

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Procesos Tecnológicos e Industriales

Sustentabilidad y tecnología

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

Programa de Apoyo a Centros de Investigación Externos II



**ITESO, Universidad
Jesuita de Guadalajara**

4G03 Programa de Apoyo a Centros de Investigación Externos II
Construcción de un modelo basado en redes neuronales profundas de
código abierto para el conteo, localización y clasificación de colonias en
placas Petri.

PRESENTAN

Programas educativos y Estudiantes

Ing. en Biotecnología, Hugo Munguía Orozco.

Profesor PAP: Dr. Melchor Arellano Plaza.

Tlaquepaque, Jalisco, diciembre, 2023.

ÍNDICE

Contenido

REPORTE PAP	2
Presentación Institucional de los Proyectos de Aplicación Profesional	2
Resumen	3
1. Ciclo participativo del Proyecto de Aplicación Profesional.....	4
1.1 Entendimiento del ámbito y del contexto	6
1.2 Caracterización de la organización.....	11
1.3 Identificación de la(s) problemática(s).....	12
1.4. Planeación de alternativa(s).....	20
1.5. Desarrollo de la propuesta de mejora	26
1.6. Valoración de productos, resultados e impactos	33
1.7. Bibliografía y otros recursos	34
2. Productos	36
3. Reflexión crítica y ética de la experiencia.....	37
3.1 Sensibilización ante las realidades	37
3.2 Aprendizajes logrados	38

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son experiencias socio-profesionales de los alumnos que desde el currículo de su formación universitaria- enfrentan retos, resuelven problemas o innovan una necesidad sociotécnica del entorno, en vinculación (colaboración) (co-participación) con grupos, instituciones, organizaciones o comunidades, en escenarios reales donde comparten saberes.

El PAP, como espacio curricular de formación vinculada, ha logrado integrar el Servicio Social (acorde con las Orientaciones Fundamentales del ITESO), los requisitos de dar cuenta de los saberes y del saber aplicar los mismos al culminar la formación profesional (Opción Terminal), mediante la realización de proyectos profesionales de cara a las necesidades y retos del entorno (Aplicación Profesional).

El PAP es un proceso acotado en el tiempo en que los estudiantes, los beneficiarios externos y los profesores se asocian colaborativamente y en red, en un proyecto, e incursionan en un mundo social, como actores que enfrentan verdaderos problemas y desafíos traducibles en demandas pertinentes y socialmente relevantes. Frente a éstas transfieren experiencia de sus saberes profesionales y demuestran que saben hacer, innovar, co-crear o transformar en distintos campos sociales.

El PAP trata de sembrar en los estudiantes una disposición permanente de encargarse de la realidad con una actitud comprometida y ética frente a las disimetrías sociales. En otras palabras, se trata del reto de “saber y aprender a transformar”.

El Reporte PAP consta de tres componentes:

El primer componente refiere al ciclo participativo del PAP, en donde se documentan las diferentes fases del proyecto y las actividades que tuvieron lugar durante el desarrollo de este y la valoración de las incidencias en el entorno.

El segundo componente presenta los productos elaborados de acuerdo con su tipología.

El tercer componente es la reflexión crítica y ética de la experiencia, el reconocimiento de las competencias y los aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

En el PAP “Construcción de un modelo basado en redes neuronales profundas de código abierto para el conteo, localización y clasificación de colonias en placas de Petri”, se buscó generar una herramienta innovadora para romper la barrera a las tecnologías de redes neuronales para la tarea repetitiva de conteo de microorganismos. El principal objetivo es otorgar acceso libre a un modelo de software y dispositivo físico que permita automatizar las tareas de microbiología de conteo de colonias en medios sólidos.

Se logró la programación en Python de un modelo básico de redes neuronales convolucionales utilizando las librerías Keras y Tensor Flow conteo de células bacterianas basándose en mapas de densidad. El modelo utilizó el error medio cuadrático como función de pérdida. Dicho modelo logró un 95% de exactitud sin presentar sobreajuste a los datos de entrenamiento utilizando un juego de datos sintéticos que emulan células bacterianas observadas mediante microscopía fluorescente

Adicionalmente, se diseñó en OpenSCAD un dispositivo capaz de alojar placas de 35 mm a 90mm para ser leídas mediante una cámara RGB de 1920 × 1080 píxeles y adicionado con iluminación LED para mejorar la obtención de imágenes. Posteriormente, el modelo fue impreso en 3D utilizando filamento de ácido poliláctico blanco para dejar un prototipo físico cuyo costo de producción fue de \$475.00 MXN.

Finalmente, los recursos de software generados fueron alojados en un repositorio de Github con licencia de uso MIT para facilitar el acceso a la información presentada en este proyecto, así como a futuras actualizaciones.

1. Ciclo participativo del Proyecto de Aplicación Profesional

El PAP es una experiencia de aprendizaje y de contribución social integrada por estudiantes, profesores, actores sociales y responsables de las organizaciones, que de manera colaborativa construyen sus conocimientos para dar respuestas a problemáticas de un contexto específico y en un tiempo delimitado. Por tanto, la experiencia PAP supone un proceso en lógica de proyecto, así como de un estilo de trabajo participativo y recíproco entre los involucrados.

El presente proyecto se desenvuelve para apoyar al Centro de Investigación y Asistencia en Tecnología y Diseño del Estado de Jalisco (CIATEJ), que forma parte de la coordinación de medio ambiente, salud y alimentación del Sistema de Centros Públicos de Investigación (CPI) del Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT). Este centro de investigación se enfoca en actividades de investigación, desarrollo tecnológico e innovación (I+D+i), así como servicios tecnológicos y de formación de recursos humanos con programas de posgrado. Dentro del centro se localiza la Unidad de Biotecnología Industrial, dirigido por la Doctora Lorena Amaya Delgado, cuyo objetivo es encontrar soluciones biotecnológicas para el desarrollo sustentable de bioprocesos en los sectores agropecuario, agua y medio ambiente, alimentos y bebidas, salud humana y animal, así como bioprocesos y energía.

Bajo la dirección del Dr. Melchor Arellano Plaza, se desarrolló el presente proyecto para develar las interacciones entre levaduras y bacterias en bebidas alcohólicas de importancia social para el estado de Jalisco, tales como raicilla, mezcal, tuba, lechuguilla, tejuino, etc. Para esto, en el PAP a llevado a cabo en el periodo de verano del 2023 por la estudiante de ingeniería Itza Xareni Noyola Hernández, se buscó vislumbrar el efecto que tienen las interacciones bacteria-bacteria, bacteria-levadura y levadura-levadura en las cualidades organolépticas de fermentos realizados con las levaduras *Saccharomyces cerevisiae*, *Pichia kluyveri* y *Kluyveromyces marxianus* y las bacterias *Limosilactobacillus fermentum* y *Lactiplantibacillus plantarum* utilizando como sustratos preparados sintéticos que simulan el jugo de agave y jugo de uva. Sin embargo, se encontró que dichas interacciones son

multifactoriales y presentan niveles de complejidad que deben ser dilucidados en campos de estudio más específicos antes de analizar el problema general.

Por lo tanto, en el presente Proyecto de Aplicación Profesional (PAP) se buscó incorporar a la investigación realizada en verano del año 2023 un mecanismo que permita el conteo, localización y clasificación de colonias generadas por microorganismos en placas de Petri para asistir al proceso de conteo de colonias en este o cualquier otro proceso en el que se requiera en el centro de investigación o en el ITESO, así como identificación de la proporción de bacterias y levaduras creciendo en medios de cultivo sólido. El desarrollo de este modelo se realizará con código abierto para que futuros estudiantes o investigadores puedan utilizarlo como herramienta educativa, interactiva y de trabajo, fomentando así la generación de un software colaborativo propietario del CIATEJ y el ITESO. De forma adicional, se busca proponer un diseño de dispositivo que albergue los componentes necesarios para hacer uso del modelo implementado. De esta manera se podrá desarrollar una versión inicial del software y hardware para hacer posible la creación de un dispositivo funcional para conteo automatizado de colonias.

El presente proyecto es una combinación de simulación, desarrollo de software y experimentación simple. La simulación involucra el diseño e impresión en 3D de un prototipo de dispositivo para leer las placas de Petri, el desarrollo de software involucra la generación del modelo de aprendizaje profundo y la experimentación consiste en la generación de un juego de datos para entrenamiento y validación del modelo de aprendizaje profundo.

Objetivo general

Desarrollar un modelo basado en redes neuronales profundas para conteo, localización y clasificación de colonias en medios sólidos, así como el hardware necesario para utilizar dicho modelo, de manera que se pueda asistir a procesos de educación e investigación en sectores biotecnológicos.

Objetivos específicos

- Desarrollar un modelo basado en redes neuronales profundas en Python que permita contar, localizar y clasificar colonias en medios sólidos a través de fotografías.
- Entrenar el modelo con un juego de datos de colonias en placas de Petri desarrollado por terceros.
- Generar un conjunto de datos para validar el modelo creado, utilizando las cepas relevantes a la investigación dirigida por el Dr. Melchor Arellano Plaza en el CIATEJ con los medios sintéticos de jugo de agave y medio de cultivo convencionales.
- Diseñar una propuesta inicial del hardware que albergará los componentes necesarios para el uso del modelo desarrollado.

1.1 Entendimiento del ámbito y del contexto

En el presente PAP se aprovechan las redes neuronales artificiales (ANN por sus siglas en inglés), en específico las redes convolucionales, para la realización de tareas de visión computacional. Las ANN se basan en la interconexión de unidades computacionales llamadas neuronas, que tratan de imitar el comportamiento de una neurona biológica [1] a través de la simulación de las funciones de cada parte de la neurona biológica mostradas en la figura 1.

En la neurona computacional, las dendritas funcionan como el “input” o entrada de datos hacia la neurona y se encuentran conectadas con órganos sensoriales o con otras dendritas a través de sinapsis. Por otro lado, el soma puede decirse que “recolecta” los datos que ingresan por las dendritas y determina si la entrada de datos generará un estímulo que pasará a través del axón. Si se dispara una señal en el axón, el núcleo, que previamente se había “excitado” por los estímulos recibidos, regresa a su estado estacionario, frenando entonces la señal que se dispara por el axón. La señal que pasa por el axón es después distribuida hacia otras neuronas en de forma unidireccional: de las terminales del axón hacia las dendritas de otra neurona [1].

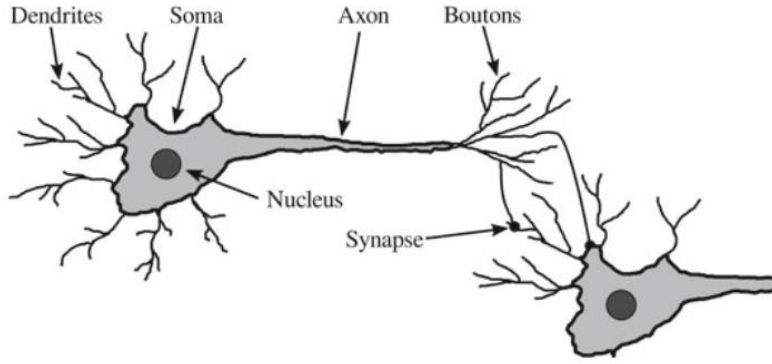


Figura 1. Diagrama simplificado de las partes de una neurona biológica [1].

De forma matemática, de acuerdo con [1], una neurona biológica se describe como $f(\mathbf{x}) = \mathcal{G}(\mathbf{w}\mathbf{x}^T + b)$, donde $\mathbf{w} \in \mathbb{R}^d$ es un vector de “pesos” que modifican a la variable $\mathbf{x} \in \mathbb{R}^d$ que representa a los “inputs” o datos de entrada; $b \in \mathbb{R}$ representa un intercepto o “bias” que ayuda a encontrar soluciones que no pasen por el origen. Finalmente, $\mathcal{G}(\mathbf{x}): \mathbb{R} \rightarrow \mathbb{R}$ es una función no lineal llamada “función de activación” que toma un número real y retorna otro número real después de aplicar una transformación no lineal. Esta función $\mathcal{G}(\mathbf{x})$ es la encargada de fijar un umbral que determinará si la neurona se excita o se inhibe dependiendo del “potencial del núcleo” representado por los datos de entrada modificados por los pesos y el término de “bias”, es decir, $\mathbf{w}\mathbf{x}^T + b$ [1]. Una representación gráfica de la expresión matemática que define a la neurona artificial, o computacional, se presenta en la figura 2.

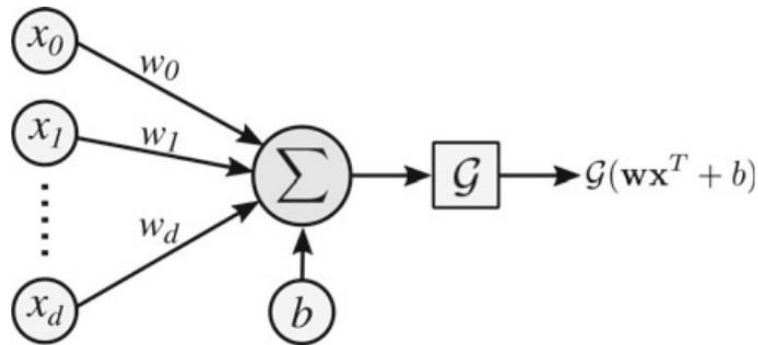


Figura 2. Representación diagramática de una neurona artificial. x_d : datos de entrada a una neurona, w_d : pesos de cada conexión entre neuronas, b : término de “bias” en la función de activación, \mathcal{G} : función de activación [1].

Ahora bien, las ANN completamente conectadas (FC por sus siglas en inglés) son conexiones entre las neuronas artificiales previamente descritas, donde existen capas de neuronas, que pueden verse como columnas i en una matriz, y cada neurona de una capa j está conectada con todas las neuronas de la capa $j + 1$, generando pesos w_{ij} para cada una de dichas conexiones [1]. De esta forma, la salida de cada función de activación de las neuronas en una capa j se convierte en la entrada de otras neuronas en la capa $j + 1$, por lo que podrían verse a estas redes neuronales como una serie de funciones compuestas o anidadas, por ejemplo, el modelo mostrado en la figura tres se representaría como la función $f(x) = \mathcal{G}(\mathcal{G}(\mathcal{G}(xW^1 + b^1)W^2 + b^2)W^3 + b^3)$, donde W^i es el vector de pesos W entre cada capa. Cabe destacar que el término x únicamente se presenta en la primera evaluación de la función anidada debido a que después de evaluar esta función, su resultado se convierte en la x para la siguiente función.

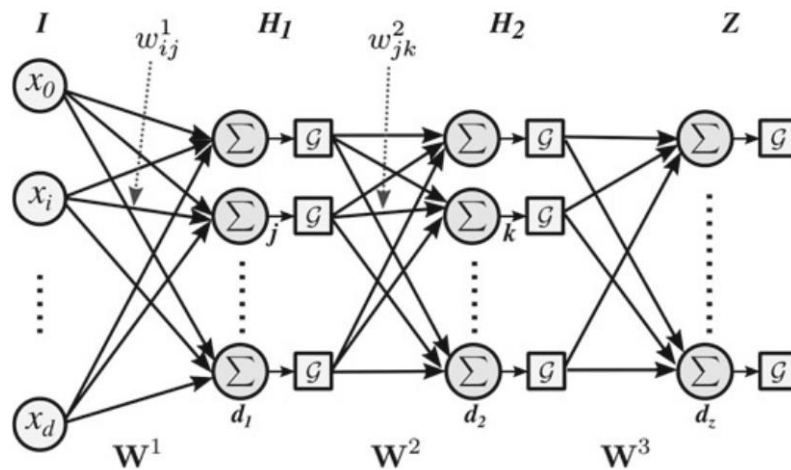


Figura 3. Grafo de una red neuronal acíclica que acepta datos d -dimensionales y retorna un vector d -dimensional. I : capa de datos de entrada, H_i : capas ocultas, Z : capa de salida [1].

Por otro lado, es importante notar que las funciones de activación para cada capa pueden ser distintas, otorgando al modelo flexibilidad para incorporar funciones lineales y no lineales dentro de su arquitectura. Algunas de las funciones de activación más utilizadas, de acuerdo con [1], son la función sigmoide $\mathcal{G}_{sigmoide}(x) = \frac{1}{1+e^{-x}}$ con derivada $\mathcal{G}'_{sigmoide}(x) = \mathcal{G}(x)(1 - \mathcal{G}(x))$, la función tangente hiperbólica $\mathcal{G}_{tanh}(x) = \frac{2}{1+e^{-2x}} - 1$ con derivada

$\mathcal{G}'_{\tanh}(x) = 1 - \mathcal{G}_{\tanh}(x)^2$, unidad lineal rectificada (ReLU) $\mathcal{G}_{ReLU}(x) = \max(0, x)$ con derivada $\mathcal{G}'_{ReLU}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$ y finalmente, softmax.

Una ventaja de las redes neuronales acíclicas es que son aproximadores universales de funciones continuas, por lo que son útiles para encontrar funciones que ayuden a encontrar funciones de transformación que ayuden a cumplir una tarea, como por ejemplo clasificación, sin la necesidad de diseñar una función de forma analítica para lograr una separabilidad lineal en el espacio de características de los datos analizados [1].

Para lograr una convergencia adecuada en una ANN es necesario implementar el algoritmo de “backpropagation”, que ayuda a ajustar los pesos del modelo de acuerdo con el cambio en la función de pérdida para minimizarla. Este algoritmo se basa en el gradiente de la función de pérdida con respecto a todos los nodos en el grafo utilizando la factorización del gradiente para moverse desde el nodo de la pérdida, hasta cada uno de los nodos de entrada. La expresión del gradiente es factorizada para simplificarla y crear un algoritmo que puede ser aplicado a cualquier grafo para calcular de forma sencilla la expresión $\frac{\delta \ell}{\delta w_i^l}$ que determina el gradiente de la función de pérdida conectada hasta los nodos iniciales [1]

Debido a que en una imagen existe alta correlación entre pixeles aledaños, se pueden reducir el número de parámetros requeridos para obtener un mapa de características representativas al agrupar neuronas por bloques y asignándoles pesos compartidos. Cada bloque (l) de $p \times q$ neuronas se conecta a todos los pixeles de la imagen y extrae información en regiones de $m \times n$ pixeles llamados campos receptivos, en vez de extraer la información de la totalidad de pixeles. Esto permite reducir drásticamente el número de parámetros requeridos para ajustar una ANN que trabaja con imágenes [1].

Los bloques de neuronas pueden ser representado matemáticamente como $f_{p,q}^l = (\mathcal{G})(\sum_{i=0}^m \sum_{j=0}^n im(p+i, q+j)w_{i,j}^l)$ [1]. A esta operación se le conoce como convolución y se denota por un asterisco (*) en la literatura. Si $X \in \mathbb{R}^{H \times W \times C}$ es una imagen con $H \times W$ pixeles (height y width) y C canales (como colores RGB) y $f \in \mathbb{R}^{h \times w \times C}$ es un filtro o bloque

de neuronas con dimensiones $h \times w$ pixeles y C canales, entonces $X * f \in \mathbb{R}^{H-h+1 \times W-w+1 \times 1}$, es decir, la matriz resultante tiene dimensiones $(H - h + 1, W - w + 1, 1)$.

Otro aspecto importante de las redes convolucionales es el “stride”, que es básicamente un modificador de la forma en que se construyen los campos receptivos sobre la matriz de datos, siendo un stride igual a uno un movimiento pixel por pixel, un stride igual a dos un movimiento que se salta un pixel, etcétera. Descrito formalmente, el stride, denotado por s , sería el incremento en m y n en la operación de convolución $(X * f)(m, n) = \sum_{i=0}^{P-1} \sum_{j=0}^{Q-1} X(m+i, n+j)f(i, j)$ con $m = 0s, 1s, 2s, \dots, H - 1$, $n = 0s, 1s, 2s, \dots, W - 1$. Al aplicar un stride las dimensiones resultantes serán $(\frac{W-P}{s} + 1 \times \frac{H-Q}{s} + 1)$, por lo que es importante elegir un stride y tamaño de filtro adecuados tal que las dimensiones resultantes sean un número entero, de lo contrario los datos de entrada X serán recortados para cumplir con lo anterior [1].

Por otro lado, existe otra operación llamada “pooling” utilizada en las redes convolucionales que ayuda a agrupar pixeles después de aplicar una convolución para extraer información y reducir la dimensionalidad de los mapas de características para facilitar el aprendizaje en el modelo. Este método consiste en tomar elementos de la matriz de datos moviéndose a través de ellos con un stride y realizando operaciones con los datos dentro del campo que se está procesando, como obteniendo el promedio de los valores o manteniendo únicamente el valor máximo [1]. Por ejemplo, realizar un “Max pooling” con stride 2 y tamaño 2 en el vector $x = [1, 10, 8, 2, 3, 6, 7, 0, 5, 4, 9, 2]$ resultaría en el vector $x_{\max_pool} = [\max(1, 10), \max(8, 2), \max(3, 6), \max(7, 0), \max(5, 4), \max(9, 2)] = [10, 8, 6, 7, 5, 9]$.

Finalmente, un elemento crucial para cualquier modelo de aprendizaje máquina es la función de pérdida o función de costo, que indica el error existente entre aquello que el modelo predice y los datos con los que está siendo entrenado. Cada función debe ser diseñada o elegida de acuerdo con la tarea que se busca realizar, ya que dirigirá la forma en que el modelo se adapta para cumplir con aquello que se le alimenta como “verdad” [1]. Para

problemas de clasificación, por ejemplo, se utiliza frecuentemente la entropía cruzada binaria o categórica, también llamadas “binary cross-entropy” y “categorical cross-entropy” en inglés, que es una forma de log-similitud negativa. Para tareas de localización se utilizan la intersección sobre la unión (IoU) o la intersección sobre la unión generalizada (GIoU), que compara áreas de las cajas delimitadoras o “bounding-boxes” que indican donde se encuentra un objeto. Finalmente, para tareas de conteo es común utilizar funciones como el error cuadrático medio (MSE), error absoluto medio (MAE) o el error de raíz cuadrada media (RMSE) [2].

Convenientemente, existen librerías en el lenguaje de programación Python, como TensorFlow [3] y Keras [4], que ayudan a diseñar y entrenar ANN utilizando un lenguaje simple. De esta manera pueden realizarse todas las operaciones previamente descritas para realizar tareas de aprendizaje profundo de forma sencilla y accesible.

El uso de las redes neuronales convolucionales en la biotecnología permite la automatización de procesos repetitivos, que requieren periodos largos de concentración y que son propensos al error humano, tales como el conteo de colonias en placa Petri. Con la incorporación de esta tecnología, los centros de investigación, educativos y la industria puede beneficiarse de reducir tiempos de proceso y enfocar el capital humano a tareas menos repetitivas.

1.2 Caracterización de la organización

El enfoque del Centro de Investigación y Asistencia en Tecnología y Diseño del Estado de Jalisco (CIATEJ) es la investigación, desarrollo tecnológica e innovación (I+D+i) para resolver problemáticas centradas en el desarrollo sostenible de la sociedad. Para esto, el centro desarrolla líneas de investigación que atienden necesidades del sector agropecuario, agua, medio ambiente, alimentos, bebidas, salud humana y animal, así como para mejorar el desarrollo sustentable mediante la mejora de bioprocesos. Las anteriores bajo las unidades de biotecnología vegetal, tecnología ambiental, tecnología alimentaria, biotecnología médica y farmacéutica y, finalmente, biotecnología industrial.

Dentro de la unidad de Biotecnología Industrial dirigida por la Dra. Amaya Delgado, se encuentran cuatro ramas de investigación enfocadas en la “bioelectrónica”, “biología sintética”, “diseño, optimización y aplicación de biocatalizadores” y “diseño y optimización de procesos fermentativos”. El presente proyecto se coloca bajo el umbral de “diseño y optimización de procesos fermentativos” por aplicarse al conteo de colonias, un procedimiento frecuentemente utilizado para evaluación de fermentaciones, a cargo del Dr. Melchor Arellano Plaza que con siete investigadores más conforman dicha rama.

1.3 Identificación de la(s) problemática(s)

Durante las investigaciones de carácter biotecnológico, el aislamiento, crecimiento y evaluación de la reproducción de microorganismos en medios sólidos es una herramienta útil que permite realizar conteo de colonias en placa, con el objetivo de realizar la caracterización de los cultivos realizados. El crecimiento microbiano requiere continuamente determinar la concentración de unidades formadoras de colonias, así como, discernir entre las morfologías de colonias [5], lo cual asiste al proceso de identificación de microorganismos, generar estudios de inhibición por acción de agentes bacteriostáticos, biocidas, antibióticos o agentes de biocontrol, así como para aislar microorganismos utilizando medios específicos.

Debido a la naturaleza del presente proyecto, son de especial interés el conteo de colonias y la determinación de la morfología de estas para su identificación, ya que forman parte de las actividades necesarias para el desarrollo de la investigación global que se plantea. El proceso de conteo de colonias es utilizado en prácticamente todos los laboratorios de microbiología del mundo, ya que es barato y sencillo, sin embargo, es propenso al error humano en dos puntos cruciales: el conteo y la transcripción de los datos a medios electrónicos [6]. Adicionalmente, el método de conteo es tardado y requiere absoluta concentración durante la realización de esta labor para evitar errores. Aún el biotecnólogo o microbiólogo más experimentado puede, después de horas de trabajo contando placas, caer en el error de omitir algunas colonias, perder la cuenta, marcar dos veces una colonia o caer en la discalculia y, por ejemplo, escribir “238” en lugar de “283”.

Por otro lado, existen dificultades intrínsecas en el proceso que pueden provocar distintos resultados cuando una placa es contada por distintos individuos, tales como falta de homogeneidad en el diámetro de las colonias, diferentes densidades en ciertas regiones de la placa, formación de películas continuas, crecimiento de colonias al borde de las placas, o empalme de dos o más colonias [7], [8]. Algunas de estas dificultades se presentan en la figura 4.

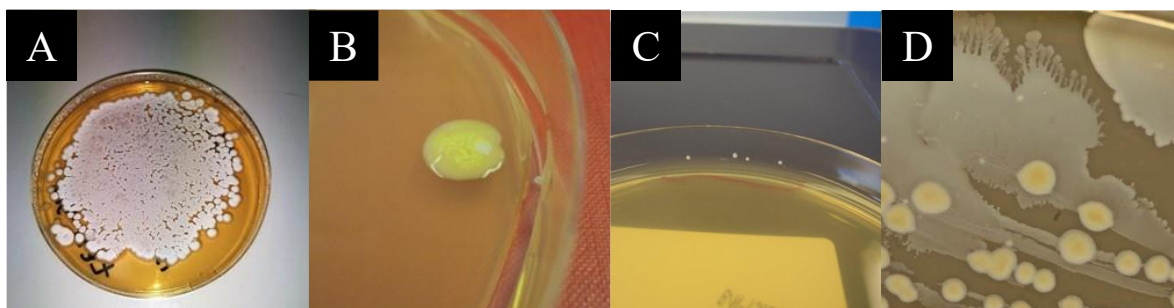


Figura 4. Dificultades encontradas durante el conteo de colonias en placa. A: Densidad de colonias más alta de la requerida. B: Fusión de colonias. C: Colonias creciendo al borde de la placa. D: formación de película bacteriana [6].

Algunas de las dificultades de cuantificación, como las relacionadas con la concentración y continuidad del recuento de colonias, pueden ser sobrellevadas con equipos para el conteo manual como los presentados en la figura 5. Sin embargo, con ellos no se resuelve la problemática del largo tiempo requerido para contar y sólo se mitigan los errores humanos hasta cierto grado, no se eliminan por completo. Además, los dispositivos más sofisticados pueden presentar costos elevados, como el equipo FE-500 con un precio de lista de 36,638.00 MXN + IVA, lo que pudiera ser una barrera para la adquisición de estos equipos en centros educativos o de investigación con presupuestos ajustados. Los equipos portátiles, por otro lado, son más accesibles en precio, como el equipo Bel-Art con precio de lista de 7,875.00 MXN, sin embargo, estos ofrecen menos ventajas que sus contrapartes de escritorio.

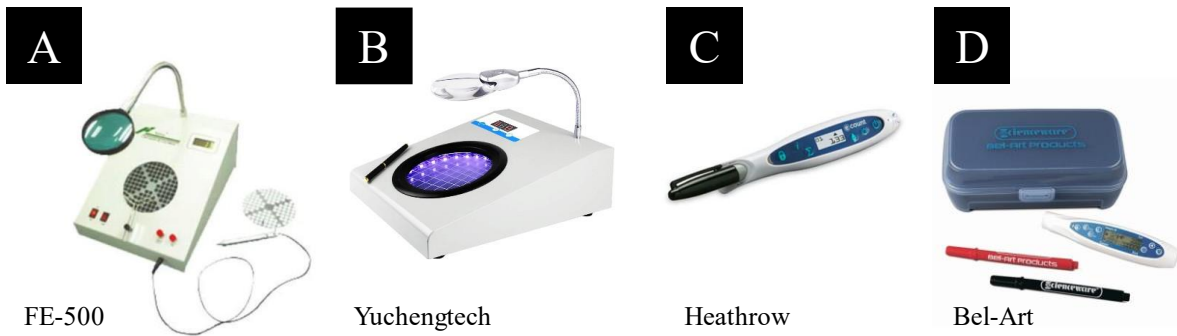


Figura 5. Contadores de colonias manuales. A y B equipos de escritorio con iluminación, amplificación, cuadrícula y alarma auditiva. C y D: Equipos portátiles.

Con ayuda de los recientes desarrollos tecnológicos en visión computacional es posible contar, localizar e identificar características morfológicas de colonias basándose en imágenes. A continuación, se presentará una breve introducción de qué es la visión computacional y cómo puede ser utilizada para la tarea automatizada de conteo, localización e identificación de colonias.

En la sociedad moderna convivimos con dispositivos que ejecutan modelos de visión computacional de forma cotidiana, incluso sin percatarnos de ello. Un ejemplo de esto es el autoenfoco en las cámaras, reconocimiento facial para vigilancia, seguridad, reconocimiento de caracteres o incluso para “etiquetar” a nuestros amigos o conocidos en redes sociales [9], algunos ejemplos se muestran en la figura 6. A pesar de esta amplia exposición a dicha tecnología, la comprensión de su funcionamiento esquivada a las masas debido a la cantidad acumulativa de conocimientos necesarios en las áreas de matemáticas y programación. Estas dos áreas alimentan los campos de la física, específicamente radiometría, óptica y diseño de sensores, así como de gráficos computacionales, sin los cuales la visión computacional sería imposible [9]. Sin embargo, ¿Qué es entonces la visión computacional?

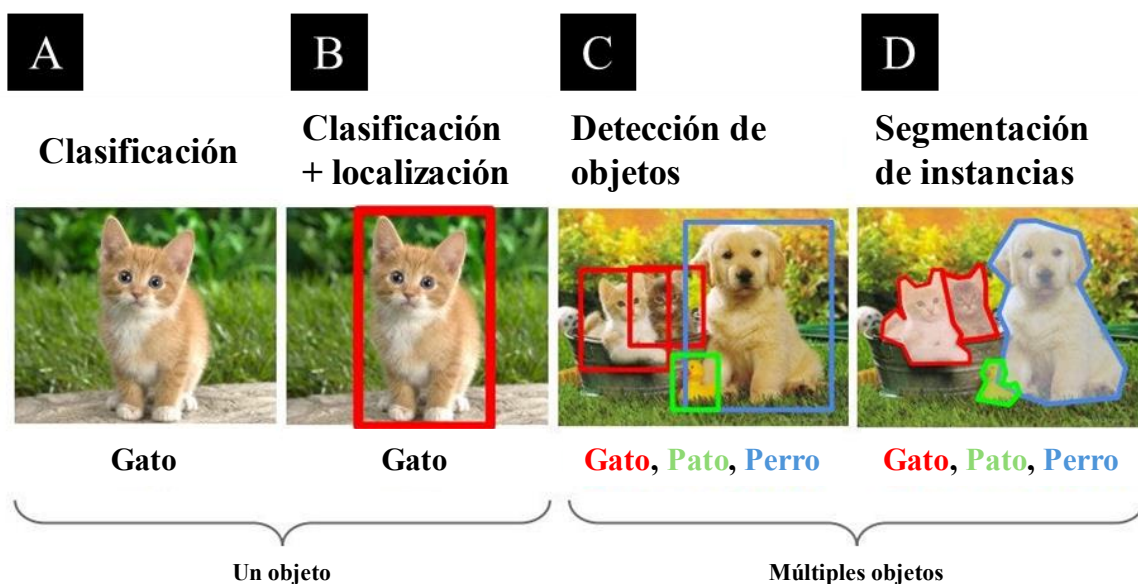


Figura 6. Ejemplos de visión computacional. A: Clasificación. B: Clasificación y localización. C: Detección de objetos. D: Segmentación de objetos [9].

Dicho brevemente, la visión computacional es el esfuerzo humano de replicar su capacidad de percibir formas, profundidad, colores y texturas utilizando cámaras como el equivalente a los ojos y computadoras como cerebros, todo esto a través de modelos matemáticos ejecutados a manera de un lenguaje de programación y como tal, es un problema de ingeniería inversa [10]. Este esfuerzo frecuentemente es abordado mediante métodos estadísticos o probabilísticos para resolver los problemas asociados a la creación de un modelo de visión computacional, por lo que existen métricas como el error medio cuadrático (MSE por sus siglas en inglés), la raíz del error cuadrático medio (RSME), Intersección sobre la Unión (IOU) o Intersección Generalizada Sobre la Unión (GIUO), por citar algunas, que ayudan a evaluar el ajuste del modelo con respecto a la realidad. Para esto último, es frecuente utilizar lenguajes de programación como Python[®], que cuenta con librerías gratuitas y de fácil utilización especializadas para resolver tareas de inteligencia artificial, aprendizaje máquina, aprendizaje profundo, visión computacional, procesamiento de audio, procesamiento de lenguaje natural, entre otros campos relacionadas. Las tareas que resuelve la visión computacional más usuales se listan a continuación [11].

1. Clasificación de objetos. Responde a la pregunta ¿Qué categoría(s) de objeto(s) existe en una imagen?

2. Identificación de objetos. ¿Qué tipo de objeto se encuentra en la imagen?
3. Verificación de objetos. ¿Existe el objeto “X” en la imagen?
4. Detección o localización de objetos. ¿En dónde se encuentran los objetos “X” en la imagen?
5. Segmentación de objetos. ¿Qué píxeles o regiones de la imagen corresponden a un objeto “X” y cuáles a “Y”?
6. Reconocimiento de objetos. Es una combinación de identificación, clasificación y detección. ¿Qué objeto está presente en la imagen y dónde?

Una relación de algunas de las tareas previamente mencionadas, así como otras más especializadas, se presentan en la figura 7. La región de interés para el presente Proyecto de Aplicación Profesional (PAP) abarca hasta el punto seis, que corresponde al reconocimiento de objetos.

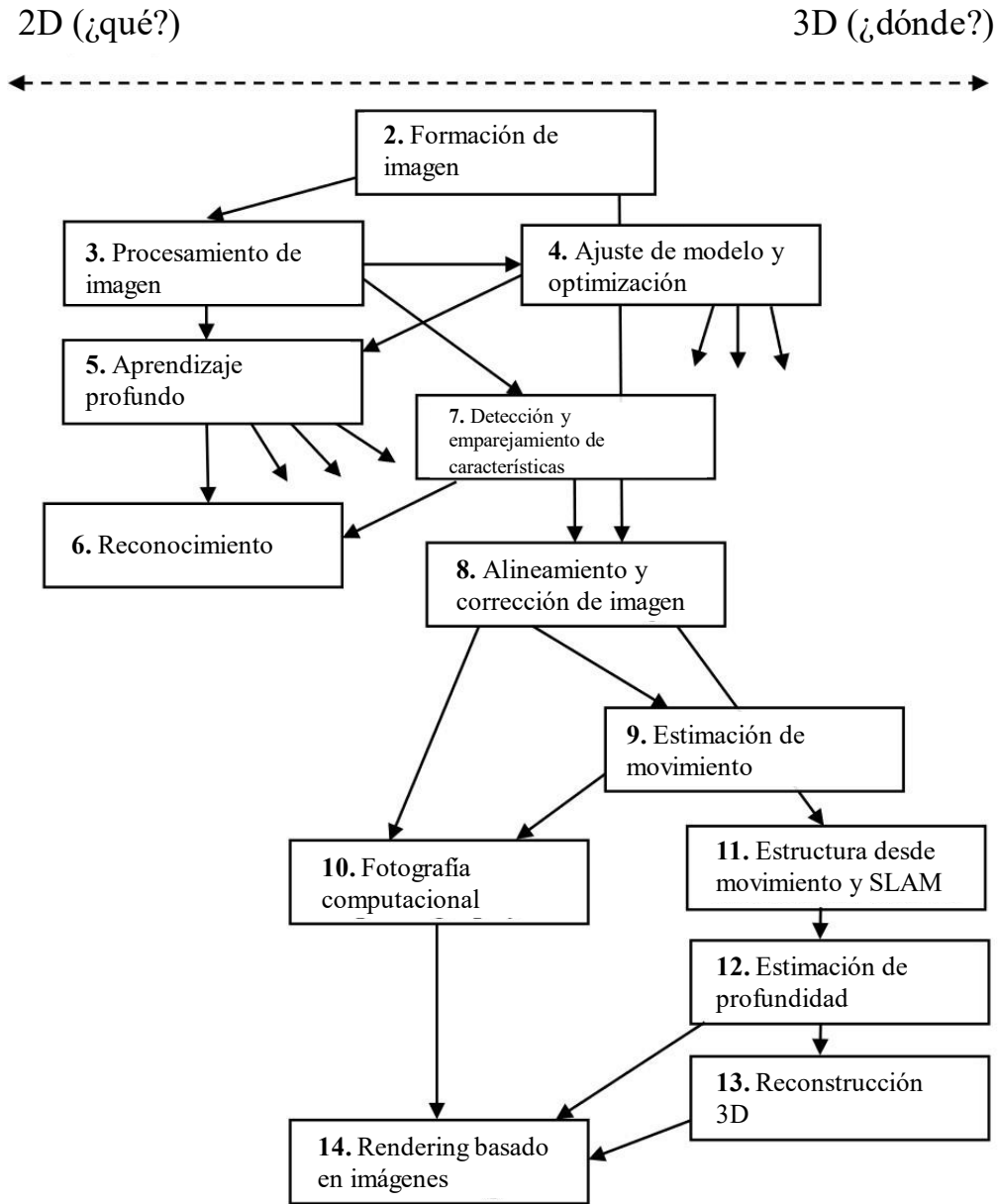


Figura 7. Taxonomía propuesta por [10] de las tareas de visión computacional cubiertas en el contexto de la publicación.

De lo anterior resulta que los algoritmos de visión computacional puedan aportar al campo de la biotecnología para resolver problemas como el abordado en este trabajo: conteo, localización y clasificación de colonias en cajas Petri. En definitiva, el mundo se ha percatado de esta posibilidad, pues ya existen en el mercado dispositivos que permiten realizar el conteo automático de colonias (figura 8), sin embargo, estos equipos aún son costosos, con precios

desde los 97,800.00 MXN para un equipo sencillo como el IncuCount[®] de Revolutionary Science[®] hasta los 2,300,000.00 MXN para un modelo sofisticado y de alta tecnología como el AutoCOL[®] de Synbiosis[®].

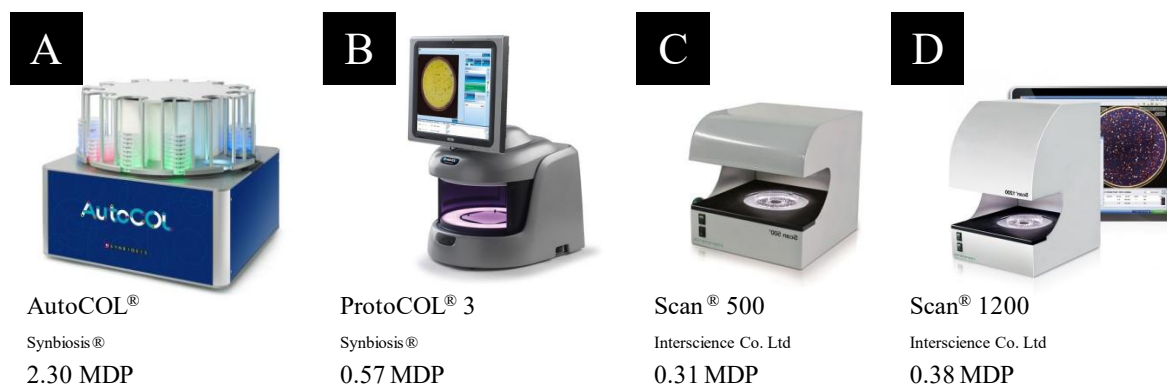


Figura 8. Contadores de colonias automáticos en el mercado. A: AutoCOL[®] de la marca Synbiosis[®], costo de 2.30 MDP. B: ProtoCOL[®], Synbiosis[®], 0.57 MDP. C: Scan[®] 500 de la empresa Interscience Co. Ltd., 0.31 MDP D: Scan[®] 1200, Interscience Co. Ltd., 0.38 MDP.

Los altos costos de los contadores representan una enorme barrera que, si bien tiene su justificación comercial, puede bloquear el acceso del público general a esta tecnología. Algunos resultados de la “privatización de la ciencia” son la reducción de la práctica científica a un instrumento de marketing, motivando a las empresas privadas, y por consecuencia a sus inversionistas, a invertir en la generación y publicación de artículos científicos que sustenten la tecnología de la cual obtienen utilidades [12]. Lo anterior se refuerza en la problemática de la reducción en la calidad experimental observada en estudios patrocinados, debido a causas como un diseño experimental ineficiente [13], que fallan en separar los procesos de aleatorización de los participantes reclutados.

Aunado a esto, el 18% de los 234 estudios que se consideraron aptos para analizar dentro de clústeres se consideraron como inadecuados [13]. Para esto, se tomó como criterio para categorización el efecto de la revista científica en que están publicados, con respecto a la calidad estadística. Dicha calidad se evaluando, tomando como criterio principal una política de “carta sellada”, en la que los estudios son realizados por terceros independientes del patrocinador. Las revistas científicas analizadas fueron BMJ, JAMA, “The Lancet” y “The

New England Journal of Medicine”. Una gráfica del estudio anterior se presenta en la figura 9, donde se observa que, en las funciones de densidad de probabilidad de cada clúster se presentan valores P (“P values” en inglés) que permiten discriminar entre cada conjunto, lo anterior de acuerdo con la distribución de eventos dentro de ellos. Además, el 26% de los estudios analizados se consideraron como inciertos debido a su método para el “enmascaramiento” de los participantes para que no afecten en la aleatorización de los experimentos realizados. Esto suma un preocupante 44%, reforzando el efecto de privatización de la ciencia mencionado en [12]. Cabe destacar que en [13] se excluyeron 97 estudios del juego de datos original a causa de ser imposible extraer obtener un valor P de ellos. Lo anterior no significa que toda investigación privada sea realizada de forma inadecuada, sino que existe motivación comercial para manipular la rigurosidad de los ensayos en los que se basan los resultados, mismos que posteriormente se vuelven una herramienta de marketing para cerrar ventas y abarcar los segmentos objetivo de la o las empresas involucradas.

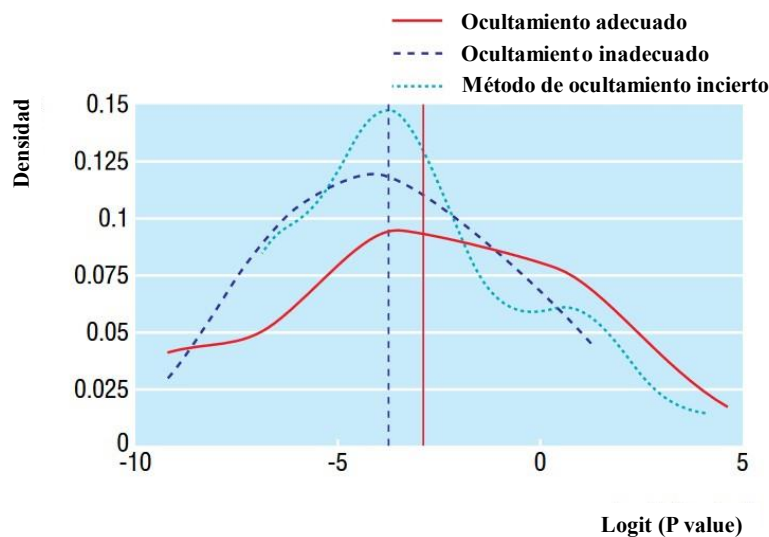


Figura 9. Funciones de densidad de probabilidad de los clústeres de datos utilizados para analizar la aleatorización del diseño experimental en estudios realizados en las revistas médicas BMJ, JAMA, “The Lancet” y “The New England Journal of Medicine” [13].

Es por la suma de los factores ya expuestos que se presenta como una necesidad poner a disposición del público general y de la comunidad científica el conocimiento de forma

gratuita y sin sesgos comerciales. El presente trabajo es un esfuerzo por comenzar a realizar esta labor al crear un proyecto comunitario a través de la transmisión integral de la información generada, tal como juegos de datos, anotaciones y el código utilizado para crear los modelos computacionales que permiten, o pretenden facilitar, el uso de tecnologías emergentes de forma pública. Esto último para fines educativos, de divulgación y, por supuesto, académicos.

1.4. Planeación de alternativa(s)

Para la creación del modelo de detección, se propone inicialmente entrenar un modelo utilizando un juego de datos sintético que emula imágenes de bacterias observadas por microscopía de fluorescencia compuesto por 200 imágenes con resolución de 256 x 256 y tres canales (RGB) generado por [14]. Posteriormente, se propone utilizar el juego de datos AGAR de 18,000 imágenes de colonias en cultivos sólidos en cajas Petri generado por [15], que incluye anotaciones de identificación de la imagen (ID), especies en toda la imagen, así como en la posición de cada colonia y, finalmente, el recuento total de colonias. En dicho juego de datos se utilizan las especies reguladas depositadas en el American Type Culture Collection (ATCC): *Streptomyces aureus* subsp. *aureus* ATCC 6548, *Bacillus subtilis* subsp. *spizizenii* ATCC 6633, *Pseudomonas aeruginosa* ATCC 9027, *Escherichia coli* ATCC 8739 y *Candida albicans* ATCC 10231 de forma aislada y en “cocultivo”. En [15] no se explica con claridad la metodología de las diluciones realizadas, sólo se menciona que se efectuaron series de diluciones 1:10 de cultivos tomando 100 µl como inóculo para placas de Agar Soya Tripticasa (TSA) en cinco réplicas. Los cultivos se incubaron a 37°C durante 18 a 24 horas.

La resolución de las imágenes en el juego de datos AGAR se considera más que suficiente para tareas de aprendizaje profundo, pues presenta una categoría de “baja-resolución” de 2,048 x 2,048 píxeles y una de “alta-resolución” de 4,000 x 6,000 píxeles. Como referencia, existen modelos viables basados en DenseNet-161 y Resnet-152 que utilizan imágenes con resoluciones de 512 x 512 píxeles con un coeficiente de correlación de Matthews (MCC) de 0.9 [16], donde un valor cercano a uno minimiza los errores en la matriz de confusión,

indicando una predicción de clases adecuada de un modelo con respecto al juego de datos de validación.

Además, se buscará incorporar, si el tiempo y hardware con el que se cuenta lo permite, incluir el juego de datos desarrollado por [17], que contiene 369 imágenes de 24 especies de bacterias, así como 56,865 anotaciones en formato JSON. El juego de datos incluye coordenadas espaciales (x , y) para la construcción del cuadro delimitador de la colonia en la imagen, mejor conocido como “bounding box” en inglés, así como dimensiones de *altura* (H) y *anchura* (W) de la imagen, para la generación de mapas de densidad. En este juego de datos se utilizan las especies *Actinobacillus equuli*, *Actinobacillus pleuropneumoniae*, *Aeromonas hydrophila*, *Bacillus cereus*, *Bibersteinia trehalosi*, *Bordetella bronchiseptica*, *Brucella ovis*, *Clostridium perfringens*, *Corynebacterium pseudotuberculosis*, *Erysipelothrix rhusiopathiae*, *Escherichia coli*, *Glaesserella parasuis*, *Klebsiella pneumoniae*, *Listeria monocytogenes*, *Paenibacillus larvae*, *Pasteurella multocida*, *Proteus mirabilis*, *Pseudomonas aeruginosa*, *Rhodococcus equi*, *Salmonella enterica*, *Staphylococcus aureus*, *Staphylococcus hyicus*, *Streptococcus agalactiae*, y *Trueperella pyogenes* cultivadas en agar sangre, agar chocolate, agar MacConkey o agar nutritivo. La cepa *Clostridium perfringens* fue la única cultivada de forma anaeróbica, el resto de forma aeróbica.

Las condiciones con las que se generó el juego de datos en [17] se describen a continuación. Los cultivos primero fueron descongelados para después ser transferidos en los medios de cultivo “apropiados”, según menciona [17]. En el segundo día, se tomaron colonias típicas para ser sembradas y cultivadas. Seguido esto, en el tercer día se utilizaron colonias para inocular caldo de soya triptona (TSB) utilizando un hisopo de algodón estéril y se incubó el caldo a 37°C durante 24 horas. El resultado del trabajo del tercer día fue utilizado para preparar diluciones seriales 1:100, tomando 0.1 mL del cultivo y transfiriéndolo a 9.9 mL de solución salina estéril, generando una dilución 10^{-2} . Se repitió el procedimiento para lograr una dilución 10^{-6} tomando muestras de 0.5 mL de cultivo y transfiriéndolas a 4.5 mL de suero salino fisiológico estéril. Cabe destacar que este juego de datos aparentemente presenta un desbalance de clases (figura 10), donde las especies *Staphylococcus aureus*, *Bordetella bronchiseptica*, *Staphylococcus hyicus*, *Rhodococcus equi*, *Streptococcus agalactiae* y

Listeria monocytogenes son las más representadas con 48, 35, 24, 24, 23 y 18 imágenes, respectivamente, cuando las demás especies son representadas por un rango entre 3 y 16 imágenes. Así mismo, las especies ya mencionadas representan el mayor número de colonias contadas a través de su clase, efecto esperado por el desbalance de clases.

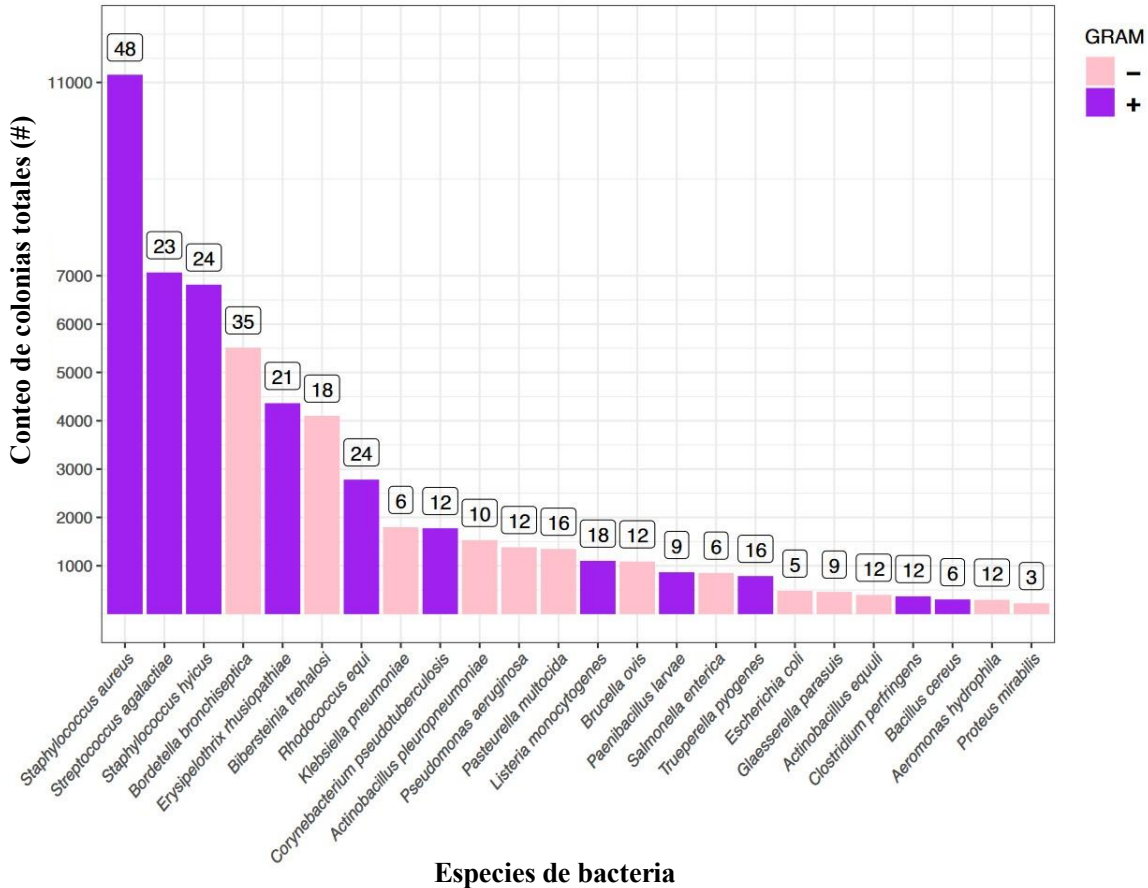


Figura 10. Clases representadas por el juego de datos en [12]. Columnas: Total de colonias contadas. Recuadro sobre la columna: total de imágenes que representan a cada especie.

Una vez aclarado el origen de los juegos de datos de entrenamiento para el modelo, se propone utilizar, como se realizó en [18], cada uno como un conjunto $\mathcal{D}_{entrenamiento} = \{(\mathcal{X}_1, \mathcal{S}_1, y_1)\}, \dots, \{(\mathcal{X}_n, \mathcal{S}_n, y_n)\}$, donde \mathcal{X}_i denota la imagen de entrada, además, $\mathcal{X}_i \in \mathbb{R}^{H \times W \times 3}$ debido a que es una imagen de dimensiones (H, W) con tres canales de color (R, G, B). Por otro lado, $\mathcal{S}_i = \{b_i\}^K$ denota las coordenadas del “bounding box” que es un arreglo con cuatro valores, es decir, $b_i^K \in \mathbb{R}^4$, representando K los ejemplos utilizados para el entrenamiento, donde $K \in \{0, 1, 2, 3 \dots\}$. Finalmente, $y_i \in \mathbb{R}^{H \times W \times 1}$ representa un mapa de

densidades espaciales binario de dimensiones (H, W) donde un valor de uno representa el centro de una colonia en $\mathbb{R}^{H \times W \times 1}$, es decir en dos dimensiones, y un cero indica ausencia de objetos en dicha posición de la matriz.

Inicialmente se buscará validar el uso de mapas de densidad desarrollados de forma sintética utilizando funciones de aleatorización para generar puntos en un espacio bidimensional en coordenadas (x, y) que posteriormente son convertidos a un mapa de densidades $\mathcal{D} \in \mathbb{R}^{256 \times 256 \times 1}$ utilizando un filtro Gaussiano $y_i = G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, utilizando desviaciones estándar $(\sigma = \sigma_x = \sigma_y = 1.5)$ en Python, para comparar el método con el utilizado por [14]. Estos mapas de densidad fungirán como datos secundarios de entrada a imágenes utilizando un modelo convolucional con una arquitectura que será desarrollada de forma libre, pero tomando como entradas una imagen con resolución 256×256 píxeles $\mathcal{X}_i \in \mathbb{R}^{256 \times 256 \times 3}$, un mapa de densidades bidimensional generado a través de un filtro Gaussiano (y_i) y, de forma opcional, un “label” $l \in \mathbb{R}$ que representa el total de colonias obtenido mediante la suma de elementos con valor 1 en y_i , es decir, $l = \sum_{a=0}^H \sum_{b=0}^W y_i(x_a, y_b)$. Esta última entrada es opcional, ya que puede calcularse en dentro de la función de pérdida del modelo $\mathcal{L} = MSE(y_{pred}, y_{ground}) = \frac{1}{K} \sum_{i=0}^K (x_{pred,i}, y_{ground,i})^2$.

Posteriormente, se validará el modelo construido a través de Transformers con la arquitectura presentada en la figura 11, donde se incluyen datos de entrada “Zero-shot”, es decir, sin segmentos de la imagen, y ejemplares “Few-shot”, que son algunos segmentos de la imagen de entrada, así como un token que el modelo puede aprender. Estos elementos pasan por un módulo donde los mapas de características interactúan y después son decodificados para retornar un mapa de densidad que permite realizar el conteo.

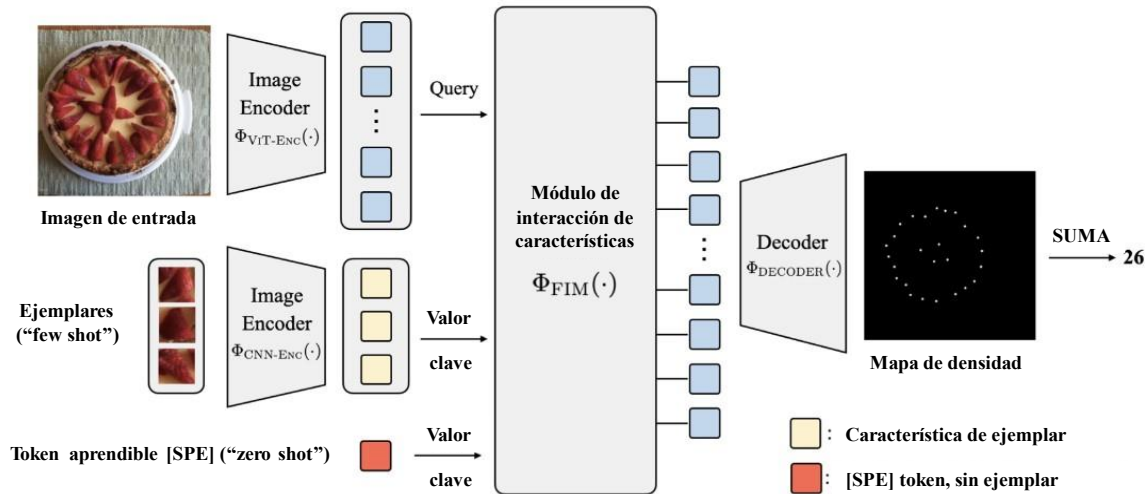


Figura 11. Arquitectura para el modelo basado en aprendizaje profundo (Deep Learning) propuesta en [13].

En cuanto a las anotaciones de las imágenes, se buscará transformar las coordenadas proporcionadas para la construcción de “bounding boxes” a coordenadas (x,y) que indiquen el centro de cada colonia. Esto se realizará encontrando la distancia media entre los bordes del bounding box presentado en cada juego de datos.

Por otra parte, el diseño del dispositivo que permitirá el uso del modelo se realizará utilizando el software libre de diseño OpenScad. Este programa permite la construcción de modelos tridimensionales utilizando figuras primitivas como cilindros, cubos y esferas y presenta la ventaja de exportar los archivos en formato de estereolitografía (STL), que pueden ser utilizados para impresión 3D. Para llevar a cabo la materialización del diseño, se utilizará la impresora K1[®] de la empresa Creality, utilizando filamentos de PLA a una temperatura de extrusión de 220°C, con un offset en Z para el G-Code de 0.02 mm, una velocidad de impresión de 300 mm/s, una temperatura de placa de 60°C y cualquier otro parámetro fuera de los mencionados en su valor por defecto en el software de corte Creality Print[®]. Cabe mencionar que, con base en experiencias previas de impresión con este equipo y el filamento por utilizar, se imprimirán las piezas retirando la cubierta superior de la impresora 3D para evitar que el filamento se sobrecaliente y escurra, formando el llamado “stringing” entre los saltos de la boquilla de extrusión.

En la tabla 1 se muestran las actividades y su organización por semanas para la realización del presente PAP. En esta tabla se muestran los recursos utilizados para cada actividad, abarcando desde recursos humanos, software y hardware.

Tabla 1. Cronograma de actividades necesarias para la realización del presente PAP.

Nombre de la actividad	Recursos	Tiempo (días)	Semana del semestre															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Reunión para planeación de objetivos	AM	1	█															
Asesorías PAP	AV AP	1	█					█				█						
Revisión de avances con investigador	AM	5			█			█			█			█				█
Redacción de entregas RPAP	TP	12			█				█			█					█	█
Revisión bibliográfica sobre redes neuronales convolucionales	TP CP	15		█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Revisión de arquitecturas para redes convoluciones de conteo	TP PY KR TF	12				█	█	█	█	█	█	█	█	█	█	█	█	█
Diseño de dispositivo en OpenSCAD	TP OS	12				█	█	█	█	█	█	█	█	█	█	█	█	█
Impresión 3D de dispositivo	TP CK1	15								█	█	█	█	█	█	█	█	█
Obtención y curado de juegos de datos	TP CP	15					█	█	█	█	█	█	█	█	█	█	█	█
Generación de funciones para obtención de datos	TP PY	3						█	█	█								
Generación de funciones generadoras para entrenamiento	TP PY KR TF	3							█	█	█							
Programación de cámara OV7670 con arduino	TP ARD	5									█	█	█	█	█			
Programación de cámara Brobotix	TP	1										█	█	█				
Pruebas con arquitecturas de redes convolucionales para conteo	TP PY KR TF	10										█	█	█	█	█	█	█
Ensamblado de dispositivo 3D	TP	2															█	█

función para obtener los tensores que ingresarán al modelo, para realizar el “split” o división de datos en juegos de entrenamiento, validación y prueba, una función generadora para el modelo que carga lotes de tamaño personalizable para facilitar el entrenamiento del modelo, así como “callbacks” de Keras que permiten guardar de forma automática los pesos del modelo durante su entrenamiento para evitar pérdidas del progreso por interrupciones inesperadas. Adicionalmente, se implementó una función para guardar el modelo y sus pesos de forma manual. El código de dihas implementaciones puede encontrarse en el repositorio de Github “Panoptic” con dirección web <https://github.com/MunguiaH/Panoptic>.

El conjunto sintético de datos para validar el uso de mapas de densidad presenta imágenes con puntos distribuidos de forma aleatoria en el espacio, representados matemáticamente como una matriz de ceros y unos $\mathcal{M} \in \mathbb{R}^{256 \times 256 \times 1}$. Estas imágenes fueron utilizadas para crear un mapa de densidad de probabilidad utilizando un filtro Gaussiano unos $\mathcal{D} \in \mathbb{R}^{256 \times 256 \times 1}$, tal como se observa en la figura 12. Estos mapas de densidad sirvieron como datos de verdad, o “ground truths” para entrenar el modelo para conteo.

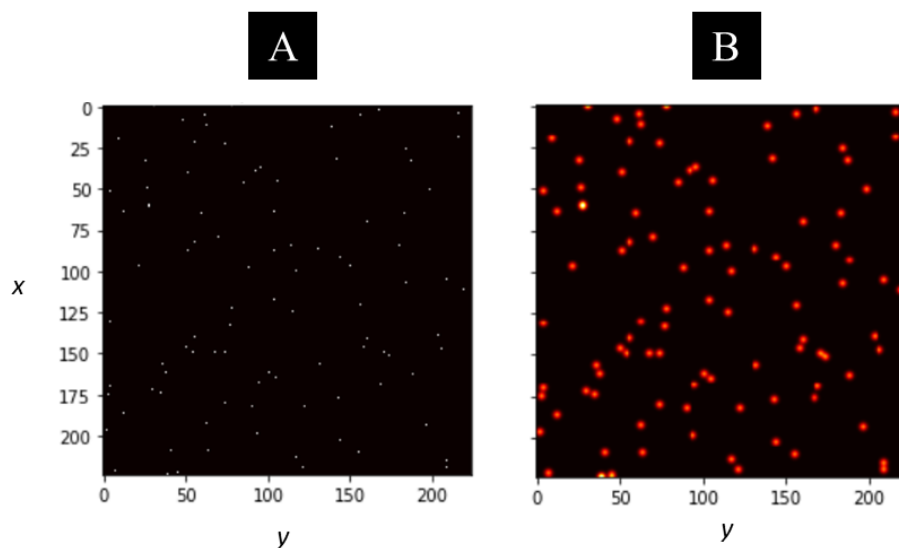


Figura 12. Gráficos de los datos de entrada para la validación del uso de mapas de densidad. A: Imagen sintética. B: Mapa de densidad de la imagen sintética.

Después de validar la generación de mapas de densidad a partir de puntos aleatorios, se utilizó el juego de datos de microscopía fluorescente generado por [14] para entrenar un modelo

simple de conteo basado en mapas de densidad utilizando la arquitectura del modelo presentad en la figura 13.

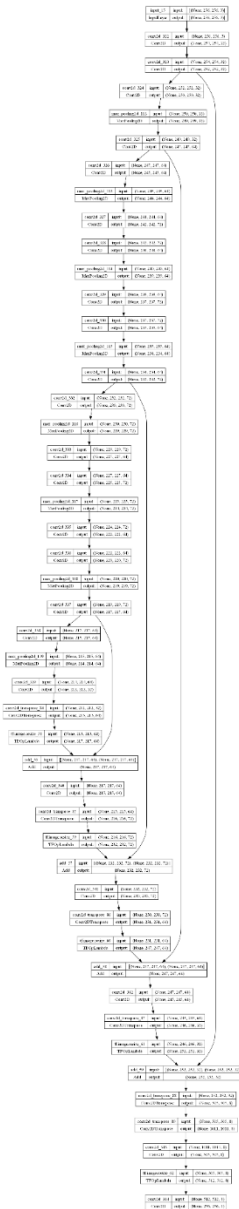


Figura 13. Arquitectura del modelo propuesto para el conteo basado en predicción de mapas de densidad. Entrada: $X \in \mathbb{R}^{256 \times 256 \times 3}$. Salida: $Z \in \mathbb{R}^{256 \times 256 \times 1}$. Parámetros: 827,201. Modelo completo disponible en <https://github.com/MunguiaH/Panoptic/blob/main/Arquitectura.png>

El modelo con la arquitectura inicial presenta 827,201 parámetros y logró obtener un 94.7% de exactitud, o “accuracy”, en la tarea de conteo utilizando el juego de datos de células fluorescentes, presentando además una minimización de la pérdida eficiente como se observa en la figura 14.

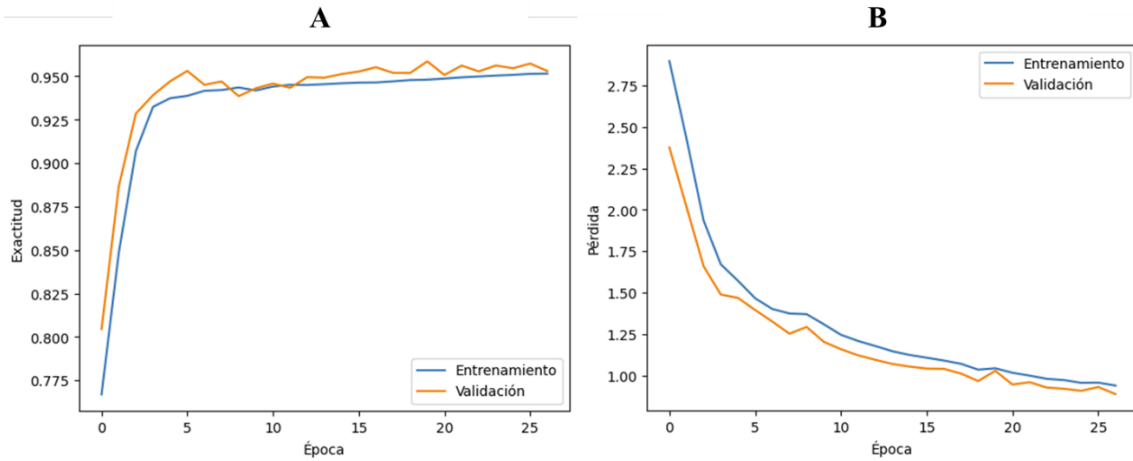


Figura 14. Funciones generadas durante el entrenamiento y validación del modelo. A: Exactitud del modelo. B: Pérdida del modelo.

En la figura 15 se presenta un ejemplo de las predicciones que el modelo genera de los mapas de densidad que representan la posición y número de colonias. En el repositorio de Github se presentan diferentes fotografías de los mapas de densidad y resultados de conteo a manera de evidencia de funcionamiento del modelo.

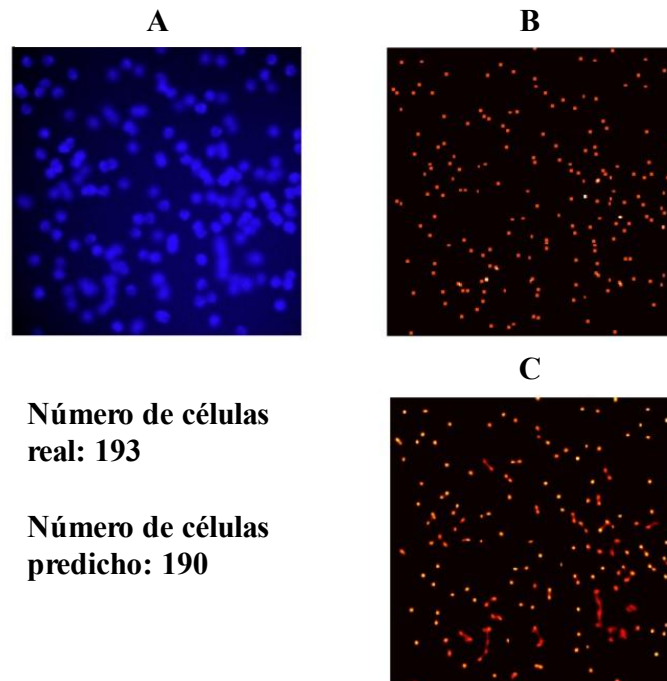


Figura 15. Mapas de densidad. A: imagen de microscopía. B: mapa de densidades real. C: mapa de densidades predicho por el modelo.

Al comparar el número de células predicho por el modelo inicial contra las células reales en cada ejemplo, se puede observar una tendencia hacia la linealidad. Esto indica que el modelo está ajustándose de forma cercana a los datos de conteo al utilizar datos fuera de los que el modelo encontró durante el entrenamiento, tal como se presenta en la figura 15.

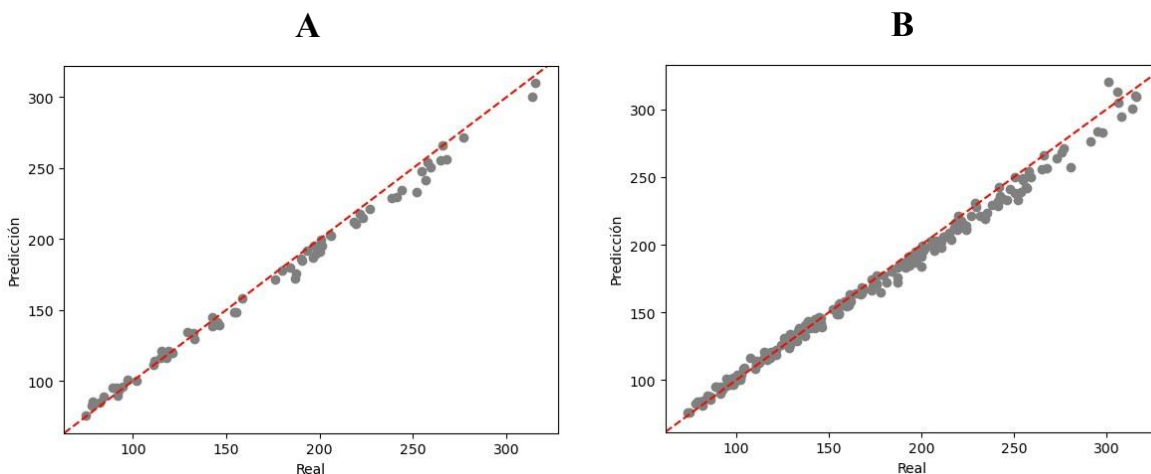


Figura 16. Comparación entre el conteo realizado por el modelo y el conteo real en el juego de datos de células fluorescentes. A: Comparación de las predicciones utilizando sólo el subconjunto de datos de prueba que no se incluyen en el entrenamiento ni validación. B: Comparación de las predicciones utilizando todo el juego de datos.

Por otro lado, se logró modelar en 3D un dispositivo para tomar fotografías de placas de Petri con diámetro de 35, 60, 65, 70 y 90 mm (figura 16) con la finalidad de tomar fotos siempre a las mismas condiciones o lo más similares posible para tener la capacidad de entrenar un modelo con un juego de datos propio de cada usuario. El modelo 3D cuenta con piezas identificadas como base, donde se coloca el LED de iluminación, bandeja, donde se colocan las placas por ser leídas y que permite el paso de la luz del LED, cuerpo, que funciona como soporte estructural, compuerta, que permite abrir y cerrar el dispositivo para introducir o retirar las placas por leer, top, que alberga la cámara con la cual se realiza la toma de fotografías y una tapa. Los archivos STL y scad del modelo pueden ser visualizados y descargados a través del repositorio de Github “Panoptic” con dirección web <https://github.com/MunguiaH/Panoptic>.

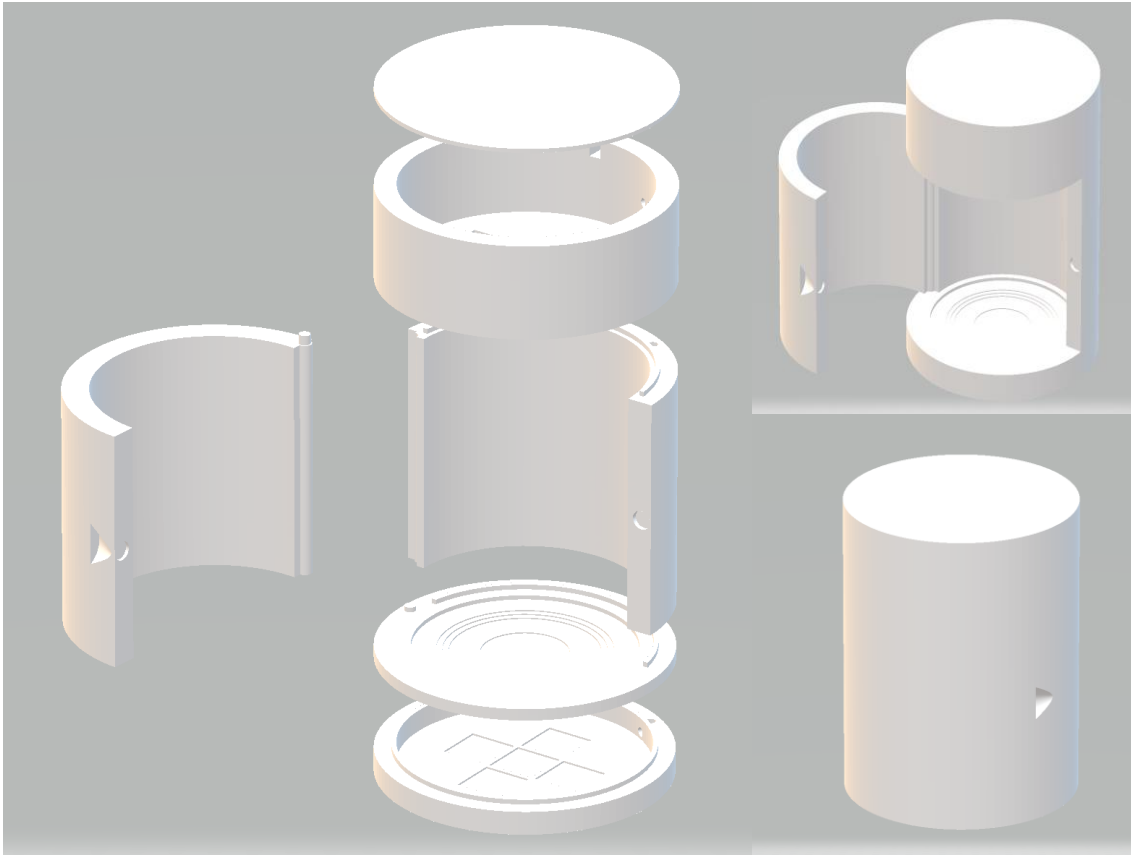


Figura 17. Diseño 3D del dispositivo para lectura de placas Petri generado con OpenSCAD.

El modelo posteriormente fue impreso en 3D con PLA blanco seda marca Amazon[®] Basics, al cual se incorporó una cámara BRobotix[®] RGB con resolución de 1920x1080 pixeles que cuenta con enfoque manual, así como un LED de alta intensidad de 3.5V, generando la primera iteración del dispositivo para lectura de placas (figura 18). Este dispositivo cuenta con conectividad USB para poder capturar imágenes a través de un código de Python generado en el presente PAP y al cual se puede acceder de forma libre a través del repositorio de Github “Panoptic” con dirección web <https://github.com/MunguiaH/Panoptic>. En la figura 19 se presenta un ejemplo de fotografías con iluminación y sin iluminación utilizando el dispositivo.

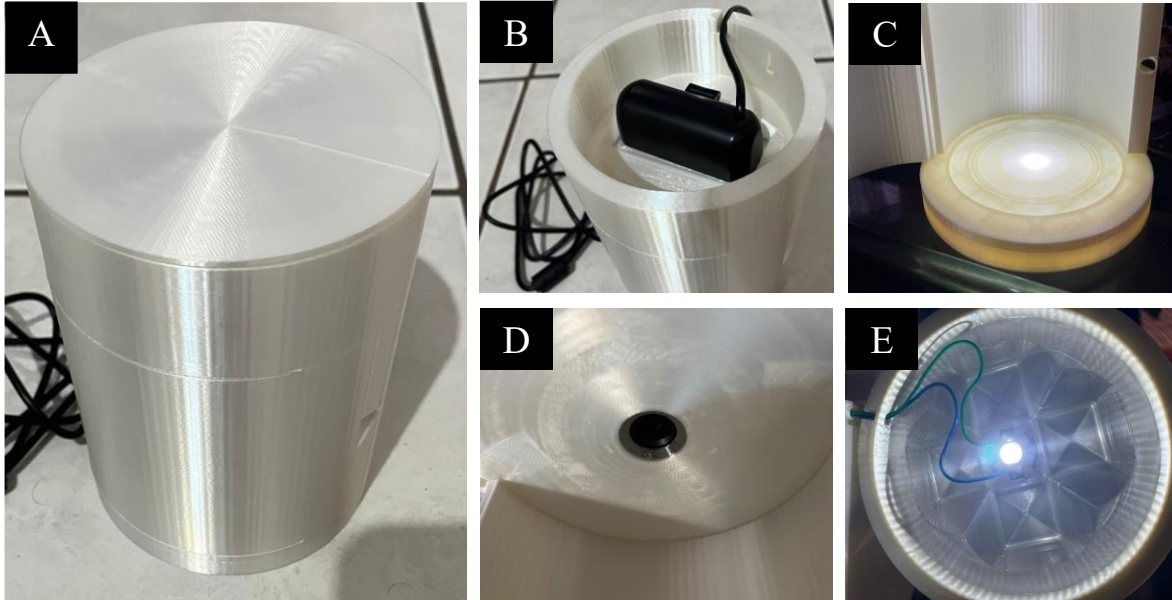


Figura 18. Prototipo del dispositivo lector de placas "Panoptic". A: Vista cerrada. B: Vista sin la tapa, exponiendo la cámara. C: Vista con la compuerta abierta, mostrando la bandeja iluminada donde se colocan las placas Petri. D: Detalle del lente de la cámara que captura la placa desde arriba. E: Detalle del LED que ilumina la bandeja.

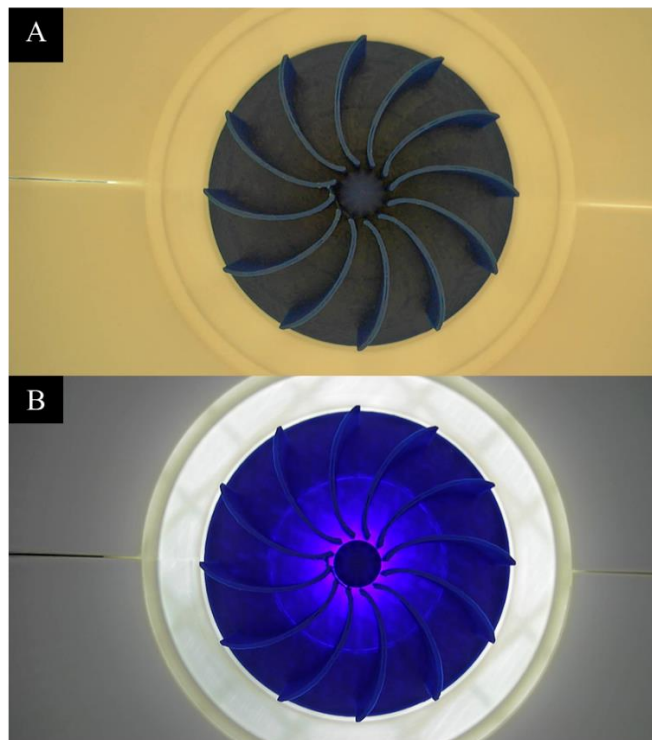


Figura 19. Ejemplo de fotografía tomada con el dispositivo lector de placas "Panoptic". A: fotografía sin iluminación. B: Fotografía con iluminación.

1.6. Valoración de productos, resultados e impactos

De acuerdo con el comportamiento de las curvas de exactitud y pérdida durante el entrenamiento y validación del modelo (figura 14), la red neuronal convolucional propuesta tiene un desempeño adecuado al alcanzar exactitudes por sobre el 95% y pérdidas debajo de 1. Cabe destacar que, como la curva de validación se encuentra sobre la curva de entrenamiento al evaluar la exactitud, el modelo es menos propenso a sufrir de sobreajuste hacia los datos de entrenamiento. Esto último se ve reforzado por el hecho de que la curva de validación al evaluar la función de pérdida del modelo. Por esta razón, se puede expresar que el modelo es eficiente en realizar la labor de conteo incluso con imágenes que aún no ha visto como indica [19], y esta aseveración se ve reforzada por el rendimiento observado en la gráfica 15.A, donde se aprecia un ajuste cercano a la linealidad entre el conteo predicho por el modelo y el conteo real en los datos de prueba, que son ajenos a aquellos utilizados en el entrenamiento y validación. Por último, el modelo podría mejorar en su exactitud, tal como el que presenta [20], sin embargo, este autor presenta resultados en el juego de validación y no en el de prueba, por lo que podría existir sobreajuste en sus datos.

El entrenamiento del modelo se detuvo a las 25 épocas debido al callback de “EarlyStopping” implementado en Keras, que se programó para detener el entrenamiento cuando después de 3 épocas no se observa una disminución de la pérdida de al menos 0.04 unidades, esto pudo haber prevenido un posible sobreajuste del modelo hacia los datos de entrenamiento, como menciona [21]. Adicionalmente, el callback de Keras “ModelCheckpoint”, programado para guardar de forma automática los pesos del modelo que presente mejor exactitud, pudo haber asistido a prevenir el sobreajuste, sin embargo, su implementación se realizó con el objetivo de evitar la pérdida del progreso del entrenamiento por interrupciones imprevistas como una “muerte” del kernel.

Si bien se logró obtener un modelo que presentó resultados favorables con el juego de datos de células fluorescentes, el modelo podría requerir ajustes al aplicarlo en fotografías de placas de Petri, por lo que la adaptación del modelo para esta tarea queda pendiente para futuros

desarrollos del PAP. Otro objetivo que no se realizó durante el presente PAP fue la adaptación del modelo para localización y clasificación de las colonias de microorganismos.

Por otro lado, el modelo de dispositivo impreso en 3D podría asistir a las tareas pendientes para cumplir con los objetivos propuestos en el presente proyecto, ya que permitirá tener uniformidad en el foco, iluminación, distancia de la cámara a la placa, resolución de la cámara y ajuste a colores para asistir al aprendizaje efectivo del modelo. Si bien el diseño del dispositivo es simple comparado con los disponibles en el mercado, puede ser una primera iteración de un dispositivo que sirva para romper la barrera de acceso a la tecnología de redes neuronales convolucionales para labores de conteo de colonias en biotecnología, principalmente porque para construir el dispositivo en su versión 1.7, que es la presentada en el actual proyecto, se invirtieron aproximadamente 70.00 MXN en filamento de PLA, 9 horas de impresión 3D con un consumo de 300 Wh, de acuerdo con su ficha técnica, que representan aproximadamente 5.00 MXN con el costo más alto por kWh de la CFE de 1.803 MXN, 390.00 MXN por la cámara Brobotix, 5.00 MXN de cableado y 5.00 MXN por un LED de alta intensidad, totalizando en 475 MXN.

1.7. Bibliografía y otros recursos

- [1] H.H. Aghdam, E.J. Heravi, *Guide to Convolutional Neural Networks: A practical application to traffic-sign detection and classification*, Springer, 2017, pp. 61-120.
- [2] L. Ciampiconi, A. Elwood, M. Leonardi, A. Mohamed, A. Rozza, “A survey and taxonomy of los functions in machine learning”. arXiv:2301, 2023.
- [3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, Craig Citro, Greg S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. “TensorFlow: Large-scale machine learning on heterogeneous systems”, 2015.
- [4] F. Chollet, “Keras”. Github repository, 2015.

XX Modificar todos los números por la adición de referencias en la sección 1.1

- [5] S. Altuntaş y H. Meral, “Süt Endüstrisinde Moleküler Metotlar İle Mikrobiyel Tanı” en Gıda Mühendisliği Kongresi, 2015, pp. 25.
- [6] T. Sandle, “Ready for The Count? Back-To-Basics Review Of Microbial Colony Counting.” Review Of Microbial Colony Counting. 24. pp. 1-10, 2020.
- [7] G. Corkidi, R. Diaz-Uribe, J.L. Folch-Mallol, y J. Nieto-Sotelo, “COVASIAM: an Image Analysis Method That Allows Detection of Confluent Microbial Colonies and Colonies of Various Sizes for Automated Counting”, Appl. Environ. Microbiol. 64(4): 1400-1404, 1998.
- [8] J. Dahle, M. Kakar, H.B. Steen y O. Kaalhus, “Automated counting of mammalian cell colonies by means of a flat bed scanner and image processing”, Cytom Part A. 60A (2):182–188, 2004.
- [9] F. Li, A. Karpathy, y J. Johnson (2016, Feb 1). Spatial Localization and Detection [Online]. Disponible en: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf.
- [10] R. Szeliski, *Computer Vision. Algorithms and Applications*, 2^a ed. Texts in Computer Vision. Springer, 2017.
- [11] I. Mihajlovic (2019, Abril). *Everything you ever wanted to know about computer vision* [Online]. Disponible en: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>
- [12] L. Gajski, “Privatization of Science – A Silent Phenomenon”, Pannoniana: Journal of Humanities 3(2), pp. 239-244, 2019.
- [13] C. Hewitt, S. Han, D. Togerson, J. Watson, J.M. Bland, “Adequacy and reporting of allocation concealment: review of recent trials published in four general medical journals”. BMJ 330. pp.1057, 2005.
- [14] V. Lempitsky y A. Zisserman, “Learning to count objects in images”, en NIPS, 2010.
- [15] S. Mjchorwska, J. Pawłowski, J. Guła, T. Bonus, A. Hanas, A. Loch, A. Pawlak, J. Roszkowiak, T. Golan y Z. Drulis-Kawa, “AGAR a microbial colony dataset for Deep learning detection”. arXiv: 2108.01234, 2021.

- [16] V. Thambawita, I. Strümke, S. Hicks, P. Halvorsen, S. Parasa, M. Riegler, “Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images.” *Diagnostics* 11(12), p. 2183, 2021
- [17] L. Makrai, B. Fodróczy, S. Nagy, P. Czeiszing, I. Csabai, G. Szita, N. Solymosi, “Annotated dataset for Deep-learning-based bacterial colony detection”. 10(497), pp. 1-6, (2023).
- [18] C. Liu, Y. Zhong, A. Zisserman y W. Xie, “CounTR: Transformer-based Generalised Visual Counting”. arXiv: 2208.13721, 2022.
- [19] T.U. Ahmed, M.N. Jamil, M.S. Hossain, K. Andersson, “An integrated real-time deep learning and belief rule base intelligent system to assess facial expression under uncertainty. ICIEV, 2020.
- [20] T. Bonus, T. Golan. (2019, Julio). “Objects counting by estimating a density map with convolutional neural networks”. *Neurosys* [Online] Disponible en: <https://neurosys.com/blog/objects-counting-by-estimating-a-density-map>
- [21] S.R. Muttineni. (2020, Septiembre). “Keras EarlyStopping Callback to train the Neural Networks Perfectly”. *Towards AI* [Online]. Disponible en: <https://pub.towardsai.net/keras-earlystopping-callback-to-train-the-neural-networks-perfectly-2a3f865148f7>

2. Productos

Nombre y código del PAP	Programa a Centros de Investigaciión Externos I. Código: PAP4G03.
Nombre del proyecto	Construcción de un modelo basado en redes neuronales profundas de código abierto para el conteo, localización y clasificación de colonias en placas Petri.
Descripción (qué es, para quién se realizó y para qué es):	Se elaboró un modelo de redes neuronales convolucionales para el conteo de colonias a partir de fotografías. Adicionalmente, se diseñó e imprimió en 3D un dispositivo para tomar fotografías a placas de Petri bajo condiciones de iluminación controladas

	para generar un juego de datos. Finalmente, se incorporaron los elementos de software generados en un repositorio de Github para facilitar el acceso a la información.
Autores:	Hugo Munguía Orozco.

3. Reflexión crítica y ética de la experiencia

El RPAP tiene también como propósito documentar la reflexión sobre los aprendizajes en sus múltiples dimensiones, las implicaciones éticas y los aportes sociales del proyecto para compartir una comprensión crítica y amplia de las problemáticas en las que se intervino.

3.1 Sensibilización ante las realidades

Personalmente fue en extremo impresionante para mí ver los precios de los dispositivos que se comercializan para realizar la tarea de conteo automatizado, siendo que la tecnología que permite lograrlo puede ser construida a partir de herramientas gratuitas. Además, la información para aprender a realizar los modelos de aprendizaje profundo se encuentra de forma libre en internet, teniendo múltiples libros, artículos y cursos disponibles para el público general. Sin embargo, la motivación de lucro lleva a las empresas a desarrollar tecnología y venderla, mas es inconcebible que se pretenda imponer precios exorbitantes por dispositivos que utilizan componentes sencillos. Esto únicamente acrecienta la barrera de entrada de las tecnologías para centros de educación e investigación y puede retrasar el avance tecnológico al convertir las nuevas herramientas en algo inaccesible.

Es por lo anterior que encontré la motivación para realizar esta primer iteración de una propuesta que, a mi forma de pensar, es un regalo para todo aquel que se interese por aportar en romper con la barrera que impide el acceso a las nuevas tecnologías. Al compartir el código y diseños considero que personas más aptas y capacitadas pueden germinar la semilla que presento y cultivar un proyecto que sea de utilidad para los trabajadores de laboratorio, investigadores, educadores y estudiantes. Es importante compartir herramientas de este tipo de forma abierta porque pueden potencialmente mejorar las condiciones de trabajo de investigadores, reduciendo las horas que deben dedicar a tareas repetitivas y cansadas,

permitiéndoles enfocar su capacidad física y mental en continuar desarrollando las tareas más demandantes de sus líneas de investigación.

3.2 Aprendizajes logrados

Durante el presente PAP fue especialmente demandante el comprender cómo funcionan las redes neuronales artificiales, cómo se programan y optimizan ython[®] utilizando las librerías de Keras y Tensorflow, así como aprender la forma de crear arquitecturas eficientes. Fue un trabajo autodidacta que trajo mucha satisfacción y un poco de frustración también por las complicaciones que surgieron durante el desarrollo del proyecto. Sin embargo, considero que se logró obtener un nivel de conocimiento básico para comenzar a generar propuestas de utilidad apoyándose con las herramientas que ofrece el aprendizaje profundo. También fue necesario afinar algunas estrategias de manejo de datos para extraer, organizar y transformar la información de los juegos de datos de forma que pudiera ser utilizada para el aprendizaje profundo.

Adicionalmente, fue todo un reto aprender a modelar en 3D utilizando OpenSCAD, ya que es una herramienta en la que todos los objetos se crean y manipulan utilizando un lenguaje de programación propio del software. Solamente se tiene una ventana de visualización, pero no es posible modificar los objetos mediante ella. Si bien esto suena como algo ineficiente, es en realidad bastante efectivo, pues el poder generar los objetos a través de programación permite tener un control sumamente estricto en las dimensiones de los objetos para lograr un ensamblado perfecto a la hora de la manufactura.

Finalmente, materializar los diseños de OpenSCAD utilizando la impresión 3D fue una tarea no menos complicada, pues la impresora K1 de Creality[®] presentó múltiples fallos de componentes durante su uso. Esto requirió que aprendiera a desarmar, limpiar, calibrar y reensamblar la impresora para poder finalmente imprimir el prototipo generado. Además, fue necesario que aprendiera cómo es que cada parámetro de impresión, como la temperatura de la boquilla, temperatura de la cama, velocidad de impresión, tasa de extrusión, offset de la boquilla en el eje Z, velocidad de ventiladores de enfriamiento, configuración de soportes de

la impresión, número de paredes, patrón de relleno, entre muchas otras, afecta a la calidad para poder siquiera comenzar la manufactura.

3.3 Inventario de competencias Inicial (ingreso del PAP) e Inventario de competencias Final (salida al PAP).

Competencias potencializadas: **color azul**.

Competencias nuevas: **color rojo**.

Competencia	Evidencia	Relevancia/Fortaleza*	
Conocimientos	Estudio de cinéticas de crecimiento.	He realizado cinéticas de crecimiento de microorganismos como bacterias y levaduras.	Puedo aportar con experimentación para establecer parámetros cinéticos que permitan optimizar y/o escalar un bioproceso.
	Diseño de biorreactores basado en cinéticas.	Realicé el diseño de un sistema de reactores en continuo en Ing. De biorreactores.	Puedo aportar el diseño de un sistema de reactores interconectados para estudiar los efectos de metabolitos en diferentes cultivos.
	Programación de sistemas de monitoreo y control.	He hecho controladores de temperatura basados en Arduino	Puedo aportar para controlar dispositivos mediante programación, como control de bombas, elementos de calefacción, etc.
	Programación y diseño de redes neuronales.	He realizado redes neuronales básicas para categorización, localización, conteo y predicción usando imágenes, video y audio.	Puedo programar redes neuronales para solucionar problemas de clasificación, identificación, conteo y predicción utilizando datos de imágenes, video y audio para cumplir con necesidades específicas.
	Diseño gráfico e industrial.	He realizado diseños independientes para productos que estoy comercializando.	Puedo poner mis habilidades básicas para iniciar diseños de reactores, componentes o materiales gráficos que ayuden a un proyecto.
	Impresión en 3D	Realicé la impresión de un prototipo de dispositivo para lectura de placas Petri que incorpora cámara web e iluminación.	Ahora puedo materializar ideas de diseños que surgen para resolver problemáticas en mi vida diaria y profesional. Esto me permite tener más herramientas que pueda usar como ingeniero.
Habilidades	Creatividad para resolución de problemas	He resuelto problemas en mi trabajo de forma creativa buscando soluciones fuera de lo evidente.	Puedo buscar soluciones diferentes a problemas actuales o nuevos mediante búsqueda de teoría y experimentación.
	Análisis de necesidades	He realizado proyectos para cumplir con necesidades específicas de un grupo de interés, adaptando los objetivos para cumplir de mejor manera con ellos.	Puedo ayudar a buscar la forma más apropiada para cumplir con las necesidades del grupo de interés para el cual se busca realizar un proyecto.

Actitudes	Liderazgo empático.	En mi trabajo y proyectos personales busco adaptarme a las condiciones de las personas con las que trabajo para generar un entorno saludable.	Dependiendo del objetivo de cada proyecto, puedo enfocarme en generar un ambiente de trabajo saludable manteniendo un ritmo de trabajo apropiado con todos los integrantes. Mi prioridad siempre es que los integrantes se sientan empoderados y motivados a hacer uso de sus conocimientos y habilidades.
	Adaptabilidad al cambio.	En mi trabajo y proyectos personales debo adaptarme a condiciones cambiantes todo el tiempo, muchas que incluso están fuera de mi control.	Puedo hacer actualizaciones de organización para lograr las metas establecidas aun cuando el, o los planes anteriores se ven comprometidos por cambios no previstos o resultados no esperados.
	Capacidad analítica.	En mi vida en general trato de analizar oportunidades y amenazas para adaptar mis planes.	Puedo ayudar a analizar y sintetizar información para presentar ideas que permitan realizar cambios a un proyecto, sean cambios técnicos, organizacionales o metodológicos.
	Tolerancia a la frustración y fracaso	Durante el PAP tuve bastantes errores, fracasos, fallas de organización por causa de tener múltiples actividades y cambios de planes. Sin embargo, aprendí a no dejar que estos problemas me frenaran o desmotivaran.	La resiliencia es para mí algo sumamente valioso, pues puede ser el diferenciador entre el fracaso y el éxito. Puedo decir que ahora me es menos devastador tener múltiples fallas en poco tiempo y que, en lugar de sentirme como un fracaso por ello, puedo sentirme bien por estar aprendiendo y experimentando.