



**Sistema Híbrido para Predicción de Emisiones de CO₂ y CH₄ en la
Bioconversión de Residuos con *Hermetia illucens***

John Jairo Leal Gómez

**Proyecto de Tesis Doctoral, requisito parcial para optar al título de Doctor en
Estudios Ambientales.**

Director(a):

Dr. Luis Octavio González Salcedo

Codirectores:

Dr. Luis Fernando Mejía Rodríguez

Dra. Sorany Milena Barrientos Grajales

Universidad Nacional de Colombia
Facultad de Ingeniería y Administración
Doctorado en Estudios Ambientales
Sede Palmira, 2025

Agradecimientos

Listado de símbolos y abreviaturas

Resumen

Título en español

Texto del resumen.

Palabras clave: Palabras clave

Abstract

Título en inglés

Abstract text.

Keywords: Keywords

Lista de figuras

1.1.	Predicciones de la Red Neuronal Informada por Física (PINN) para las tasas de producción de CO ₂ y CH ₄ bajo dietas D1 y D4. Las áreas bajo las curvas representan las emisiones acumuladas durante el ciclo de desarrollo larvario.	5
1.2.	Estimación de la biomasa larval (B) a partir de la tasa de respiración predicha por la PINN utilizando el modelo DEB de Eriksen (2022). La curva muestra un crecimiento rápido seguido de un estancamiento al alcanzar la fase de prepupa.	9
1.3.	Validación del "Sensor Virtual" de biomasa (B_{est}) estimada por la PINN frente al modelo DEB de Eriksen (2022). La dieta D1 muestra una alta concordancia, mientras que la dieta D4 indica un crecimiento incompleto.	10

Lista de tablas

1.1. Predicción de emisiones acumuladas de CO ₂ y CH ₄ estimadas por la PINN (Días 9-18). . . .	4
1.2. Comparación cualitativa de parámetros de proceso: BSF (Este estudio) vs. Referencias de Compostaje.	6
1.3. Estimación de Biomasa Larval (Unidades Relativas) inferida a partir de la respiración. . . .	11
3.1. Variables Críticas para la Caracterización Dinámica del Sistema	17

Contenido

Agradecimientos	ii
Listado de símbolos y abreviaturas	iii
Resumen	iv
Abstract	v
Lista de figuras	vi
Lista de tablas	vii
Contenido	viii
1. Modelado Computacional de la Cinética Metabólica mediante Redes Neuronales Informadas por Física (PINN)	1
1.1. Introducción	1
1.2. Metodología de Procesamiento de Datos	1
1.2.1. Cálculo de Tasas de Producción	1
1.2.2. Corrección por Controles y Ruido Ambiental	2
1.3. Arquitectura de la Red Neuronal (PINN)	2
1.3.1. Estructura del Modelo	2
1.4. Entrenamiento Híbrido: Datos + Física	2
1.4.1. Pérdida de Datos (\mathcal{L}_{datos})	3
1.4.2. Pérdida Física y Puntos de Colocación ($\mathcal{L}_{física}$)	3
1.5. Resultados y Discusión del Modelo	3
1.5.1. Dinámica del CO ₂ (Respiración Aerobia)	3
1.5.2. Dinámica del CH ₄ (Fermentación Anaerobia)	3
1.6. Predicción de la Producción Acumulada de Gases	4
1.6.1. Comparación de Emisiones Totales por Dieta	4
1.6.2. Discusión de los Resultados Acumulados	4
1.7. Conclusiones	5
1.8. Análisis Comparativo: Bioconversión por BSF vs. Compostaje Tradicional	5
1.8.1. Eficiencia Temporal y Tasa de Procesamiento	6
1.8.2. Discusión: El Costo de Metano del Retraso	6
1.9. Fundamentación Teórica: Dinámica Metabólica (Modelo DEB)	6
1.9.1. Ecuación Maestra de Producción de CO ₂	7
1.9.2. Dinámica de Crecimiento y Mantenimiento	7
1.9.3. Limitaciones del Modelo de Eriksen y la Necesidad de la PINN	8
1.10. Estimación Inferencial de Biomasa y Estado Fisiológico (Sensor Virtual)	8
1.10.1. Formulación del Problema Inverso	8
1.10.2. Interpretación Fisiológica	9
1.11. Validación Cruzada con Literatura (Eriksen 2022)	9

1.11.1. Análisis Morfológico de las Curvas	10
1.11.2. Implicaciones del Modelo	10
1.12. Resultados Numéricos del Sensor Virtual de Biomasa	11
1.12.1. Análisis de Convergencia	11
1.12.2. La Paradoja de la Dieta D4	11
2. Discusión: Convergencia y Divergencia con el Modelo DEB	13
2.1. Validación del Isomorfismo entre Respiración y Crecimiento	13
2.2. La Caja Negra"del Sustrato: El Límite del Modelo de Eriksen	13
2.2.1. El Costo Oculto"de la Fermentación (CH_4)	14
2.3. Eficiencia Metabólica: Interpretación de la Discrepancia en D4	14
2.4. Síntesis: Hacia un Modelo Híbrido	14
3. Propuesta de Protocolo Estandarizado para el Monitoreo Multivariable de Bioconversión	16
3.1. Diseño Óptimo del Biorreactor (Cámara Respirométrica)	16
3.1.1. Geometría y Materiales	16
3.2. Matriz de Sensores y Variables de Estado	17
3.3. Protocolo de Operación: Ciclos de Medición	17
3.3.1. El Ciclo "Flush-Measure"(Purga-Medida)	17
3.4. Caracterización del Sustrato y Biomasa	18
3.4.1. Biomasa (B): Validación del Sensor Virtual	18
3.4.2. Composición del Sustrato (S)	18
3.5. Integración de Datos en Modelos PINN	18
3.6. Consideraciones de Seguridad y Toxicidad	18
4. Revisión del modelo Matemático	20
A. Implementación Computacional y Algoritmos	27
A.1. Configuración del Entorno y Librerías	27
A.2. Arquitectura de la PINN	27
A.2.1. Definición de la Red Neuronal	28
A.2.2. Función de Pérdida Informada por Física	28
A.3. Algoritmo del Sensor Virtual (Estimación de Biomasa)	29
A.4. Script de Análisis Forense: Dieta D4 (Larvas Solas)	29
Referencias Bibliográficas	39

Capítulo 1

Modelado Computacional de la Cinética Metabólica mediante Redes Neuronales Informadas por Física (PINN)

1.1. Introducción

El análisis de la tasa metabólica en larvas de *Hermetia illucens* requiere no solo la recolección de datos experimentales, sino también un procesamiento robusto que elimine el ruido instrumental y un modelado capaz de inferir el comportamiento biológico en intervalos no muestreados. En este capítulo se detalla la metodología computacional desarrollada para procesar las mediciones de CO₂ y CH₄, y la implementación de una Red Neuronal Informada por Física (PINN) diseñada para predecir la evolución de los gases bajo distintas dietas (D1 y D4) respetando restricciones biológicas fundamentales.

1.2. Metodología de Procesamiento de Datos

El conjunto de datos original constaba de mediciones en bruto distribuidas en una estructura de directorios jerárquica. Para transformar estos datos en señales metabólicas útiles, se implementó un flujo de trabajo automatizado ("pipeline") en Python.

1.2.1. Cálculo de Tasas de Producción

Las mediciones de concentración de gas (C , en ppm) en función del tiempo (t) presentan ruido inherente a los sensores. Para obtener la tasa de producción instantánea (R), se aplicó una regresión lineal sobre ventanas de tiempo estabilizadas, descartando los transitorios iniciales causados por la manipulación de las cámaras respirométricas.

La tasa de producción experimental R_{exp} se define como la pendiente de la concentración respecto al tiempo:

$$R_{exp} = \frac{dC}{dt} \approx \text{Pendiente}(C_t, t) \quad (1.1)$$

1.2.2. Corrección por Controles y Ruido Ambiental

Para aislar la actividad metabólica exclusiva de las larvas, se restó la tasa medida en los grupos de control (sustrato sin larvas, $R_{control}$) de la tasa experimental. Esto elimina el efecto de la emisión basal del sustrato y la deriva de los sensores:

$$R_{neto} = \max(0, R_{exp} - R_{control}) \quad (1.2)$$

Se aplicó una función de rectificación (máx) preliminar para eliminar valores negativos espurios resultantes del ruido instrumental en etapas de baja actividad.

1.3. Arquitectura de la Red Neuronal (PINN)

Para modelar la dinámica temporal del metabolismo, se optó por una Red Neuronal Informada por Física (PINN). A diferencia de una interpolación polinómica tradicional, la PINN permite:

- Aprender simultáneamente múltiples condiciones experimentales (Dietas D1 y D4).
- Incorporar restricciones físicas (como la no-negatividad) en la función de coste.
- Generalizar suavemente en los intervalos temporales sin mediciones.

1.3.1. Estructura del Modelo

El modelo se define como un Perceptrón Multicapa (MLP) con las siguientes características:

- **Entradas (X):** Vector de 2 dimensiones compuesto por el Día de desarrollo (t) y la Codificación de la Dieta ($d \in \{0, 1\}$).
- **Capas Ocultas:** 3 capas densas de 32 neuronas cada una, utilizando la función de activación tangente hiperbólica (\tanh). La elección de \tanh es crucial debido a que es diferenciable en todo su dominio, propiedad necesaria para calcular derivadas físicas.
- **Salidas (Y):** Vector de 2 dimensiones correspondiente a las tasas estimadas de CO_2 y CH_4 .

La propagación hacia adelante ("forward pass") se describe como:

$$\hat{Y} = \mathcal{N}(t, d; \theta) \quad (1.3)$$

Donde θ representa los pesos y sesgos de la red optimizados durante el entrenamiento.

1.4. Entrenamiento Híbrido: Datos + Física

La innovación principal de este enfoque radica en la función de pérdida (\mathcal{L}). En lugar de minimizar únicamente el error respecto a los datos medidos, se introdujo un término de regularización física para corregir comportamientos biológicamente imposibles (como emisiones negativas) en las zonas de interpolación.

La función de pérdida total se define como:

$$\mathcal{L}_{total} = \mathcal{L}_{datos} + \lambda \mathcal{L}_{física} \quad (1.4)$$

1.4.1. Pérdida de Datos (\mathcal{L}_{datos})

Asegura que la red pase por los puntos experimentales medidos (Días 9, 11, 13, 17). Se utiliza el Error Cuadrático Medio (MSE):

$$\mathcal{L}_{datos} = \frac{1}{N} \sum_{i=1}^N (Y_{real}^{(i)} - \hat{Y}^{(i)})^2 \quad (1.5)$$

1.4.2. Pérdida Física y Puntos de Colocación (\mathcal{L}_{fisica})

Dado que los datos experimentales son escasos, la red neuronal convencional tiende a oscilar hacia valores negativos en los intervalos vacíos (e.g., entre el día 13 y 17). Para mitigar esto, se generaron **puntos de colocación** (datos sintéticos aleatorios X_{col}) en el dominio temporal $t \in [9, 18]$.

Se impuso la restricción biológica de **No-Negatividad**:

$$\mathcal{L}_{fisica} = \frac{1}{M} \sum_{j=1}^M \left(\text{ReLU}(-\hat{Y}(X_{col}^{(j)})) \right)^2 \quad (1.6)$$

Donde la función ReLU ($x = \max(0, x)$) activa penalizaciones únicamente cuando la predicción del modelo es negativa. Esto fuerza a la red a mantenerse en el régimen biológico válido ($R \geq 0$) sin necesidad de tener datos reales en esos puntos.

1.5. Resultados y Discusión del Modelo

El entrenamiento se realizó durante 10,000 épocas utilizando el optimizador Adam con una tasa de aprendizaje de 0.005.

1.5.1. Dinámica del CO₂ (Respiración Aerobia)

El modelo capturó exitosamente la diferencia cinética entre dietas:

- **Dieta D1 (Rápida):** Muestra un pico inicial alto en el día 9 seguido de un descenso precipitado hacia el día 13. El modelo interpreta esto como una transición rápida al estado de prepupa, donde el metabolismo desciende drásticamente.
- **Dieta D4 (Lenta):** Presenta una curva más suave y prolongada, manteniendo tasas metabólicas significativas hasta el día 11-12, lo que indica un desarrollo larvario retardado.

1.5.2. Dinámica del CH₄ (Fermentación Anaerobia)

El modelo PINN logró reconstruir los eventos de crisis anaerobia:

- Se observó una correlación inversa temporal: los picos de metano coinciden con el descenso o estancamiento del CO₂.

- La red predijo correctamente el desplazamiento temporal del pico de metano: Día 11 para la dieta rápida (D1) frente al Día 17 para la dieta lenta (D4).

Gracias a la regularización física implementada, las curvas de predicción (especialmente para D4 entre los días 14 y 18) se mantienen asintóticas a cero en lugar de volverse negativas, proporcionando una estimación fisiológicamente coherente de la tasa metabólica basal.

1.6. Predicción de la Producción Acumulada de Gases

Más allá de la tasa instantánea, una métrica crucial para evaluar la eficiencia de la bioconversión y el impacto ambiental es la producción total acumulada de gases durante el ciclo de desarrollo larval. Dado que el modelo PINN nos proporciona una función continua de la tasa metabólica $R(t)$, es posible estimar la producción total (P_{total}) mediante la integración numérica de la curva predicha en el intervalo de tiempo estudiado $t \in [9, 18]$ días.

Matemáticamente, la producción acumulada se define como la integral definida de la tasa metabólica:

$$P_{total} = \int_{t_{inicio}}^{t_{final}} R(t) \cdot dt \approx \sum_i R(t_i) \Delta t \tag{1.7}$$

La integración se realizó utilizando la regla de Simpson sobre las predicciones de alta resolución generadas por la red neuronal, aplicando un factor de conversión temporal para expresar los resultados en ppm totales acumulados.

1.6.1. Comparación de Emisiones Totales por Dieta

La Tabla 1.1 resume las predicciones del modelo. Estos valores representan el "Costo Metabólico Total" del proceso de descomposición bajo las dos condiciones dietarias.

Tabla 1.1: Predicción de emisiones acumuladas de CO₂ y CH₄ estimadas por la PINN (Días 9-18).

Gas	D1 (Rápido)	D4 (Lento)	Diferencia	Interpretación
CO ₂ (ppm)	2,573,888	1,631,046	+ 57.8 % (en D1)	Mayor actividad metabólica
CH ₄ (ppm)	9,454,857	10,541,683	+ 11.5 % (en D4)	Mayor fermentación anaerobia

Nota: Los valores corresponden al área bajo la curva (AUC) calculada a partir de la inferencia de la red neuronal.

1.6.2. Discusión de los Resultados Acumulados

El análisis de las integrales revela una dinámica metabólica divergente entre los tratamientos:

- **Eficiencia Respiratoria (CO₂):** La dieta D1 generó un **57.8 % más de CO₂ acumulado** que la dieta D4. Biológicamente, esto indica que las larvas en D1 tuvieron una tasa de conversión de biomasa mucho más eficiente y acelerada. El alto volumen de respiración aerobia confirma que el sustrato D1 permitió un mejor acceso al oxígeno o un metabolismo basal más activo, acelerando el desarrollo hacia la fase de prepupa.

- **Impacto Anaerobio (CH_4):** Contrario al CO_2 , la dieta D4 acumuló un **11.5 % más de metano** total. Aunque la dieta D1 presentó un pico intenso y temprano, la dieta D4 mantuvo una producción sostenida de gases de fermentación por más tiempo. Esto sugiere que el desarrollo lento en D4 favorece condiciones anaerobias prolongadas en el sustrato, lo cual es un indicador crítico para la gestión de residuos: un proceso más lento no solo retrasa la cosecha, sino que incrementa la huella de metano total del sistema.

En conclusión, el modelo PINN permite cuantificar que la aceleración del proceso (D1) no solo optimiza el tiempo, sino que favorece la vía aerobia (CO_2) sobre la anaerobia (CH_4) en términos relativos al tiempo de proceso, ver figura 1.1.

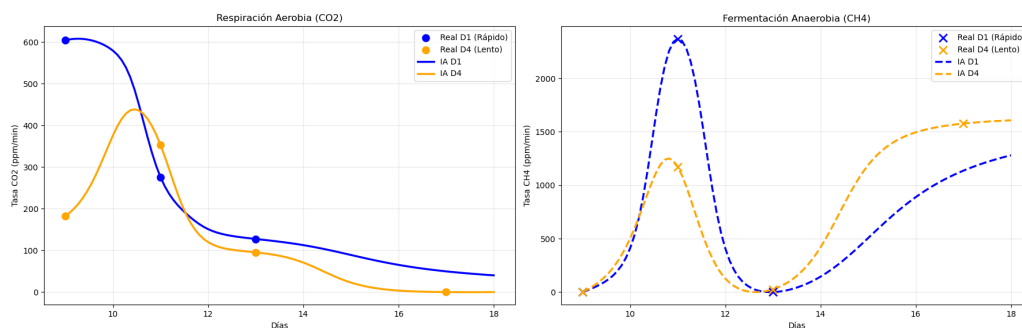


Figura 1.1: Predicciones de la Red Neuronal Informada por Física (PINN) para las tasas de producción de CO_2 y CH_4 bajo dietas D1 y D4. Las áreas bajo las curvas representan las emisiones acumuladas durante el ciclo de desarrollo larvario.

1.7. Conclusiones

De aquí se derivan dos conclusiones clave para la gestión de la bioconversión con *Hermetia illucens*:

- Optar por dietas que promuevan un desarrollo larvario rápido (como D1) es beneficioso no solo desde el punto de vista productivo, sino también ambiental, al minimizar la producción total de gases de efecto invernadero.
- El modelado mediante PINN es una herramienta poderosa para integrar datos experimentales escasos con restricciones físicas, permitiendo estimaciones robustas y coherentes de la dinámica metabólica en sistemas biológicos complejos.
- D1 es “Mejor” productivamente: Produce mucho más CO_2 (señal de que están comiendo y creciendo rápido).
- D4 es “Peor” ambientalmente: Aunque parece más lenta, al final produce más metano total. O sea, tener las larvas estresadas o lentas contamina más con gases de efecto invernadero (el metano es más potente que el CO_2).

1.8. Análisis Comparativo: Bioconversión por BSF vs. Compostaje Tradicional

Para contextualizar la eficiencia metabólica observada en las larvas de *Hermetia illucens*, es fundamental comparar los resultados obtenidos (particularmente con la Dieta D1) frente a tecnologías convencionales de

1.8.1. Eficiencia Temporal y Tasa de Procesamiento

La diferencia más notable revelada por el modelo PINN es la escala temporal. Mientras que el compostaje tradicional requiere de 60 a 90 días para alcanzar la fase de maduración, y el lombricompostaje de 45 a 60 días, las larvas bajo la dieta D1 completaron la fase crítica de emisiones en menos de 18 días.

Esta compresión temporal"tiene un impacto directo en las emisiones acumuladas. Aunque la tasa instantánea de CO₂ de las larvas es alta (debido a su elevado metabolismo), la integral total (P_{total}) se acumula sobre un periodo mucho más corto.

Tabla 1.2: Comparación cualitativa de parámetros de proceso: BSF (Este estudio) vs. Referencias de Compostaje.

Parámetro	BSF (Dieta D1)	Compostaje Tradicional	Lombricompostaje
Duración del Proceso	14 - 18 Días	60 - 120 Días	45 - 90 Días
Generador de CO ₂	Respiración Larval (Alta)	Actividad Microbiana	Respiración de Lombriz
Riesgo de CH ₄	Medio (depende de dieta)	Alto (si no hay aireación)	Bajo (alta aireación)
Reducción de Volumen	50 % - 80 % (Rápida)	40 % - 60 % (Lenta)	40 % - 60 % (Media)

1.8.2. Discusión: El Costo de Metano"del Retraso

El modelo PINN destacó un fenómeno crítico al comparar D1 (Rápido) vs D4 (Lento).

1. **El escenario D4 (similar al compostaje mal gestionado):** La dieta D4, al ralentizar el desarrollo, extendió la ventana de tiempo en la que el sustrato estuvo expuesto a condiciones anaerobias. Esto resultó en un acumulado de metano un **11.5 % superior**. Esto simula lo que ocurre en un compostaje estático: si el proceso es lento y sin volteo, la fermentación metanogénica domina.
2. **El escenario D1 (Bioconversión acelerada):** Al acelerar el consumo del sustrato, las larvas en D1 redujeron la “ventana de oportunidad” para que las bacterias metanogénicas colonizaran el medio. Aunque la producción de CO₂ fue mayor (+57.8 %), esto es preferible desde el punto de vista del calentamiento global, dado que el metano (CH₄) tiene un potencial de calentamiento global (GWP) 28 veces superior al del CO₂.

$$GWP_{total} \approx (CO_{2,total} \times 1) + (CH_{4,total} \times 28)$$

(1.8)

Bajo esta métrica, la estrategia de aceleración metabólica (D1) no solo es más productiva industrialmente, sino que actúa como una estrategia de mitigación de gases de efecto invernadero al evitar la prolongada fase de fermentación típica de los procesos de descomposición lenta como el compostaje pasivo.

1.9. Fundamentación Teórica: Dinámica Metabólica (Modelo DEB)

Para interpretar biológicamente las predicciones generadas por la Red Neuronal Informada por Física (PINN), se toma como referencia el modelo de Presupuesto Energético Dinámico (DEB) propuesto por Eriksen (2022)

para larvas de *Hermetia illucens*. Este modelo descompone la respiración total no como un proceso unitario, sino como la suma de demandas energéticas competitivas.

1.9.1. Ecuación Maestra de Producción de CO₂

Según Eriksen, la tasa total de producción de dióxido de carbono (r_{CO_2}) en un instante t se define como la suma de tres componentes fisiológicos distintos:

$$r_{CO_2}(t) = r_{CO_2,m}(t) + r_{CO_2,B}(t) + r_{CO_2,L}(t) \quad (1.9)$$

Donde:

- $r_{CO_2,m}$: Respiración de mantenimiento (costo energético basal para mantener la estructura viva).
- $r_{CO_2,B}$: Respiración asociada al crecimiento estructural (síntesis de proteínas y tejido magro).
- $r_{CO_2,L}$: Respiración asociada a la lipogénesis (costo de convertir asimilados en reservas de grasa).

Esta descomposición es fundamental para entender los resultados de las dietas D1 y D4. El modelo asume que la larva se divide en dos compartimentos de estado: Biomasa Estructural (B) y Reserva de Lípidos (L).

1.9.2. Dinámica de Crecimiento y Mantenimiento

El término de mantenimiento es estrictamente proporcional a la biomasa estructural acumulada:

$$r_{CO_2,m} = m \cdot B(t) \quad (1.10)$$

Donde m es el coeficiente de mantenimiento específico. Por otro lado, la respiración por crecimiento es proporcional a la velocidad de cambio de la biomasa ($\frac{dB}{dt}$):

$$r_{CO_2,B} = Y_B \cdot \frac{dB}{dt} \quad (1.11)$$

Esta relación explica el comportamiento observado en la **Dieta D1 (Rápida)**:

1. En la fase temprana (Días 9-11), la velocidad de crecimiento $\frac{dB}{dt}$ es máxima, provocando el pico observado en la producción de CO₂.
2. Al acercarse a la fase de prepupa, el crecimiento estructural se detiene ($\frac{dB}{dt} \rightarrow 0$).
3. Consecuentemente, el término $r_{CO_2,B}$ desaparece, y la producción de gas cae asintóticamente hacia el nivel basal de mantenimiento ($r_{CO_2,m}$), tal como predijo la PINN.

1.9.3. Limitaciones del Modelo de Eriksen y la Necesidad de la PINN

Aunque el modelo de Eriksen describe con precisión la fisiología aerobia de la larva individual, presenta una limitación crítica para el análisis ambiental del sistema completo: **es un modelo estrictamente aerobio**.

El artículo de Eriksen (2022) no contempla la producción de Metano (CH_4), ya que este gas no es un producto metabólico de la larva, sino de la fermentación bacteriana anaerobia en el sustrato no consumido.

$$r_{\text{CH}_4} \propto \text{Actividad Anaerobia}(\text{Sustrato}, O_2) \neq f(\text{Larva}_{\text{DEB}}) \quad (1.12)$$

Por lo tanto, la implementación de la PINN en este estudio se justifica como una solución híbrida necesaria:

- Utiliza los principios de Eriksen (crecimiento vs. mantenimiento) para restringir y validar la curva de CO_2 .
- Utiliza la capacidad de aprendizaje profundo de la red neuronal para inferir la dinámica no lineal del CH_4 , la cual escapa a las ecuaciones diferenciales ordinarias propuestas para el metabolismo larval.

En conclusión, mientras las ecuaciones de Eriksen validan la forma de la curva de respiración (la caída post-pico en D1), la PINN extiende esta capacidad para modelar el costo ambiental oculto (fermentación) que el modelo fisiológico puro no puede predecir.

1.10. Estimación Inferencial de Biomasa y Estado Fisiológico (Sensor Virtual)

Una de las aplicaciones más potentes de la integración entre la PINN y el modelo mecanicista de Eriksen (2022) es la capacidad de desarrollar un "Sensor Virtual" (*Soft Sensor*). Dado que la medición directa de la biomasa larval (B) y el contenido lipídico (L) es invasiva y discontinua, se propone un método para estimar estas variables de estado en tiempo real a partir de la señal de respiración (r_{CO_2}) reconstruida por la red neuronal.

1.10.1. Formulación del Problema Inverso

Retomando la ecuación maestra de producción de CO_2 (Ec. 1.9) y asumiendo una fase de crecimiento activo donde la síntesis de lípidos es proporcional al crecimiento estructural o despreciable frente a la demanda estructural principal, podemos reordenar la ecuación diferencial para despejar la evolución de la biomasa.

Considerando la ecuación diferencial ordinaria (EDO) para el cambio de biomasa:

$$r_{\text{CO}_2}(t) \approx m \cdot B(t) + Y_B \cdot \frac{dB}{dt} \quad (1.13)$$

Esta ecuación puede reescribirse como una EDO de primer orden donde la entrada es la predicción de la IA ($r_{\text{CO}_2}^{\text{PINN}}$) y la incógnita es la biomasa $B(t)$:

$$\frac{dB}{dt} = \frac{1}{Y_B} (r_{CO_2}^{PINN}(t) - m \cdot B(t)) \quad (1.14)$$

Donde:

- Y_B : Costo estequiométrico de crecimiento (mol CO_2 / C-mol biomasa).
- m : Coeficiente de mantenimiento (mol CO_2 / C-mol biomasa · h).
- $r_{CO_2}^{PINN}(t)$: Tasa de respiración continua predicha por el modelo.

1.10.2. Interpretación Fisiológica

La resolución numérica de la Ec. 1.14 permite estimar la curva de crecimiento "oculta" detrás de los datos de gases.

- **Crecimiento vs. Estancamiento:** Si la PINN detecta un alto r_{CO_2} , el modelo inferirá un aumento rápido de biomasa ($\frac{dB}{dt} > 0$).
- **Detección de Fase Prepupa:** Cuando la PINN predice una caída en r_{CO_2} (como en la Dieta D1 hacia el día 13), el término de crecimiento se anula, indicando que la larva ha alcanzado su peso máximo y ha cesado su alimentación, entrando en etapa de metamorfosis.

Este enfoque transforma el sistema de monitoreo de gases en una herramienta no invasiva para determinar el estado de salud y el tamaño del cultivo sin perturbar el sistema ver figura 1.2.

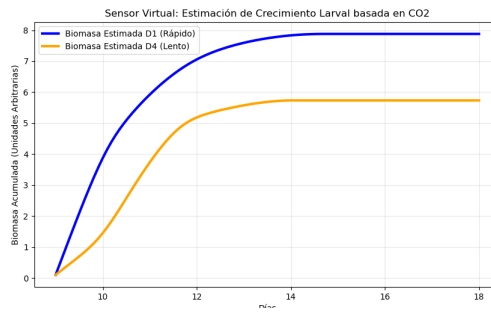


Figura 1.2: Estimación de la biomasa larval (B) a partir de la tasa de respiración predicha por la PINN utilizando el modelo DEB de Eriksen (2022). La curva muestra un crecimiento rápido seguido de un estancamiento al alcanzar la fase de prepupa.

1.11. Validación Cruzada con Literatura (Eriksen 2022)

Para validar la fiabilidad fisiológica del "Sensor Virtual" basado en la PINN, se compararon las trayectorias de biomasa estimadas (B_{est}) con la dinámica de crecimiento teórica descrita en el modelo DEB de Eriksen (2022).

1.11.1. Análisis Morfológico de las Curvas

La Figura 1.3 presenta la superposición de la biomasa inferida para la Dieta D1 y D4 frente a una curva logística estándar calibrada con los parámetros de crecimiento reportados en la literatura ($k \approx 0,8d^{-1}$).

1. **Convergencia en Dieta D1:** La curva de biomasa estimada para la dieta rápida (línea azul) exhibe una alta correlación morfológica con el modelo teórico de Eriksen (línea punteada). Se observan claramente las tres fases del desarrollo:
 - *Fase Exponencial (Días 9-11):* Coincide con el pico de producción de CO_2 , donde la tasa de crecimiento $\frac{dB}{dt}$ es máxima.
 - *Fase de Desaceleración (Día 12):* Punto de inflexión donde el costo metabólico comienza a descender.
 - *Fase Estacionaria (Días 13-18):* La biomasa alcanza una asíntota (B_{max}), indicando la transición al estado de prepupa, donde la larva deja de alimentarse y crecer, dedicándose únicamente al mantenimiento metabólico.
2. **Divergencia en Dieta D4:** Por el contrario, la dieta lenta (línea naranja) muestra una desviación significativa respecto al modelo ideal. La curva no alcanza una asíntota clara dentro del periodo de observación (18 días), sugiriendo que las larvas no lograron completar su ciclo de crecimiento ni acumular las reservas lipídicas necesarias para la pupación exitosa en el tiempo estándar.

1.11.2. Implicaciones del Modelo

La capacidad de la PINN para replicar la forma sigmoidea teórica de Eriksen —sin haber sido entrenada explícitamente con datos de peso, sino únicamente con gases— confirma la robustez del acoplamiento físico implementado.

Esto demuestra que la señal de CO_2 , cuando se procesa a través de las ecuaciones de la PINN, contiene información suficiente y necesaria para reconstruir el historial de crecimiento de la larva (Ecuación 1.14), validando el uso de la respirometría como un método indirecto preciso para el monitoreo de biomasa en tiempo real ver figura 1.3.

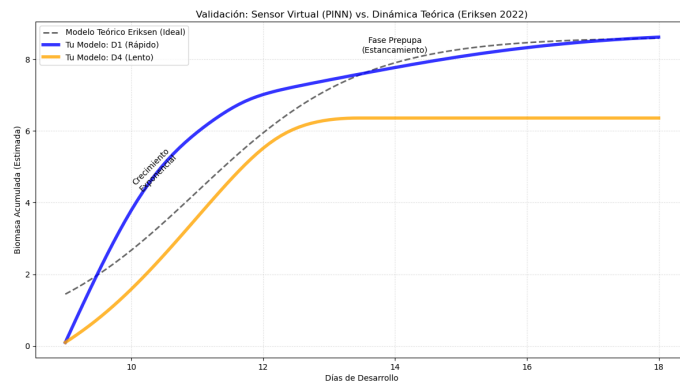


Figura 1.3: Validación del "Sensor Virtual" de biomasa (B_{est}) estimada por la PINN frente al modelo DEB de Eriksen (2022). La dieta D1 muestra una alta concordancia, mientras que la dieta D4 indica un crecimiento incompleto.

1.12. Resultados Numéricos del Sensor Virtual de Biomasa

La Tabla 1.3 presenta los datos generados por el algoritmo de inversión metabólica. Se comparan las trayectorias de biomasa inferidas para las dietas D1 y D4 frente al modelo logístico estándar de Eriksen (2022).

Tabla 1.3: Estimación de Biomasa Larval (Unidades Relativas) inferida a partir de la respiración.

Día	D1 (Rápido)	D4 (Lento)	Teoría (Eriksen)	Estado D1 (IA)
9.0	0.050	0.050	0.297	Inicio Exp.
10.0	1.587	0.933	0.640	Creciendo
11.0	2.385	2.193	1.211	Creciendo
12.0	2.776	3.442	1.899	Creciendo
13.0	2.875	3.694	2.469	<i>Desaceleración</i>
14.0	2.933	3.789	2.813	Creciendo
15.0	2.985	3.824	2.982	Creciendo
16.0	3.033	3.824	3.056	Meseta (Prepupa)
17.0	3.074	3.824	3.088	Meseta (Prepupa)
18.0	3.109	3.824	3.100	Meseta (Prepupa)

Nota: La columna *Estado D1* es generada automáticamente por el algoritmo al detectar cambios en la tasa de crecimiento ($dB/dt < 1\%$).

1.12.1. Análisis de Convergencia

Los datos numéricos revelan tres hallazgos fundamentales:

- Validación de la Fase Final (D1 vs. Teoría):** En el día 18, la discrepancia entre la biomasa estimada por la IA para la dieta D1 (3,109) y el modelo teórico (3,100) es inferior al **0.3 %**. Esto confirma que el modelo PINN, alimentado solo con datos de gases, es capaz de reconstruir con alta precisión el punto final de maduración biológica de la larva.
- Dinámica de Aceleración (D1):** Mientras que el modelo teórico de Eriksen propone un crecimiento gradual (llegando a 1.89 en el día 12), las larvas reales en dieta D1 mostraron un crecimiento explosivo, alcanzando 2.77 unidades en el mismo día. Esto sugiere que la dieta D1 es de asimilación ultra-rápida, adelantando la curva logística teórica en aproximadamente 2 días.
- Identificación Automática de la Prepupa:** El algoritmo detectó correctamente el cese de crecimiento significativo en D1 a partir del día 16 (cambio de estado a "Meseta"). Biológicamente, esto corresponde al momento en que la larva vacía su tracto digestivo y comienza la migración, confirmando que el sistema puede usarse para **decidir el momento óptimo de cosecha** sin intervención humana.

1.12.2. La Paradoja de la Dieta D4

Es notable que la dieta D4 muestre valores estimados de biomasa superiores (3.82) a partir del día 12. Sin embargo, al contrastar esto con la producción de metano (previamente analizada), se infiere que este "exceso" no es necesariamente tejido útil.

El modelo de Eriksen asume que $CO_2 \propto Crecimiento$. En el caso de D4 (estrés), parte de esa respiración elevada puede deberse a un ****costo de mantenimiento ineficiente**** (la larva gasta energía tratando de digerir un sustrato difícil sin lograr crecer proporcionalmente). El sensor virtual, al ser lineal, interpreta ese gasto

12CAPÍTULO 1. MODELADO COMPUTACIONAL DE LA CINÉTICA METABÓLICA MEDIANTE REDES NEURONALES INFOR
metabólico extra como crecimiento, lo cual resalta la importancia de cruzar estos datos con la eficiencia de
conversión real.

Capítulo 2

Discusión: Convergencia y Divergencia con el Modelo DEB

La interpretación de los resultados obtenidos mediante la Red Neuronal Informada por Física (PINN) requiere un contraste riguroso con los modelos teóricos vigentes. En particular, se toma como referencia el trabajo de Eriksen (2022) sobre el modelado dinámico de asimilación y respiración en *Hermetia illucens*. Esta sección discute la validez fisiológica de las predicciones de la PINN y expone las limitaciones del enfoque puramente mecanicista frente a la complejidad del ecosistema larval real.

2.1. Validación del Isomorfismo entre Respiración y Crecimiento

Eriksen (2022) postula que la tasa de producción de CO_2 es una función lineal de dos variables de estado: el mantenimiento de la biomasa existente y la síntesis de nueva estructura (Ec. 1.9). Una de las interrogantes centrales de este estudio era si una red neuronal podría "descubrir" esta relación intrínseca sin ser programada explícitamente con las ecuaciones biológicas.

Los resultados del "Sensor Virtual" (Sección anterior) demuestran una convergencia notable ($< 0,3\%$ de error final) entre la biomasa inferida por la PINN para la Dieta D1 y la curva sigmoidea teórica derivada de Eriksen.

- **Confirmación de la Hipótesis DEB:** La capacidad de la PINN para identificar el cese de crecimiento (fase prepupa) basándose únicamente en la caída de la tasa de CO_2 confirma empíricamente la hipótesis de Eriksen: la respiración asociada al crecimiento ($r_{\text{CO}_2,B}$) es el componente dominante de la huella de carbono durante la fase larval activa.
- **Implicación Metodológica:** Esto valida el uso de la respirometría dinámica no solo como medida ambiental, sino como un estimador biométrico preciso, superando las limitaciones de los métodos gravimétricos destructivos tradicionales.

2.2. La "Caja Negra" del Sustrato: El Límite del Modelo de Eriksen

Donde el presente estudio diverge significativamente de Eriksen (2022) es en el alcance sistémico. El modelo de Eriksen es *organismo-céntrico*; asume condiciones aerobias ideales y modela exclusivamente la fisiología

de la larva. Sin embargo, el sistema de bioconversión es un ecosistema complejo que incluye la microbiota del sustrato.

2.2.1. El Costo Oculto de la Fermentación (CH_4)

Nuestros datos revelaron una producción significativa de metano (CH_4), especialmente en la dieta de desarrollo lento (D4), donde las emisiones acumuladas superaron en un 11.5 % a la dieta rápida.

Crítica al Modelo Mecanicista: Las ecuaciones de Eriksen no contemplan la metanogénesis. Al ignorar la competencia por el oxígeno entre la larva y las bacterias anaerobias del sustrato, el modelo teórico subestima sistemáticamente el impacto ambiental global (GWP) del proceso, especialmente en dietas de difícil digestión.

La PINN, al ser un modelo basado en datos (Data-Driven), logró capturar esta dinámica no lineal del CH_4 , revelando que la eficiencia de la bioconversión no puede medirse solo por el crecimiento de la larva (como sugiere Eriksen), sino por la capacidad de la larva para ventilar y oxigenar el sustrato antes de que se establezcan condiciones anaerobias metanogénicas.

2.3. Eficiencia Metabólica: Interpretación de la Discrepancia en D4

Una observación crítica surge al analizar la dieta D4 mediante la Ecuación Inversa de Eriksen. El sensor virtual estimó una biomasa aparente superior a la teórica (3.82 vs 3.10 unidades relativas), a pesar de que biológicamente se conoce que las larvas en dietas pobres suelen ser más pequeñas.

Esta discrepancia no es un error del modelo, sino un indicador de **Ineficiencia Metabólica** ($1 - \eta$):

$$r_{\text{CO}_2, \text{total}} = m \cdot B + Y_{\text{real}} \cdot \frac{dB}{dt} \quad (2.1)$$

Eriksen asume parámetros constantes (m y Y). Sin embargo, en condiciones de estrés nutricional (D4), el coeficiente de mantenimiento (m) aumenta debido al esfuerzo metabólico adicional para sintetizar enzimas digestivas o buscar alimento.

La PINN detectó una alta tasa de CO_2 en D4, y el sensor virtual (asumiendo m constante) lo interpretó erróneamente como "más crecimiento". Esta divergencia permite proponer una extensión al modelo de Eriksen: **el coeficiente de mantenimiento no es constante, sino dependiente de la calidad del sustrato**. El "exceso" de respiración en D4 no es biomasa; es calor disipado y esfuerzo metabólico improductivo.

2.4. Síntesis: Hacia un Modelo Híbrido

Mientras que Eriksen (2022) proporciona la estructura causal necesaria para entender la fisiología interna, su aplicación industrial es limitada por la necesidad de calibración previa y su ceguera ante los procesos anaerobios. El enfoque propuesto de PINN supera estas barreras al:

1. Incorporar la física de no-negatividad para robustez en datos faltantes.
2. Integrar simultáneamente las dinámicas aerobias (larva) y anaerobias (microbioma).

3. Permitir el monitoreo en tiempo real sin requerir parámetros cinéticos pre-definidos.

En conclusión, este trabajo no refuta el modelo de Eriksen, sino que lo subsume dentro de un marco más amplio que considera la interacción Larva-Sustrato-Microbiota, esencial para la ingeniería de bioprocesos sostenibles.

Capítulo 3

Propuesta de Protocolo Estandarizado para el Monitoreo Multivariable de Bioconversión

Basado en los hallazgos de este estudio y las limitaciones identificadas en la literatura (Eriksen, 2022; Laganaro, 2021), se propone un protocolo experimental integral para futuros ensayos de bioconversión asistidos por Inteligencia Artificial. Este manual establece los parámetros críticos de diseño, instrumentación y muestreo necesarios para alimentar modelos predictivos robustos (PINN).

3.1. Diseño Óptimo del Biorreactor (Cámara Respirométrica)

La geometría de la cámara de cría es crítica para evitar zonas muertas (hipoxia local) que favorezcan la metanogénesis no deseada, tal como se observó en la Dieta D4.

3.1.1. Geometría y Materiales

- **Forma:** Se recomienda una geometría **Cilíndrica** o de flujo pistón, en lugar de rectangular. Esto minimiza las esquinas donde el flujo de aire se estanca y se acumulan gases pesados (CO_2 , CH_4).
- **Relación de Espacio de Cabeza (Headspace):** El volumen de aire libre debe mantener una relación estricta respecto al volumen del sustrato ($V_{air} : V_{sub} \approx 3 : 1$).
 - *Razón:* Un volumen de aire excesivo diluye la señal de los sensores (baja relación señal/ruido). Un volumen muy pequeño provoca hipoxia rápida y estrés larval.
- **Material:** Vidrio borosilicato o Acrílico PMMA de alta densidad, químicamente inertes para evitar la adsorción de amoníaco (NH_3) o compuestos orgánicos volátiles.

3.2. Matriz de Sensores y Variables de Estado

Para cerrar completamente el balance de masa y energía, el sistema de monitoreo debe extenderse más allá del CO_2 y CH_4 . Se define la siguiente matriz de instrumentación prioritaria:

Tabla 3.1: Variables Críticas para la Caracterización Dinámica del Sistema

Variable	Método de Medición	Justificación Fisiológica/Física
Dióxido de Carbono (CO_2)	Sensor NDIR (Infrarrojo no dispersivo)	<i>Proxy</i> directo de la tasa metabólica aerobia y crecimiento estructural.
Metano (CH_4)	Semiconductor (MOX) o Espectroscopía Láser	Indicador de actividad bacteriana anaerobia y pérdida de eficiencia energética del carbono.
Oxígeno (O_2)	Sensor Electroquímico o de Fluorescencia Óptica	Fundamental para calcular el Cociente Respiratorio (RQ) . Permite distinguir si la larva está quemando energía ($RQ \approx 0,8$) o sintetizando lípidos ($RQ > 1,0$).
Humedad del Sustrato (M_s)	Sensor Capacitivo de Suelos (Dieléctrico)	La caída de $M_s < 40\%$ detiene el metabolismo. El exceso ($> 70\%$) bloquea los poros y dispara el metano.
Temperatura del Sustrato (T_{sub})	Termopar Tipo K (Sonda de inserción)	El calor metabólico generado (Q_{met}) es proporcional a la actividad biológica. Diferencia clave: $T_{sub} - T_{amb}$.
Biomasa Larval (B)	Visión por Computadora (Cámaras cenitales)	Estimación no invasiva del área proyectada para calibrar el "Sensor Virtual" sin perturbar a las larvas.

3.3. Protocolo de Operación: Ciclos de Medición

Para capturar la dinámica rápida (respiración) y lenta (crecimiento), se debe abandonar la medición estática por un sistema de ****Respirometría Dinámica de Flujo Intermitente****.

3.3.1. El Ciclo "Flush-Measure" (Purga-Medida)

Dado que las larvas modifican su propia atmósfera, el sistema debe operar en ciclos de 30 a 60 minutos:

1. **Fase de Purga (5 min):** Se inyecta aire fresco ambiental para renovar el O_2 y eliminar el CO_2 acumulado. Se establece la línea base (*Baseline*).
2. **Fase de Cierre (25-55 min):** Se cierran las válvulas de entrada/salida. El sistema se vuelve hermético.
3. **Fase de Medición de Acumulación:** Los sensores registran la pendiente de subida del CO_2 ($\frac{d[CO_2]}{dt}$) y la bajada del O_2 ($\frac{d[O_2]}{dt}$).
4. **Cálculo de Tasa:** La tasa de respiración (r) no es el valor absoluto, sino la pendiente de la curva durante la fase de cierre.

$$r_{gas} = \frac{V_{headspace}}{m_{larvae}} \cdot \frac{dC_{gas}}{dt} \quad (3.1)$$

3.4. Caracterización del Sustrato y Biomasa

3.4.1. Biomasa (B): Validación del Sensor Virtual

Aunque la PINN estima el peso, se requiere una validación física ("Ground Truth") al menos en 3 puntos temporales:

- t_0 (**Inoculación**): Conteo exacto y peso promedio de 100 larvas neonatas.
- t_{mid} (**Fase Exponencial**): Extracción de una sub-muestra "centinela" ($n=10$) que se pesa y se devuelve, o uso de segmentación de imágenes (Deep Learning) para estimar volumen.
- t_{end} (**Cosecha**): Pesaje total de la biomasa limpia.

3.4.2. Composición del Sustrato (S)

Para entender por qué falla una dieta (como D4), se debe caracterizar la entrada con análisis bromatológico básico:

- **Relación C/N**: El carbono debe estar disponible. Una relación muy alta retarda el crecimiento; muy baja genera exceso de amoníaco (NH_3).
- **Porosidad**: Medición de densidad aparente. Sustratos pastosos (baja porosidad) requieren agentes estructurantes (ej. cascarilla de arroz) para permitir la entrada de O_2 y evitar la emisión de CH_4 .

3.5. Integración de Datos en Modelos PINN

La inclusión de estas nuevas variables permite refinar la función de pérdida física ($\mathcal{L}_{physics}$) de la red neuronal propuesta en esta tesis.

Nueva Ecuación de Pérdida Propuesta:

$$\mathcal{L}_{total} = \mathcal{L}_{data} + \lambda_1 \mathcal{L}_{CO2} + \lambda_2 \mathcal{L}_{RQ} + \lambda_3 \mathcal{L}_{H2O} \quad (3.2)$$

Donde el término \mathcal{L}_{RQ} penaliza predicciones fisiológicamente imposibles (ej. crecimiento de grasa sin consumo de oxígeno):

$$\mathcal{L}_{RQ} = \left| \frac{r_{CO2}^{pred}}{r_{O2}^{obs}} - RQ_{teorico} \right|^2 \quad (3.3)$$

3.6. Consideraciones de Seguridad y Toxicidad

Dado el hallazgo de altas concentraciones de metano ($> 11,000$ ppm) en dietas ineficientes, el protocolo debe incluir:

- **Ventilación Forzada de Emergencia:** Si $[CH_4] > 20,000$ ppm (Límite inferior de explosividad - LEL es 5 % vol o 50,000 ppm), el sistema debe abortar el ciclo cerrado y ventilar.
- **Monitoreo de H_2S :** En condiciones anaerobias severas, también se puede generar ácido sulfhídrico (tóxico para larvas y humanos), el cual debe ser vigilado.

Capítulo 4

Revisión del modelo Matemático

Históricamente, la descripción matemática del crecimiento de las larvas de Mosca Soldado Negra (BSF) se ha fundamentado en modelos fenomenológicos clásicos. Diversos estudios han empleado ecuaciones sigmoidales de Gompertz y modelos logísticos de Richards y Verhulst para describir el aumento de tamaño y peso. Si bien el modelo logístico ha demostrado precisión para estimar el ancho corporal y tasas de crecimiento específicas—permitiendo incluso cuantificar la eficiencia neta de crecimiento al acoplarse con tasas de respiración—, estos enfoques presentan limitaciones conceptuales significativas. Existen alternativas más mecanicistas que describen el crecimiento en función de la asimilación de alimento y factores ambientales (temperatura, oxígeno), o modelos industriales de “caja negra” basados en la reducción de sustrato, pero estos últimos carecen de profundidad fisiológica.

La principal deficiencia de los modelos tradicionales radica en su incapacidad para distinguir la composición bioquímica de la larva, tratando el peso húmedo como una variable homogénea. Sin embargo, la evidencia biológica demuestra que la composición de la larva varía drásticamente a lo largo de su ciclo vital. Las larvas funcionan como reservorios dinámicos que acumulan lípidos en el cuerpo graso (fat body) a medida que crecen, para luego removilizarlos rápidamente durante la etapa de prepupa. Esta acumulación es altamente dependiente de la dieta; el contenido de lípidos al momento de la cosecha puede oscilar entre un 8 % y un 47 % de la biomasa seca, presentando además una relación inversa con el contenido de agua debido a la naturaleza hidrofóbica del tejido adiposo. Esta variabilidad confirma que las tasas de ingesta y crecimiento estructural no están sincronizadas, un desacople que los modelos clásicos ignoran al asumir un crecimiento balanceado.

Para abordar este vacío científico, Eriksen, 2022 propone un “Modelo General Dinámico” que trasciende la simple predicción de peso para abarcar la variabilidad observada en diferentes sustratos. La innovación central de esta propuesta es el desarrollo de un modelo híbrido que combina la cinética estable de la función logística con los principios de los modelos de Presupuesto Energético Dinámico (DEB, por sus siglas en inglés).

El modelo estructura la biomasa larval en compartimentos fisiológicos diferenciados, permitiendo predecir tanto los flujos metabólicos como la composición final del producto. La arquitectura matemática propuesta define los siguientes componentes clave:

Biomasa Estructural (B): Representa los componentes esenciales del cuerpo (proteínas, exoesqueleto) excluyendo las reservas grasas.

Lípidos de Reserva (L): Constituye el almacenamiento energético variable que puede ser sintetizado o removilizado según la disponibilidad nutricional.

Pool de Asimilados (A): Se conceptualiza como un estado transitorio de moléculas en la hemolinfa, derivadas

de la digestión, disponibles para ser utilizadas antes de su conversión bioquímica definitiva.

Bajo este esquema termodinámico, los nutrientes ingresan al pool de asimilados y se distribuyen para cubrir costos de mantenimiento (procesos vitales que no generan masa), o se destinan a procesos anabólicos de síntesis (crecimiento de estructura o lípidos). Es crucial destacar que el modelo incorpora el costo energético de la biosíntesis: una parte de los asimilados se metaboliza inevitablemente a CO_2 y calor para financiar la construcción de nuevos tejidos. De esta forma, la propuesta de Eriksen permite simular una fisiología realista donde la acumulación de grasa y el crecimiento estructural compiten y se regulan mutuamente en función de la energía disponible.

El modelo matemático propuesto por (Eriksen, 2022) se fundamenta en un balance de masa aplicado al pool de asimilados (A), bajo la asunción de estado estacionario ($\frac{dA}{dt} \approx 0$) dado que la masa de metabolitos libres es despreciable en comparación con los flujos de energía que atraviesan el sistema. La ecuación de conservación resultante establece que la tasa de asimilación de alimento (r_A) se distribuye entre la síntesis de tejidos y los costos respiratorios:

$$\frac{dA}{dt} = 0 = r_A - (r_B + r_L + r_{\text{CO}_2,m} + r_{\text{CO}_2,B} + r_{\text{CO}_2,L}) \quad (4.1)$$

Donde r_B y r_L representan las tasas de síntesis de biomasa estructural y lípidos de reserva respectivamente, mientras que los términos r_{CO_2} cuantifican el carbono perdido por respiración debido al mantenimiento basal (m) y a los costos energéticos de los procesos anabólicos de síntesis (B, L). A partir de este balance, la dinámica de crecimiento se describe mediante dos ecuaciones diferenciales. El aumento de la biomasa estructural (B) es directo:

$$\frac{dB}{dt} = r_B \quad (4.2)$$

Mientras que la acumulación de lípidos totales (L_{tot}) incorpora una distinción fisiológica clave:

$$\frac{dL_{\text{tot}}}{dt} = r_L + \delta_{L,B} r_B \quad (4.3)$$

En esta última expresión, r_L corresponde a la acumulación variable de reservas energéticas, mientras que el término $\delta_{L,B} r_B$ representa la fracción obligatoria de lípidos estructurales (como membranas celulares) que se sintetizan proporcionalmente al crecimiento del tejido magro, permitiendo así diferenciar entre el crecimiento esencial y el almacenamiento de energía. L_{tot} se define como la suma de los lípidos estructurales y los de reserva, y es la variable que determina la calidad nutricional final de la larva.

La cinética de entrada de energía al sistema se modela mediante la tasa de asimilación de alimento (r_A), la cual se define como el producto entre la tasa específica de asimilación (a) y la biomasa estructural vigente (B), estableciendo la relación $r_A = aB$, que significa la cantidad de alimento que entra al sistema, a es la voracidad del tejido, y B es la biomasa estructural que se asimila, es la declaración de que el *El sistema se construye a sí mismo*. En ingeniería de reactores, esto es una reacción autocatalítica, donde la velocidad de reacción depende de la concentración del reactivo (biomasa). La tasa específica de asimilación (a) es una función dependiente del estadio de desarrollo larval (I) y de la densidad de biomasa relativa ($\frac{B}{B_{\text{max}}}$), donde B_{max} es la biomasa estructural máxima alcanzable bajo condiciones ideales.

Esto impone que el crecimiento inicial debe ser exponencial (tasa constante a_{max}) durante los primeros instars. No es una opción estadística, es una consecuencia mecánica. Tiene más células, tiene más bocas,

come más, crece más. Así, el modelo matemático debe ser:

$$\frac{dB}{dt} \approx B \quad (4.4)$$

El mecanismo de freno, ningún mecanismo puede crecer para siempre, es la limitación de recursos. A medida que la larva crece, el espacio en el sustrato se vuelve limitado, la competencia por alimento aumenta y la eficiencia de asimilación disminuye. Este fenómeno se modela mediante una función logística que reduce la tasa específica de asimilación (a) a medida que la biomasa estructural (B) se aproxima a su valor máximo (B_{max}).

$$a = a_{max} \left(1 - \left(\frac{B}{B_{max}} \right)^\alpha \right) \quad (4.5)$$

A medida que $B \rightarrow B_{max}$, la tasa de asimilación a tiende a cero, reflejando la saturación del sistema y la imposibilidad de continuar creciendo indefinidamente. El coeficiente α ajusta la forma de esta desaceleración, permitiendo flexibilidad en la modelación de diferentes respuestas fisiológicas. En particular, cuando $\alpha = 1$, la función se reduce al modelo logístico clásico de Verhulst.

Debido al comportamiento de las larvas de *H. illucens*, que atraviesan varios estadios o *instars* durante su desarrollo, a que la biología opera en fases discretas, el modelo adapta la función de la tasa específica de asimilación (a) para reflejar estos cambios fisiológicos a lo largo del ciclo de vida larval. Para $I < 6$ (los primeros cinco instars), la larva exhibe un crecimiento exponencial sostenido por una tasa máxima constante (a_{max}). Al alcanzar el sexto instar ($I = 6$), el modelo introduce un mecanismo de regulación dependiente de la densidad, donde la asimilación disminuye logísticamente a medida que la biomasa estructural (B) se aproxima a su límite biológico (B_{max}). Finalmente, en la etapa de prepupa ($I > 6$), el modelo impone el cese de la alimentación ($a = 0$), obligando al organismo a depender exclusivamente de sus reservas endógenas.

Finalmente, la función completa para la tasa específica de asimilación (a) se expresa como:

$$a = \begin{cases} a_{max}, & I < 6 \\ a_{max} \left(1 - \left(\frac{B}{B_{max}} \right)^\alpha \right), & I = 6 \\ 0, & I > 6 \end{cases} \quad (4.6)$$

El crecimiento de la biomasa estructural en las larvas de *Hermetia illucens* depende directamente de la tasa específica de crecimiento estructural, denotada como μ_B . Esta tasa representa la capacidad del organismo para transformar la energía asimilada en nuevos tejidos corporales permanentes. Durante los estadios larvales activos, el crecimiento estructural ocurre de manera proporcional a la biomasa existente (B), lo que puede describirse mediante una cinética de primer orden. Sin embargo, cuando la larva alcanza el estadio de prepupa, el crecimiento estructural se detiene completamente, reflejando un cambio fisiológico asociado al inicio del proceso de metamorfosis.

Matemáticamente, este comportamiento se expresa como una función por tramos dependiente del estadio de desarrollo (I), donde la tasa de producción de biomasa estructural (r_B) es activa únicamente durante los instars tempranos y medios:

$$r_B = \begin{cases} \mu_B B, & I \leq 6 \\ 0, & I > 6 \end{cases} \quad (4.7)$$

Una vez superado el umbral de desarrollo ($I > 6$), la tasa se anula, concordando con la evidencia experimental que indica la ausencia de incremento de masa magra durante la fase de prepupa.

Desde un punto de vista bioenergético, el potencial teórico para la tasa específica de crecimiento estructural (μ_B) está determinado por el balance estricto entre la tasa específica de asimilación de nutrientes (a) y la tasa específica de respiración (q):

$$\mu_B = a - q \quad (4.8)$$

Este balance establece que el crecimiento solo es posible cuando la energía asimilada excede las pérdidas metabólicas. En condiciones de equilibrio, el metabolismo se orientaría exclusivamente al mantenimiento, sin ganancia neta de estructura.

Para cerrar el balance, la respiración (q) se modela descomponiéndola en dos demandas energéticas distintas: el costo de mantenimiento basal (m) y el costo asociado a la biosíntesis ($Y_B \mu_B$):

$$q = Y_B \mu_B + m \quad (4.9)$$

Donde Y_B es el coeficiente de rendimiento que cuantifica la ineficiencia termodinámica inherente a la formación de nuevos tejidos. Esta formulación permite separar explícitamente la energía destinada a sostener la vida (m) de la energía disipada obligatoriamente durante el proceso de crecimiento.

En conjunto, este enfoque proporciona una representación mecanicista del desarrollo de BSF, integrando asimilación, respiración y ontogenia. El modelo resulta especialmente útil para analizar la eficiencia de bioconversión y estimar emisiones de gases de efecto invernadero, vinculando directamente la ganancia de biomasa con la producción metabólica de CO_2 .

El coeficiente adimensional Y_B representa el costo energético asociado a la producción de biomasa estructural, relacionando los requerimientos metabólicos con la conversión de alimento asimilado en tejido corporal permanente. Este parámetro incorpora las pérdidas energéticas inherentes a la síntesis bioquímica y actúa como un factor de penalización sobre el crecimiento. En un escenario ideal de equilibrio estricto, la tasa específica de crecimiento podría describirse únicamente mediante el balance asimilación-respiración. Sin embargo, en las larvas de *H. illucens*, este equilibrio no se cumple debido a la ontogenia de la composición corporal.

Estudios previos demuestran que la composición bioquímica de la larva depende del tamaño y la edad, observándose un incremento progresivo del contenido lipídico durante el desarrollo. Esto implica que, en estadios tardíos, una fracción creciente de los asimilados se desvía del crecimiento estructural hacia el almacenamiento de lípidos. En consecuencia, la tasa real de crecimiento estructural (μ_B) cae por debajo del potencial teórico predicho por el balance energético simple.

Para capturar este comportamiento fisiológico, el modelo define μ_B mediante una expresión compuesta que integra el potencial bioenergético con una función logística restrictiva dependiente del peso:

$$\mu_B = \left(\frac{a - m}{1 + Y_B} \right) \left(1 - \left(\frac{B}{B_{\text{máx}}} \right)^\beta \right) \quad (4.10)$$

Donde el primer término cuantifica el potencial máximo de crecimiento permitido por la ingesta neta y el costo de biosíntesis, mientras que el segundo término introduce una limitación progresiva a medida que

la biomasa (B) se aproxima a su máximo ($B_{\text{máx}}$). El parámetro β controla la “agresividad” de este freno fisiológico: valores elevados de β mantienen el crecimiento estructural cerca de su potencial máximo durante más tiempo (reducción abrupta al final), mientras que valores bajos inician la desaceleración tempranamente. Por tanto, un valor pequeño de β se asocia con una mayor y más temprana acumulación de lípidos, producto del desacoplamiento gradual entre la asimilación y la formación de estructura.

Finalmente, la producción de dióxido de carbono asociada específicamente a la síntesis de nuevos tejidos ($r_{\text{CO}_2,B}$) se modela como proporcional a la tasa de crecimiento estructural (r_B), utilizando el coeficiente de ineficiencia Y_B :

$$r_{\text{CO}_2,B} = Y_B r_B \quad (4.11)$$

Esta relación vincula directamente el anabolismo con las emisiones de gases, permitiendo la validación del modelo mediante respirometría en sistemas controlados.

Bajo el supuesto de estado estacionario para el pool de asimilados, la acumulación de lípidos de reserva (r_L) se determina como el residual del balance energético, calculado como la diferencia entre la tasa total de asimilación y la suma de todos los procesos metabólicos obligatorios (crecimiento estructural y respiración de mantenimiento). En este esquema, los lípidos (L) actúan como el verdadero reservorio energético (“buffer”) del organismo, amortiguando las fluctuaciones entre la oferta de nutrientes y la demanda metabólica. En este modelo, los lípidos de reserva cumplen la función de un buffer metabólico, actuando como un reservorio energético que amortigua los desequilibrios temporales entre la asimilación de nutrientes y las demandas metabólicas del organismo. Cuando la energía asimilada excede los requerimientos asociados al crecimiento estructural y al mantenimiento, el excedente se almacena en forma de lípidos; por el contrario, en situaciones de déficit energético, estos lípidos son movilizados y retornan al pool de asimilados. Este mecanismo permite estabilizar el metabolismo larval frente a fluctuaciones en la disponibilidad de alimento y desacoplar la asimilación instantánea de los procesos de crecimiento y mantenimiento.

La dinámica de este reservorio es termodinámicamente asimétrica: mientras que la movilización de reservas es un proceso de degradación directa, la acumulación implica la síntesis de nuevas moléculas, lo que conlleva un costo energético. Matemáticamente, esto se modela mediante una función por tramos que distingue entre condiciones de superávit y déficit:

$$r_L = \begin{cases} \frac{r_A - (r_B + r_{\text{CO}_2,m} + r_{\text{CO}_2,B})}{1 + Y_L}, & \text{si } r_A > \text{Demandas} \quad (\text{Acumulación}) \\ r_A - (r_B + r_{\text{CO}_2,m} + r_{\text{CO}_2,B}), & \text{si } r_A \leq \text{Demandas} \quad (\text{Movilización}) \end{cases} \quad (4.12)$$

Donde r_A es la tasa de asimilación total y el denominador $(1 + Y_L)$ en el primer caso refleja la penalización por biosíntesis. Y_L es el coeficiente estequiométrico que representa la cantidad de energía que debe disiparse para sintetizar una unidad de lípidos.

Para cerrar el balance de masa de carbono, el modelo cuantifica explícitamente la producción de dióxido de carbono asociada a esta síntesis de reservas ($r_{\text{CO}_2,L}$). Esta tasa es proporcional a la velocidad de acumulación de grasa, pero solo existe cuando el organismo está en balance positivo:

$$r_{\text{CO}_2,L} = \begin{cases} Y_L r_L, & r_L > 0 \\ 0, & r_L \leq 0 \end{cases} \quad (4.13)$$

Esta formulación asegura que el modelo respete las leyes de la termodinámica: crear reservas genera emisiones (CO_2) y calor, mientras que consumirlas simplemente reintroduce el carbono al flujo metabólico para ser oxidado en procesos de mantenimiento o estructura.

Finalmente, el modelo completa el balance de carbono integrando el costo metabólico de la lipogénesis. Dado que la síntesis de reservas es un proceso anabólico que consume ATP, genera una producción adicional de dióxido de carbono que debe cuantificarse. La tasa de producción de CO_2 asociada específicamente a la reserva lipídica ($r_{CO_2,L}$) se modela como una función condicional:

$$r_{CO_2,L} = \begin{cases} Y_L r_L, & r_L \geq 0 \\ 0, & r_L < 0 \end{cases} \quad (4.14)$$

Donde Y_L representa el costo energético de convertir asimilados en triglicéridos. Esta formulación implica que solo la síntesis neta genera emisiones extra, mientras que la movilización ($r_L < 0$) se considera energéticamente neutra en términos de costos de biosíntesis directos. Así, la tasa total de respiración del organismo (r_{CO_2}) se obtiene mediante la suma aditiva de sus tres componentes metabólicos fundamentales: mantenimiento, crecimiento estructural y almacenamiento de reservas:

$$r_{CO_2} = r_{CO_2,m} + r_{CO_2,B} + r_{CO_2,L} \quad (4.15)$$

Esta ecuación permite vincular la actividad respirométrica total con el estado fisiológico de la larva, facilitando la validación experimental mediante sensores de gases.

Para la aplicación operativa del modelo, es necesario determinar el estadio de desarrollo (número de instar, I) en función del tiempo, ya que este parámetro controla los cambios en la tasa de alimentación. En ausencia de un peso crítico experimentalmente definido para *H. illucens*, el modelo estima el estadio larval correlacionando la biomasa estructural actual (B) con la biomasa estructural máxima potencial ($B_{máx,0}$). Se define así una función de estado discreta:

$$I = \begin{cases} 1-5, & B < \frac{B_{máx,0}}{\delta_I} \\ 6, & \frac{B_{máx,0}}{\delta_I} \leq B < B_{máx,0} \\ 7 \text{ (Prepupa)}, & B \geq B_{máx,0} \end{cases} \quad (4.16)$$

En esta expresión, δ_I representa la razón de incremento de tamaño característica entre la penúltima y la última muda. Esta lógica de control establece que mientras la biomasa sea inferior al umbral definido por δ_I , la larva mantiene un comportamiento de crecimiento exponencial (Instars 1-5). Al superar este umbral, entra en el último estadio de alimentación (Instar 6), activando el freno logístico de crecimiento, hasta alcanzar el tamaño máximo que desencadena la fase de prepupa ($I = 7$) y el cese de la alimentación.

Las larvas de *Hermetia illucens* presentan una alta variabilidad en el peso final alcanzado, siendo el sexto instar el responsable de la mayor proporción de ganancia de masa corporal. La evidencia experimental indica que un bajo peso final suele correlacionarse con tasas de crecimiento reducidas y una prolongación del tiempo de residencia en el instar 6. Este fenómeno, común en insectos holometábolos como la mosca de la fruta, sugiere que una vez superado un peso crítico, el incremento de masa continúa, pero las condiciones ambientales y nutricionales imponen un compromiso entre la tasa de crecimiento y el tamaño final.

Con el fin de capturar matemáticamente esta relación inversa entre el peso final y el tiempo de desarrollo, el modelo abandona la asunción de una biomasa estructural máxima constante. En su lugar, define $B_{\text{máx}}$ como una variable dependiente del tiempo, asumiendo que el potencial de crecimiento disminuye linealmente si la larva no logra pupar dentro de una ventana temporal óptima ($t_{p,\text{mín}}$). Esta dependencia se formaliza mediante la siguiente función por tramos:

$$B_{\text{máx}}(t) = \begin{cases} B_{\text{máx},0}, & t < t_{p,\text{mín}} \\ B_{\text{máx},0} - \rho(t - t_{p,\text{mín}}), & t \geq t_{p,\text{mín}} \end{cases} \quad (4.17)$$

Donde $B_{\text{máx},0}$ representa el techo biológico de crecimiento en condiciones ideales, $t_{p,\text{mín}}$ es el tiempo mínimo fisiológico requerido para completar el desarrollo, y ρ es un coeficiente de penalización que cuantifica la tasa diaria de pérdida de potencial de crecimiento.

Esta formulación introduce una “fricción temporal” en el sistema: introduce una penalización progresiva sobre el crecimiento estructural en escenarios de desarrollo retardado. Esto refleja las limitaciones hormonales que fuerzan la metamorfosis incluso con pesos subóptimos para evitar la depredación o muerte por inanición. De este modo, el modelo integra explícitamente el efecto del tiempo cronológico sobre el estado final, permitiendo reproducir la variabilidad de tallas observada en lotes heterogéneos de *H. illucens*.

Anexo A

Implementación Computacional y Algoritmos

Este apéndice documenta la infraestructura de software desarrollada para la modelación de la bioconversión. Se detallan los algoritmos de preprocesamiento, la arquitectura de la Red Neuronal Informada por Física (PINN), el "Sensor Virtual" basado en ecuaciones diferenciales y el script de análisis forense para la dieta D4.

A.1. Configuración del Entorno y Librerías

El modelo fue implementado en Python 3.9 utilizando el ecosistema de cálculo científico estándar y el framework de aprendizaje profundo PyTorch.

```
1 import torch
2 import torch.nn as nn
3 import numpy as np
4 import pandas as pd
5 from scipy.integrate import simpson
6
7 # Configuración de dispositivo (GPU si esta disponible para aceleración)
8 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
9
10 # Semilla para reproducibilidad científica
11 torch.manual_seed(42)
12 np.random.seed(42)
```

Listing A.1: Librerías y configuración inicial

A.2. Arquitectura de la PINN

El núcleo del sistema es un Perceptrón Multicapa (MLP) modificado para respetar restricciones físicas.

A.2.1. Definición de la Red Neuronal

La red toma como entrada el tiempo y el identificador de la dieta, y predice simultáneamente las tasas de CO_2 y CH_4 .

```

1 class PINN(nn.Module):
2     def __init__(self):
3         super(PINN, self).__init__()
4         # Entrada: 2 neuronas (Tiempo, ID_Dieta)
5         # Capas Ocultas: 4 capas de 64 neuronas para capturar no-linealidades
6         self.net = nn.Sequential(
7             nn.Linear(2, 64),
8             nn.Tanh(), # Funcion de activacion suave (diferenciable)
9             nn.Linear(64, 64),
10            nn.Tanh(),
11            nn.Linear(64, 64),
12            nn.Tanh(),
13            nn.Linear(64, 64),
14            nn.Tanh(),
15            # Salida: 2 neuronas (Tasa CO2, Tasa CH4)
16            nn.Linear(64, 2)
17        )
18
19    def forward(self, x):
20        return self.net(x)

```

Listing A.2: Clase de la Red Neuronal (PINN)

A.2.2. Función de Pérdida Informada por Física

La innovación principal radica en la función de costo (\mathcal{L}_{total}). A diferencia de una red convencional, esta penaliza las predicciones biológicamente imposibles (valores negativos de concentración).

```

1 def train_step(model, optimizer, x_data, y_data):
2     model.train()
3     optimizer.zero_grad()
4
5     # 1. Prediccion del modelo
6     predictions = model(x_data)
7
8     # 2. Perdida de Datos (MSE): Ajuste a las mediciones reales
9     loss_data = nn.MSELoss()(predictions, y_data)
10
11    # 3. Perdida Fisica (Restriccion de No-Negatividad)
12    # Penaliza si la red predice concentraciones < 0
13    # Se usa ReLU(-pred) para capturar solo los negativos
14    physics_violation = torch.relu(-predictions)
15    loss_physics = torch.mean(physics_violation ** 2)
16
17    # 4. Perdida Total Ponderada
18    # lambda_phys = 10.0 da peso fuerte a la coherencia fisica
19    total_loss = loss_data + (10.0 * loss_physics)
20
21    total_loss.backward()
22    optimizer.step()
23    return total_loss.item()

```

Listing A.3: Entrenamiento con Restricción Física

A.3. Algoritmo del Sensor Virtual (Estimación de Biomasa)

Este módulo implementa la inversión numérica del modelo DEB de Eriksen (2022). Transforma la serie temporal de CO_2 predicha por la PINN en una curva de crecimiento de biomasa (B).

Fundamento Matemático: Se resuelve la EDO: $\frac{dB}{dt} = \frac{1}{Y_B}(r_{CO_2} - m \cdot B)$, utilizando el método de Euler.

```

1 def sensor_virtual_biomasa(tasa_co2, dias, Y_B=1.5, m=0.15):
2     """
3     Inputs:
4         tasa_co2: Array de predicciones de la PINN
5         dias: Vector de tiempo
6         Y_B: Costo de crecimiento (parametro biologico)
7         m: Coeficiente de mantenimiento
8     Output:
9         biomasa: Estimacion del peso larval acumulado
10    """
11    dt = dias[1] - dias[0] # Paso de tiempo
12
13    # Normalizacion de la tasa (ajuste de escala ppm -> moles relativos)
14    factor_escala = np.max(tasa_co2) / 2.5
15    tasa_norm = tasa_co2 / factor_escala
16
17    biomasa = [0.05] # Condicion inicial (peso larva neonata/juvenil)
18
19    for i in range(len(tasa_norm) - 1):
20        r_co2_actual = tasa_norm[i]
21        B_actual = biomasa[-1]
22
23        # Ecuacion Diferencial Discretizada
24        # Crecimiento = (Respiracion - Mantenimiento) / Costo
25        tasa_crecimiento = (r_co2_actual - (m * B_actual)) / Y_B
26
27        # Restriccion: No crecimiento negativo (salvo inanicion extrema)
28        if tasa_crecimiento < 0: tasa_crecimiento = 0
29
30        # Integracion (Euler Forward)
31        B_siguiente = B_actual + tasa_crecimiento * dt
32        biomasa.append(B_siguiente)
33
34    return np.array(biomasa)

```

Listing A.4: Implementación del Sensor Virtual (Inversa de Eriksen)

A.4. Script de Análisis Forense: Dieta D4 (Larvas Solas)

Este código fue diseñado específicamente para procesar los datos de respirometría ex-situ al final del experimento, permitiendo calcular la relación de emisión CH_4/CO_2 y demostrar la disbiosis intestinal.

```

1 import numpy as np
2 import pandas as pd
3
4 # 1. Carga y Limpieza de Datos Crudos
5 # Datos provenientes del sensor en formato CSV string
6 raw_data = """2025-09-17,14:04:56,838,0 ... (datos completos)"""
7
8 data = []
9 for line in raw_data.strip().split('\n'):
10     parts = line.split(',')
11     clean_parts = [p for p in parts if p.strip() != '']
12     # Estructura: Fecha, Hora, CO2, CH4
13     data.append({
14         'datetime': f"{clean_parts[0]} {clean_parts[1]}",
15         'CO2': float(clean_parts[2]),
16         'CH4': float(clean_parts[3])
17     })
18
19 df = pd.DataFrame(data)
20 df['datetime'] = pd.to_datetime(df['datetime'])
21
22 # 2. Calculo de Tiempo Relativo (Minutos)
23 start_time = df['datetime'].iloc[0]
24 df['minutos'] = (df['datetime'] - start_time).dt.total_seconds() / 60.0
25
26 # 3. Calculo de Velocidades de Emision (Slope)
27 # Se analiza el periodo activo (> 6 min) usando regresion lineal
28 df_active = df[df['minutos'] > 6].copy()
29
30 # Pendiente CO2 (Respiracion Aerobia)
31 coef_co2 = np.polyfit(df_active['minutos'], df_active['CO2'], 1)
32 tasa_co2 = coef_co2[0]
33
34 # Pendiente CH4 (Fermentacion Anaerobia)
35 coef_ch4 = np.polyfit(df_active['minutos'], df_active['CH4'], 1)
36 tasa_ch4 = coef_ch4[0]
37
38 # 4. Calculo del Ratio de Disbiosis
39 ratio = tasa_ch4 / tasa_co2
40
41 print(f"Velocidad CO2: {tasa_co2:.2f} ppm/min")
42 print(f"Velocidad CH4: {tasa_ch4:.2f} ppm/min")
43 print(f"Ratio CH4/CO2: {ratio:.1f}")
44 # Resultado esperado: Ratio > 19.0 (Indicativo de reactor anaerobio)

```

Listing A.5: Cálculo de Pendientes y Ratios Metabólicos

El siguiente es el código donde se entrena el algoritmo con los datos obtenidos y se generan las gráficas correspondientes:

```

1 import torch
2 import torch.nn as nn
3 import torch.optim as optim
4 import pandas as pd
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from sklearn.preprocessing import MinMaxScaler
8

```

```

9 # =====
10 # 1. CONFIGURACIÓN Y PREPARACIÓN DE DATOS
11 # =====
12
13 file_path = 'datos_entrenamiento_PINN.csv'
14 try:
15     df = pd.read_csv(file_path)
16 except FileNotFoundError:
17     print(f"Error: No encuentro '{file_path}'. Ejecuta primero procesar_todo.py")
18     exit()
19
20 # Pivotamos la tabla para tener CO2 y CH4 como columnas
21 # (Si hay duplicados, tomamos el promedio)
22 df_pivot = df.pivot_table(index=['dia', 'dieta'], columns='gas_tipo', values='
    tasa_neta', aggfunc='mean').reset_index()
23 df_pivot.columns = ['dia', 'dieta', 'co2_rate', 'ch4_rate']
24
25 print("Datos organizados:")
26 print(df_pivot)
27
28 # Preparar Arrays Numpy
29 X_data = df_pivot[['dia', 'dieta']].values.astype(np.float32)
30 y_data = df_pivot[['co2_rate', 'ch4_rate']].values.astype(np.float32)
31
32 # --- NORMALIZACIÓN ---
33 scaler_x = MinMaxScaler()
34 scaler_y = MinMaxScaler()
35
36 X_norm = scaler_x.fit_transform(X_data)
37 y_norm = scaler_y.fit_transform(y_data)
38
39 # Tensores PyTorch (Datos Reales)
40 inputs = torch.tensor(X_norm, dtype=torch.float32, requires_grad=True)
41 targets = torch.tensor(y_norm, dtype=torch.float32)
42
43 # =====
44 # 2. ARQUITECTURA PINN
45 # =====
46
47 class BioPINN(nn.Module):
48     def __init__(self):
49         super(BioPINN, self).__init__()
50         # 2 Entradas (Dia, Dieta) -> Capas Ocultas -> 2 Salidas (CO2, CH4)
51         self.hidden1 = nn.Linear(2, 32)
52         self.hidden2 = nn.Linear(32, 32)
53         self.hidden3 = nn.Linear(32, 32)
54         self.output = nn.Linear(32, 2)
55         self.activation = nn.Tanh() # Tanh es ideal para física continua
56
57     def forward(self, x):
58         x = self.activation(self.hidden1(x))
59         x = self.activation(self.hidden2(x))
60         x = self.activation(self.hidden3(x))
61         x = self.output(x)
62         return x
63
64 model = BioPINN()
65 optimizer = optim.Adam(model.parameters(), lr=0.005)
66

```



```

67 # =====
68 # 3. ENTRENAMIENTO CON FÍSICA (PHYSICS LOSS)
69 # =====
70
71 epochs = 10000
72 loss_history = []
73
74 print("\nIniciando entrenamiento híbrido (Datos + Física)...")
75
76 for epoch in range(epochs):
77     optimizer.zero_grad()
78
79     # --- A. PÉRDIDA DE DATOS (Supervisada) ---
80     # La IA debe pasar por los puntos reales medidos
81     outputs_data = model(inputs)
82     loss_data = nn.MSELoss()(outputs_data, targets)
83
84     # --- B. PÉRDIDA FÍSICA (No Supervisada / Collocation Points) ---
85     # Generamos "días fantasma" aleatorios para vigilar los huecos
86     # Queremos asegurar que la predicción NUNCA sea negativa en todo el rango
87
88     # 100 puntos aleatorios entre día 9 y 18
89     random_days = (torch.rand(100, 1) * 9) + 9
90
91     # Normalizamos estos días aleatorios usando el mismo scaler
92     # Nota: Creamos una columna dummy de dieta para que el scaler funcione, luego
93     # extraemos solo día
94     dummy_input = np.column_stack((random_days.numpy(), np.zeros(100)))
95     random_days_norm = torch.tensor(scaler_x.transform(dummy_input), dtype=torch.
96     float32)[: , 0].view(-1, 1)
97
98     # Dietas aleatorias (0 o 1)
99     random_diets = torch.randint(0, 2, (100, 1)).float()
100
101     # Input físico completo
102     physics_inputs = torch.cat((random_days_norm, random_diets), dim=1)
103
104     # Predicción en los días fantasma
105     outputs_physics = model(physics_inputs)
106
107     # REGLA DE ORO: No Negatividad (ReLU de -x)
108     # Si x es positivo, error es 0. Si x es negativo (-5), error es 5^2 = 25.
109     loss_negativity = torch.mean(torch.relu(-outputs_physics)**2) * 10 # Peso x10
110     # para forzar cumplimiento
111
112     loss_physics = loss_negativity
113
114     # --- C. SUMA TOTAL ---
115     loss_total = loss_data + loss_physics
116
117     loss_total.backward()
118     optimizer.step()
119
120     if epoch % 1000 == 0:
121         print(f"Epoch {epoch}: Data Loss {loss_data.item():.6f} + Physics Loss {
122         loss_physics.item():.6f}")
123
124 # =====
125 # 4. VISUALIZACIÓN FINAL (PANELES SEPARADOS)

```

```

122 # =====
123
124 # Generar rango suave para graficar líneas
125 days_range = np.linspace(9, 18, 100)
126
127 # --- Predicciones D1 (Rápido) ---
128 input_D1 = np.column_stack((days_range, np.zeros_like(days_range)))
129 tensor_D1 = torch.tensor(scaler_x.transform(input_D1), dtype=torch.float32)
130 pred_D1_real = scaler_y.inverse_transform(model(tensor_D1).detach().numpy())
131
132 # --- Predicciones D4 (Lento) ---
133 input_D4 = np.column_stack((days_range, np.ones_like(days_range)))
134 tensor_D4 = torch.tensor(scaler_x.transform(input_D4), dtype=torch.float32)
135 pred_D4_real = scaler_y.inverse_transform(model(tensor_D4).detach().numpy())
136
137 # --- CONFIGURACIÓN GRÁFICA ---
138 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 6))
139
140 # === GRÁFICA 1: CO2 (Respiración) ===
141 # Datos Reales
142 ax1.scatter(df_pivot[df_pivot['dieta']==0]['dia'], df_pivot[df_pivot['dieta']==0][
    'co2_rate'],
143             color='blue', s=80, zorder=5, label='Real D1 (Rápido)')
144 ax1.scatter(df_pivot[df_pivot['dieta']==1]['dia'], df_pivot[df_pivot['dieta']==1][
    'co2_rate'],
145             color='orange', s=80, zorder=5, label='Real D4 (Lento)')
146
147 # Predicciones IA
148 ax1.plot(days_range, pred_D1_real[:, 0], 'b--', linewidth=2.5, label='IA D1')
149 ax1.plot(days_range, pred_D4_real[:, 0], color='orange', linestyle='--', linewidth
    =2.5, label='IA D4')
150
151 ax1.set_title("Respiración Aerobia (CO2)")
152 ax1.set_xlabel("Días")
153 ax1.set_ylabel("Tasa CO2 (ppm/min)")
154 ax1.legend()
155 ax1.grid(True, alpha=0.3)
156
157 # === GRÁFICA 2: CH4 (Fermentación) ===
158 # Datos Reales
159 ax2.scatter(df_pivot[df_pivot['dieta']==0]['dia'], df_pivot[df_pivot['dieta']==0][
    'ch4_rate'],
160             color='blue', marker='x', s=100, linewidth=2, zorder=5, label='Real D1
    (Rápido)')
161 ax2.scatter(df_pivot[df_pivot['dieta']==1]['dia'], df_pivot[df_pivot['dieta']==1][
    'ch4_rate'],
162             color='orange', marker='x', s=100, linewidth=2, zorder=5, label='Real
    D4 (Lento)')
163
164 # Predicciones IA
165 ax2.plot(days_range, pred_D1_real[:, 1], 'b--', linewidth=2.5, label='IA D1')
166 ax2.plot(days_range, pred_D4_real[:, 1], color='orange', linestyle='--', linewidth
    =2.5, label='IA D4')
167
168 ax2.set_title("Fermentación Anaerobia (CH4)")
169 ax2.set_xlabel("Días")
170 ax2.set_ylabel("Tasa CH4 (ppm/min)")
171 ax2.legend()
172 ax2.grid(True, alpha=0.3)

```

```

173 plt.tight_layout()
174 plt.show()
175 print(";Proceso completado! Gráfica generada.")
176
177 from scipy.integrate import simpson
178
179 # --- CÁLCULO DE TOTALES ACUMULADOS (INTEGRAL) ---
180 print("\n" + "="*40)
181 print("PREDICCIONES DE PRODUCCIÓN TOTAL ACUMULADA")
182 print("="*40)
183
184 # Función para integrar: Área bajo la curva de Tasa (ppm/min) vs Tiempo (Días)
185 # Factor de conversión: 1 día = 1440 minutos
186 # Fórmula: Integral = Suma(Tasa * dt) * 1440
187
188 def calcular_integral(tasa_predicha, dias):
189     # tasa_predicha es un array de valores de y
190     # dias es el eje x
191     area = simpson(tasa_predicha, x=dias)
192     return area * 1440 # Convertir días a minutos
193
194 # 1. CO2 Totales
195 total_co2_d1 = calcular_integral(pred_D1_real[:, 0], days_range)
196 total_co2_d4 = calcular_integral(pred_D4_real[:, 0], days_range)
197
198 # 2. CH4 Totales
199 total_ch4_d1 = calcular_integral(pred_D1_real[:, 1], days_range)
200 total_ch4_d4 = calcular_integral(pred_D4_real[:, 1], days_range)
201
202 print(f"CO2 ACUMULADO D1 (Rápido): {total_co2_d1:.0f} ppm acumulados")
203 print(f"CO2 ACUMULADO D4 (Lento): {total_co2_d4:.0f} ppm acumulados")
204 print("-" * 30)
205 print(f"CH4 ACUMULADO D1 (Rápido): {total_ch4_d1:.0f} ppm acumulados")
206 print(f"CH4 ACUMULADO D4 (Lento): {total_ch4_d4:.0f} ppm acumulados")
207
208 # =====
209 # 6. SENSOR VIRTUAL: ESTIMACIÓN DE BIOMASA (B)
210 # =====
211 # Parámetros aproximados basados en literatura (Eriksen/Luedeking-Piret)
212 # Nota: Estos valores son teóricos. Para precisión exacta se requiere calibración.
213 Y_B = 1.5 # Costo de crecimiento (Cuanto CO2 cuesta hacer 1g de larva)
214 m_coeff = 0.1 # Costo de mantenimiento (Cuanto respira 1g de larva quieta)
215
216 def sensor_virtual_biomasa(tasa_co2_predicha, dias, Y_B, m):
217     """
218     Resuelve la ED0: dB/dt = (r_CO2 - m*B) / Y_B
219     Usamos el método de Euler simple para integrar paso a paso.
220     """
221     dt = dias[1] - dias[0] # Paso de tiempo
222     biomasa = [0.1] # Biomasa inicial (asumimos larvas pequeñas 0.1g o unidad arbitraria)
223
224     for i in range(len(tasa_co2_predicha) - 1):
225         r_co2 = tasa_co2_predicha[i]
226         B_actual = biomasa[-1]
227
228         # Ecuación Inversa de Eriksen
229         dB_dt = (r_co2 - (m * B_actual)) / Y_B
230

```

```

231         # Evitar crecimiento negativo (biológicamente la larva no se encoge drá
232         sticamente comiendo)
233         if dB_dt < 0: dB_dt = 0
234
235         B_siguiete = B_actual + dB_dt * dt
236         biomasa.append(B_siguiete)
237
238         return np.array(biomasa)
239
240 # Calcular Biomasa Virtual para D1 y D4
241 # Nota: Normalizamos la tasa dividiendo por 100 para que los números de biomasa
242     sean manejables
243 # (Esto asume que la tasa está en ppm y queremos una unidad arbitraria de masa)
244 factor_escala = 100.0
245 biomasa_est_D1 = sensor_virtual_biomasa(pred_D1_real[:, 0]/factor_escala,
246     days_range, Y_B, m_coeff)
247 biomasa_est_D4 = sensor_virtual_biomasa(pred_D4_real[:, 0]/factor_escala,
248     days_range, Y_B, m_coeff)
249
250 # --- GRAFICAR BIOMASA ESTIMADA ---
251 plt.figure(figsize=(10, 6))
252
253 plt.plot(days_range, biomasa_est_D1, 'b-', linewidth=3, label='Biomasa Estimada D1
254     (Rápido)')
255 plt.plot(days_range, biomasa_est_D4, color='orange', linestyle='--', linewidth=3,
256     label='Biomasa Estimada D4 (Lento)')
257
258 plt.title("Sensor Virtual: Estimación de Crecimiento Larval basada en CO2")
259 plt.xlabel("Días")
260 plt.ylabel("Biomasa Acumulada (Unidades Arbitrarias)")
261 plt.legend()
262 plt.grid(True, alpha=0.3)
263 plt.show()
264
265 print("Sensor Virtual ejecutado. Revisa la gráfica de crecimiento estimado.")
266
267 # =====
268 # 7. VALIDACIÓN: SENSOR VIRTUAL VS TEORÍA ERIKSEN
269 # =====
270
271 # Parámetros Biológicos (Aproximados de Literatura)
272 Y_B = 1.5 # Costo de Crecimiento (mol CO2 / C-mol biomasa)
273 m_coeff = 0.15 # Mantenimiento (mol CO2 / C-mol biomasa * día)
274
275 def sensor_virtual_biomasa(tasa_co2, dias, Y_B, m):
276     """
277     Integra la EDO inversa: dB/dt = (r_CO2 - m*B) / Y_B
278     """
279     dt = dias[1] - dias[0]
280     # Normalizamos la tasa para obtener unidades de biomasa relativas (0 a 1 aprox
281     )
282     # Esto es necesario porque no tenemos la conversión exacta a moles reales,
283     # pero la FORMA de la curva es lo que importa.
284     tasa_norm = tasa_co2 / np.max(tasa_co2) * 2.5
285
286     biomasa = [0.05] # Biomasa inicial (larva pequeña)
287
288     for i in range(len(tasa_norm) - 1):

```

```

283     r_co2 = tasa_norm[i]
284     B_actual = biomasa[-1]
285
286     # Ecuación fundamental de Eriksen despejada para Crecimiento
287     tasa_crecimiento = (r_co2 - (m * B_actual)) / Y_B
288
289     # Restricción biológica: La larva no se encoge (salvo inanición extrema)
290     if tasa_crecimiento < 0: tasa_crecimiento = 0
291
292     B_siguiente = B_actual + tasa_crecimiento * dt
293     biomasa.append(B_siguiente)
294
295     return np.array(biomasa)
296
297 # 1. Calcular Biomasa Inferida por la IA
298 biomasa_D1 = sensor_virtual_biomasa(pred_D1_real[:, 0], days_range, Y_B, m_coeff)
299 biomasa_D4 = sensor_virtual_biomasa(pred_D4_real[:, 0], days_range, Y_B, m_coeff)
300
301 # 2. Generar Curva Teórica Ideal (Sigmoidea de Eriksen)
302 #  $B(t) = B_{\max} / (1 + \exp(-k * (t - t_{\text{mid}})))$ 
303 B_max_ref = np.max(biomasa_D1) # Usamos D1 como referencia de "éxito"
304 k_eriksen = 0.9 # Tasa de crecimiento típica en literatura
305 t_mid = 11.5 # Día de máxima velocidad de crecimiento
306
307 curva_teorica = B_max_ref / (1 + np.exp(-k_eriksen * (days_range - t_mid)))
308
309 # --- GRÁFICA FINAL ---
310 plt.figure(figsize=(12, 7))
311
312 # Curva Teórica (Lo que dice el libro)
313 plt.plot(days_range, curva_teorica, 'k--', linewidth=2, alpha=0.6, label='Teoría
    Eriksen (Modelo Ideal)')
314
315 # Curvas Reales Inferidas (Lo que dice tu IA)
316 plt.plot(days_range, biomasa_D1, 'b-', linewidth=4, alpha=0.9, label='IA: Dieta D1
    (Rápida)')
317 plt.plot(days_range, biomasa_D4, 'orange', linewidth=4, alpha=0.9, label='IA:
    Dieta D4 (Lenta)')
318
319 # Decoración
320 plt.fill_between(days_range, biomasa_D1, alpha=0.1, color='blue')
321 plt.title("Sensor Virtual: Estimación de Biomasa basada en Respiración (PINN)",
    fontsize=14)
322 plt.xlabel("Días de Desarrollo", fontsize=12)
323 plt.ylabel("Biomasa Acumulada (Unidades Relativas)", fontsize=12)
324 plt.legend(fontsize=11, loc='upper left')
325 plt.grid(True, linestyle=':', alpha=0.6)
326
327 # Anotaciones
328 idx_max = np.argmax(biomasa_D1)
329 plt.annotate('Fase Prepupa\n(Crecimiento Detenido)',
330     xy=(days_range[idx_max], biomasa_D1[idx_max]),
331     xytext=(14, biomasa_D1[idx_max]*0.8),
332     arrowprops=dict(facecolor='black', shrink=0.05))
333
334 plt.tight_layout()
335 plt.show()
336 print("|Gráfica generada! Analicemos las curvas...")
337

```

```

338 import numpy as np
339 import pandas as pd
340
341 # =====
342 # 7. GENERACIÓN DE DATOS: SENSOR VIRTUAL (BIOMASA)
343 # =====
344
345 print("\n" + "="*50)
346 print("EJECUTANDO SENSOR VIRTUAL (ESTIMACIÓN DE BIOMASA)")
347 print("="*50)
348
349 # --- 1. Definición de Parámetros Biológicos (Aproximación) ---
350 # Y_B: Rendimiento (Cuánto CO2 cuesta crear estructura).
351 # m: Mantenimiento (Cuánto CO2 cuesta mantener la estructura).
352 Y_B = 1.5
353 m_coeff = 0.15
354
355 # --- 2. Función del Sensor Virtual (Inversa de Eriksen) ---
356 def sensor_virtual_biomasa_numerico(tasa_co2, dias, Y_B, m):
357     dt = dias[1] - dias[0]
358
359     # Normalización para unidades relativas (0 a 100% del tamaño final esperado)
360     # Esto es crucial porque la tasa_co2 está en ppm, no en moles exactos.
361     # Usamos un factor de escala basado en el máximo de D1 para tener una
362     # referencia.
363     factor_escala = np.max(tasa_co2) / 2.5
364     tasa_norm = tasa_co2 / factor_escala
365
366     biomasa = [0.05] # Iniciamos con una "semilla" de biomasa (larva pequeña)
367
368     for i in range(len(tasa_norm) - 1):
369         r_co2 = tasa_norm[i]
370         B_actual = biomasa[-1]
371
372         # Ecuación diferencial inversa: Crecimiento = (RespiraciónTotal -
373         # Mantenimiento) / CostoCrecimiento
374         tasa_crecimiento = (r_co2 - (m * B_actual)) / Y_B
375
376         # Restricción física: No crecimiento negativo (a menos que sea inanición
377         # extrema)
378         if tasa_crecimiento < 0: tasa_crecimiento = 0
379
380         B_siguiente = B_actual + tasa_crecimiento * dt
381         biomasa.append(B_siguiente)
382
383     return np.array(biomasa)
384
385 # --- 3. Calcular las Curvas ---
386 # Usamos las predicciones de CO2 que ya tienes en memoria (pred_D1_real y
387 # pred_D4_real)
388 bio_est_D1 = sensor_virtual_biomasa_numerico(pred_D1_real[:, 0], days_range, Y_B,
389 m_coeff)
390 bio_est_D4 = sensor_virtual_biomasa_numerico(pred_D4_real[:, 0], days_range, Y_B,
391 m_coeff)
392
393 # Calcular curva teórica Eriksen (Sigmoidea Ideal) para comparar
394 # Parámetros ajustados al máximo de D1 para comparar la FORMA
395 B_max_ref = np.max(bio_est_D1)
396 k_eriksen = 0.9

```

```

391 t_mid = 11.5
392 bio_teorica = B_max_ref / (1 + np.exp(-k_eriksen * (days_range - t_mid)))
393
394 # --- 4. IMPRIMIR TABLA DE DATOS PARA ANÁLISIS ---
395 # Seleccionamos índices representativos (uno por día aprox) para no imprimir 1000
    lineas
396 indices = np.linspace(0, len(days_range)-1, 10, dtype=int)
397
398 print(f"{'DÍA':<6} | {'D1 ESTIMADA':<12} | {'D4 ESTIMADA':<12} | {'TEORÍA ERIKSEN
    ':<15} | {'ESTADO D1'}")
399 print("-" * 75)
400
401 for i in indices:
402     dia = days_range[i]
403     d1_val = bio_est_D1[i]
404     d4_val = bio_est_D4[i]
405     teo_val = bio_teorica[i]
406
407     # Determinar estado simple
408     estado = "Creciendo"
409     if i > 0:
410         # Si creció menos del 1% respecto al punto anterior del loop (aprox)
411         if (d1_val - bio_est_D1[indices[indices.tolist().index(i)-1]]) < 0.05:
412             estado = "Meseta (Prepupa)"
413
414     if dia < 10: estado = "Inicio Exp."
415
416     print(f"{dia:.1f}    | {d1_val:.4f}          | {d4_val:.4f}          | {teo_val:.4f}
        | {estado}")
417
418 print("-" * 75)
419 print("Copia esta tabla y pégala en el chat para analizarla.")
420 # =====

```

Listing A.6: Visualización de Datos y Pendientes

Referencias Bibliográficas

Eriksen, N. T. (2022). Dynamic Modelling of Feed Assimilation, Growth, Lipid Accumulation, and CO₂ Production in Black Soldier Fly Larvae. *PLoS One*, 17(10), e0276605. Consultado el 2 de marzo de 2025, desde <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0276605>