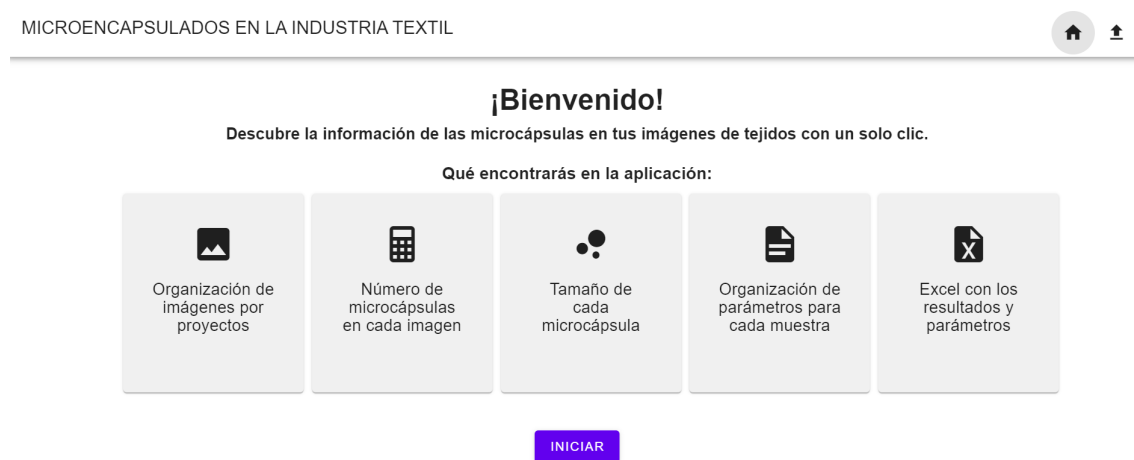


Figura A.1: Página inicial de la aplicación web

3. El siguiente paso consiste en hacer clic en el botón “Iniciar” para ingresar a la aplicación:

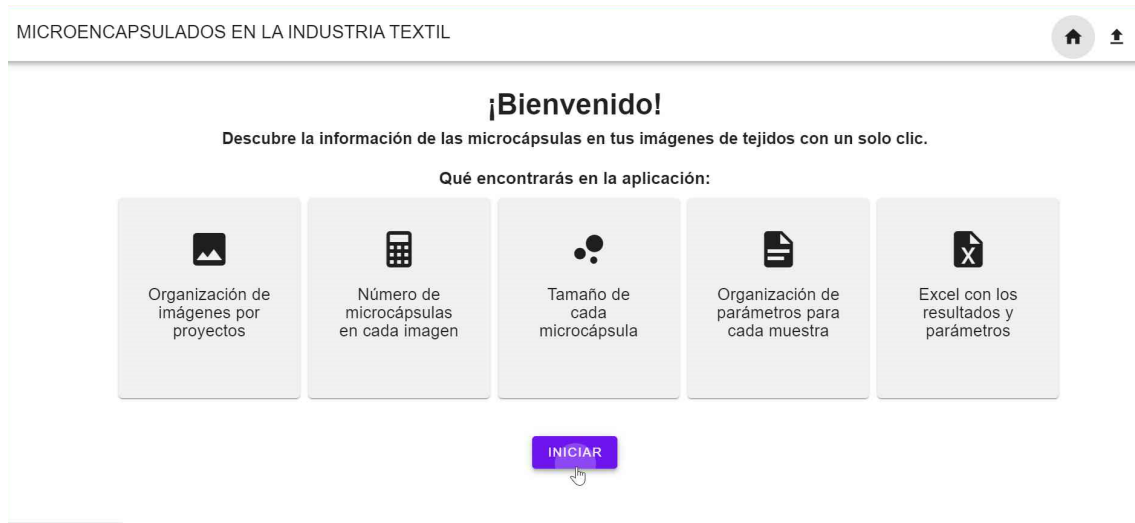


Figura A.2: Pulsar el botón de “Iniciar”

4. Una vez que esté en esta página, deberá crear el proyecto al que desea acceder en caso de que aún no esté creado. Para ello debe rellenar el campo de “Nombre de proyecto” y pulsar el botón “Crear”:

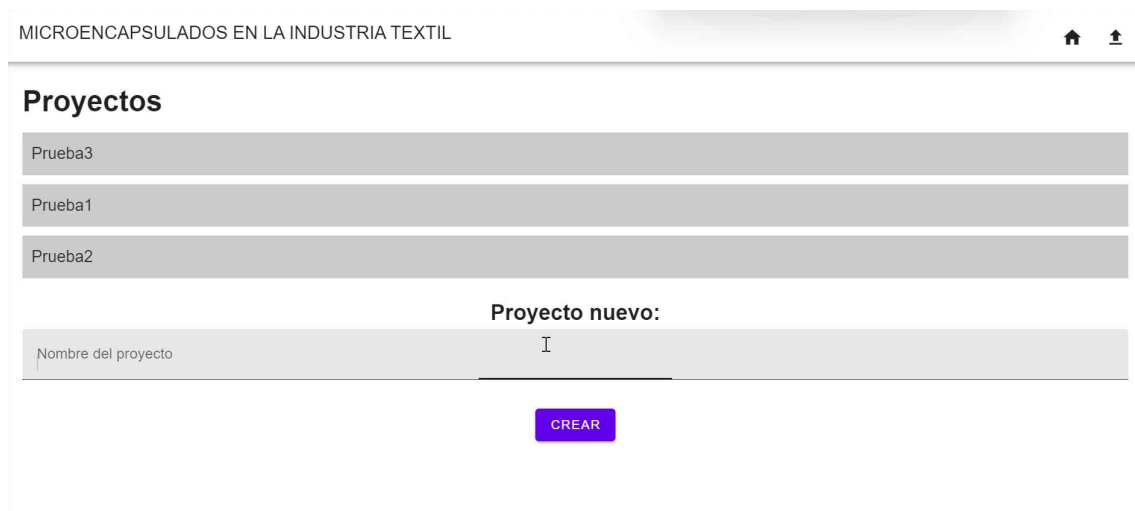


Figura A.3: Rellenar el campo de “Nombre de proyecto”

Proyectos

Prueba3

Prueba1

Prueba2

Proyecto nuevo:

Nombre del proyecto
Prueba 4

CREAR



Figura A.4: Pulsar el botón de “Crear”

5. Después de crear el proyecto, podrá acceder al proyecto que desee:

Proyectos

Prueba3

Prueba1

Prueba2

Prueba 4

Proyecto nuevo:

Nombre del proyecto

CREAR

Figura A.5: Acceder al proyecto

6. Dentro del proyecto, deberá cargar las imágenes que le interesen. Para cargar las imágenes, simplemente haga clic en el botón con el icono de subir imágenes:



Figura A.6: Pulsar el botón para subir imágenes

7. Se abrirá una página en la que, al hacer clic en el botón “Elegir imágenes”, se mostrarán sus carpetas locales, desde donde podrá seleccionar las imágenes que desea cargar en el proyecto.

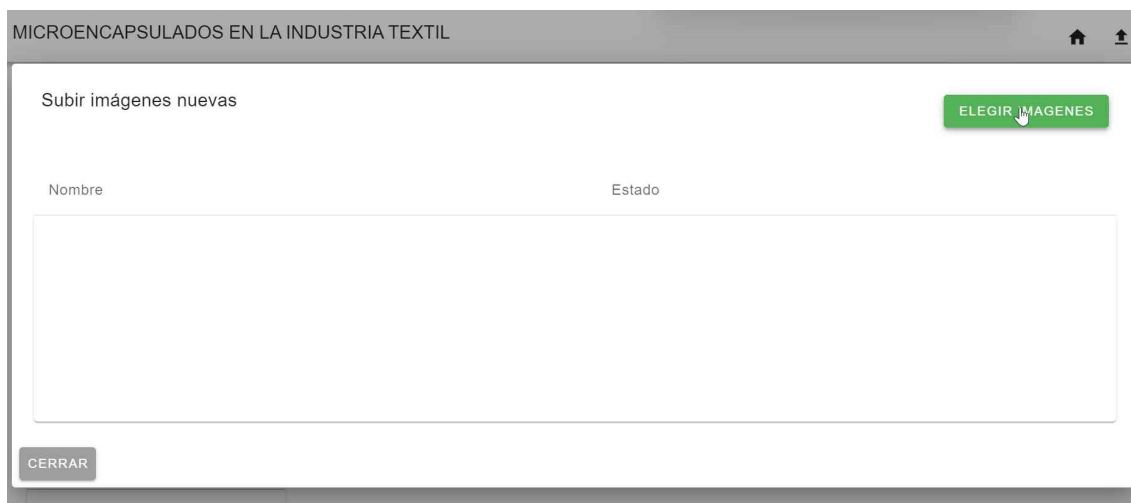


Figura A.7: Subir imágenes nuevas

8. Para verificar si las imágenes se han cargado correctamente, verá una barra de progreso que indicará el estado del proceso. Una vez que las imágenes se hayan cargado con éxito, verá la palabra “Uploaded”. Después de esto, puede hacer clic en el botón “Cerrar” para regresar a la página principal del proyecto:

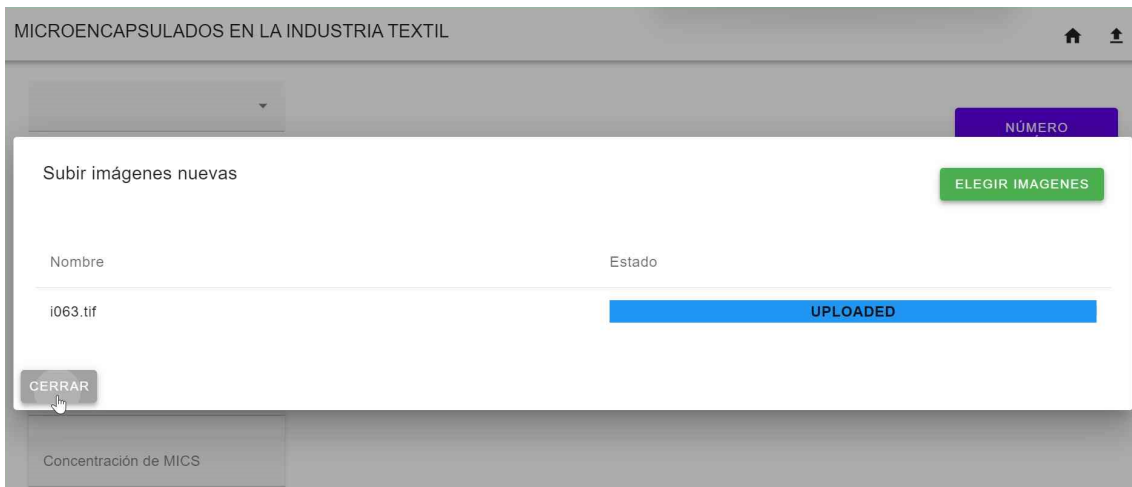


Figura A.8: Imágenes subidas

9. En el menú desplegable de la izquierda, puede seleccionar la imagen que desea analizar. Luego, esta imagen aparecerá en grande en la página principal del proyecto:

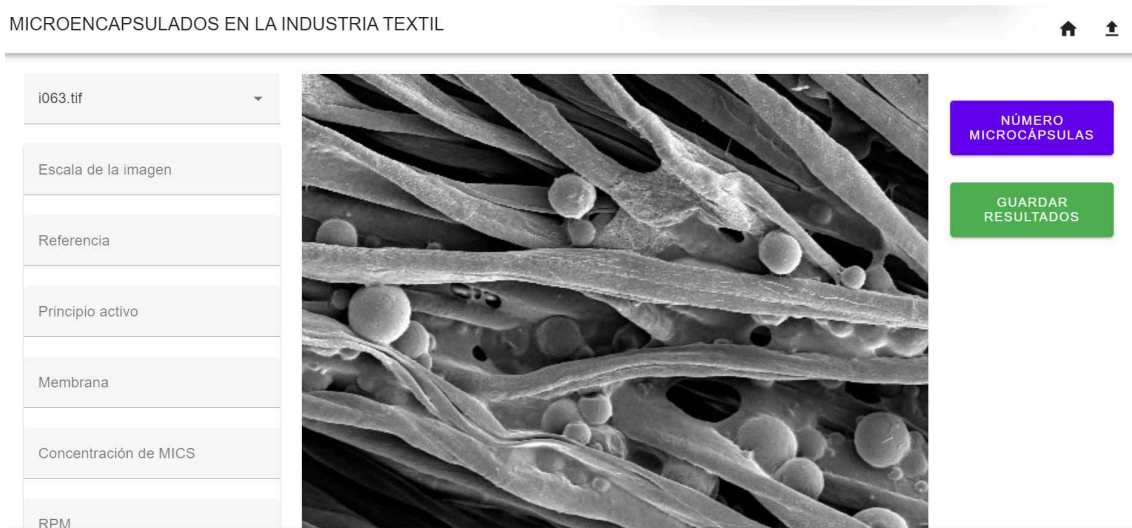


Figura A.9: Seleccionar la imagen que desea

10. A continuación, complete los campos de los parámetros de la imagen:

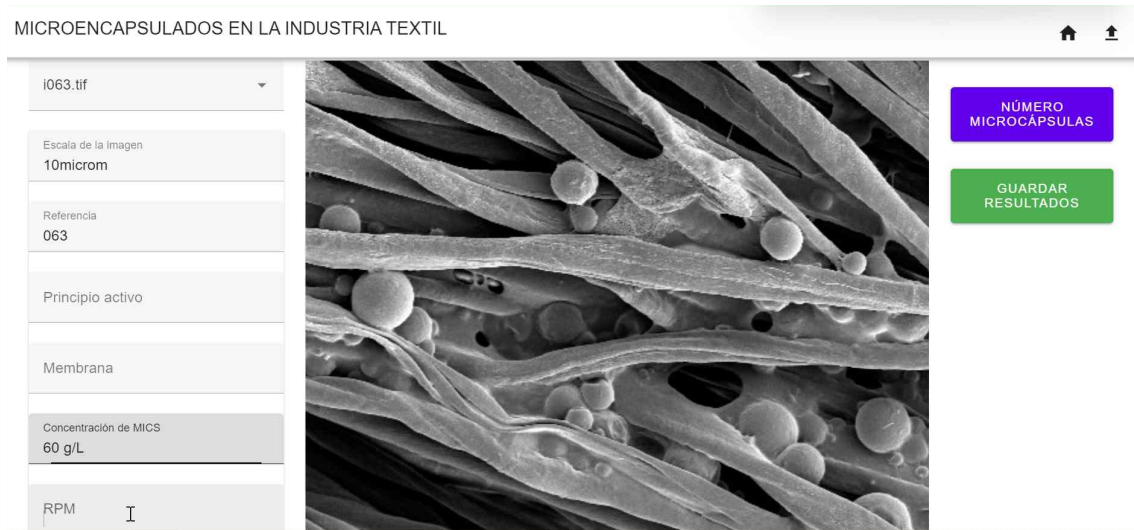


Figura A.10: Parámetros de la imagen

11. Para calcular el número de microcápsulas, simplemente haga clic en el botón “Número de microcápsulas”, y se abrirá una ventana que mostrará el resultado final.



Figura A.11: Pulsar el botón de “Número de microcápsulas”

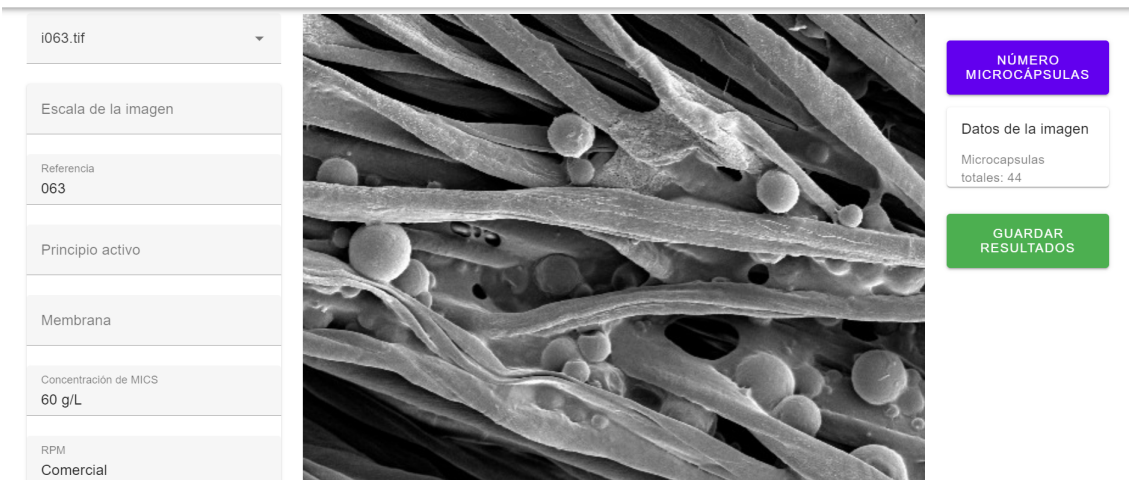


Figura A.12: Resultado final

12. Por último, si desea guardar los resultados obtenidos y los parámetros de la imagen, simplemente haga clic en el botón “Guardar resultados”:

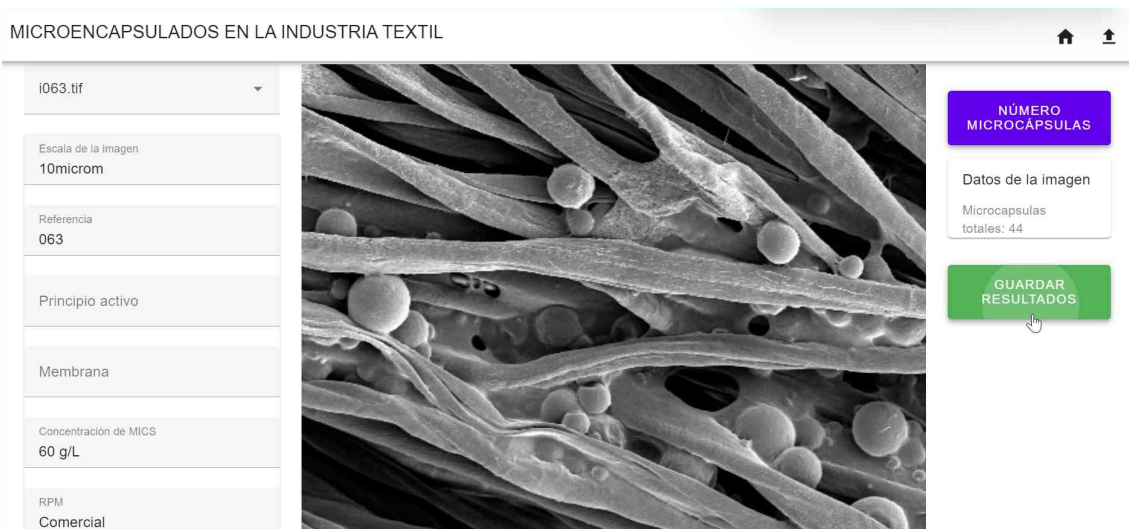


Figura A.13: Pulsar el botón de “Guardar resultados”

Se guardará un archivo Excel local con un nombre personalizado, que seguirá el formato “NombreDelProyecto_NombreDeLaImagen_resultados.xlsx”. En este formato, el primer parámetro corresponderá al nombre del proyecto, y el segundo al nombre de la imagen que ha sido analizada.

Por último, el archivo Excel tendrá el siguiente formato:

	A	B
1	Proyecto	Prueba 4
2	Imagen	i063.tif
3		
4	Parámetro	Valor
5	Escala de la imagen	10microm
6	Referencia	063
7	Principio activo	
8	Membrana	
9	Concentración de MICS	60 g/L
10	RPM	Comercial
11		
12	Número de microcápsulas	44

Figura A.14: Ejemplo del archivo Excel