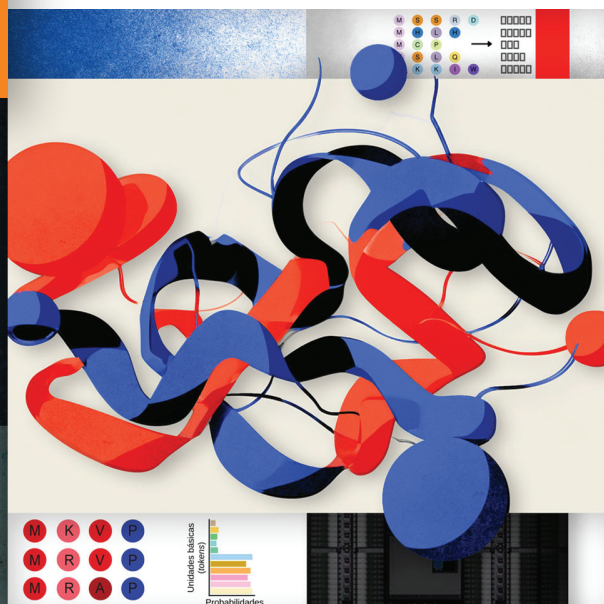
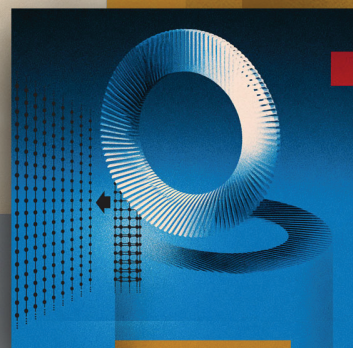
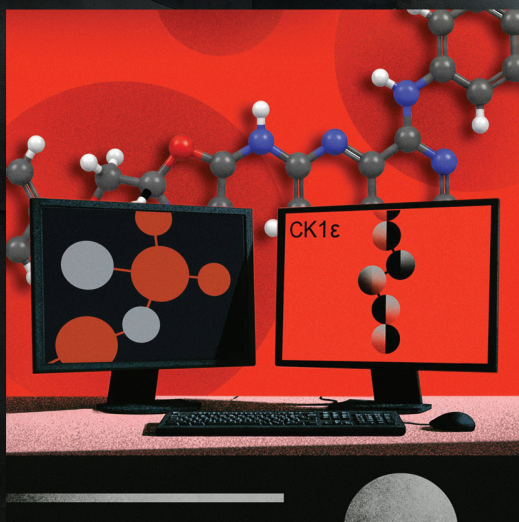




Congreso Internacional de SUPERCÓMPUTO

El Supercómputo en los Tiempos
de la Inteligencia Artificial **2**



Agosto 2026 • DOI: 10.22201/dgtic.26832968e.2026.17 • ISSN: 2683-2968.

TIES es una revista de acceso abierto bajo la licencia Creative Commons Atribución-No Comercial 4.0 (CC BY-NC 4.0).

© 2026 TIES, Revista de Tecnología e Innovación en Educación Superior es editada por la Universidad Nacional Autónoma de México (UNAM) a través de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC). Circuito exterior s/n, Ciudad Universitaria, Alcaldía Coyoacán, C.P. 04510, Ciudad de México, México. Número de reserva de Derechos otorgado por INDAUTOR: 04-2019-011816190900-203.

El contenido de los artículos es responsabilidad de los autores y no refleja el punto de vista del Comité editorial, del Editor o de la UNAM.



Índice

Editorial

Descifrando el Lenguaje de las Proteínas: Modelos de Lenguaje a Gran Escala para el Diseño y Optimización de Nuevas Macromoléculas	I
Uso de supercómputo para el descubrimiento de fármacos: un caso de estudio sobre CK1ε	1
Supercómputo para el medio ambiente: paralelizando simulaciones de derrames en el mar	36
Potenciando pronósticos ambientales en México: supercómputo como pilar en modelación de calidad del aire y de dispersión de cenizas	51
Superconductividad en nanoestructuras vía matriz por bloques	62
	75

Congreso Internacional de Supercómputo 2025, volumen 2.

El Supercómputo en la Ciencia de Frontera

La investigación científica ha experimentado un cambio fundamental en las últimas décadas. Ha ido de un modelo experimental al "cuarto paradigma" de la ciencia, caracterizado por la presencia de datos masivos y el modelado con computadoras. En este escenario, el supercómputo se ha consolidado como un agente epistemológico determinante, en el que no se trata de procesar datos a gran velocidad, sino de abrir metodologías científicas para simular la realidad y descifrar la complejidad de la naturaleza y de la biología humana. Las instituciones de educación superior desempeñan un papel estratégico en este contexto, tanto en el desarrollo de recursos e infraestructura como en la formación de especialistas que apuntalan el progreso y los avances frente a los desafíos tecnológicos.

Este número de la *Revista TIES* da testimonio del impacto del supercómputo y la inteligencia artificial en la resolución de problemas complejos. A través de una narrativa que atraviesa la nanoescala, los sistemas ambientales y el desarrollo de tecnologías, este número ilustra cómo la simulación numérica y los modelos predictivos impactan nuestra relación con el entorno y la salud humana.

La primera frontera se sitúa en la escala molecular con el artículo "Descifrando el lenguaje de las proteínas: modelos de lenguaje a gran escala para el diseño y optimización de nuevas macromoléculas", en el que Diego Pérez Villanueva, Luz Mariana González Vega, Diego Muñoz y Sergio Romero Romero, abordan un problema de la bioquímica computacional. Tradicionalmente, la ingeniería de proteínas se ha limitado a la evolución dirigida o la mutación racional, procesos que pueden resultar lentos y costosos. Las personas autoras muestran cómo los modelos de lenguaje de gran tamaño (LLM), basados en la arquitectura Transformer y entrenados automáticamente con millones de secuencias, aprenden el "lenguaje" de los aminoácidos. Este paralelismo lingüístico coadyuva a la creación de nuevas macromoléculas con funciones innovadoras en los campos de biotecnología y biomedicina.

El potencial de modelado se manifiesta en una aplicación clínica que es trascendental en la lucha contra el cáncer. Luis Córdova, Marco Velasco y Adrián Sánchez, en el artículo "Uso de supercómputo para el descubrimiento de fármacos: un caso de estudio sobre CK1ε", describen el uso de recursos computacionales de alto rendimiento en el cribado virtual y la simulación de dinámica molecular. La caseína cinasa 1 ε (CK1ε) es una enzima asociada con la progresión tumoral, por lo que constituye una

diana terapéutica de interés. No obstante, identificar compuestos químicos capaces de inhibirla mediante métodos convencionales toma más de una década. A partir de simulaciones de dinámica molecular ejecutadas en la supercomputadora Miztli de la UNAM, los autores caracterizaron las interacciones moleculares y propusieron candidatos viables.

El supercómputo también encuentra aplicaciones directas en el estudio del medio ambiente y en la capacidad de respuesta ante contingencias ecológicas. Gabriela Reséndiz Colorado, Rosario de Lourdes Romero, Jorge Zavala, Miguel Ángel Robles y Pavel Ernesto Oropeza presentan en "Supercómputo para el medio ambiente: paralelizando simulaciones de derrames en el mar" una estrategia para generar pronósticos rápidos ante incidentes por hidrocarburos en el Golfo de México. A través del uso de la biblioteca PyGnome y la paralelización de las simulaciones mediante *scripts* en Python y Bash en el clúster Ometéotl de la UNAM, las personas autoras redujeron en un 77.8 % el tiempo de cómputo frente a los métodos secuenciales. Esto hizo posible ejecutar más de 66 000 simulaciones en apenas 36 horas, una capacidad particularmente relevante cuando la oportunidad de los resultados puede incidir en las decisiones de los equipos responsables de atender una contingencia.

José Agustín García aborda el tema de la modelación atmosférica y la protección civil ante riesgos naturales, donde el cómputo de alto rendimiento puede salvar vidas. En "Potenciando pronósticos ambientales en México: supercómputo como pilar en modelación de calidad del aire y de dispersión de cenizas", los investigadores del ICAYCC de la UNAM exponen la implementación diaria de los modelos WRF-Chem y Fall3D para resolver numéricamente las ecuaciones de Navier-Stokes y de transporte químico, lo que permite predecir cómo se dispersan contaminantes como el ozono o las cenizas del volcán Popocatepetl. Para ello, se necesita una infraestructura de almacenamiento paralelo como LUSTRE y gestores de trabajos como SLURM.

El número concluye con la participación de las personas autoras Germán Emiliano López, Vicenta Sánchez y Chumin Wang, quienes abordan la escala cuántica en el artículo "Superconductividad en nanoestructuras vía matriz por bloques". Se trata de un fenómeno en el que los electrones cooperan sin resistencia eléctrica para transportar energía sin pérdidas. Tradicionalmente, resolver las ecuaciones de Bogoliubov-de Gennes (BdG) mediante métodos autoconsistentes es muy complejo desde el punto de vista computacional; sin embargo, con la supercomputadora Miztli de la UNAM, lograron dividir las matrices complejas en bloques independientes. El resultado es un problema de 19 átomos que antes requería tres semanas de procesamiento continuo y que se resolvió en apenas nueve minutos, lo que representa una aceleración de más de tres mil veces. El trabajo muestra un futuro en el que la combinación de álgebra lineal, supercómputo e inteligencia artificial impulsa el diseño cuántico de nuevos superconductores que operan a temperatura ambiente.

Con este segundo volumen, la *Revista TIES* cierra la publicación de los trabajos derivados del Congreso Internacional de Supercómputo 2025, organizado por la Universidad Nacional Autónoma de México. Los artículos reunidos en ambos números muestran un campo que difícilmente puede entenderse

sólo desde la capacidad de procesamiento: el supercómputo participa en la manera de formular preguntas, construir modelos y explorar fenómenos cuya escala o complejidad rebasa los límites de otros métodos de análisis. Desde las ciencias ambientales y biomédicas hasta el estudio de materiales y sistemas cuánticos, estos trabajos dejan constancia de algunas de las líneas presentadas durante el congreso y, al mismo tiempo, del conocimiento que se desarrolla en la Universidad en torno a la computación de alto rendimiento.

Dr. Héctor Benítez Pérez



Descifrando el Lenguaje de las Proteínas: Modelos de Lenguaje a Gran Escala para el Diseño y Optimización de Nuevas Macromoléculas

Diego Pérez-Villanueva*

Universidad Nacional Autónoma de México, Instituto de Fisiología Celular, Departamento de Bioquímica y Biología Estructural, Ciudad de México, México.

ORCID: 0000-0002-8002-5464

Luz Mariana Gonzalez-Vega*

Universidad Nacional Autónoma de México, Instituto de Fisiología Celular, Departamento de Bioquímica y Biología Estructural, Ciudad de México, México.

ORCID: 0009-0003-7933-2760

Diego Muñoz Beltran*

Universidad Nacional Autónoma de México, Instituto de Fisiología Celular, Departamento de Bioquímica y Biología Estructural, Ciudad de México, México.

ORCID: 0009-0008-3790-7286

Sergio Romero-Romero

Universidad Nacional Autónoma de México, Instituto de Fisiología Celular, Departamento de Bioquímica y Biología Estructural, Ciudad de México, México.

ORCID: 0000-0003-2144-7912

** Estos autores contribuyeron por igual a este trabajo*

Recepción: 02 de marzo de 2026.

Aceptación: 26 de junio de 2026.

Agosto 2026 • número de revista 17 • DOI: 10.22201/dgtic.26832968e.2026.160

Descifrando el Lenguaje de las Proteínas: Modelos de Lenguaje a Gran Escala para el Diseño y Optimización de Nuevas Macromoléculas

Resumen

Los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) han transformado la inteligencia artificial al demostrar que pueden aprender patrones estadísticos complejos a partir de datos secuenciales masivos. De manera análoga al lenguaje natural, las proteínas pueden entenderse como secuencias de aminoácidos cuya organización jerárquica determina su estructura y función. Este paralelismo ha impulsado el uso de LLMs para la generación de secuencias proteicas de novo, permitiendo explorar regiones del espacio de secuencias más allá de los métodos tradicionales. En esta revisión, presentamos los fundamentos de los LLMs aplicados a ciencia de proteínas, en particular, la arquitectura Transformer y el modelado autorregresivo, describiendo cómo los mecanismos de atención, las representaciones vectoriales y las estrategias de entrenamiento posibilitan la generación de proteínas residuo por residuo, de manera condicional o incondicional. Asimismo, analizamos los enfoques generativos y los criterios de validación computacional y experimental de las secuencias obtenidas. Finalmente, destacamos la importancia del supercómputo y de infraestructuras con aceleradores especializados para el entrenamiento y despliegue de estos modelos, así como otras aplicaciones relevantes en ciencia de proteínas, como la anotación funcional y la predicción estructural.

Palabras Clave: modelos de lenguaje, generación de secuencias proteicas, diseño de proteínas, inteligencia artificial generativa, cómputo de alto desempeño, biología computacional.

Decoding the Language of Proteins: Large Language Models for the Design and Optimization of Novel Macromolecules

Abstract

Large Language Models (LLMs) have transformed Artificial Intelligence by demonstrating the ability to learn complex statistical patterns from massive sequential datasets. Analogous to natural language, proteins can be understood as sequences of amino acids whose hierarchical organization determines their structure and function. This parallelism has driven the application of LLMs to de novo protein sequence generation, enabling the exploration of regions of sequence space beyond the reach of traditional methods. In this review, we present the foundations of LLMs applied to protein science, with particular emphasis on transformer-based architectures and autoregressive modeling. We describe how attention mechanisms, vector representations, and training strategies enable residue-by-residue protein generation, either in conditional or unconditional settings. We further analyze generative approaches and the computational and experimental validation strategies used to assess the resulting sequences. Finally, we highlight the critical role of high-performance computing and accelerator-based infrastructures in training and deploying these models at scale, and briefly discuss other applications in protein science, including functional annotation and structural prediction.

Keywords: *language models, protein sequence generation, protein design, generative artificial intelligence, high-performance computing, computational biology.*

Introducción

Las proteínas son moléculas esenciales para el funcionamiento de todos los organismos. Participan en funciones diversas, como determinar la estructura de una célula, catalizar reacciones químicas o participar en la respuesta inmunitaria. Son indispensables para la vida y, dada la diversidad de funciones que desempeñan, su diseño computacional se ha convertido en foco central de investigación en los últimos años.

Históricamente, uno de los mayores retos dentro del área ha sido el diseño de proteínas nuevas con funciones específicas. A diferencia de las proteínas naturales, que son el resultado de millones de años de evolución, las proteínas sintéticas dependen del entendimiento humano de la relación entre secuencia, estructura y función. Debido a esta complejidad, la ingeniería de proteínas ha dependido de la modificación y selección experimental de proteínas naturales para descubrir y optimizar funciones bioquímicas [1]. Tradicionalmente, el diseño de proteínas se ha basado tanto en la evolución dirigida como en el diseño racional y, aunque estos enfoques han sido efectivos, suelen ser costosos y requieren demasiado tiempo, lo que ha limitado la velocidad con la que pueden explorarse nuevas funciones proteicas [2].

Es por ello que la inteligencia artificial (IA) y los modelos de lenguaje a gran escala (LLMs, *Large Language Models* en inglés) han revolucionado la manera en que se pueden generar proteínas mediante enfoques de diseño computacional. Específicamente, estas herramientas permiten un desarrollo experimental más eficiente, al acelerar la creación de nuevas secuencias de proteínas capaces de formar estructuras y funciones novedosas. Gracias a este avance, hoy es posible abordar problemas en áreas como biotecnología, biología sintética o biomedicina desde una perspectiva distinta, en la que el diseño de proteínas puede realizarse de forma más eficiente [3], [4], [5], [6].

Entre las herramientas de IA, los LLMs basados en el procesamiento de lenguaje natural, que aprenden a modelar los patrones y reglas del lenguaje humano a partir de grandes cantidades de texto, han demostrado la capacidad de aprender y modelar el “lenguaje” de las proteínas [7]. En este marco, el diseño computacional permite explorar secuencias ausentes en la naturaleza, tratándolas como combinaciones de aminoácidos cuya organización determina su estructura y función. Esto plantea una pregunta central: ¿es posible “escribir” nuevas secuencias para generar proteínas novedosas con formas y funciones distintas?

Modelos de lenguaje a gran escala y su aplicación en ciencia de proteínas

Un modelo de lenguaje es un modelo probabilístico que aprende los patrones estadísticos de las secuencias a partir de grandes volúmenes de datos, con el objetivo de estimar la

probabilidad de una secuencia completa o de sus elementos dado un contexto [8], [9], [10], [11]. Los primeros enfoques dependían de un número fijo de elementos previos, lo que limitaba la captura de dependencias de largo alcance y el manejo de combinaciones poco frecuentes [7]. Con el auge del aprendizaje profundo, surgieron arquitecturas neuronales más expresivas que superaron estas limitaciones y dieron origen a los LLMs actuales [12], [13], [14].

Los LLMs modernos se basan principalmente en la arquitectura Transformer y pueden entrenarse con millones de secuencias para distintos objetivos. Durante el entrenamiento, las secuencias se dividen en unidades básicas denominadas *tokens*, que posteriormente se proyectan en representaciones vectoriales conocidas como *embeddings*, permitiendo capturar relaciones semánticas y estructurales complejas dentro de ventanas de contexto amplias [8], [15], [16] (Tabla I).

Tabla I

Glosario de términos clave en LLMs y diseño de proteínas

Término	Definición
Afinamiento (<i>fine-tuning</i>)	Proceso de ajuste adicional de un modelo preentrenado utilizando datos específicos para adaptarlo a una tarea particular.
Ciclo de entrenamiento (<i>epoch</i>)	Ciclo completo en el que el modelo procesa la totalidad del conjunto de datos de entrenamiento una vez.
Enmascaramiento	Proceso del entrenamiento de un LLM mediante el cual se ocultan partes de una secuencia para que el modelo prediga los elementos faltantes a partir de un contexto.
Hiperparámetro	Variable configurable previa al entrenamiento que controla el aprendizaje del modelo.
<i>in silico</i>	Experimentos o análisis realizados a través de una computadora.

Término	Definición
Instrucción de entrada (<i>prompt</i>)	Instrucción que se le da a un modelo para que genere una respuesta.
<i>in vitro</i>	Experimentos dentro de un laboratorio.
Mecanismo de atención	Proceso mediante el cual el modelo pondera la relevancia de otros <i>tokens</i> al construir la representación de cada uno.
Modelo de lenguaje	Modelo probabilístico que estima la probabilidad de una secuencia o de sus elementos dado un contexto.
Modelo de lenguaje a gran escala (LLM)	Modelo de lenguaje basado en arquitecturas neuronales profundas y entrenado con grandes volúmenes de datos para capturar patrones complejos en secuencias.
Modelo de lenguaje autorregresivo	Modelo generativo de secuencias, capaz de predecir cada elemento siguiente condicionado al contexto pasado previamente definido en un orden determinado.
Modelo de lenguaje enmascarado	Modelo que aprende a predecir elementos ocultos (<i>tokens</i>) dentro de una secuencia, utilizando el contexto disponible.
Parámetro o Peso	Variable interna del modelo que se ajusta automáticamente durante el entrenamiento.
Perplejidad	Métrica que mide la calidad de las predicciones de un modelo sobre una secuencia.
Pruebas de referencia (<i>benchmark</i>)	Conjunto de secuencias o tareas de referencia empleado para evaluar y comparar el desempeño de modelos.
Representación vectorial (<i>embedding</i>)	Representación vectorial numérica de una secuencia o residuo de proteína que captura relaciones contextuales y funcionales.

Término	Definición
Semilla aleatoria (<i>seed</i>)	Valor numérico inicial utilizado para controlar los procesos aleatorios durante el entrenamiento o la generación de un modelo; establecer una misma seed permite reproducir resultados similares entre distintas ejecuciones del modelo.
Temperatura	Hiperparámetro que regula la aleatoriedad en la generación de una secuencia; valores altos aumentan la diversidad y valores bajos producen secuencias menos aleatorias.
<i>Transformer</i>	Arquitectura base de los modelos de lenguaje que aprende relaciones entre los elementos de una secuencia identificando los más relevantes.
Unidad básica de secuencia (<i>token</i>)	Unidad básica que un modelo de lenguaje utiliza para aprender, procesar y generar información.

Esta capacidad vuelve a los LLMs ideales para el diseño de proteínas. Al estar conformadas por un código alfabético, donde cada letra representa un aminoácido, las secuencias proteicas pueden modelarse de forma análoga a la predicción de palabras en una oración. Así, estos modelos predicen el siguiente aminoácido en una secuencia, permitiendo la formación de motivos estructurales, dominios y proteínas completas, de la misma forma en que un conjunto de palabras forma enunciados complejos y conectados (Fig. 1A). Dado que existen millones de secuencias disponibles en diversas bases de datos, los modelos disponen de suficiente información para captar la relación secuencia-estructura-función de las proteínas [10], [11], [17]. Además, esta analogía se extiende a otros fenómenos compartidos entre lenguaje y proteínas como mutaciones comparables a “errores tipográficos”, permutaciones circulares (cambios de orden) y homología entre secuencias (Fig. 1B).

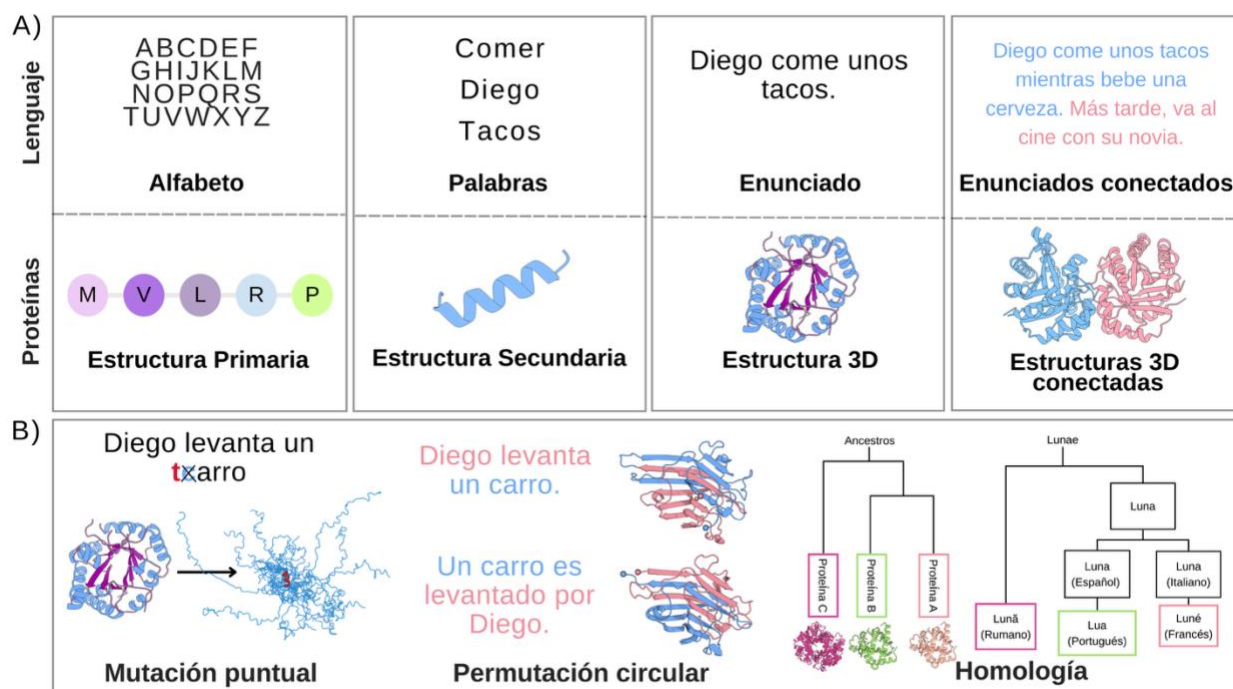


Fig. 1. Analogía entre el lenguaje natural y la organización estructural de las proteínas. A) En la parte superior, se muestra la jerarquía del lenguaje, desde el alfabeto hasta enunciados conectados, ilustrando cómo las unidades básicas se combinan para generar significado en niveles superiores. En la parte inferior, se presenta la organización de las proteínas, donde los aminoácidos (estructura primaria) forman motivos locales (estructura secundaria), que se pliegan en estructuras tridimensionales y pueden ensamblarse en complejos multiméricos. B) Además de su organización jerárquica, comparan otras similitudes que existen entre lenguajes humanos y proteínas: procesos de variación como las mutaciones puntuales similares a lo que sucede en cambios de una letra (izquierda), permutaciones que reorganizan la secuencia y conservan el significado global (centro), y homología como divergencia desde un ancestro común (derecha). Esta analogía resalta los paralelismos entre sistemas secuenciales jerárquicos y su relevancia para los LLMs aplicados a proteínas.

Si bien los LLMs se popularizaron a partir de 2020, su desarrollo se remonta varias décadas atrás. En los años cincuenta, tras múltiples avances en procesamiento de lenguaje natural, investigadores de IBM desarrollaron el primer sistema mecánico de traducción. Posteriormente, en los ochenta, surgieron los primeros modelos estadísticos relevantes para la predicción de texto y, en los años 2000, los primeros modelos neuronales de lenguaje [19]. El avance decisivo en el campo fue la introducción de la arquitectura Transformer en 2017 [18]. A diferencia de las redes recurrentes o convolucionales, Transformer analiza secuencias completas de forma simultánea mediante el mecanismo de atención [20], [21], lo que

permite capturar dependencias a larga distancia dentro de una misma secuencia, clave para procesar secuencias. Precisamente, la introducción de la arquitectura Transformer, junto con los avances en hardware, han impulsado un rápido desarrollo de LLMs en distintas áreas, entre ellas, el diseño y generación de proteínas.

Generación de secuencias proteicas de novo mediante LLMs

Como mencionamos anteriormente, la arquitectura Transformer es la tecnología detrás del éxito de los LLMs en la actualidad. Dicha arquitectura se divide en dos grandes módulos: el codificador y el decodificador [22]. El codificador es el encargado de procesar los datos de entrada, secuencias de aminoácidos, en el caso de los LLMs de proteínas, y generar representaciones. El decodificador toma estas representaciones y, con ellas, genera los datos de salida. Ambos módulos están conformados por varias capas, con subcapas en cada una de ellas. Las subcapas corresponden a un conjunto de cabezas de autoatención y a las redes neuronales prealimentadas, respectivamente. La primera es donde se ejecuta el mecanismo de autoatención, el cual permite a cada unidad básica de secuencia (*token*) comunicarse con las demás para determinar las dependencias que existen entre ellas. La segunda expande y filtra las representaciones provenientes de las cabezas de atención de cada unidad básica de secuencia por separado, generando las representaciones finales de su propia capa.

Originalmente, Transformer utiliza una arquitectura codificador-decodificador conjunta; sin embargo, esta arquitectura ha evolucionado hacia configuraciones que utilizan cada módulo por separado. Por un lado, las arquitecturas sólo codificadoras se emplean para generar representaciones útiles para tareas de inferencia de propiedades; por otro, las sólo decodificadoras se centran en la generación de texto [23], [24]. En esta sección, nos enfocamos en la generación de secuencias proteicas.

Para que un LLM generativo sea capaz de diseñar secuencias, requiere pasar previamente por un proceso con múltiples fases (Fig. 2A). Este proceso comienza con la recolección de millones de secuencias provenientes de bases de datos de referencia [12]. Entre las bases de datos de proteínas, la más popular es UniProt, la cual cuenta con alrededor de 203,130,941 secuencias en total (enero, 2026). De éstas, aunque en menor número, podemos encontrar secuencias con etiquetas de acuerdo a su función, estructura,

localización subcelular, entre otros [25]. Posteriormente, tras recolectar y definir los datos de entrada, es necesario dividir el texto en sus unidades básicas de secuencia que serán claves para el entrenamiento [12], [16], [26]. Dichas unidades básicas en el lenguaje natural pueden ser palabras, caracteres o sílabas y, en el lenguaje de las proteínas, aminoácidos individuales o un conjunto de ellos. Por ejemplo, la oración *¡Hola mundo!* podría dividirse en; (¡)-(Hola)-(mundo)-(!); mientras que, en una proteína de secuencia *MTPK*, podría dividirse como (M)-(T)-(P)-(K). Una vez que las unidades básicas de secuencia son definidas, éstas se representan en el LLM durante su entrenamiento mediante representaciones vectoriales, las cuales capturan relaciones estadísticas a partir de su aparición en contextos similares en un espacio multidimensional [14], [15] (Fig. 2B). Dichas representaciones forman parte de los parámetros entrenables del modelo y suelen incorporarse en sus etapas iniciales de procesamiento [8], [15], [27]. La transformación y optimización de las representaciones vectoriales dependen de los pesos o parámetros del modelo, valores numéricos ajustables que se actualizan durante el entrenamiento [8], [28].

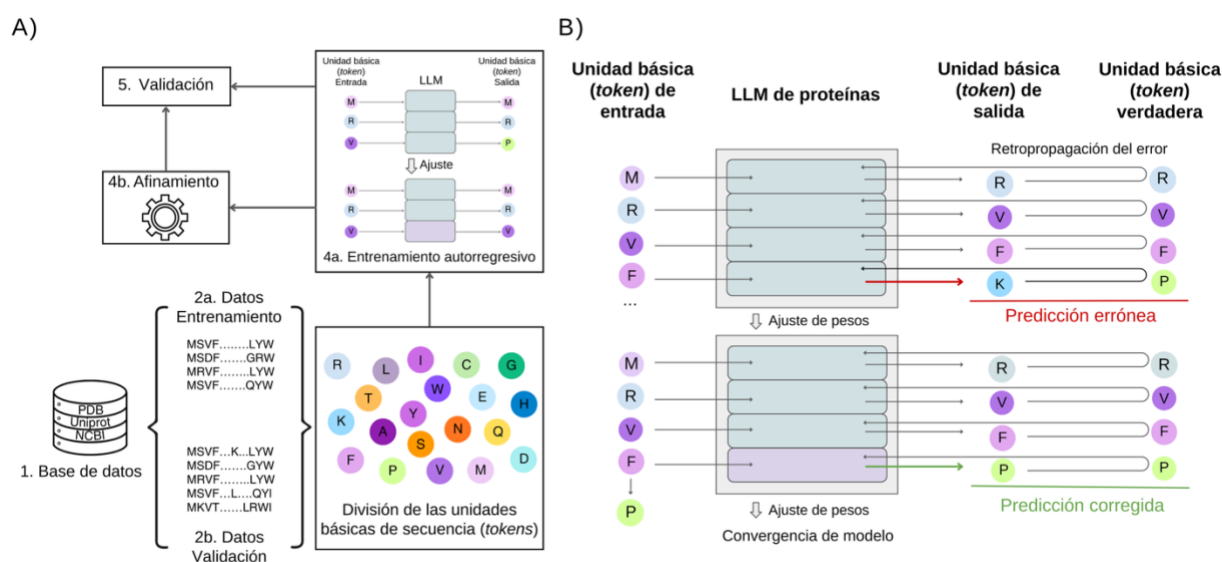


Fig. 2. Representación gráfica del proceso de entrenamiento de un LLM para la generación de secuencias de proteínas. A) Flujo de trabajo para el entrenamiento y validación del modelo. Las secuencias proteicas se obtienen de bases de datos estructurales y de secuencia, se dividen en conjuntos de entrenamiento y validación y posteriormente se dividen en unidades básicas de secuencia (*tokens*), asignándoles un carácter numérico a cada aminoácido incorporado al vocabulario del modelo. El modelo de lenguaje se entrena de manera autorregresiva mediante la predicción del siguiente aminoácido en la secuencia, seguido de un proceso de afinamiento y evaluación del desempeño. Dependiendo del rendimiento del modelo, tras esta fase, es posible

proseguir con la validación del modelo y su despliegue o utilizar el paso de afinamiento como un punto intermedio, permitiendo el uso de datos etiquetados y el ajuste de los hiperparámetros. B) Esquema del aprendizaje autorregresivo a nivel de secuencia. A partir de un conjunto de unidades básicas de secuencia de entrada que representan aminoácidos previamente observados, el modelo predice la siguiente unidad básica en la secuencia. La predicción se compara con la unidad básica de secuencia verdadera y el error calculado se retropropaga a través de la red para actualizar los pesos del modelo. Este proceso iterativo de predicción y ajuste conduce progresivamente a la convergencia del modelo y a la mejora en la capacidad de generar secuencias coherentes con las propiedades estadísticas del conjunto de entrenamiento.

El ajuste de estos valores numéricos se logra mediante el proceso de retropropagación, en coordinación con la estrategia de entrenamiento autorregresiva, específica de LLMs generativos (Fig. 3) [27], [29], [30]. En la estrategia autorregresiva, se predice el residuo siguiente a partir de los residuos anteriores en una secuencia; por ejemplo, si el LLM se encuentra prediciendo el residuo número 34 de una proteína, generará su predicción a partir de los 33 residuos anteriores. De ser errónea la predicción, el error se retropropaga por el LLM y los pesos se ajustan para dirigir la predicción hacia el residuo correcto [28], [29] (Fig. 2B). Una vez completado el entrenamiento, en caso de que sea necesario, puede realizarse un afinamiento o *fine-tuning*, una extensión del entrenamiento utilizando datos etiquetados específicos para una tarea en concreto. En este paso, los pesos ya aprendidos se vuelven a ajustar para especializar el modelo en un problema específico [12], [28], [31]. Finalmente, para evaluar el desempeño del modelo, una fracción de las secuencias iniciales que no se usó durante el entrenamiento se utiliza para validación. Los datos de validación permiten medir la capacidad del modelo de generalizar, utilizando datos no vistos previamente. Asimismo, esta fase permite detectar problemas de ajuste de parámetros y comparar las configuraciones del modelo [31].

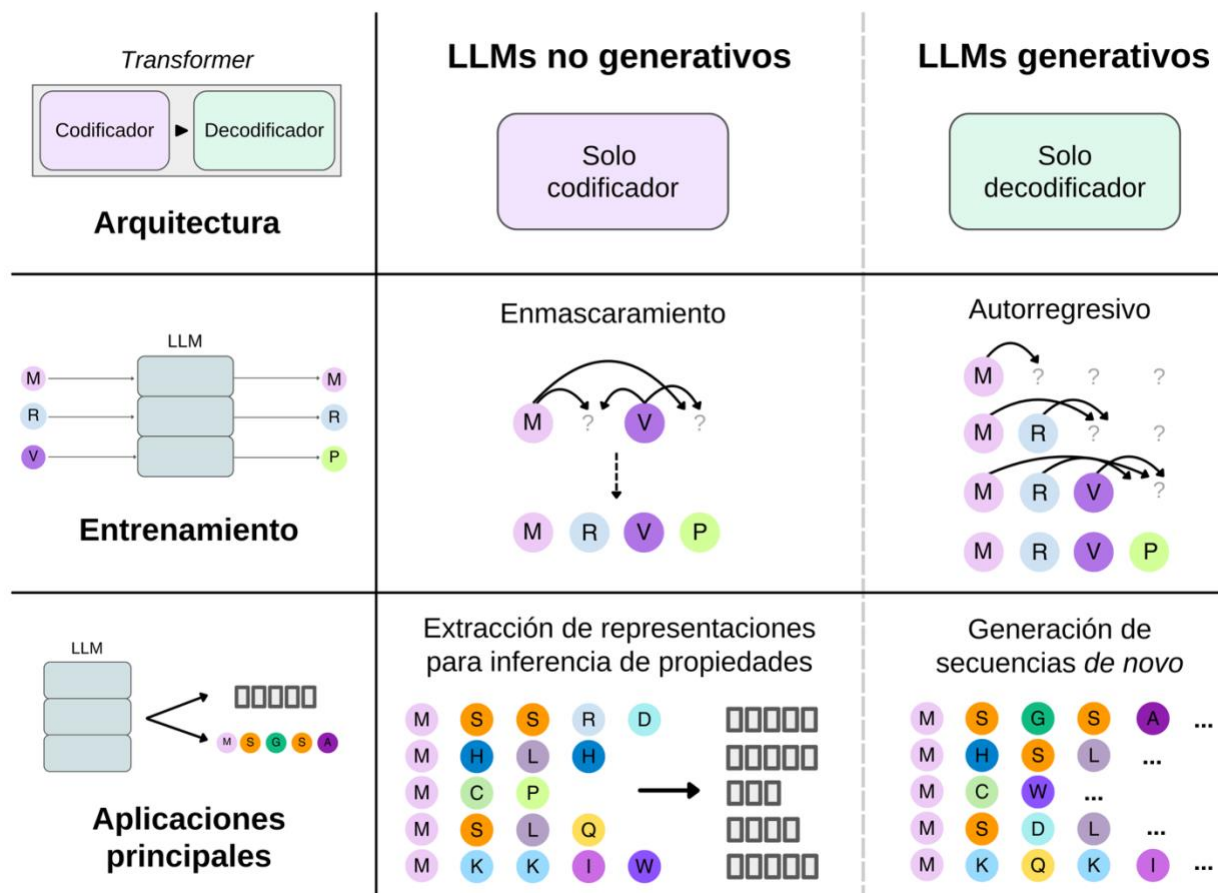


Fig. 3. Arquitecturas basadas en Transformer utilizadas en el campo del diseño de proteínas. La arquitectura Transformer original utiliza un módulo codificador que procesa y genera representaciones de las secuencias de entrada, y un módulo decodificador que toma las representaciones para generar las secuencias de salida.

Actualmente, los LLMs no generativos utilizan la arquitectura sólo codificadora, entrenada por enmascaramiento de residuos para extraer representaciones de las secuencias para predecir o detectar propiedades específicas. Los LLMs generativos, por el contrario, utilizan la arquitectura sólo decodificadora, entrenada por autorregresión para generar secuencias condicional o incondicionalmente.

Entendiendo el proceso de generación, es importante tener en cuenta que podemos configurar la manera en la que entrenamos al modelo. Las variables que permiten configurarlo son los hiperparámetros de entrenamiento. Entre ellos, se incluyen la tasa de aprendizaje, el tamaño de lote y el número de ciclos de entrenamiento o *epochs*. La tasa de aprendizaje determina la magnitud de los ajustes aplicados a los parámetros en cada interacción; el tamaño de lote define cuántos ejemplos se procesan antes de ajustar los

parámetros; y el número de ciclos de entrenamiento controla cuántas veces el modelo recorre por completo el conjunto de datos de entrenamiento [32], [33].

Una vez que los LLMs generativos son entrenados, entonces podemos generar secuencias desde cero y configurar su generación por medio de los hiperparámetros de generación, los cuales modifican la factibilidad de escoger cada unidad básica de secuencia durante la generación mediante el ajuste de sus respectivas probabilidades (Fig. 4B). La estrategia principal para generar secuencias, al igual que en el entrenamiento, es la autorregresiva, en la cual cada predicción depende de las unidades básicas de secuencia anteriores (Fig. 4A) [30]. A partir de un contexto inicial, el modelo estima una distribución de probabilidad sobre la siguiente unidad básica de secuencia y, en función de los hiperparámetros de generación, selecciona uno de su vocabulario. Por ejemplo, la temperatura modula esta distribución de probabilidades: valores elevados la suavizan, favoreciendo la diversidad en sus respuestas, mientras que valores bajos la vuelven más concentrada, produciendo secuencias más conservadoras (Fig. 4B).

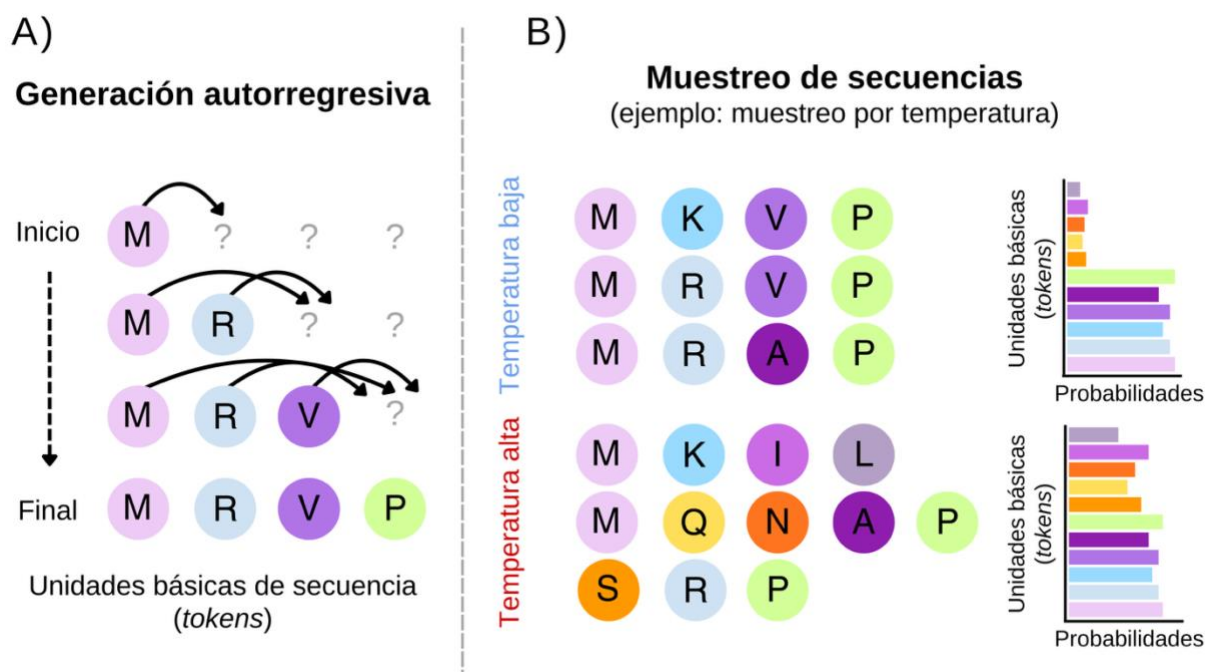


Fig. 4. Generación autorregresiva y estrategias de muestreo para la generación de secuencias proteicas. A) La generación se realiza de manera secuencial, prediciendo iterativamente la siguiente unidad básica de secuencia

condicionada a las unidades previamente generadas. A partir de una unidad básica inicial, el modelo predice sucesivamente los siguientes aminoácidos hasta completar la secuencia final. Cada nueva unidad básica se incorpora al contexto y condiciona la siguiente predicción. B) Ejemplo de muestreo por temperatura durante la generación. La temperatura modula la distribución de probabilidades sobre las posibles unidades básicas de secuencia de salida: valores bajos favorecen las unidades básicas con mayor probabilidad (generación más conservadora y determinista), mientras que valores altos suavizan la distribución y permiten explorar unidades básicas menos probables, incrementando la diversidad de las secuencias generadas.

En la actualidad, existen múltiples LLMs con la capacidad de generar secuencias desde cero, donde cada uno de ellos presenta variaciones arquitectónicas que van dirigidas a mejorar el entrenamiento y/o la generación de secuencias (Tabla II). En general, estos modelos corresponden a arquitecturas sólo decodificadoras y emplean una estrategia autorregresiva para generar secuencias. Dentro de los LLMs generativos, pueden distinguirse dos grandes categorías según su capacidad de formar secuencias condicionadas en una o más tareas: no condicionales y condicionales [34]. Algunos ejemplos de los primeros son ProtGPT2, ProGen2, y ProGen3, los cuales se especializan en la generación de secuencias *de novo* [35], [36], [37]. Cabe mencionar que estos modelos pueden condicionarse a una tarea específica si se reentrenan o afinan con datos etiquetados para ese fin. Por otro lado, entre los modelos condicionales más importantes, podemos encontrar ZymCTRL (enzimas), PoET (homología), ProGen (familias de proteínas), ProLLaMA (función) y IgLM (anticuerpos) [38], [39], [40], [41], [42]. Asimismo, aunque los modelos EVO y EVO 2 no forman únicamente proteínas, son un ejemplo relevante de cómo los LLMs generativos comprenden y utilizan el dogma central de la biología molecular para generar condicionalmente ADN, ARN y proteínas [43], [44].

Tabla II

Algunos ejemplos relevantes de LLMs de proteínas y sus características principales

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
ZymCTRL	Decodificador (autorregresivo)	UniProt/BRENDA (enzimas anotadas)	Generación controlada de secuencias enzimáticas	[39]

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
ProtGPT2	Decodificador (GPT-2, autorregresivo)	UniRef50	Generación de proteínas <i>de novo</i>	[35]
ProGen	Decodificador (autorregresivo)	UniParc, UniprotKB, Pfam, e información taxonómica de NCBI	Generación condicional de proteínas	[38]
ProGen2	Decodificador (autorregresivo)	Combinaciones de Uniref90, Big Fantastic Database, y la base de datos Observed Antibody Space	Generación de proteínas <i>de novo</i> y predicción de aptitud	[36]
ProGen3	Decodificador (autorregresivo, y de relleno)	Profluent Protein Atlas v1	Generación de proteínas <i>de novo</i> y predicción de aptitud	[37]
PoET	Decodificador (autorregresivo)	UniRef50	Generación condicional de secuencias y modelado evolutivo	[40]
PoET-2	Codificador (enmascaramiento) y Decodificador dual (enmascaramiento y autorregresivo)	UniRef50, secuencias asociadas a estructuras de la AlphaFold DataBase	Generación condicional de secuencias, modelado evolutivo y predicción de aptitud	[45]
ProstT5	Codificador (enmascaramiento) y Decodificador (autorregresivo)	AlphaFold DataBase	Clasificación estructural, homología, plegamiento inverso y plegamiento	[46]

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
DPLM	Codificador y Decodificador (difusión)	UniRef50	Generación de secuencias <i>de novo</i> y condicionalmente por motivos estructurales	[48]
EVO	Decodificador (autorregresivo)	Genomas bacterianos y de arqueas del Genome Taxonomy Database, secuencias de fagos del IMG/VR v4 database, y secuencias plasmídicas de IMG/PR database	Generación condicional de secuencias de ADN, generalizando entre ADN, ARN, y proteínas de organismos procariontes. Modelado evolutivo y predicción de aptitud a nivel genoma	[74]
EVO2	Decodificador (autorregresivo)	OpenGenome2 dataset	Generación condicional de secuencias de ADN, generalizando entre ADN, ARN y proteínas de todos los dominios de la vida. Modelado evolutivo y predicción de aptitud a nivel genoma	[44]
ProLLama	Decodificador (autorregresivo)	UniRef50 y protein2ipr de InterPro	Generación de secuencias incondicionalmente y condicionalmente y predicción de propiedades	[41]

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
IgLM	Decodificador (autorregresivo, y de relleno)	Observed Antibody Space	Generación condicional de secuencias de anticuerpos	[42]
PALM-H3	Codificador (enmascaramiento) y Decodificador (autorregresivo)	Observed Antibody Space, The Coronavirus Antibody Database, 14H y 14L, BioMap	Generación <i>de novo</i> de la secuencia CDRH3 de anticuerpos	[47]
HelixFold-Single	Codificador (enmascaramiento)	UniRef30, Uniclust30 y AlphaFold DataBase	Modelado tridimensional	[75]
ESM2/ESMfold	Codificador (enmascaramiento)	UniRef50	Modelado tridimensional	[50]
TooT-BERT-M	Codificador (enmascaramiento)	Swiss-Prot (proteínas de membrana)	Detección de proteínas de membrana	[76]
ProteinBERT	Codificador (enmascaramiento)	UniProtKB/UniRef90	Generación de representaciones	[77]
EnzBERT	Codificador (enmascaramiento)	Secuencias enzimáticas	Clasificación de enzimas por su clase EC	[78]
InterLabelGO+	Codificador (enmascaramiento)	Gene Ontology Annotation	Predicción funcional	[79]
ProSTAGE	Codificador (enmascaramiento) + Red	MPTherm, ProthermDB,	Predicción de estabilidad mutacional	[80]

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
	de grafos convolucional (GCN)	ThermoMutDB, y FireProtDB		
MutaPLM	Codificador (enmascaramiento) y Decodificador (autorregresivo).	MutaDescribe UniProtKB/SwissProt	Generación de representaciones y secuencias optimizadas enfocadas a mutaciones de proteínas	[81]
SignalP 6.0	Codificador (enmascaramiento) + Campos aleatorios condicionales (CRF)	UniProt, Prosite, y Topology Database of Transmembrane Proteins	Predicción de péptidos señal	[82]
Dyna-1	Codificador (enmascaramiento) + clasificador	Biological Magnetic Resonance Bank	Predicción de la dinámica de proteínas	[83]
NetSolP	Codificador (enmascaramiento) + Clasificador	Protein Structure Initiative database y la base de datos de solubilidad y usabilidad de Price	Clasificación de proteínas por solubilidad y usabilidad	[62]
PLMSearch/PLMalign	Codificador (enmascaramiento) + Modelo de regresión	SCOPe40, Swiss-Prot, AlphaFold DataBase, y CATHS40	Detección de homología	[84]
DiffPALM	Codificador (enmascaramiento)	UniRef50	Predicción coevolutiva	[85]
GO2Sum	Codificador y Decodificador	SwissProt	Generación de descripciones	[86]

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
			funcionales a partir de términos GO	
PTM-GPT2	Decodificador (autorregresivo)	Protein Post-translational modifications DataBase	Predicción de regiones con modificaciones postraduccionales	[63]

Por otro lado, una estrategia innovadora para dotar a los modelos generativos de la capacidad de predecir regiones específicas, utilizando contexto bidireccional, es el rellenado de secuencia (*infilling*). En este enfoque, la región objetivo se desplaza al final de la secuencia y se genera usando el contexto restante; ejemplos de esta estrategia son ProGen3 e IgLM. Otra aproximación que amplía las capacidades de los LLMs combina el modelado autorregresivo con el enmascaramiento mediante arquitecturas codificador-decodificador. Modelos como PALM-H3, PoET-2 y ProstT5 potencian la generación contextual al aprovechar representaciones producidas por el codificador [45], [46], [47].

Por último, la estrategia autorregresiva no es la única con capacidad generativa. La estrategia de eliminación de ruido, en la que el modelo se entrena degradando progresivamente una secuencia para luego reconstruirla, también ha demostrado ser eficaz para generar proteínas. Un ejemplo es DPLM, que permite la generación condicionada a varias tareas a partir de representaciones contextuales enriquecidas [48].

Es importante mencionar que, aun cuando los LLMs antes mencionados se entrenan y validan internamente con métricas basadas en la predicción de secuencias naturales, no es suficiente para demostrar la robustez de los modelos en escenarios reales. Para ello, es necesario evaluarlos en sus predicciones, utilizando puntos de referencia *in silico* o *in vitro*.

Validación computacional y experimental de secuencias generadas por LLMs

La generación de nuevas secuencias mediante LLMs representa sólo el primer paso en el diseño computacional de proteínas. Para determinar si estas secuencias son estructural y funcionalmente viables, es indispensable someter tanto al modelo de lenguaje como a las secuencias resultantes a un proceso de validación.

LLMs generativos, como ProtGPT2, ProGen, ZymCTRL o PoET-2, se han utilizado en el diseño de nuevas proteínas [49], [35], [38], [39], [50], siendo capaces de aprender patrones evolutivos y estructurales implícitos y, por lo tanto, han podido generar secuencias *de novo* siguiendo reglas aprendidas. Sin embargo, es fundamental validarlos más allá de sus métricas de entrenamiento, ya que pueden producir secuencias que parecen correctas desde el punto de vista bioinformático, pero que no necesariamente funcionan en un contexto biológico. Por ello, es necesario complementar esta evaluación con pruebas de referencia estandarizadas (*benchmarks*), métricas adicionales de desempeño y, finalmente, estudios de expresión y caracterización experimental que permitan determinar su funcionalidad real. Por ejemplo, para evaluar su desempeño, se utilizan estrategias ortogonales, en este caso, pruebas como Protein GYM o TAPE, que comparan las predicciones del modelo con datos experimentales en propiedades como estabilidad térmica o afinidad, lo cual permite confirmar la robustez de los resultados desde distintos enfoques experimentales y computacionales [51], [52].

Después de validar el modelo de lenguaje y antes de llevar una proteína diseñada en la computadora al laboratorio, es necesario filtrar las secuencias generadas mediante validaciones *in silico* e *in vitro*. En la etapa *in silico*, es importante distinguir entre métricas internas del modelo y métricas externas de evaluación. Entre las primeras, una de las más utilizadas es la perplejidad, que refleja el grado de confianza en sus propias predicciones [53]. Sin embargo, esta métrica no evalúa directamente la viabilidad bioquímica de la proteína. Por ello, es fundamental incorporar métricas externas, como la evaluación computacional de propiedades, entre ellas, la solubilidad, la tendencia a la agregación y el modelado estructural [57], [58], [56], las cuales permiten incrementar las tasas de éxito experimental.

Finalmente, las secuencias seleccionadas deben validarse experimentalmente. En esta etapa, las secuencias se caracterizan mediante métodos bioquímicos en condiciones controladas para confirmar si presentan la estructura y función esperadas. Como ejemplo

concreto, un estudio reciente de nuestro grupo de trabajo utilizó los modelos generativos ProtGPT2 y ZymCTRL para diseñar enzimas *de novo*, en específico, triosafosfato isomerasas, las cuales se generaron y probaron experimentalmente, demostrando proteínas con expresión soluble y actividad catalítica [49], lo que evidencia la capacidad de estos modelos para generar proteínas plausibles y funcionales.

Aplicaciones adicionales de los LLMs en ciencia de proteínas

Además de la generación de proteínas *de novo*, los LLMs son utilizados para múltiples tareas relacionadas al estudio de las éstas, incluyendo la anotación funcional; la predicción de interacciones moleculares; el modelado de la estructura tridimensional; la predicción del efecto de mutaciones; la determinación de homología entre secuencias; e, incluso, como asistentes en el diseño, ejecución y análisis de experimentos [57], [58].

Existen múltiples modelos capaces de predecir o describir las aplicaciones antes mencionadas. Sin embargo, a diferencia de la generación *de novo* de proteínas, éstos comparten el uso de LLMs pre-entrenados con millones de secuencias mediante enmascaramiento; es decir, ocultando residuos durante el entrenamiento para que el modelo aprenda a predecirlos utilizando como contexto el resto de la secuencia (Fig. 3) [59]. La ventaja de este acercamiento permite que los modelos se entrenen bidireccionalmente, a diferencia del entrenamiento autorregresivo donde sólo se utiliza el contexto previo al aminoácido generado. No obstante, por sí solos son incapaces de generar nuevas secuencias, por lo que se consideran como modelos “no generativos”. Gracias a este entrenamiento por enmascaramiento, han demostrado capacidad para abordar tareas complejas como predecir los efectos de mutaciones en la función de una proteína, identificar sitios de unión e incluso generar mapas de contactos de residuos para la predicción estructural tridimensional [50], [60].

En términos generales, existen tres formas de utilizar LLMs preentrenados no generativos en el campo de las proteínas:

1. **Afinamiento (*fine-tuning*):** consiste en una etapa adicional de entrenamiento del modelo, utilizando un nuevo conjunto de datos específico para una aplicación

determinada (Fig. 2A). A modo de ilustración, un modelo de este tipo es ProteinBERT, el cual, por ejemplo, al ser afinado con bases de datos de referencia que describen características estructurales y relaciones evolutivas entre proteínas, mejora su capacidad predictiva en ambas tareas [61].

2. Extracción y utilización de representaciones vectoriales (*embeddings*): los LLMs preentrenados generan representaciones capaces de capturar propiedades estructurales y funcionales de las proteínas. Estas representaciones pueden utilizarse como entrada para modelos externos encargados de realizar tareas específicas, como predecir solubilidad o estabilidad. Por ejemplo, NetSolP utiliza secuencias de proteínas etiquetadas según su expresión y solubilidad para distinguir entre proteínas solubles e insolubles. Para ello, obtiene y promedia las representaciones generadas por un conjunto de modelos de la familia ESM1b, los cuales, posteriormente, son procesados mediante un clasificador lineal ajustado para dicha tarea [62].
3. LLMs en arquitecturas multimodales: En este enfoque, los LLMs se integran con otros módulos especializados dentro de una misma arquitectura, donde cada componente contribuye a la inferencia final. Por ejemplo, el modelo ESMfold predice la estructura tridimensional de una proteína a partir de las representaciones vectoriales generadas por un LLM preentrenado (ESM2), que se procesan secuencialmente en módulos de plegamiento y estructura hasta generar las coordenadas tridimensionales [50].

Por último, es importante mencionar que, si bien los modelos no generativos se utilizan más, los modelos generativos pueden aplicarse para realizar tareas distintas a la generación de secuencias. Un ejemplo es PTM-GPT2, modelo que, mediante instrucciones de entrada (*prompts*), utiliza ProtGPT2 para inferir modificaciones postraduccionales [63]. Como se puede observar en el número de modelos que se crean, la diversidad de aplicaciones posibles y la cantidad masiva de operaciones que se requieren para entrenarlos, surge la necesidad de una infraestructura académica con la capacidad de solventarla.

El papel del supercómputo en los LLMs de proteínas

El uso de los LLMs para el diseño de proteínas y otras aplicaciones no sería posible sin el acceso a infraestructura de cómputo de alto desempeño. A diferencia de los modelos clásicos

de bioinformática, los LLMs requieren ser entrenados con volúmenes masivos de datos y mediante procesos de optimización altamente demandantes desde el punto de vista computacional [64] [65]. En el caso de las proteínas, estos datos incluyen miles de secuencias y estructuras tridimensionales, cuya diversidad y complejidad permiten a los modelos aprender las “reglas” implícitas que gobiernan el plegamiento, la estabilidad y la función de estas macromoléculas [14].

El entrenamiento de estos modelos implica billones de operaciones matemáticas, el manejo eficiente de grandes volúmenes de memoria y el uso intensivo de unidades de procesamiento especializadas, unidades de procesamiento gráfico (GPU, *Graphics Processing Units*) o aceleradores dedicados [66]. Este tipo de cargas de trabajo excede por mucho las capacidades de estaciones de trabajo convencionales y hace indispensable el uso de centros de supercómputo, donde es posible paralelizar cálculos, reducir tiempos de entrenamiento y explorar arquitecturas de modelos cada vez más complejas [67].

En este contexto, los centros de supercómputo académicos juegan un papel estratégico. No sólo democratizan el acceso a capacidades computacionales avanzadas para grupos de investigación y estudiantes, sino que también permiten el desarrollo de modelos abiertos y reproducibles, en contraste con enfoques cerrados dependientes exclusivamente de la industria. Para universidades e instituciones públicas, invertir en infraestructura de supercómputo y en la formación de recursos humanos capaces de aprovecharla representa una oportunidad clave para posicionarse en la frontera del diseño computacional de biomoléculas y de la IA aplicada a las ciencias de la vida [68] [69].

En el contexto de México y América Latina, las universidades públicas han desempeñado históricamente un papel central en el desarrollo del supercómputo académico [70]. Sin embargo, el surgimiento de los LLMs y otras técnicas modernas de IA han introducido nuevas demandas computacionales, particularmente de acceso a GPUs. La ausencia de infraestructura institucional competitiva limita la capacidad académica para entrenar y adaptar modelos avanzados, generando dependencia de recursos externos o comerciales. Esta situación remarca la necesidad de inversión estratégica en infraestructura pública con aceleradores, tanto para fortalecer la investigación y la formación de estudiantes como para asegurar la soberanía tecnológica. Así, el uso de los LLMs muestra que el

supercómputo no es un fin en sí mismo, sino un habilitador de nuevas formas de hacer ciencia.

Desafíos técnicos, reproducibilidad y ética en LLMs de proteínas

A pesar del auge de la IA, persisten desafíos importantes en su aplicación al diseño de proteínas. Muchos procesos biológicos dependen de estados dinámicos difíciles de predecir, lo que refuerza la necesidad de validación experimental rigurosa [1]. Además, los altos costos de entrenamiento y la tendencia hacia modelos con mayor número de parámetros representan una barrera significativa para su implementación en entornos de laboratorio [71].

Los LLMs también pueden heredar sesgos de los datos utilizados en su entrenamiento, lo que limita su capacidad para generar proteínas estables en condiciones extremas o variantes *de novo* alejadas del espacio de secuencias original [72]. Asimismo, la reproducibilidad y la transparencia constituyen retos clave, ya que la falta de documentación de parámetros como semillas aleatorias (*seeds*) o instrucciones de entrada (*prompts*) pueden introducir variabilidad y dificultar la replicación de resultados [43]. A ello, se suman consideraciones éticas y sociales, incluyendo el acceso desigual debido al alto costo computacional, el posible uso indebido con implicaciones en bioseguridad y el impacto ambiental asociado al entrenamiento a gran escala [68].

A pesar de esos retos, la IA y los LLMs continuarán desempeñando un papel central en la investigación científica. Su uso responsable requiere validación experimental, documentación transparente de los métodos empleados y evaluación de los diseños desde una perspectiva de bioseguridad [73]. Más que sustituir el criterio científico, estos modelos deben entenderse como herramientas que amplifican y exigen rigor, responsabilidad y reflexión crítica.

Conclusiones y perspectivas

El uso de LLMs en el diseño de proteínas representa un cambio conceptual importante en la bioquímica y biología computacional. Sistemas originalmente desarrollados para modelar lenguaje humano han demostrado ser capaces de aprender las reglas que organizan secuencias biológicas a partir de grandes volúmenes de datos. Al identificar patrones análogos a los sintácticos y semánticos, estos modelos capturan señales evolutivas, estructurales y funcionales presentes en millones de proteínas naturales, permitiendo generar nuevas secuencias dentro del vasto espacio teórico posible.

Este enfoque modifica la manera en que concebimos el diseño de proteínas, dado que los modelos generativos permiten explorar regiones inexploradas del espacio de secuencia y diseñar proteínas con propiedades específicas. Una diferencia clave entre la generación de secuencias *de novo* mediante LLMs y los enfoques tradicionales como la mutación racional radica en el alcance y rapidez del espacio de exploración. La mutación racional parte de una proteína conocida que consiste en introducir cambios puntuales guiados por conocimiento estructural o funcional, lo que restringe la búsqueda a regiones cercanas del espacio de secuencias y favorece la conservación del plegamiento. En contraste, los LLMs generativos pueden construir secuencias completas residuo por residuo, sin depender de un andamiaje preexistente.

Al aprender regularidades globales a partir de millones de proteínas naturales, estos modelos pueden proponer combinaciones inéditas coherentes con las reglas naturales del plegamiento proteico y funciones biológicas. Así, la convergencia entre IA generativa, grandes bases de datos biológicas y cómputo de alto desempeño configura un nuevo paradigma en el área, con implicaciones potenciales en biotecnología y biomedicina. Sin embargo, estos avances presentan limitaciones importantes. La generación computacional de secuencias no sustituye la validación experimental rigurosa, y aspectos como los sesgos en los datos de entrenamiento, la reproducibilidad, la transparencia metodológica, los costos y el acceso desigual a la infraestructura deben considerarse cuidadosamente. Por ello, el desarrollo y aplicación de LLMs en proteínas requieren estándares claros de evaluación, documentación y bioseguridad. En conjunto, los LLMs constituyen no sólo una herramienta, sino también un nuevo marco conceptual para el diseño de macromoléculas en la era del supercómputo.

Financiamiento

Este trabajo ha sido financiado por SECIHTI (subvención: CBF2023-2024-271) y PAPIIT UNAM (subvenciones: IA203925 e IF201526).

Declaración del uso de IA

Se emplearon herramientas de IA únicamente como apoyo en la revisión de estilo y corrección gramatical. El contenido científico y la redacción original fueron realizados exclusivamente por los autores.

Referencias

- [1] D. Listov, C. A. Goverde, B. E. Correia, and S. J. Fleishman, "Opportunities and challenges in design and optimization of protein function," *Nat. Rev. Mol. Cell Biol.*, vol. 25, no. 8, pp. 639–653, Aug. 2024, doi: 10.1038/s41580-024-00718-y.
- [2] H. Y. Koh *et al.*, "AI-driven protein design," *Nat. Rev. Bioeng.*, vol. 3, no. 12, pp. 1034–1056, Sep. 2025, doi: 10.1038/s44222-025-00349-8.
- [3] K. I. Albanese, S. Barbe, S. Tagami, D. N. Woolfson, and T. Schiex, "Computational protein design," *Nat. Rev. Methods Primer*, vol. 5, no. 1, p. 13, Feb. 2025, doi: 10.1038/s43586-025-00383-1.
- [4] H. Khakzad, I. Igashov, A. Schneuing, C. Goverde, M. Bronstein, and B. Correia, "A new age in protein design empowered by deep learning," *Cell Syst.*, vol. 14, no. 11, pp. 925–939, Nov. 2023, doi: 10.1016/j.cels.2023.10.006.
- [5] T. Kortemme, "De novo protein design—From new structures to programmable functions," *Cell*, vol. 187, no. 3, pp. 526–544, Feb. 2024, doi: 10.1016/j.cell.2023.12.028.

- [6] J. Beck and S. Romero-Romero, "Designing *de novo* TIM barrels: insights into stabilization, diversification, and functionalization strategies," *Biochem. Soc. Trans.*, vol. 54, no. 2, p. BST20253060, Feb. 2026, doi: 10.1042/BST20253060.
- [7] A. Zubiaga, "Natural language processing in the era of large language models," *Front. Artif. Intell.*, vol. 6, p. 1350306, Jan. 2024, doi: 10.3389/frai.2023.1350306.
- [8] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed. 2026. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3>.
- [9] M. Raza, "LLMs vs. SLMs: The Differences in Large & Small Language Models," *Splunk-Blogs*. [Online]. Available: https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html
- [10] J. S. Lee, O. Abdin, and P. M. Kim, "Language models for protein design," *Curr. Opin. Struct. Biol.*, vol. 92, p. 103027, Jun. 2025, doi: 10.1016/j.sbi.2025.103027.
- [11] E. Simon, K. Swanson, and J. Zou, "Language models for biological research: a primer," *Nat. Methods*, vol. 21, no. 8, pp. 1422–1429, Aug. 2024, doi: 10.1038/s41592-024-02354-y.
- [12] Stanford University, "AI Demystified: Introduction to large language models," *Techology training*. [Online]. Available: <https://uit.stanford.edu/service/techtraining/ai-demystified/llm>
- [13] I. A. Blank, "What are large language models supposed to model?," *Trends Cogn. Sci.*, vol. 27, no. 11, pp. 987–989, Nov. 2023, doi: 10.1016/j.tics.2023.08.006.
- [14] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," Jul. 22, 2020, *arXiv:2005.14165*. doi: 10.48550.
- [15] Microsoft, "How Embeddings Extend Your AI Model's Reach," *Learn Microsoft*. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/ai/conceptual/embeddings>

- [16] Microsoft, "Understanding tokens," *Learn Microsoft*. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/ai/conceptual/understanding-tokens>
- [17] G. Zhang, C. Liu, J. Lu, S. Zhang, and L. Zhu, "The Role of AI-Driven De Novo Protein Design in the Exploration of the Protein Functional Universe," *Biology*, vol. 14, no. 9, p. 1268, Sep. 2025, doi: 10.3390/biology14091268.
- [18] Toloka Team, "History of LLMs: Complete Timeline & Evolution (1950-2026)," *Toloka AI Blog*. [Online]. Available: <https://toloka.ai/blog/history-of-llms/>
- [19] K. D. Foote, "A Brief History of Natural Language Processing," *DATAVERSITY*. [Online]. Available: <https://www.dataversity.net/articles/a-brief-history-of-natural-language-processing-nlp/>
- [20] F. Tahir, "Deep Learning Models: CNN, RNN and Transformers," *Medium*. [Online]. Available: <https://medium.com/@fatima.tahir511/deep-learning-models-e07492b02bb0>
- [21] D. Mashru, "Comparative Analysis of CNN, RNN, LSTM, and Transformer Architectures in Deep Learning," *Educ. Adm. Theory Pract.*, pp. 5439–5443, Apr. 2023, doi: 10.53555/kuey.v29i4.10364.
- [22] A. Vaswani *et al.*, "Attention Is All You Need," 2017, *arXiv:1706.03762*. doi: 10.48550
- [23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, *arXiv:1810.04805*. doi: 10.48550.
- [24] A. R. K. Narasimhan, "Improving Language Understanding by Generative Pre-Training," 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:49313245>
- [25] The UniProt Consortium *et al.*, "UniProt: the Universal Protein Knowledgebase in 2025," *Nucleic Acids Res.*, vol. 53, no. D1, pp. D609–D617, Jan. 2025, doi: 10.1093/nar/gkaf1010.

- [26] OpenAI, "What Are Tokens and How to Count them?," *Help-OpenAI*. [Online]. Available: <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>
- [27] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," 2020, *arXiv*:2005.14165. doi: 10.48550.
- [28] Simplifying Future Tech, "LLM Parameters and Hyperparameters Explained," *Medium*. [Online]. Available: <https://medium.com/@SimplifyingFutureTech/llm-parameters-and-hyperparameters-explained-2ba876dbae29>
- [29] IBM, "What is backpropagation?," *IBM-Topics*. [Online]. Available: <https://www.ibm.com/think/topics/backpropagation>
- [30] [30] Z. ul Abideen, "Autoregressive Models for Natural Language Processing," *Medium*. [Online]. Available: <https://medium.com/@zaiinn440/autoregressive-models-for-natural-language-processing-b95e5f933e1f>
- [31] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities," 2024, *arXiv*: 2408.13296. doi: 10.48550.
- [32] A. Takyar, "Optimize to actualize: The impact of hyperparameter tuning on AI," *LeewayHertz-AI Development Company*. [Online]. Available: <https://www.leewayhertz.com/hyperparameter-tuning/>
- [33] A. Babu, "A Comprehensive Guide to Hyperparameter Tuning in Machine Learning," *Medium*. [Online]. Available: <https://medium.com/@aditib259/a-comprehensive-guide-to-hyperparameter-tuning-in-machine-learning-dd9bb8072d02>
- [34] S. Romero-Romero, S. Lindner, and N. Ferruz, "Exploring the Protein Sequence Space with Global Generative Models," *Cold Spring Harb. Perspect. Biol.*, vol. 15, no. 11, p. a041471, Nov. 2023, doi: 10.1101/cshperspect.a041471.

- [35] N. Ferruz, S. Schmidt, and B. Höcker, "ProtGPT2 is a deep unsupervised language model for protein design," *Nat. Commun.*, vol. 13, no. 1, p. 4348, Jul. 2022, doi: 10.1038/s41467-022-32007-7.
- [36] E. Nijkamp, J. Ruffolo, E. N. Weinstein, N. Naik, and A. Madani, "ProGen2: Exploring the Boundaries of Protein Language Models," 2022, *arXiv:2206.13517*. doi: 10.48550.
- [37] A. Bhatnagar *et al.*, "Scaling Unlocks Broader Generation and Deeper Functional Understanding of Proteins," Apr. 16, 2025, *Synthetic Biology*. doi: 10.1101/2025.04.15.649055.
- [38] A. Madani *et al.*, "Large language models generate functional protein sequences across diverse families," *Nat. Biotechnol.*, vol. 41, no. 8, pp. 1099–1106, Aug. 2023, doi: 10.1038/s41587-022-01618-2.
- [39] G. Munsamy *et al.*, "Conditional language models enable the efficient design of proficient enzymes," *Bioinformatics*, May 05, 2024, doi: 10.1101/2024.05.03.592223.
- [40] T. F. Truong and T. Bepler, "PoET: A generative model of protein families as sequences-of-sequences," 2023, *arXiv: 2306.06156*. doi: 10.48550.
- [41] L. Lv *et al.*, "ProLLaMA: A Protein Large Language Model for Multi-Task Protein Language Processing," 2024, *arXiv:2402.16445*. doi: 10.48550/ARXIV.2402.16445.
- [42] R. W. Shuai, J. A. Ruffolo, and J. J. Gray, "IgLM: Infilling language modeling for antibody sequence design," *Cell Syst.*, vol. 14, no. 11, pp. 979-989.e4, Nov. 2023, doi: 10.1016/j.cels.2023.10.001.
- [43] E. Nguyen *et al.*, "Sequence modeling and design from molecular to genome scale with Evo," *Science*, vol. 386, no. 6723, p. eado9336, Nov. 2024, doi: 10.1126/science.ado9336.
- [44] G. Brix *et al.*, "Genome modeling and design across all domains of life with Evo 2," *Genomics*, Feb. 21, 2025, doi: 10.1101/2025.02.18.638918.

- [45] T. F. Truong and T. Bepler, "Understanding protein function with a multimodal retrieval-augmented foundation model," 2025, *arXiv:2508.04724*. doi: 10.48550.
- [46] M. Heinzinger *et al.*, "Bilingual language model for protein sequence and structure," *NAR Genomics and Bioinformatics*, vol. 6, no. 4, p. lqae150, Nov. 2024, doi: 10.1093/nargab/lqae150.
- [47] H. He *et al.*, "De novo generation of SARS-CoV-2 antibody CDRH3 with a pre-trained generative large language model," *Nat. Commun.*, vol. 15, no. 1, p. 6867, Aug. 2024, doi: 10.1038/s41467-024-50903-y.
- [48] X. Wang, Z. Zheng, F. Ye, D. Xue, S. Huang, and Q. Gu, "Diffusion Language Models Are Versatile Protein Learners," 2024, *arXiv:2402.18567*. doi: 10.48550.
- [49] S. Romero-Romero, A. E. Braun, T. Kossendey, N. Ferruz, S. Schmidt, and B. Höcker, "De novo design of triosephosphate isomerases using generative language models," *bioRxiv* Nov. 10, 2024, doi: 10.1101/2024.11.10.622869.
- [50] Z. Lin *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, Mar. 2023, doi: 10.1126/science.ade2574.
- [51] P. Notin *et al.*, "ProteinGym: Large-Scale Benchmarks for Protein Design and Fitness Prediction," *bioRxiv*, Dec. 08, 2023, doi: 10.1101/2023.12.07.570727.
- [52] R. Rao *et al.*, "Evaluating Protein Transfer Learning with TAPE," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 9689–9701, Dec. 2019.
- [53] F. L. Bronnec, A. Verine, B. Negrevertgne, Y. Chevaleyre, and A. Allauzen, "Exploring Precision and Recall to assess the quality and diversity of LLMs," Jun. 04, 2024, *arXiv:2402.10693*. doi: 10.48550/arXiv.2402.10693.
- [54] J. Hon *et al.*, "SoluProt: prediction of soluble protein expression in *Escherichia coli*," *Bioinformatics*, vol. 37, no. 1, pp. 23–28, Apr. 2021, doi: 10.1093/bioinformatics/btaa1102.

- [55] O. Conchillo-Solé, N. S. de Groot, F. X. Avilés, J. Vendrell, X. Daura, and S. Ventura, "AGGRESKAN: a server for the prediction and evaluation of 'hot spots' of aggregation in polypeptides," *BMC Bioinformatics*, vol. 8, p. 65, Feb. 2007, doi: 10.1186/1471-2105-8-65.
- [56] J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, doi: 10.1038/s41586-021-03819-2.
- [57] M. Leclercq and A. Droit, "Protein Language Models: Applications and Perspectives," *J. Proteome Res.*, Dec. 2025, doi: 10.1021/acs.jproteome.5c00506.
- [58] N. Ghosh, D. Santoni, D. Nawn, E. Ottaviani, and G. Felici, "A Comprehensive Review of Transformer-based language models for Protein Sequence Analysis and Design," 2025, *arXiv:2507.13646*. doi: 10.48550.
- [59] J.-Y. Chen, J.-F. Wang, Y. Hu, X.-H. Li, Y.-R. Qian, and C.-L. Song, "Evaluating the advancements in protein language models for encoding strategies in protein function prediction: a comprehensive review," *Front. Bioeng. Biotechnol.*, vol. 13, p. 1506508, Jan. 2025, doi: 10.3389/fbioe.2025.1506508.
- [60] Y. Liu and B. Tian, "Protein–DNA binding sites prediction based on pre-trained protein language model and contrastive learning," *Brief. Bioinform.*, vol. 25, no. 1, p. bbad488, Nov. 2023, doi: 10.1093/bib/bbad488.
- [61] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, "ProteinBERT: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, Apr. 2022, doi: 10.1093/bioinformatics/btac020.
- [62] V. Thumuluri, H.-M. Martiny, J. J. Almagro Armenteros, J. Salomon, H. Nielsen, and A. R. Johansen, "NetSolP: predicting protein solubility in *Escherichia coli* using language models," *Bioinformatics*, vol. 38, no. 4, pp. 941–946, Jan. 2022, doi: 10.1093/bioinformatics/btab801.
- [63] P. Shrestha, J. Kandel, H. Tayara, and K. T. Chong, "Post-translational modification prediction via prompt-based fine-tuning of a GPT-2 model," *Nat. Commun.*, vol. 15, no. 1, p. 6699, Aug. 2024, doi: 10.1038/s41467-024-51071-9.

- [64] K. M. Tolle, D. S. W. Tansley, and A. J. G. Hey, "The Fourth Paradigm: Data-Intensive Scientific Discovery [Point of View]," *Proc. IEEE*, vol. 99, no. 8, pp. 1334–1337, Aug. 2011, doi: 10.1109/JPROC.2011.2155130.
- [65] J. Dongarra *et al.*, "The International Exascale Software Project roadmap," *Int. J. High Perform. Comput. Appl.*, vol. 25, no. 1, pp. 3–60, Feb. 2011, doi: 10.1177/1094342010391989.
- [66] C. T. Lee and R. E. Amaro, "Exascale Computing: A New Dawn for Computational Biology," *Comput. Sci. Eng.*, vol. 20, no. 5, pp. 18–25, Sep. 2018, doi: 10.1109/MCSE.2018.05329812.
- [67] Big Data Interagency Working Group, High End Computing Interagency Working Group, Networking & Information Technology Research & Development Subcommittee, and Committee On Science & Technology Enterprise Of The National Science & Technology Council, *The Convergence of High Performance Computing, Big Data, and Machine Learning*. U.S Government, Sep. 2019. [Online]. Available: <https://www.nitrd.gov/pubs/Convergence-HPC-BD-ML-JointWSreport-2019.pdf>
- [68] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models," Jul. 12, 2022, *arXiv:2108.07258*. doi: 10.48550.
- [69] UNESCO, "Implementation of the UNESCO Recommendation on Open Science," UNESCO - Open Science. [Online]. Available: <https://www.unesco.org/en/open-science/implementation>
- [70] A. Santillán González and L. Hernández-Cervantes, "Hitos del supercómputo: del Giga al Exascale," *EPISTEMUS*, vol. 18, no. 35, Dec. 2023, doi: 10.36790/epistemus.v18i35.300.
- [71] S. Ren, B. Tomlinson, R. W. Black, and A. W. Torrance, "Reconciling the contrasting narratives on the environmental impact of large language models," *Sci. Rep.*, vol. 14, no. 1, p. 26310, Nov. 2024, doi: 10.1038/s41598-024-76682-6.

- [72] Y. Shen, G. Kudla, and D. A. Oyarzún, "Improving the generalization of protein expression models with mechanistic sequence information," *Nucleic Acids Res.*, vol. 53, no. 3, p. gkaf020, Jan. 2025, doi: 10.1093/nar/gkaf020.
- [73] P. Hunter, "Security challenges by AI-assisted protein design: The ability to design proteins in silico could pose a new threat for biosecurity and biosafety," *EMBO Rep.*, vol. 25, no. 5, pp. 2168–2171, Mar. 2024, doi: 10.1038/s44319-024-00124-7.
- [74] X. Fang *et al.*, "A method for multiple-sequence-alignment-free protein structure prediction using a protein language model," *Nat. Mach. Intell.*, vol. 5, no. 10, pp. 1087–1096, Oct. 2023, doi: 10.1038/s42256-023-00721-6.
- [75] H. Ghazikhani and G. Butler, "TooT-BERT-M: Discriminating Membrane Proteins from Non-Membrane Proteins using a BERT Representation of Protein Primary Sequences," in *2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Ottawa, ON, Canada: IEEE, Aug. 2022, pp. 1–8. doi: 10.1109/CIBCB55180.2022.9863026.
- [76] N. Buton, F. Coste, and Y. Le Cunff, "Predicting enzymatic function of protein sequences with attention," *Bioinformatics*, vol. 39, no. 10, p. btad620, Oct. 2023, doi: 10.1093/bioinformatics/btad620.
- [77] Q. Liu, C. Zhang, and L. Freddolino, "InterLabelGO+: unraveling label correlations in protein function prediction," *Bioinformatics*, vol. 40, no. 11, p. btae655, Nov. 2024, doi: 10.1093/bioinformatics/btae655.
- [78] G. Li, S. Yao, and L. Fan, "ProSTAGE: Predicting Effects of Mutations on Protein Stability by Using Protein Embeddings and Graph Convolutional Networks," *J. Chem. Inf. Model.*, vol. 64, no. 2, pp. 340–347, Jan. 2024, doi: 10.1021/acs.jcim.3c01697.
- [79] Y. Luo, Z. Nie, M. Hong, S. Zhao, H. Zhou, and Z. Nie, "MutaPLM: Protein Language Modeling for Mutation Explanation and Engineering," 2024, *arXiv:2410.22949*. doi: 10.48550.

- [80] F. Teufel *et al.*, "SignalP 6.0 predicts all five types of signal peptides using protein language models," *Nat. Biotechnol.*, vol. 40, no. 7, pp. 1023–1025, Jul. 2022, doi: 10.1038/s41587-021-01156-3.
- [81] H. K. Wayment-Steele *et al.*, "Learning millisecond protein dynamics from what is missing in NMR spectra," *bioRxiv*, Mar. 19, 2025, doi: 10.1101/2025.03.19.642801.
- [82] W. Liu *et al.*, "PLMSearch and PLMAlign: Protein Language Model (PLM)-Based Homologous Protein Sequence Search and Alignment," in *Large Language Models (LLMs) in Protein Bioinformatics*, vol. 2941, D. B. Kc, Ed., in *Methods in Molecular Biology*, vol. 2941, New York, NY: Springer US, 2025, pp. 227–241. doi: 10.1007/978-1-0716-4623-6_14.
- [83] U. Lupo, D. Sgarbossa, and A.-F. Bitbol, "Pairing interacting protein sequences using masked language modeling," *Proc. Natl. Acad. Sci.*, vol. 121, no. 27, p. e2311887121, Jul. 2024, doi: 10.1073/pnas.2311887121.
- [84] S. J. Giri, N. Ibtehaz, and D. Kihara, "GO2Sum: generating human-readable functional summary of proteins from GO terms," *Npj Syst. Biol. Appl.*, vol. 10, no. 1, p. 29, Mar. 2024, doi: 10.1038/s41540-024-00358-0.



Uso de supercómputo para el descubrimiento de fármacos: un caso de estudio sobre CK1ε

Luis Córdova Bahena

Universidad Nacional Autónoma de México, Facultad de Medicina,
Ciudad de México, México.

ORCID: 0000-0003-2666-6559

Axel Adrián Sánchez Álvarez

Universidad Nacional Autónoma de México, Facultad de Medicina,
Ciudad de México, México.

ORCID: 0009-0009-7395-5960

Marco Antonio Velasco Velázquez

Universidad Nacional Autónoma de México, Facultad de Medicina,
Ciudad de México, México.

ORCID: 0000-0001-9717-0265

Recepción: 23 de enero de 2026.

Aceptación: 21 de agosto de 2026.

Agosto 2026 • número de revista 17 • DOI: 10.22201/dgtic.26832968e.2026.17.129

Uso de supercómputo para el descubrimiento de fármacos: un caso de estudio sobre CK1ε

Resumen

La caseína cinasa 1 épsilon (CK1ε) es una enzima que promueve la progresión de diversos tumores, por lo que es considerada una diana terapéutica. Sin embargo, aún no existen fármacos con uso clínico contra CK1ε. En este estudio, utilizamos una estrategia de cribado virtual para identificar potenciales inhibidores de CK1ε. Utilizando recursos de supercómputo para realizar simulaciones de dinámica molecular, identificamos las interacciones que median la unión entre CK1ε e inhibidores reportados. Con estos datos, generamos un modelo de farmacóforo que empleamos para identificar candidatos dentro de grandes bases de compuestos. A continuación, realizamos acoplamiento molecular seguido de cálculos de propiedades ADMET y microespecies predominantes. Finalmente, usamos nuevamente recursos de supercómputo para caracterizar detalladamente el modo de unión de los compuestos más prometedores para inhibir a CK1ε, mediante simulaciones de dinámica molecular. Como resultado, identificamos tres compuestos con potencial actividad inhibitoria de CK1ε. Nuestros hallazgos ilustran cómo el proceso de cribado virtual, empleando modelos de farmacóforo y simulaciones de dinámica molecular realizadas con recursos de supercómputo permite identificar compuestos con potencial actividad inhibitoria sobre CK1ε. Además, discutimos el papel del supercómputo en la viabilidad de este tipo de estrategias, particularmente en estudios que requieren realizar simulaciones de dinámica molecular. La estrategia empleada en este trabajo aprovecha las capacidades de supercomputadoras para el descubrimiento de nuevos fármacos.

Palabras Clave: cáncer, cribado virtual, descubrimiento de fármacos asistido por computadora, reposicionamiento de fármacos, supercómputo.

Use of supercomputing for drug discovery: A case study on CK1ε

Abstract

Casein kinase 1 epsilon (CK1ε) is an enzyme that promotes the progression of various tumors and is therefore considered a therapeutic target. However, there are currently no drugs approved for clinical use against CK1ε. In this study, we employed a virtual screening strategy to identify potential CK1ε inhibitors. Using supercomputing resources to perform molecular dynamics simulations, we identified the interactions that mediate binding between CK1ε and previously reported inhibitors. Based on these data, we generated a pharmacophore model that was used to identify candidate compounds within ultra-large chemical libraries. Subsequently, we carried out molecular docking followed by calculations of ADMET properties and predominant microspecies. Finally, we again used supercomputing resources to characterize in detail the binding modes of the most promising compounds for CK1ε inhibition through molecular dynamics simulations. As a result, we identified three compounds with potential inhibitory activity against CK1ε. Our findings illustrate how the combined use of pharmacophore models and molecular dynamics simulations supported by supercomputing resources can be used to prioritize candidates for drug discovery. In addition, we discuss the role of supercomputing in enabling these strategies, particularly in studies that require the performance of molecular dynamics simulations. The strategy employed in this work takes advantage of supercomputer capabilities for drug discovery.

Keywords: Cancer, Virtual screening, Computer-aided drug discovery, Drug repurposing, Supercomputing.

Introducción

1. Descubrimiento de fármacos asistido por computadora

El descubrimiento de fármacos es un proceso largo y costoso. Se estima que el desarrollo de cada fármaco toma de 10-17 años y requiere de 1000-2000 millones de dólares en

investigación [1]. Ante esta problemática, el uso de herramientas computacionales es una alternativa para acelerar y optimizar estos procesos. El descubrimiento de fármacos asistido por computadora emplea distintas técnicas para predecir el potencial como fármaco de compuestos candidatos. En este estudio, empleamos diversos métodos computacionales, que describimos brevemente.

1.1 Acoplamiento molecular. Esta técnica se emplea para descifrar el modo de unión entre dos o más moléculas. En el campo de la farmacología, usamos acoplamiento molecular para predecir el modo de unión entre una molécula tipo fármaco (también llamada ligando) y una proteína que participa en el desarrollo o progresión de una enfermedad y, por lo tanto, es una diana terapéutica de interés. Estos experimentos computacionales usan programas diseñados específicamente para modelar acoplamiento molecular. Los distintos programas disponibles tienen en común que: i) usan algoritmos de búsqueda definidos para explorar las distintas conformaciones espaciales en las que un ligando puede acoplarse en una diana terapéutica rígida; e ii) incluyen funciones de evaluación para “calificar” cuáles de esas conformaciones son mejores. La matemática subyacente de las funciones de evaluación suele sobresimplificar algunos fenómenos, haciendo que los cálculos necesarios sean sencillos de llevar a cabo [2]. Por ejemplo, el efecto del solvente en el sistema de estudio puede ser emulado, ajustando los valores obtenidos usando una constante dieléctrica. La ventaja de estas simplificaciones es que hacen que los acoplamiento molecular tengan un bajo costo computacional. Típicamente, el proceso conlleva unos cuantos miles de cálculos, por lo que una computadora personal puede realizarlos en pocos minutos [3]. Entonces, los acoplamiento molecular frecuentemente se emplean como una primera aproximación para caracterizar la posible interacción entre un fármaco y su objetivo farmacológico. Sin embargo, este enfoque simplifica en exceso el problema biológico, entre otros aspectos, al considerar la presencia de sólo dos actores (la biomolécula y el ligando). Por esta razón, es necesario recurrir a otras técnicas para profundizar en el estudio de estas interacciones.

1.2 Simulaciones de Dinámica Molecular (SDM). Las SDM son métodos computacionales que predicen el movimiento de un sistema en función del tiempo. En el proceso de descubrimiento de fármacos, los sistemas a estudiar son la proteína de interés (en ausencia de un ligando) y el complejo ligando-proteína. Dado que en este método se estudia la movilidad de ambos sistemas, es posible observar cambios conformacionales que afectan la

estabilidad del complejo. Para mejorar la simulación, se incorporan iones al sistema para emular el ambiente de un sistema fisiológico. Además, se usa un modelo de solvente explícito en los cálculos.

En una SDM, se modela cada átomo de las moléculas en el sistema como una esfera individual. Un ligando tiene entre 100 y 200 átomos, y una proteína blanco cientos de miles. Además, deben considerarse las centenas de moléculas que se agregan al sistema: moléculas de agua como solvente (cada una con tres átomos) y un átomo por cada ion agregado.

Asimismo, en las SDM se lleva a cabo, para cada átomo del sistema, la resolución numérica de las ecuaciones de movimiento de Newton. Las fuerzas que actúan sobre cada átomo tienen diversas contribuciones; por ejemplo, el potencial de Lennard-Jones —que modela las fuerzas de atracción y repulsión entre átomos no enlazados como una función continua de la distancia interatómica— y los términos electrostáticos. A partir del cálculo de esas fuerzas, se integran las ecuaciones de movimiento para obtener la posición y la velocidad de cada átomo en función del tiempo. Este procedimiento permite simular, entre otras cosas, los cambios conformacionales que tendrán los componentes del sistema en el tiempo. Debido a todo lo anterior, las SDM son costosas desde el punto de vista computacional. El uso de laboratorios de supercómputo permite destinar múltiples procesadores (32 o 64 procesadores CPU) trabajando simultáneamente en la resolución de los sistemas de ecuaciones de los campos de fuerza. En esas condiciones, es posible simular 10 nanosegundos (un cienmillonésimo de segundo) por día. Sin embargo, para estudiar la unión de compuestos tipo fármaco a su proteína blanco, se requieren SDM que van de centenas de nanosegundos hasta un microsegundo (1000 nanosegundos). Por otro lado, el desarrollo de arquitecturas de cómputo basadas en procesadores gráficos (GPU) ha permitido acelerar de manera significativa la producción de SDM. En comparación con implementaciones tradicionales basadas únicamente en CPU, el uso de GPU puede incrementar la eficiencia computacional en uno o más órdenes de magnitud, reduciendo de forma sustancial los tiempos de cálculo requeridos. Además, es común realizar SDM independientes adicionales debido a que las velocidades iniciales de los átomos son tratadas como vectores y las direcciones de éstos son asignadas aleatoriamente. Aunque es prácticamente imposible replicar el experimento, estas repeticiones permiten evaluar la convergencia y reproducibilidad de los resultados. Si bien es cierto que algunas etapas del descubrimiento de fármacos asistido por computadora pueden realizarse en equipos

convencionales, los estudios que involucran simulaciones de dinámica molecular requieren recursos de supercómputo. Por ejemplo, una simulación de cientos de nanosegundos puede requerir semanas o meses de cálculo en una computadora personal, mientras que el uso de arquitecturas paralelas basadas en CPU y GPU permite reducir significativamente estos tiempos. Esto no sólo acelera los cálculos, sino que también hace posible realizar réplicas, analizar múltiples sistemas y explorar una mayor parte del espacio químico.

1.3 Generación de modelos de farmacóforo. Para que un fármaco tenga actividad biológica, debe unirse a su diana terapéutica. La unión de fármacos a biomoléculas ocurre a través de interacciones intermoleculares entre las que destacan los enlaces de hidrógeno, interacciones de apilamiento, e interacciones hidrofóbicas. Los enlaces de hidrógeno son interacciones atractivas en las que participan tres átomos organizados en cierta conformación espacial: un átomo aceptor de enlaces de hidrógeno (AH), que cuenta con un par electrónico libre; un átomo de hidrógeno; y un átomo donador de enlaces de hidrógeno (DH), que es al que está unido covalentemente el hidrógeno. Por otro lado, los apilamientos son interacciones atractivas entre las nubes electrónicas de dos anillos aromáticos (Aro). Finalmente, las interacciones hidrofóbicas (Hid) ocurren principalmente entre regiones apolares de las moléculas. Estas interacciones están asociadas a la tendencia del agua a favorecer contactos con grupos polares, promoviendo así la asociación entre regiones hidrofóbicas. Las interacciones intermoleculares también permiten que la unión entre el ligando y la proteína sea estable en el tiempo.

Si se definen las interacciones que permiten el óptimo anclaje de un ligando en una diana terapéutica, entonces se puede construir un modelo que ubique determinadas características electrónicas (las que definen las interacciones intermoleculares) en el espacio. Ese modelo, al que se le conoce como farmacóforo, se puede utilizar como plantilla para buscar compuestos que tengan las características deseadas con la hipótesis de que podrán unirse a la proteína de interés.

1.4 Cálculo de propiedades ADMET. Las propiedades ADMET son características de los compuestos asociadas a los procesos de absorción, distribución, metabolismo, eliminación y toxicidad. Entonces, el cálculo de esas propiedades permite predecir el comportamiento de un compuesto una vez que se administre a un sistema biológico. En el descubrimiento de fármacos, la evaluación temprana de estas propiedades es esencial para estimar la

probabilidad de que un compuesto pueda llegar a su proteína objetivo sin generar efectos adversos. Diversas herramientas computacionales permiten calcular propiedades ADMET, como solubilidad, permeabilidad, lipofilidad, estabilidad metabólica, y capacidad de atravesar barreras biológicas, partiendo únicamente de la estructura química de los compuestos candidatos.

1.5 Cálculo de microespecies predominantes. Una misma molécula puede existir en diferentes formas de protonación en función del pH del medio en el que se encuentre. A esas formas con distinto estado de protonación, les llamamos microespecies. En el descubrimiento de fármacos, la identificación de microespecies es crucial pues cambios en la carga y la distribución electrónica de un ligando influyen en su capacidad para establecer interacciones intermoleculares (por ejemplo, enlaces de hidrógeno) con una biomolécula. En particular, nos interesa conocer cuál es la especie que existiría más frecuentemente en las condiciones de pH en las que se encuentra la proteína blanco durante la enfermedad que pretendemos tratar. De esta manera, podemos describir de manera más realista el fenómeno de unión ligando-proteína.

2. La caseína cinasa 1 épsilon (CK1ε) es una diana terapéutica.

CK1ε es una proteína perteneciente a la superfamilia de cinasas (enzimas que fosforilan a otras moléculas). CK1ε participa en la regulación de múltiples procesos fisiológicos entre los que destacan el control de la proliferación celular y la señalización circadiana. Específicamente, CK1ε modula vías de señalización como la Wnt/β-catenina y Hedgehog, mediante la fosforilación de proteínas que regulan múltiples procesos celulares. En el contexto oncológico, la desregulación de CK1ε se ha asociado con la progresión tumoral en distintos tipos de cáncer. Su actividad aberrante favorece la proliferación celular, resistencia a la apoptosis, migración e invasión tumoral, pues activa sostenidamente rutas de señalización oncogénicas. Diversos estudios han demostrado que la inhibición farmacológica o genética de CK1ε reduce la progresión de tumores en modelos experimentales. Por esto, CK1ε es considerada una diana terapéutica de interés en cáncer [4].

La estructura tridimensional de CK1ε ya ha sido reportada. Está constituida por dos lóbulos, uno organizado principalmente en hojas beta y otro en hélices alfa. Como todas las

cinastas, CK1ε requiere la unión de una molécula de ATP, la cual actúa como donador del grupo fosfato que es transferido a las proteínas blanco durante la reacción enzimática. El sitio de unión a ATP en CK1ε se localiza en la hendidura formada entre los dos lóbulos. Esa región reconoce tanto el grupo adenina como los grupos fosfato del ATP. Si se bloquea la unión entre el ATP y CK1ε, se inhibe la actividad enzimática [5]. Convenientemente, existe una región cóncava contigua al sitio de unión a ATP que tiene propiedades adecuadas para unir a moléculas tipo fármaco, por lo que se han descubierto inhibidores de la enzima que se unen a esa cavidad hidrofóbica e impiden la unión del ATP.

En este estudio, aprovechamos el conocimiento existente de la estructura de CK1ε para identificar candidatos con potencial actividad farmacológica inhibitoria sobre la enzima. Para esto, implementamos una estrategia computacional que se conoce como cribado virtual (CV). El CV integra de manera secuencial distintos filtros computacionales que emplean los métodos descritos arriba. Cada uno de estos filtros contribuye a refinar progresivamente el conjunto de compuestos candidatos, lo que permite evaluar grandes bibliotecas de compuestos. Al final, se espera que los candidatos seleccionados por CV presenten afinidad y estabilidad en la formación de complejos proteína-ligando y tengan un perfil farmacocinético favorable. Por lo tanto, el CV es una estrategia racional y eficiente que se usa en etapas tempranas del descubrimiento de fármacos.

Los resultados farmacológicos descritos en este trabajo fueron publicados previamente por nuestro grupo en estudios especializados [6,7]. En el presente artículo, utilizamos dichos resultados como caso de estudio para ilustrar el papel del supercómputo en el proceso de cribado virtual y su contribución al descubrimiento de fármacos asistido por computadora.

Desarrollo

En este proyecto, se implementó una metodología computacional para la identificación de compuestos con potencial para inhibir la actividad de la enzima CK1ε. Para comenzar, seleccionamos cinco inhibidores de CK1ε reportados previamente en la literatura científica. El primero de los inhibidores (IN1) se encuentra co-cristalizado con CK1ε [8] y la estructura del complejo está disponible en la base de datos de Protein Data Bank con código PDB 4HNI

(Figura 1A). Los inhibidores 2, 3, 4, y 5 (IN2-5) [9]—[11] fueron seleccionados por su selectividad (capacidad para inhibir a CK1ε y no hacerlo en otros miembros de la familia) y su potencia (concentración mínima de fármaco que puede ejercer efecto inhibitorio). Los modos de unión de IN2-5 en CK1ε fueron obtenidos mediante acoplamiento molecular. Los resultados indican que todos los IN1-5 se unen a la región de CK1ε que media la unión de ATP. Cabe resaltar que el conjunto completo de inhibidores forma enlaces de hidrógeno con el asa que interconecta los lóbulos alfa y beta. Por otro lado, el IN1 mantiene un anillo fuera de la cavidad hidrofóbica contigua al sitio de unión a ATP en CK1ε, mientras que los IN2, IN3, e IN5 insertan sus anillos en esa cavidad. El IN4 cubre ambas regiones con sus anillos (Figura 1A).

Tras definir los modos de unión de los IN1-5, realizamos una serie de SDM empleando el equipo de supercómputo del Laboratorio Nacional de Cómputo de Alto Desempeño (LANCAD) Mitzli. La cantidad de conformaciones analizadas y la duración de las trayectorias hacen que este tipo de estudios resulte difícil de implementar en equipos de cómputo convencionales dentro de tiempos razonables. El uso de supercómputo permitió procesar simultáneamente múltiples SDM y analizar decenas de miles de conformaciones moleculares. Nuestras simulaciones incluyeron 20,000 conformaciones CK1ε-IN para cada uno de los complejos, correspondientes a trayectorias de 200 ns. En cada una de esas 20,000 conformaciones, estudiamos las interacciones intermoleculares que se forman entre el ligando y la proteína. Encontramos que todos los inhibidores forman enlaces de hidrógeno con el residuo E85 —localizado en el asa de interconexión de los lóbulos— con una prevalencia superior al 90 % del total de conformaciones en cada simulación. Esto indica que dicha asa juega un papel clave en el modo de unión de los distintos inhibidores. Además, todos los inhibidores forman interacciones hidrofóbicas con el residuo I23 —localizado al fondo de la cavidad hidrofóbica— con una prevalencia superior al 50 % y con el residuo I148 —localizado en la región opuesta a la cavidad hidrofóbica— con una prevalencia de alrededor del 20 %. Por otro lado, IN1 frecuentemente forma enlaces de hidrógeno con el residuo E83, reforzando la participación del asa de interconexión en el modo de unión del inhibidor. Asimismo, IN1-3 e IN5 forman interacciones hidrofóbicas de alta prevalencia con el residuo A36, resaltando el papel de la cavidad hidrofóbica como sitio de estabilización para el modo de anclaje de los inhibidores. Finalmente, los inhibidores IN3-5 forman un apilamiento con el residuo F20, que, espacialmente, se encuentra cercano a I148 en la región

expuesta al solvente, lo que indica que esa región puede formar interacciones hidrofóbicas y/o de apilamiento (Figura 1B).

Los datos generados en el análisis de interacciones intermoleculares soportaron la generación de un modelo de farmacóforo. Nuestro modelo incluyó seis elementos. Los tres primeros son: un donador de enlaces de hidrógeno (DH1), un aceptor de enlaces de hidrógeno (AH), y un segundo donador de enlaces de hidrógeno (DH2) en una configuración espacial lineal. Estos elementos son necesarios para formar interacciones óptimas con el asa de interconexión de los lóbulos de CK1ε. El cuarto elemento del farmacóforo es un anillo aromático (Aro1) que da rigidez y estabilidad a los tres primeros elementos. El quinto elemento es una característica hidrofóbica para ocupar dicha cavidad de CK1ε, y el sexto elemento es un segundo anillo aromático (Aro2) para formar apilamientos y/o interacciones hidrofóbicas en la región expuesta al solvente (Figura 1C).

Posteriormente, seleccionamos dos bases de datos de compuestos como universo de búsqueda. Una de ellas (DrugBank) contiene fármacos aprobados por la FDA y fue seleccionada con el objetivo de identificar nuevas actividades de fármacos con uso clínico. La otra (PubChem) incluye millones de compuestos diversos (Figura 1D).

Entonces, partiendo de un gran grupo de compuestos, implementamos una estrategia de cribado virtual (CV). El CV utiliza secuencialmente diferentes técnicas computacionales para identificar los compuestos con mejor potencial para convertirse en fármacos. En nuestro primer filtrado, comparamos las propiedades químicas y espaciales del modelo de farmacóforo contra más de 100 millones de compuestos, lo que nos permitió identificar 298 compuestos con las características deseadas. En el segundo filtrado, calculamos computacionalmente las propiedades fisicoquímicas de éstos para encontrar aquellos que cumplieran con perfiles favorables previamente reportados, como las reglas de Lipinski y/o reglas de Veber que establecen criterios recomendados para fármacos con vía de administración oral. De los 290 compuestos identificados en el paso anterior, sólo 74 tuvieron propiedades adecuadas para permanecer en el estudio. En el tercer filtro, calculamos las especies predominantes de cada candidato. Una molécula puede existir con diferentes cargas en diferentes condiciones; por esto, los fármacos deben diseñarse para que, en las condiciones del organismo, tengan las características deseadas. Seleccionamos aquellos compuestos que, a los valores de pH que existen en el ambiente tumoral, tuvieron

cargas (estados de protonación) compatibles con las características DH1-AH-DH2 del farmacóforo. De los 74 compuestos identificados en el paso anterior, sólo 31 presentaron la configuración deseada y fueron conservados. Entre estos 31 compuestos, se incluyen dos fármacos con uso clínico indicado como antivirales más 29 compuestos diversos (C1-29). Finalmente, los clasificamos en grupos y llevamos a cabo una serie de SDM de algunos representativos, usando nuevamente recursos de supercómputo provistos por el LANCAD (Figura 1E). En las trayectorias de las SDM, identificamos subintervalos donde los compuestos mantenían un modo de unión estable y calculamos el cambio en la energía libre de Gibbs (ΔG de unión). Valores más negativos de ΔG de unión indican que la formación del complejo proteína-ligando es termodinámicamente más favorable y, por lo tanto, sugieren una mayor afinidad entre ambas moléculas. Identificamos que la etravirina —un fármaco empleado para el tratamiento de pacientes con VIH— presentó una afinidad hacia CK1ε equiparable a la de IN1, el cual usamos como referencia comparativa [6]. Por otro lado, los compuestos C14 y C15, que comparten un núcleo estructural común, mostraron un ΔG de unión favorable [7] (Figura 1F).

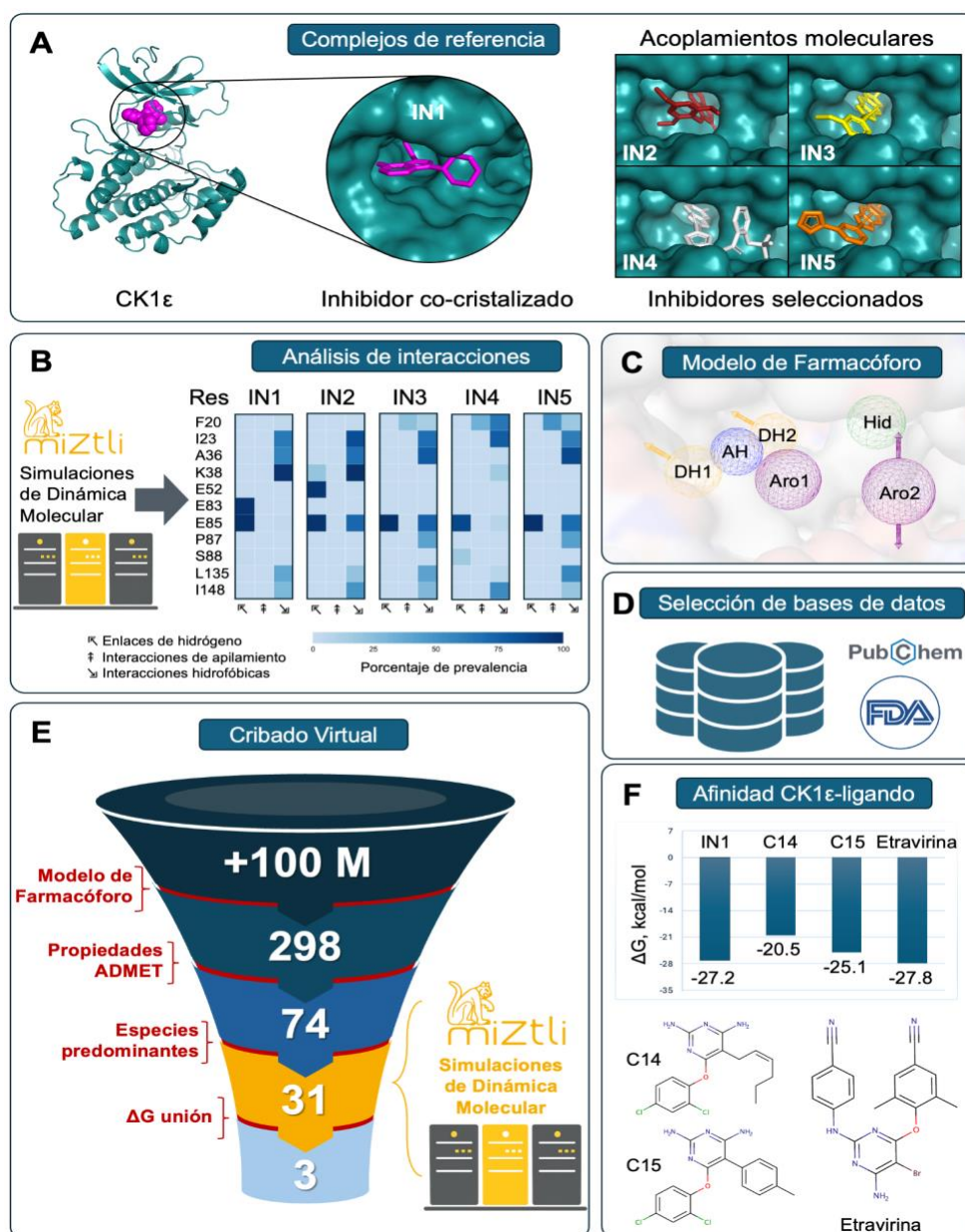


Fig. 1. Identificación de potenciales agentes antineoplásicos mediante cribado virtual. A) Selección de los inhibidores de referencia e identificación de su modo de unión a CK1ε. El complejo CK1ε-inhibidor 1 se reportó previamente, mientras que los complejos de los inhibidores 2-5 fueron obtenidos por acoplamiento molecular.

B) Análisis de las interacciones intermoleculares obtenidas de las simulaciones de dinámica molecular. C) Modelo de farmacóforo generado a partir de inhibidores de CK1ε. D) Selección de universo de búsqueda. E) Protocolo de cribado virtual. F) Energía de unión de los mejores candidatos, calculada a partir de simulaciones de dinámica molecular.

Aunque las herramientas computacionales permiten reducir costos y acelerar etapas tempranas del descubrimiento de fármacos, sus resultados dependen de aproximaciones realizadas con modelos matemáticos que simplifican sistemas biológicos complejos. Por ello, es importante interpretar los resultados de cribado virtual como hipótesis de trabajo que requieren validación experimental posterior.

Conclusión

Presentamos aquí un flujo de trabajo para la identificación de potenciales inhibidores de CK1ε, una enzima clave en la activación de genes asociados a la progresión de diversos tipos de tumores. El conjunto de experimentos computacionales empleados incluye simulaciones complejas que requieren el uso de equipos de supercómputo, resaltando el papel del supercómputo en el descubrimiento de fármacos asistido por computadora.

Nuestros resultados ponen de manifiesto cómo el uso de estrategias de cribado virtual apoyadas en recursos de supercómputo permiten priorizar compuestos con potencial actividad inhibitoria sobre CK1ε. Sin embargo, los candidatos identificados requieren validación experimental adicional para confirmar su actividad biológica y evaluar su posible aplicación terapéutica. Los resultados relacionados con la identificación de etravirina como potencial inhibidor de CK1ε fueron reportados previamente por nuestro grupo [6]. De manera similar, la identificación de los compuestos C14 y C15 fue publicada posteriormente en [7]. En el presente trabajo, utilizamos estos estudios como caso de estudio para ilustrar cómo el uso combinado de cribado virtual y recursos de supercómputo puede emplearse en el descubrimiento de fármacos asistido por computadora. Este caso de estudio ilustra el papel del supercómputo en el proceso de descubrimiento de fármacos asistido por computadora.

Agradecimientos

Este proyecto fue realizado gracias a los recursos otorgados por el proyecto LANCAD-UNAM-DGTIC-389.

Referencias

- [1] M. Dickson and J. P. Gagnon, "Key factors in the rising cost of new drug discovery and development," *Nat. Rev. Drug Discov.* vol. 3, No. 5, pp. 417-429, May 2004, doi:10.1038/nrd1382.
- [2] R. Wang, Y. Lu , and S. Wang, "Comparative Evaluation of 11 Scoring Functions for Molecular Docking," *J. Med. Chem.*, vol. 46 no. 12, pp. 2287-2303, May 2003, doi:10.1021/jm0203783.
- [3] F. D. Prieto-Martínez, M. Arciniega, J. L. Medina-Franco. "Acoplamiento Molecular: Avances Recientes y Retos," *TIP Rev. Espec. en Ciencias Químico-Biológicas.* vol. 21, pp. 65-87, 2018, doi:10.22201/fesz.23958723e.2018.0.143.
- [4] N. H. Long and S.-J. Lee, "Targeting casein kinase 1 for cancer therapy: current strategies and future perspectives," *Front. Oncol.*, vol. 13, pp. 1-8 Nov. 2023, doi: 10.3389/fonc.2023.1244775.
- [5] Y. K. Sunkari, L. Meijer, and M. Flajolet, "The protein kinase CK1: Inhibition, activation, and possible allosteric modulation," *Front. Mol. Biosci.*, vol. 9, pp. 1-10 Aug. 2022, doi: 10.3389/fmolb.2022.916232.
- [6] L. Córdova-Bahena, A. A. Sánchez-Álvarez, A. J. Ruiz-Moreno, and M. A. Velasco-Velázquez, "Repositioning of Etravirine as a Potential CK1ε Inhibitor by Virtual Screening," *Pharmaceuticals (Basel)*, vol. 15, no. 1, p. 8, Jan. 2022, doi: 10.3390/ph15010008.
- [7] A. A. Sánchez-Álvarez, M. A. Velasco-Velázquez, and L. Cordova-Bahena, "In Silico Identification of 2,4-Diaminopyrimidine-Based Compounds as Potential CK1ε Inhibitors," *Pharmaceuticals (Basel)*, vol. 18, no. 5, p. 741, May 2025, doi: 10.3390/ph18050741.

- [8] K. M. Walton, K. Fisher, D. Rubitski, M. Marconi, Q. J. Meng, M. Sládek, *et al.*, "Selective Inhibition of Casein Kinase 1 ϵ Minimally Alters Circadian Clock Period," *J. Pharmacol. Exp. Ther.*, vol. 330, no. 2, pp. 430–439, Aug. 2009, doi: 10.1124/jpet.109.151415.
- [9] J. Bischof, J. Leban, M. Zaja, A. Grothey, B. Radunsky, O. Othersen *et al.*, "2-Benzamido-N-(1H-benzo[d]imidazol-2-yl)thiazole-4-carboxamide derivatives as potent inhibitors of CK1 δ/ϵ ," *Amino Acids*, vol. 43, no. 4, pp. 1577–1591, Oct. 2012, doi: 10.1007/s00726-012-1234-x.
- [10] N. Mashhoon, A. J. DeMaggio, V. Tereshko, S. C. Bergmeier, M. Egli, M. F. Hoekstra *et al.*, "Crystal Structure of a Conformation-selective Casein Kinase-1 Inhibitor," *J. Biol. Chem.*, vol. 275, no. 26, pp. 20052–20060, Jun. 2000, doi: 10.1074/jbc.M001713200.
- [11] J. Richter, J. Bischof, M. Zaja, H. Kohlhof, O. Othersen, D. Vitt, *et al.*, "Difluoro-dioxolo-benzoimidazol-benzamides As Potent Inhibitors of CK1 δ and ϵ with Nanomolar Inhibitory Activity on Cancer Cell Proliferation," *J. Med. Chem.*, vol. 57, no. 19, pp. 7933–7946, Oct. 2014, doi: 10.1021/jm500600b.



Supercómputo para el medio ambiente: paralelizando simulaciones de derrames en el mar

Gabriela Reséndiz Colorado

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Atmósfera y Cambio Climático,
Ciudad de México, México.

ORCID: 0000-0002-2421-6987

Rosario Romero Centeno

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Atmósfera y Cambio Climático,
Ciudad de México, México.

ORCID: 0000-0003-3840-5655

Jorge Zavala Hidalgo

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Atmósfera y Cambio Climático,
Ciudad de México, México.

ORCID: 0000-0002-2737-434X

Miguel Ángel Robles Roldán

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Atmósfera y Cambio Climático,
Ciudad de México, México.

ORCID: 0000-0003-1193-5576

Pável Ernesto Oropeza Alfaro

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Atmósfera y Cambio Climático,
Ciudad de México, México.

ORCID: 0000-0002-7464-1973

Recepción: 27 de febrero de 2026.

Aceptación: 25 de agosto de 2026.

Agosto 2026 • número de revista 17 • DOI: 10.22201/dgtic.26832968e.2026.17.133

Supercómputo para el medio ambiente: paralelizando simulaciones de derrames en el mar

Resumen

Las simulaciones numéricas de escenarios y pronósticos de trayectorias de derrames de hidrocarburos juegan un papel preponderante en la atención de estos incidentes. Uno de los principales retos es obtener estas simulaciones en corto tiempo. En este trabajo, se aborda la paralelización de simulaciones de derrames por medio del desarrollo de *scripts* en Bash y Python, lo cual permite aprovechar los múltiples nodos del *cluster* Ometéotl del Grupo Interacción Océano-Atmósfera del Instituto de Ciencias de la Atmósfera y Cambio Climático (ICAYCC) de la UNAM. Los resultados de la implementación de cómputo paralelo muestran una reducción del 77.8% en el tiempo de pared respecto a la versión secuencial, lo que permitió obtener 66 430 simulaciones de derrames en 36 horas. Por otro lado, las simulaciones de un solo pozo, considerando un derrame diario durante 26 años, tardaron 3 horas y 20 minutos en ejecutarse.

Palabras Clave: simulaciones de derrame, paralelización, PyGnome, derrames de hidrocarburo.

High-Performance Computing for the environment: Parallelizing oil spill simulations at sea

Abstract

Numerical simulations of oil spill scenarios and trajectory forecasts play a key role in responding to these incidents. An important challenge is producing these scenarios in a short time. This work presents the parallelisation of oil spill simulations through the

development of Bash and Python scripts, which enable the use of multiple nodes of the Ometéotl cluster at the Grupo Interacción Océano-Atmósfera of the ICAyCC, UNAM. The results of the parallel computing implementation showed a 77.8% reduction in the wall-clock time compared to the sequential version, enabling 66,430 simulations to be completed in 36 hours. Meanwhile, simulations of a single well with a daily oil spill over 26 years took 3 hours and 20 minutes to execute.

Keywords: oil spill simulations, parallelization, PyGnome, hydrocarbon spills.

Introducción

En México, la explotación de hidrocarburos es un sector importante de las actividades económicas. En 2024, poco más del 74% de la producción correspondió a campos ubicados en el Golfo de México (GoM, por sus siglas en inglés), lo que representa 1103 miles de barriles diarios [1]. Si bien la explotación petrolera en México genera importantes beneficios económicos, también conlleva considerables impactos ambientales cuando ocurren incidentes como los derrames de hidrocarburos. El GoM se encuentra en riesgo de sufrir tales impactos y, de hecho, dos de los más grandes derrames a nivel mundial ocurrieron ahí. El primero ocurrió en 1979, durante la perforación del pozo Ixtoc-I en la Sonda de Campeche. Se perdió el control del pozo, lo que resultó en un derrame que estuvo activo 289 días, periodo en el que se vertieron 3.4 millones de barriles [2]. Más recientemente, durante la perforación del pozo Macondo en aguas profundas estadounidenses, se presentó un derrame de 3.19 millones de barriles, que duró activo 87 días [3].

Cuando estos incidentes ocurren, se producen afectaciones en los ecosistemas marinos y se ha comprobado su efecto tóxico en una amplia diversidad de organismos, que van desde el plancton, los invertebrados, las aves y los peces, hasta los mamíferos marinos [4], [5]. Además, cuando el derrame no es controlado y el hidrocarburo arriba a las costas, se generan impactos negativos tanto en las actividades económicas como en las poblaciones que habitan las zonas costeras.

Para controlar el derrame y minimizar los impactos negativos de estos incidentes, es necesario planear la atención y respuesta; con este objetivo, el pronóstico y los escenarios de la trayectoria del derrame juegan un papel preponderante. Por lo anterior, a nivel mundial, se han desarrollado diferentes programas de cómputo enfocados en esta tarea, algunos de código abierto y otros que requieren una licencia de pago. Sin embargo, en todos los casos, el costo computacional de ejecutar estos escenarios puede retrasar la obtención de la trayectoria probable y, por tanto, demorar la toma de decisiones para minimizar los impactos. En este contexto, el objetivo de este trabajo fue implementar la paralelización de simulaciones de derrames de petróleo para construir escenarios probables basados en 9490 simulaciones, correspondientes a un derrame diario durante un periodo de 26 años, en tiempos reducidos de ejecución.

Configuración de las simulaciones de derrames

Las simulaciones de derrames de hidrocarburo fueron realizadas mediante el paquete PyGnome de la NOAA [6], que es una biblioteca de Python que usa una aproximación lagrangiana [7]. En esta aproximación, el hidrocarburo es representado por partículas virtuales, las cuales tienen una masa que depende del tipo de hidrocarburo que se esté simulando y se degrada por evaporación. La configuración de estas simulaciones requiere datos de viento a 10 m sobre la superficie y de corrientes oceánicas superficiales, obtenidos de simulaciones desarrolladas por el Grupo Interacción Océano-Atmósfera del ICAYCC-UNAM [8], [9], usando el modelo atmosférico Weather Research & Forecasting (WRF) y el modelo oceánico HYbrid Coordinate Ocean Model (HYCOM), respectivamente.

Para las simulaciones, se seleccionaron siete ubicaciones en tres regiones de explotación petrolera activa en el GoM: Plataforma de Tamaulipas, Plataforma de Tabasco y Sonda de Campeche. Los puntos de siembra de partículas corresponden a siete pozos activos reportados por la Secretaría de Energía de México [10] en el mapa de hidrocarburos [11] (Figura 1). Las simulaciones se configuraron en 2D, es decir, superficiales, considerando que todas las áreas de explotación se encuentran en aguas someras. Para cada pozo, se configuró una simulación diaria durante un periodo de 26 años, lo que resultó en 9490 simulaciones por pozo y 66 430 en total.

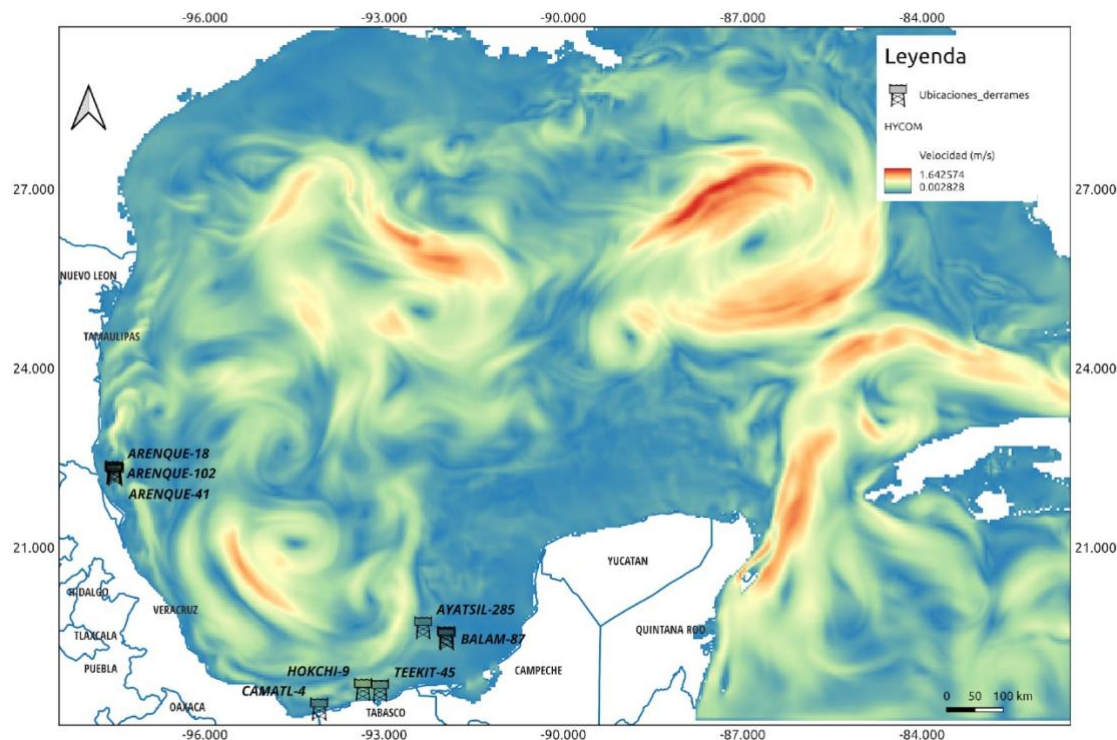


Fig. 1 - Área de estudio y ubicación de los pozos usados para simular los derrames hipotéticos. Para las simulaciones, se usaron los pozos ilustrados a excepción de Arenque-102 debido a la cercanía entre pozos en la región.

Estrategia de ejecución en modo secuencial

Las simulaciones de derrames realizadas con PyGnome usualmente se corren en modo secuencial, de acuerdo con el diagrama de bloques de la Figura 2. Bajo esta estrategia, se realiza sólo una simulación y, cuando ésta termina, se inicia la del siguiente día; todo ello por medio de un solo *script* en Python en el que las simulaciones son controladas por la estructura de control secuencial, conocida como *ciclo for*.

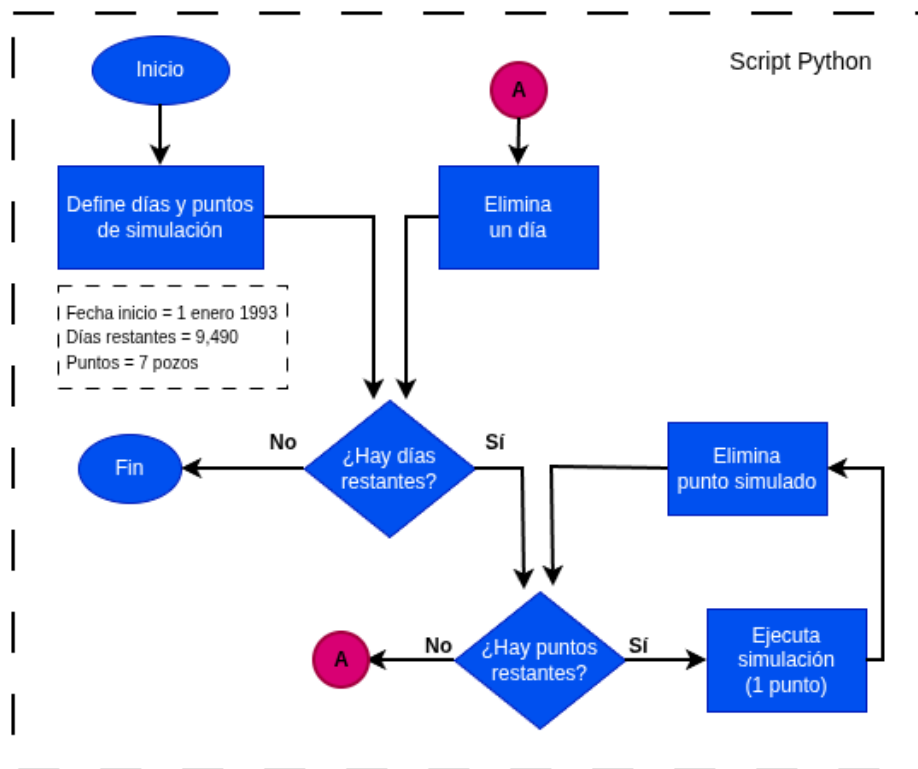


Fig. 2 - Flujo de trabajo para las simulaciones en modo secuencial.

Tomando en cuenta que los equipos de cómputo actuales cuentan con múltiples núcleos, esta forma de trabajo no es eficiente cuando es necesario realizar miles de simulaciones, como en el caso de este trabajo. Por ello, se desarrolló e implementó una estrategia de paralelización que permite hacer uso de los recursos computacionales de múltiples nodos de cómputo en un *cluster*.

Estrategia de ejecución en paralelo

Debido a la cantidad de simulaciones necesarias para construir los escenarios, se implementó una estrategia de paralelización en el *cluster* Ometéotl del Grupo Interacción Océano-Atmósfera del ICAYCC, UNAM. Para las ejecuciones, se usaron 10 nodos de cómputo del *cluster* con 24 núcleos por nodo. Cada nodo cuenta con procesadores de doble *socket*,

Intel Xeon E5-2670 V3 a 2.3 GHz y 128 GB de RAM. El sistema de archivos utilizado en el *cluster* es LUSTRE 2.15.52, implementado en una red Infiniband FDR a 56 GB/s. La gestión de trabajos de los usuarios en el *cluster* es administrada por el software gestor de recursos SLURM.

La metodología se implementó usando tres *scripts*: dos en Bash y uno en Python. El primero es un *script* maestro que se encarga de distribuir y enumerar la ejecución de todas las simulaciones (o tareas) en los núcleos disponibles. Cabe resaltar que, dado que las simulaciones se ejecutan en una infraestructura de cómputo compartida, no siempre se encuentran disponibles todos los recursos solicitados. El *script* maestro ejecuta por lotes el segundo *script*, que define, para cada tarea, las variables de entorno correspondientes al pozo y al número de día que se simulará, con base en el número de tarea asignado por la rutina maestra y el SLURM. Finalmente, el segundo *script* llama a la ejecución del *script* de Python (Figura 3).

El *script* de Python realiza la simulación para el pozo y el día especificados en las variables de entorno definidas previamente. Además, en este *script* se establecen los otros parámetros de la simulación: duración, volumen de hidrocarburo derramado, tipo de hidrocarburo, si se considerará degradación natural, datos de corrientes y viento, así como la configuración de los archivos de salida.

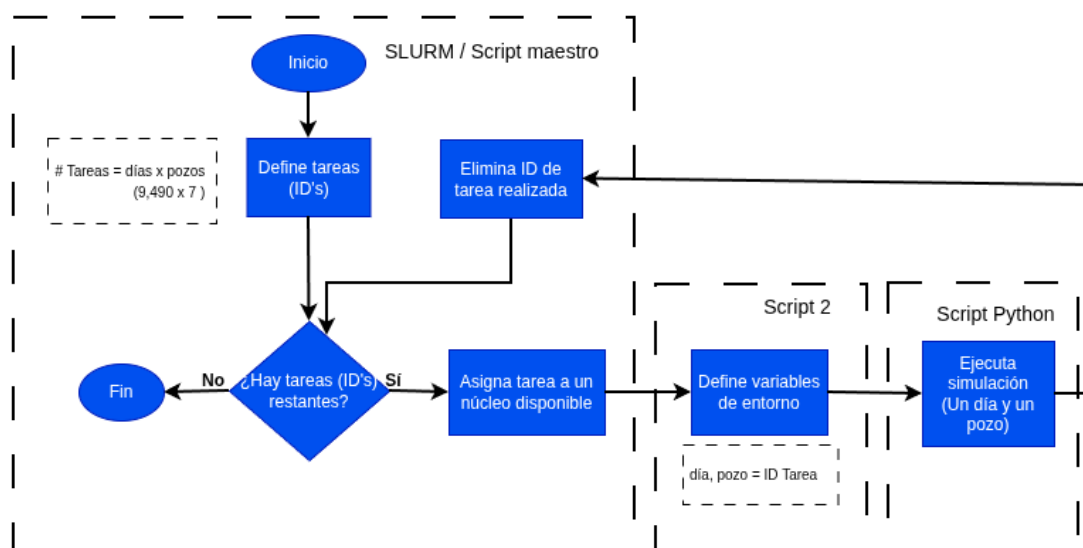


Fig. 3 - Flujo de trabajo para las simulaciones en modo paralelo.

Basados en este desarrollo e implementación, se realizó la paralelización del análisis de resultados. Estos análisis consisten en que, por medio de una malla que cubre el GoM, se cuente la cantidad de partículas y se estime la probabilidad de la presencia de una partícula en cada píxel, así como la masa por píxel para cada uno de los días de la simulación. Estos cálculos son realizados para cada una de las cuatro temporadas del año. Considerando que se tienen 7 pozos, esto nos da como resultado la ejecución de 28 trabajos simultáneos.

Mejora en la eficiencia de ejecución

Las simulaciones implementadas de forma secuencial, en donde se corría una simulación por ejecución a través de un código en Python, resultaba en que el tiempo para obtener todas las simulaciones fuera de 6.75 días. Sin embargo, con la implementación en paralelo, se ejecutan de manera simultánea 240 simulaciones con un flujo de trabajo ininterrumpido, lo que permitió obtener una reducción en tiempo de ejecución de 4.5 veces (Figura 4). Con esto, el tiempo de cómputo se redujo en 77.8%, pasando de un tiempo de pared de casi una semana a 36 horas.

De esta manera, se pueden construir numerosos escenarios y analizarlos en cuestión de algunos días. Esta implementación es la base para la realización de escenarios bajo condiciones reales en caso de la ocurrencia de un incidente en México. En un caso realista, la ejecución se limitaría a un solo pozo y las simulaciones se requerirían para horizontes de pronóstico más cortos, por ejemplo, a siete o diez días. Bajo este supuesto, se realizaron simulaciones para un derrame activo de 90 días de duración para un solo pozo, considerando de nuevo un derrame diario durante 26 años, con un tiempo de ejecución de 3 horas y 20 minutos.

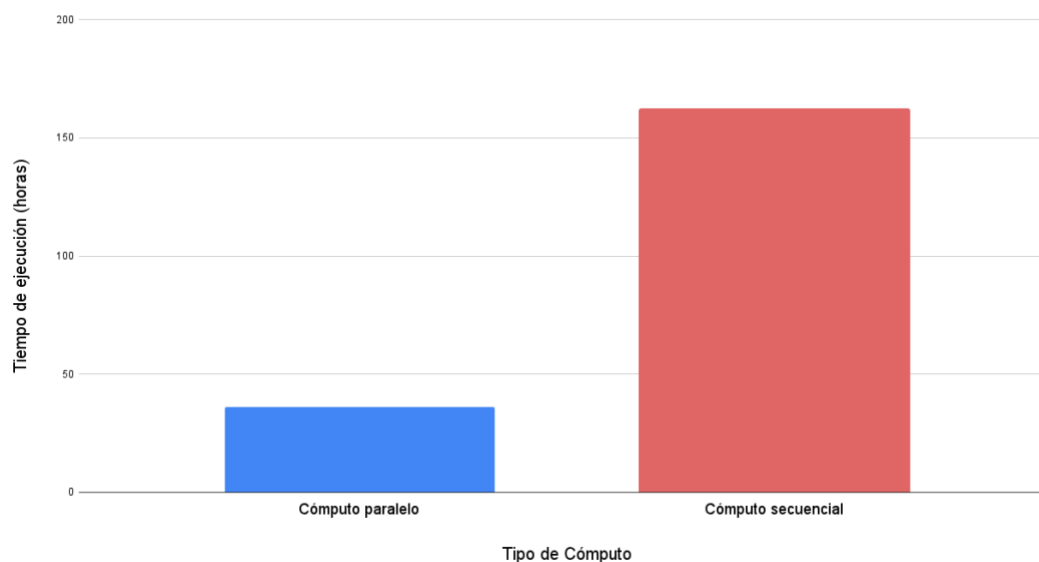


Fig. 4 - Comparativa del tiempo de ejecución con un núcleo contra el cómputo en paralelo utilizando 10 nodos con 24 núcleos por nodo.

Conclusiones

El uso del supercómputo para paralelizar la ejecución de simulaciones de derrames de hidrocarburos permitió reducir los tiempos de procesamiento, convirtiendo un proceso que duraba varios días en uno de pocas horas. Esta optimización permite aprovechar los recursos del *cluster* Ometéotl para generar miles de simulaciones en poco tiempo. La paralelización implementada en este trabajo puede servir como base para el desarrollo de sistemas operacionales que complementen los pronósticos de trayectorias con la generación de escenarios probabilísticos en caso de un derrame real. La importancia de este avance radica en que, ante una contingencia ambiental, el factor tiempo y el conocimiento de la posible trayectoria de la mancha son críticos para mitigar daños ecológicos y socioeconómicos. Esto permitirá ofrecer de manera eficiente información rápida y oportuna a las autoridades y equipos de respuesta para la protección de los ecosistemas marinos y poblaciones costeras.

Agradecimientos

Estancia posdoctoral realizada gracias al programa de Becas Posdoctorales de la Universidad Nacional Autónoma de México (POSDOC).

Referencias

- [1] Petróleos Mexicanos (PEMEX), *Anuario estadístico 2024*, 2024. [Online]. Available: <https://www.pemex.com/ri/Publicaciones/Anuario%20Estadistico%20Archivos/Anuario%20Estad%3%ADstico%202024.pdf>. Accessed: Sep. 2025.
- [2] L. A. Soto, A. V. Botello, S. Licea-Durán, M. L. Lizárraga-Partida, and A. Yáñez-Arancibia, "The environmental legacy of the Ixtoc-I oil spill in Campeche Sound, southwestern Gulf of Mexico," *Front. Mar. Sci.*, vol. 1, p. 57, 2014, doi: 10.3389/fmars.2014.00057.
- [3] J. Beyer, H. C. Trannum, T. Bakke, P. V. Hodson, and T. K. Collier, "Environmental effects of the Deepwater Horizon oil spill: A review," *Mar. Pollut. Bull.*, vol. 110, no. 1, pp. 28–51, 2016, doi: 10.1016/j.marpolbul.2016.06.027.
- [4] National Research Council, "Understanding the Risk," in *Oil in the Sea III: Inputs, Fates, and Effects*, Washington, DC, USA: The National Academies Press, 2003, pp. 19–62.
- [5] P. Chapman, T. L. Wade, and A. H. Knap, "Environmental Effects and Management of Oil Spills on Marine Ecosystems," in *Handbook of Oil Spill Science and Technology*, F. Fingas, Ed., Boca Raton, FL, USA: CRC Press, 2018, pp. 119–145, doi: 10.1201/9781315153933-5.
- [6] NOAA Office of Response and Restoration (ORR), "PyGnome: General NOAA Operational Modeling Environment," [Online]. Available: <https://github.com/NOAA-ORR-ERD/PyGnome>. Accessed: Aug. 2025.

- [7] NOAA Office of Response and Restoration (ORR), *PyGnome Documentation*, [Online]. Available: <https://gnome.orr.noaa.gov/doc/pygnome/index.html>. Accessed: Sep. 2025.
- [8] R. Romero-Centeno, J. Zavala-Hidalgo, and J. P. Osuna-Cañedo, Eds., "Circulación oceánica y oleaje," in *Atlas de línea base ambiental del golfo de México*, vol. II, S. Z. Herzka *et al.*, Coords., México: CIGoM, 2021.
- [9] R. Romero-Centeno and J. Zavala-Hidalgo, Eds., "Meteorología," in *Atlas de línea base ambiental del golfo de México*, vol. I, S. Z. Herzka *et al.*, Coords., México: CIGoM, 2021.
- [10] Secretaría de Energía (SENER), "Sistema de Información de Hidrocarburos," [Online]. Available: <https://sih-hidrocarburos.energia.gob.mx/>. Accessed: May 2025.
- [11] Secretaría de Energía (SENER), "Mapa Interactivo de Contratos y Asignaciones de Hidrocarburos," [Online]. Available: <https://mapa-hidrocarburos.energia.gob.mx/>. Accessed: May 2025.



Potenciando pronósticos ambientales en México: supercómputo como pilar en modelación de calidad del aire y de dispersión de cenizas

José Agustín García Reynoso

Universidad Nacional Autónoma de México, Instituto de Ciencias de la Tierra y Cambio
Climático, Ciudad de México, México.
ORCID: 0000-0001-7718-9241

Recepción: 14 de enero de 2026

Aceptación: 21 de agosto 2026

Agosto 2026 • número de revista 17 • DOI: 10.22201/dgtic.26832968e.2026.17.127

Potenciando pronósticos ambientales en México: supercómputo como pilar en modelación de calidad del aire y de dispersión de cenizas

Resumen

La calidad del aire es el resultado de la interacción entre emisiones contaminantes y procesos meteorológicos. Para predecirla, es necesario resolver numéricamente las ecuaciones fundamentales que describen el comportamiento atmosférico, incluyendo conservación de masa, momento, energía y reacciones químicas. Estas simulaciones requieren gran capacidad de supercómputo.

En el Instituto de Ciencias de la Atmósfera y Cambio Climático (ICAYCC) de la UNAM, se utiliza el modelo WRF-Chem, que integra meteorología, transporte y química. Este modelo se ejecuta en un sistema de cómputo de alto rendimiento donde se utilizan 81 núcleos para el procesamiento en paralelo. El WRF-Chem utiliza un dominio único, dividido en 90x90 celdas de 3x3 km. Su ejecución requiere 90 minutos de procesamiento para 72 horas de pronóstico.

El modelo Fall3D se emplea para la dispersión y el depósito de ceniza volcánica. Efectúa 72 corridas que incluyen tres niveles de columna (3, 5 y 10 km) en 24 horas. Cada simulación usa 12 núcleos y emplea cerca de 5 minutos, completando todas las simulaciones en ~70 minutos.

Para soportar estas tareas, se utiliza el WRF-Chem, compilado para la infraestructura disponible. Su ejecución se realiza mediante scripts automatizados en Bash, que emplean el sistema de gestión de trabajos (SLURM) y de archivos paralelos (LUSTRE). Los productos generados se despliegan posteriormente en una página web.

Palabras Clave: ozono, Popocatépetl, aerosol, contaminante secundario, SLURM, LUSTRE.

Enhancing environmental forecasting in Mexico: Supercomputing as a pillar for air quality and ash dispersion modeling

Abstract

Air quality is the result of the interaction between pollutant emissions and meteorological processes. To predict it, it is necessary to numerically solve the fundamental equations that describe atmospheric behavior, including conservation of mass, momentum, energy, and chemical reactions. These simulations require significant supercomputing power.

At the ICAYCC of UNAM, the WRF-Chem model is used, which integrates meteorology, transport, and chemistry. This model runs on a high-performance computing system using 81 cores for parallel processing. WRF-Chem uses a single domain divided into 90x90 cells of 3x3 km. Its execution requires 90 minutes of processing for a 72-hour forecast.

The Fall3D model is used for volcanic ash dispersion and deposition. It performs 72 runs that include three column levels (3, 5, and 10 km) over 24 hours. Each simulation uses 12 cores and takes approximately 5 minutes, completing all simulations in ~70 minutes. To support these tasks, WRF-Chem compiled for the available infrastructure is used; automated Bash scripts are used for execution, employing a job management system (SLURM) and a parallel file management system (LUSTRE); the products are deployed on a web page.

Keywords: *ozone, Popocatépetl, aerosol, secondary pollutant, SLURM, LUSTRE.*

Introducción

Los fenómenos físicos que ocurren en la naturaleza pueden describirse mediante ecuaciones como la de conservación de masa, *momentum* y de cinética química.

Así, para comprender los desplazamientos del aire, vapor de agua y contaminantes en la atmósfera, se emplea la ecuación de conservación de masa [1], la cual es representada por:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

Donde

ρ - es la densidad del fluido (kg m^{-3})

t - es el tiempo (s)

$\mathbf{u} = (u, v, w)$ - vector de velocidad del fluido

Por otra parte, para estudiar y predecir el comportamiento del viento, se emplea la ecuación de conservación del momento lineal (ecuaciones de Navier-Stokes) [1]:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \text{Fricción}$$

ρ Densidad del fluido

$\mathbf{u}$ Vector velocidad

p Presión estática [Pa]

$\boldsymbol{\tau}$ Tensor de esfuerzos viscosos (τ_{ij})

t Tiempo (s)

$\mathbf{g}$ Vector aceleración gravitacional [m s^{-2}]

∇p Gradiente de presión

$\nabla \cdot (\rho uu)$ Divergencia del tensor de momento convectivo

Estas ecuaciones se emplean para estimar el estado actual y futuro de la atmósfera. En la predicción meteorológica, se manejan modelos eulerianos que resuelven estas ecuaciones mediante métodos numéricos, como en el modelo WRF [2]-[4].

Por otra parte, la calidad del aire en un sitio depende principalmente de las condiciones meteorológicas y de las emisiones. Por ello, también se emplean modelos para estimar las emisiones [5]-[7] y generar su distribución espacial, temporal y por especies químicas [8]. En el caso de contaminación por partículas, ocurren procesos de condensación, coagulación, absorción, sedimentación y depósito que se deben considerar para poder describir los niveles de concentración y el transporte de las emisiones. Para ello, se emplean modelos como el Fall3D [9], que incluye el término de sedimentación, como se muestra en la siguiente ecuación:

$$\frac{\partial C}{\partial t} + \nabla \cdot [C(u + v_s)] = \nabla \cdot (K \nabla C) + S$$

Donde

$C(x, t)$ Concentración de partículas [kg, m^{-3}] o [m^{-3}] según el caso

u Velocidad del fluido [m, s^{-1}]

v_s Velocidad de sedimentación de partículas (generalmente vertical y dirigida hacia abajo)

s Tiempo [s]

K Tensor o coeficiente escalar de difusión turbulenta [$m^2 s^{-1}$]

S Términos fuente o sumidero (emisiones, deposición, reacciones)

$\nabla \cdot$ Divergencia

∇C Gradiente de concentración

Si se desea conocer el estado del tiempo o calidad del aire a partir de condiciones iniciales y de frontera para una región en particular, la resolución de las ecuaciones anteriores requiere de una gran capacidad de cálculo para obtener resultados en un tiempo razonable. Por ello, se requiere de sistemas de procesamiento numérico en paralelo, conocidos como equipos de supercómputo (HPCC).

Desarrollo

En el Instituto de Ciencias de la Atmósfera y Cambio Climático (ICAyCC), se cuenta con un sistema HPCC compuesto por 20 nodos, cada uno con 24 *cores* (Intel Xeon) a 2.3 GHz, que alcanza un rendimiento de 17.4 TFP. El sistema cuenta con Debian 5.10, compiladores Intel de 2019 y alrededor de 960 TB de almacenamiento en paralelo. El entorno de software posee 50 librerías científicas administradas con Lmod, versión 6.6 [10], mientras que la gestión de trabajos se administra con LSRM [11]. Esta capacidad de supercómputo permite generar las emisiones que se emplean en el pronóstico de calidad del aire [12], así como realizar la dispersión y depósito de cenizas [13] en un tiempo lo suficientemente corto (calidad del aire en 90 min y cenizas en 15 min) para generar pronósticos diarios y publicar los resultados en la web. En contraste, la ejecución de estos pronósticos en un sistema con un procesador de 2.3 GHz, incluso considerando una escalabilidad óptima, implicaría tiempos de procesamiento aproximados de cinco días para el modelo de calidad del aire y alrededor de tres días para el modelo de dispersión de cenizas. Estos tiempos resultan poco viables para su aplicación operativa en tiempo real. Para el caso de dispersión de cenizas, se realizó un trabajo previo para implementar una estrategia de ejecución del modelo Fall3D, en donde se optimiza el uso del recurso computacional mediante la asignación eficiente del número óptimo de núcleos [14]. Con base en ese trabajo, se identificó que se requieren 12 núcleos para este equipo. Mediante el empleo de Crontab, se programa la ejecución del script principal en Bash, el cual llama a los diversos scripts encargados de realizar los diferentes procesos necesarios para la generación de los pronósticos. Los pasos a seguir son el procesamiento de datos meteorológicos de modelos globales (WPS,REAL), la generación de emisiones de los compuestos contaminantes en la región (DiETE) [8] y la ejecución del modelo de calidad del aire (WRF/chem) [15]. Para el caso de la dispersión de cenizas, el modelo Fall3D [16] requiere de los programas auxiliares que realizan el procesamiento de la

información de meteorología (SetDBS), la distribución de tamaños (SetTGSD) y la emisión de cenizas (SetSrc). Una vez ejecutados los pronósticos, se utilizan scripts para almacenar los resultados, generar las imágenes que serán publicadas y, finalmente, transferirlas del clúster al equipo donde está el sitio web. Una descripción del sistema de pronóstico de calidad del aire se puede encontrar en [12], mientras que la correspondiente al sistema de dispersión de cenizas volcánicas en [14]. En la Fig. 1, se puede apreciar el diagrama de flujo del proceso de elaboración de los pronósticos descritos anteriormente, donde se incluye el uso de los pronósticos del depósito de ceniza para la elaboración de mapas de peligro (HazardMaps) [17].

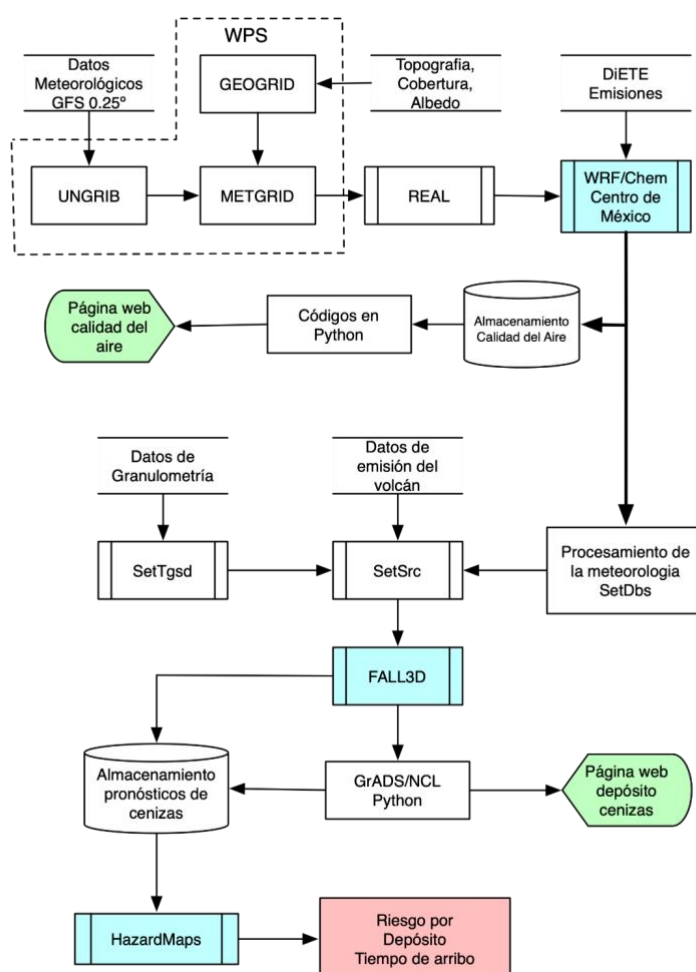


Fig. 1. Esquema del proceso de pronóstico de calidad del aire (WRF/Chem) y su acoplamiento con el pronóstico de dispersión de cenizas (Fall3D).

Resultados

Una vez que han terminado los procesos de generación de los pronósticos, los resultados de estos se pueden observar en la página web <https://lamca.atmosfera.unam.mx>. En este sitio, se presentan las imágenes del pronóstico de calidad del aire y del de dispersión de cenizas. En el caso de calidad del aire, se tienen imágenes horarias de tres días para los contaminantes monóxido de carbono (CO), partículas menores de 10 micras (PM₁₀), ozono, dióxido de nitrógeno (NO₂) y dióxido de azufre (SO₂). Un ejemplo de los productos obtenidos se presenta en la Fig. 2, la cual muestra la distribución espacial de la concentración de ozono para las 14 horas.

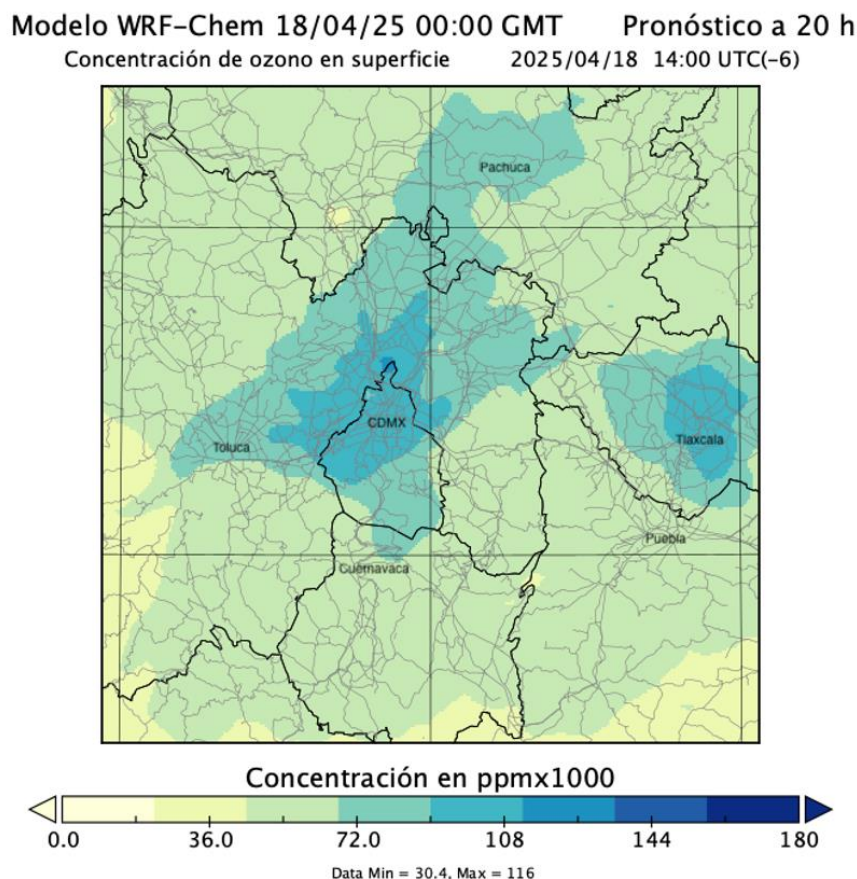


Fig. 2. Distribución superficial de ozono en el Centro de México a las 14:00 locales.

En el caso del pronóstico de dispersión de cenizas, a partir de erupciones hipotéticas, se obtienen productos que muestran el depósito de ceniza, las concentraciones ambientales a diferentes niveles de vuelo y otros productos de interés como cortes de norte a sur (lat) y de oeste a este (long).

En la Fig. 3, se muestra el panel de concentraciones a los diferentes niveles de vuelo, el nivel de vuelo de FL100 corresponde a 10 000 pies de altitud.

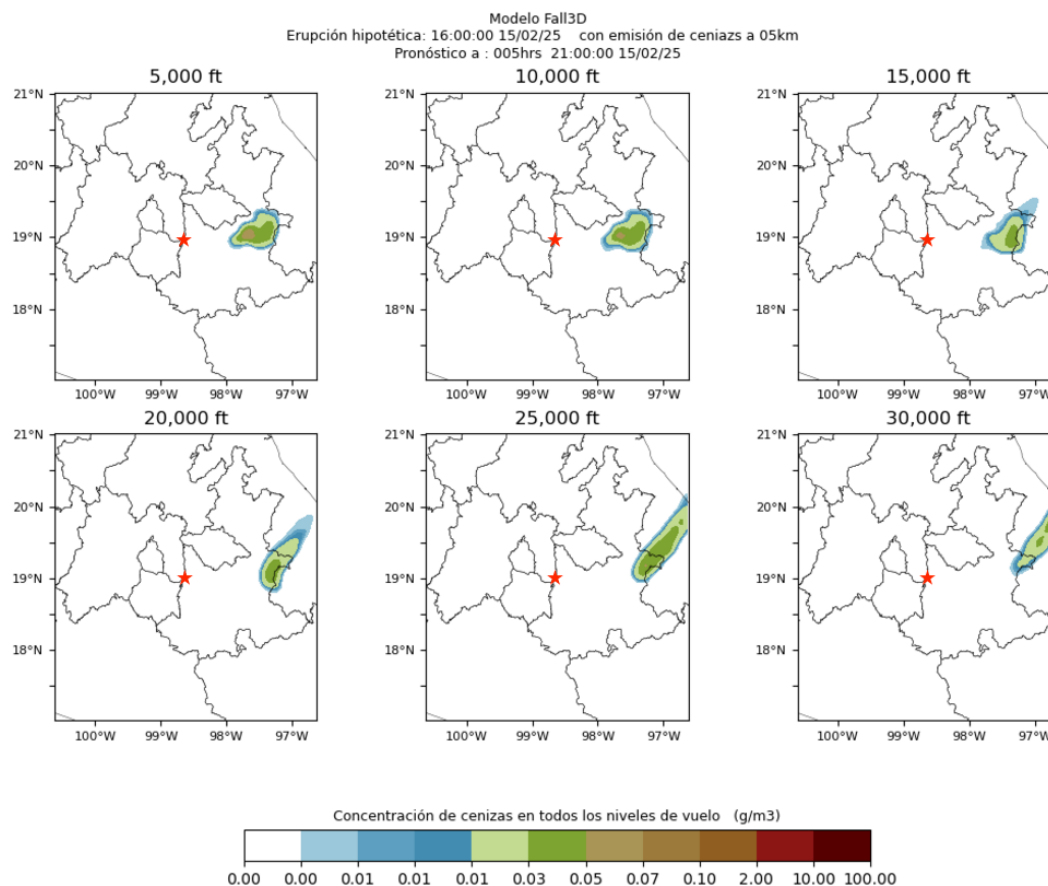


Fig. 3. Panel de concentración de ceniza a diferentes niveles de vuelo, la estrella indica la ubicación del cráter del Popocatepetl.

El sistema de pronóstico de calidad del aire se ha evaluado para conocer su desempeño [18]. Los resultados de las simulaciones del depósito de ceniza se han empleado mediante el código HazarMaps [19] para generar mapas donde se presentan las zonas con la probabilidad de depósito de ceniza y tiempo de arribo [20].

Conclusión

Los procesos atmosféricos se pueden describir con ecuaciones matemáticas. Para resolver estas ecuaciones y conocer las condiciones actuales y futuras de la atmósfera, se necesita una capacidad considerable de cálculo numérico; debido a esto, el uso de herramientas de alto desempeño es una necesidad más que una opción. La transformación de los resultados de estas ecuaciones complejas en imágenes comprensibles contribuye al acceso público de la ciencia y la protección civil. La automatización de los procesos mediante *Crontab* y *scripts* en Bash para la ejecución de los modelos garantiza la generación diaria de los pronósticos. El uso de middleware como SLURM y Lmod demuestra la especialización técnica en el manejo de Tecnologías de la Información y Comunicación (TIC) aplicadas en las ciencias atmosféricas. Así, con HPCC y la automatización, se facilita la generación de pronósticos en tiempos razonables, lo que permite una toma de decisiones más efectiva tanto de las autoridades como de la sociedad.

Agradecimientos

Se reconce y agradece al Lic. Pedro Damián Cruz Santiago quien administra el equipo de supercómputo donde se realizan los pronósticos, a la Mtra. Dulce Rosario Herrera Moro quien mantiene las páginas web y a la unidad de cómputo y supercómputo del Instituto de Ciencias de la Atmósfera.

Sobre el uso de herramientas de Inteligencia Artificial Generativa:

Se empleó ChatGpt version 1.2025.330 para generar el código de Latex de las ecuaciones y términos de las mismas.

Referencias

- [1] J. M. Wallace and P. V. Hobbs, *Atmospheric science: an introductory survey*, 2nd ed. Amsterdam, Netherlands: Elsevier Academic Press, 2006, p. 483.
- [2] W. C. Skamarock *et al.*, *A Description of the Advanced Research WRF Model Version 4.3* Boulder, CO, USA: National Center for Atmospheric Research, 2021, ed, 2022.

- [3] W. C. Skamarock *et al.*, *A description of the Advanced Research WRF version 3*. Boulder, CO, USA: National Center for Atmospheric Research , 2008.
- [4] W. C. Skamarock *et al.*, *A description of the advanced research WRF model version 4*, Boulder, CO, USA: National Center for Atmospheric Research, p. 145, 2019.
- [5] U.S. Environmental Protection Agency, *Compilation of Air Pollutant Emissions Factors 1.1 Bituminous And Subbituminous Coal Combustion*, Environmental Protection Agency, US, AP-42, April, 1998 1998, vol. 1. [Online]. Available: <https://www.epa.gov/air-emissions-factors-and-quantification/ap-42-fifth-edition-volume-i-chapter-1-external-0>
- [6] Dirección General de Gestión de la Calidad del Aire, *Inventario de emisiones de la Ciudad de México 2016*, 1 ed. (no. 1). Ciudad de México: Secretaria del Medio Ambiente, 2018, p. 220.
- [7] Dirección General de Gestión de la Calidad del Aire, *Inventario de emisiones de la Zona Metropolitana del Valle de México 2018*, Ciudad de México, México: Gobierno de la Ciudad de México, 2021, p. 168. [Online]. Available: <http://www.aire.cdmx.gob.mx/descargas/publicaciones/flippingbook/inventario-emisiones-cdmx-2018/Inventario-de-emisiones-cdmx-2018.pdf>. Accessed on: 25/nov/2021.
- [8] J. A. García-Reynoso, B. E. Mar-Morales, and L. G. Ruiz-Suárez, "Spatial, temporal and speciation model of the Mexico national emissions inventory (base year 2008) for its use in air quality modeling (DiETE)," *Revista Internacional de Contaminacion Ambiental*, Article vol. 34, no. 4, pp. 635-649, 2018, doi: 10.20937/RICA.2018.34.04.07.
- [9] A. Folch, A. Costa, and G. Macedonio, "FALL3D: A computational model for transport and deposition of volcanic ash," *Computers and Geosciences*, vol. 35, no. 6, pp. 1334-1342, 2009, doi: 10.1016/j.cageo.2008.08.008.
- [10] *A New Environment Module System — Lmod 6.6 Documentation*. (2020). Texas Advanced Computing Center. [Online]. Available: <https://lmod.readthedocs.io/en/6.6/>

- [11] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management," in *Job Scheduling Strategies for Parallel Processing*, Berlin, Heidelberg, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds., 2003// 2003: Springer Berlin Heidelberg, pp. 44-60.
- [12] A. R. Garcia, B. E. Mar-Morales, and L. G. Ruiz-Suárez. *Numerical air quality forecast modeling system: Mexico case study*, pp. 195-202, 2016.
- [13] A. García-Reynoso, J. Zavala-Hidalgo, H. Delgado-Granados, and D. R. Herrera-Moro, "The Use of HPC on Volcanic Tephra Dispersion Operational Forecast System," in *Communications in Computer and Information Science*, vol. 948. Cham, Switzerland: Springer, 2019, pp. 110-117, doi: 10.1007/978-3-030-10448-1_10.
- [14] A. R. García, J. Zavala-Hidalgo, H. Delgado-Granados, J. Garcia-Escalante, O. Gómez-Ramos, and D. Herrera-Moro, "An automated ash dispersion forecast system: case study Popocatepetl volcano, Mexico," *Journal of Applied Volcanology*, vol. 12, no. 1, Art no. 9, 2023, doi: 10.1186/s13617-023-00135-4.
- [15] G. A. Grell *et al.*, "Fully coupled 'online' chemistry within the WRF model," *Atmospheric Environment*, vol. 39, no. 37, pp. 6957-6975, 2005, doi: 10.1016/j.atmosenv.2005.04.027.
- [16] A. Folch, A. Costa, and G. Macedonio, *FALL3D-7.1 User's Manual*, Barcelona, Spain: Barcelona Supercomputing Center (Bsc-CNS), 2016, p. 45. [Online]. Available: https://thehub.org/resources/3950/download/fall3d_7.1.pdf. Accessed on: 26/07/2023.
- [17] A. Folch, *HazardMaps*, Fortran code, HazardMaps.f90, 2016.
- [18] J. A. Rodríguez-Zas and J. A. García-Reynoso, "Updating of the 2013 national emissions inventory for air quality modeling in central Mexico," *Revista Internacional de Contaminacion Ambiental*, vol. 37, pp. 463-487, 2021, doi: 10.20937/RICA.53865.
- [19] R. Bonasia *et al.*, "Long-range hazard assessment of volcanic ash dispersal for a Plinian eruptive scenario at Popocatepetl volcano (Mexico): implications for civil

aviation safety," *Bulletin of Volcanology*, vol. 76, no. 1, p. 789, 2014, doi: 10.1007/s00445-013-0789-z.

- [20] J. A. García Reynoso and B. E. Mar-Morales, "Spatial analysis of volcanic ash dispersion from Popocatépetl: probability estimation and influence area mapping in a risk context," *Terra digitalis*, vol. 9, no. 1, pp. 1-6, 2025, doi: 10.22201/igg.25940694e.2025.1.125.



Superconductividad en nanoestructuras mediante matrices por bloques

Germán Emiliano López

Universidad Nacional Autónoma de México, Facultad de Ciencias, Departamento de Física, Ciudad de México, México.

ORCID: 0009-0002-4884-7772

Vicenta Sánchez

Universidad Nacional Autónoma de México, Facultad de Ciencias, Departamento de Física, Ciudad de México, México.

ORCID: 0000-0001-6898-6636

Chumin Wang

Universidad Nacional Autónoma de México, Instituto de Investigaciones en Materiales, Ciudad de México, México.

ORCID: 0000-0002-2737-9722

Recepción: 28 de febrero de 2026.

Aceptación: 24 de agosto de 2026.

Agosto 2026 • número de revista 17 • DOI: 10.22201/dgtic.26832968e.2026. 17.155

Superconductividad en nanoestructuras mediante matrices por bloques

Resumen

La superconductividad es un fenómeno cuántico extraordinario en el que millones de electrones se organizan y cooperan para transportar electricidad sin resistencia. Para facilitar su estudio en nanoestructuras, se utiliza la transformación a matrices organizadas por bloques, siendo una herramienta matemática que separa las distintas "partes" del sistema sin sacrificar precisión. Esto podría conducir al diseño cuántico de dispositivos a escala macroscópica, tales como cables y bobinas superconductoras, usando la capacidad de cómputo actual.

Palabras Clave: superconductividad, condensado de pares, formalismo de Bogoliubov-de Gennes, transformación unitaria, nanotecnología.

Superconductivity in nanostructures through blocked matrices

Abstract

Superconductivity is an extraordinary quantum phenomenon in which millions of electrons organize and cooperate to transport electricity without resistance. To facilitate its study in nanostructures, the transformation into block-organized matrices is used, serving as a mathematical tool that separates the different "parts" of the system without sacrificing precision. This could enable the quantum design of macroscopic-scale devices, such as superconducting cables and coils, using current computational capabilities.

Keywords: *Superconductivity, Cooper pair condensate, Bogoliubov-de Gennes formalism, Unitary transformation, Nanotechnology.*

Introducción

Imagina un mundo donde la electricidad fluye sin perder energía, los trenes flotan sobre las vías y los escáneres médicos son increíblemente precisos. Esto puede ser posible gracias a la superconductividad, un fenómeno descubierto hace más de un siglo, mientras se exploraban los secretos del frío extremo. A continuación, descubre la fascinante historia de cómo se llega a comprender este extraño comportamiento de la materia, usando diversas herramientas tales como la mecánica cuántica y el cómputo de alto rendimiento.

Todo comenzó en 1908 en los Países Bajos. Ahí, el físico Heike Kamerlingh Onnes logró licuar helio, el cual es usualmente un gas y se vuelve líquido a una temperatura de -269°C , equivalentemente a 4.2 K. Onnes tenía el refrigerador más potente del mundo y le permitía estudiar lo que pasaba con los materiales cuando se enfrían a temperaturas extremadamente bajas. En aquella época, los científicos debatían si la resistencia eléctrica desaparecía a 0 K o tiende a infinito dado que los electrones se quedan congelados. Onnes eligió mercurio por tener una muestra de alta pureza y observó una desaparición abrupta de la resistencia eléctrica al llegar a los 4.19 K, descubriendo un nuevo estado de la materia que llamó superconductividad. En 1913 recibió el Premio Nobel por este hallazgo. Pronto se dio cuenta de que, si inducía una corriente eléctrica en un anillo superconductor, esta seguía fluyendo sin perder intensidad, creando una especie de "movimiento perpetuo" eléctrico con una cota inferior de 100,000 años [1].

En 1933, el físico alemán Walther Meissner y su colega Robert Ochsenfeld realizaron otro hallazgo clave sobre el estado superconductor, el cual expulsa siempre el campo magnético de su interior. Este fenómeno se conoce como Efecto Meissner y es la razón por la que vemos fotografías de imanes levitando sobre un superconductor, demostrando que la superconductividad es mucho más que "resistencia cero" [2].

Este fenómeno no era exclusivo del mercurio; pronto se descubrió que otros metales puros como el vanadio se vuelve superconductor a 5.4 K o el plomo a 7.2 K, los cuales

también perdían la resistencia en el frío extremo. Décadas después, el niobio metálico puro se convertiría en el campeón de los elementos, con una temperatura crítica (T_c) de 9.2 K. Sin embargo, el verdadero salto hacia las aplicaciones prácticas no vendría de los metales puros, sino de la combinación de varios de ellos en aleaciones metálicas. A principios de la década de 1960, los investigadores descubrieron una aleación de niobio-titanio (NbTi) con T_c de 9.5 K, pero un campo magnético crítico que supera a 10 teslas (T) a una temperatura de 4.2 K. Una década después, en 1973, la carrera por batir récords de T_c llevó al descubrimiento del Nb₃Ge (niobio-germanio), un compuesto con una estructura cristalina A15 que logró alcanzar los 23.2 K, convirtiéndose en el superconductor con la T_c más alta durante más de una década.

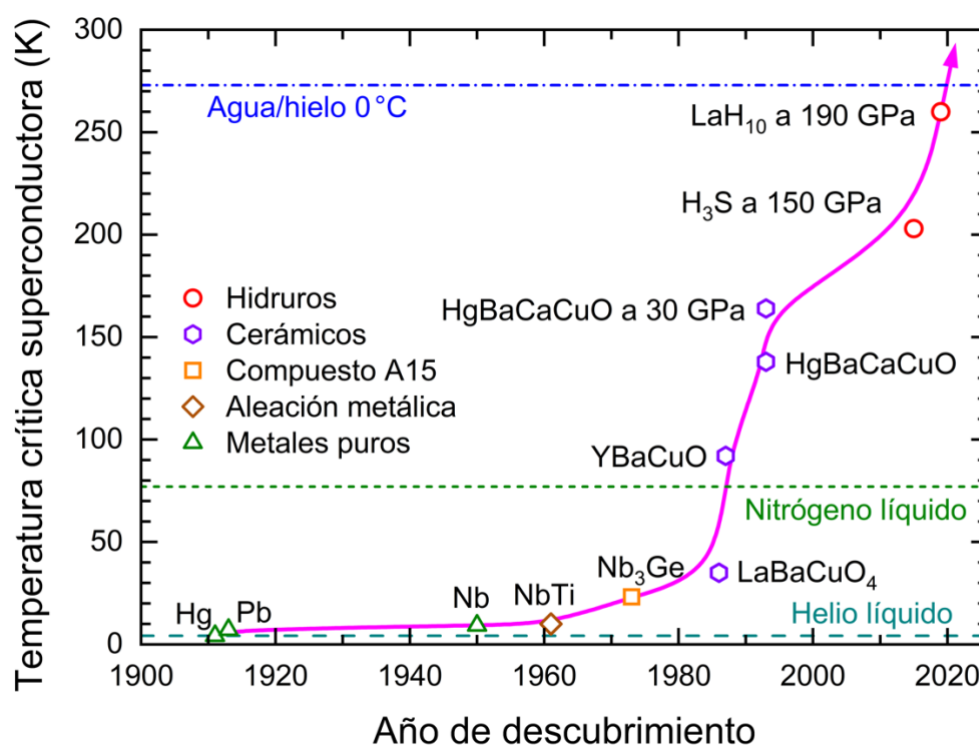


Fig. 1. Año de descubrimiento de diversos materiales superconductores y su correspondiente temperatura crítica.

En 1986, los investigadores de IBM Georg Bednorz y Alex Müller descubrieron un material cerámico LaBaCuO₄ (óxido de lantano-bario-cobre) que se volvía superconductor a 35 K. Lo

revolucionario fue que las cerámicas suelen ser aislantes, es decir, no conducen la electricidad. Este hallazgo fue tan impactante que recibieron el Premio Nobel al año siguiente, en 1987, algo inusualmente rápido.

Unos meses después, un equipo liderado por Paul Chu sustituyó el lantano por itrio y creó el YBaCuO (óxido de itrio-bario-cobre), un material que alcanzó la superconductividad a 92 K (-181°C). Este salto fue crucial, porque permitía usar nitrógeno líquido para enfriarlos, un refrigerante tan barato como la leche, en lugar del carísimo helio líquido. Así, nacieron los superconductores de alta temperatura, un fenómeno que, 40 años después, la comunidad científica aún no tiene una explicación convincente.

Después del descubrimiento de los cupratos en 1986, la siguiente gran revolución llegó de la mano de los hidruros, materiales que combinan hidrógeno con otros elementos formando un sólido cuando se somete a altas presiones. La idea de los hidruros fue propuesta por N. W. Ashcroft en 1968 y el primer hidruro superconductor experimentalmente confirmado fue el sulfuro de hidrógeno (H₃S) en 2015, con una asombrosa T_c de -70°C, eso sí, bajo una presión descomunal. Poco después, en 2019, llegó el decahidruro de lantano (LaH₁₀), que elevó la temperatura crítica hasta los -13°C, acercándose peligrosamente al punto de congelación del agua, aunque siempre dentro de un yunque de diamante.

El salto de los superconductores de laboratorio a la vida cotidiana se encuentra en continuo progreso. Se están fabricando dos tipos de cables superconductores: (I) basados en multifilamentos trenzados, como se ilustra en la Figura 2(a) [3]; y (II) basados en cintas de cupratos mostradas en la Figura 2(b). Por ejemplo, se han instalado decenas de kilómetros de cables superconductores basados en las cintas mencionadas para transmitir electricidad sin pérdida desde las plantas hasta zonas urbanas, los cuales ya se encuentran en funcionamiento en los Estados Unidos, Alemania y China [4].



Fig. 2. Aplicaciones de la superconductividad: (a) Cable de multifilamentos trenzados para la generación de campo magnético; (b) Cable de transmisión de potencia con nitrógeno líquido circulando en el centro; (c) Máquina de resonancia magnética; (d) Fusión nuclear controlada; (e) Tren de levitación magnética.

Asimismo, los ingenieros incrustan cientos de filamentos entrelazados de NbTi dentro de una matriz metálica, para construir electroimanes, los cuales se utilizan en los equipos de resonancia magnética de los hospitales, como se muestra en la Figura 2(c), creando campos magnéticos intensos y estables de 1.5 y 3 T. Estos electroimanes también se usan, por ejemplo, en el gran colisionador de hadrones y en la fusión nuclear controlada para confinar el plasma a 100 millones de grados, creando un “pequeño sol en la Tierra” (ver Figura 2(d)). Por último, los prototipos de trenes de levitación magnética (Maglev), han sido presentados en Japón, China y Brasil, donde sus potentes imanes superconductores los hacen flotar sobre las vías, eliminando el rozamiento y alcanzando velocidades superiores a 600 km/h, como se muestra en la Figura 2(e).

Para entender por qué materiales como los metales puros y los compuestos A15 conducen la electricidad sin resistencia alguna, tenemos que asomarnos a la teoría BCS, desarrollada en 1957 por John Bardeen, Leon Cooper y Robert Schrieffer, la cual descubrió

el "truco" de la superconductividad: los electrones no viajan solos, sino que lo hacen en parejas, formando los llamados pares de Cooper, por ejemplo, atraídos mediante fonones (cuantos de la vibración atómica) [5]. Este proceso de apareamiento puede visualizarse como un electrón moviéndose por dentro de un metal; al pasar, atrae a los iones positivos de la red, creando una pequeña estela de carga positiva justo a su paso. Un segundo electrón es atraído por esa estela, y así ambos avanzan como si bailaran un vals tomándose de las manos. Lo fascinante es que todos estos pares de Cooper se comportan al unísono, como un gran baile coreografiado.

Como una evolución de la teoría BCS, en 1964, se desarrolló el formalismo de Bogoliubov-de Gennes (BdG), el cual permite a los científicos estudiar en el espacio real los superconductores nanoestructurados y con múltiples interfaces entre distintos materiales, a diferencia de la teoría BCS formulada para materiales perfectos y periódicos [6]. Sin embargo, las ecuaciones BdG para las brechas superconductoras locales se resuelven generalmente de manera autoconsistente, es decir, se hace una suposición inicial de la brecha superconductora (Δ) y se calcula mediante las ecuaciones BdG una nueva Δ . Si no coinciden las dos Δ , se ajusta la suposición y se repite el proceso hasta que las dos coincidan. Es bien conocido que el procedimiento autoconsistente es computacionalmente costoso y, en general, no se tiene la garantía de encontrar una solución física.

Método de matriz por bloques

Para entender el método, imaginemos un nanotubo superconductor infinitamente largo cuya sección transversal contiene 9 átomos, como se observa en la parte superior de la Figura 3. En lugar de analizar toda su estructura al mismo tiempo, aplicamos una transformación unitaria que permite "reordenar" la descripción del sistema de modo que la matriz asociada a las ecuaciones de BdG se divide en bloques pequeños e independientes. Como se ilustra en la parte inferior de la Figura 3, el nanotubo puede entonces visualizarse como un conjunto de cadenas efectivas que no interactúan entre sí. Así, un sistema tridimensional se convierte en un conjunto de cadenas unidimensionales, lo que facilita enormemente su análisis y resolución.

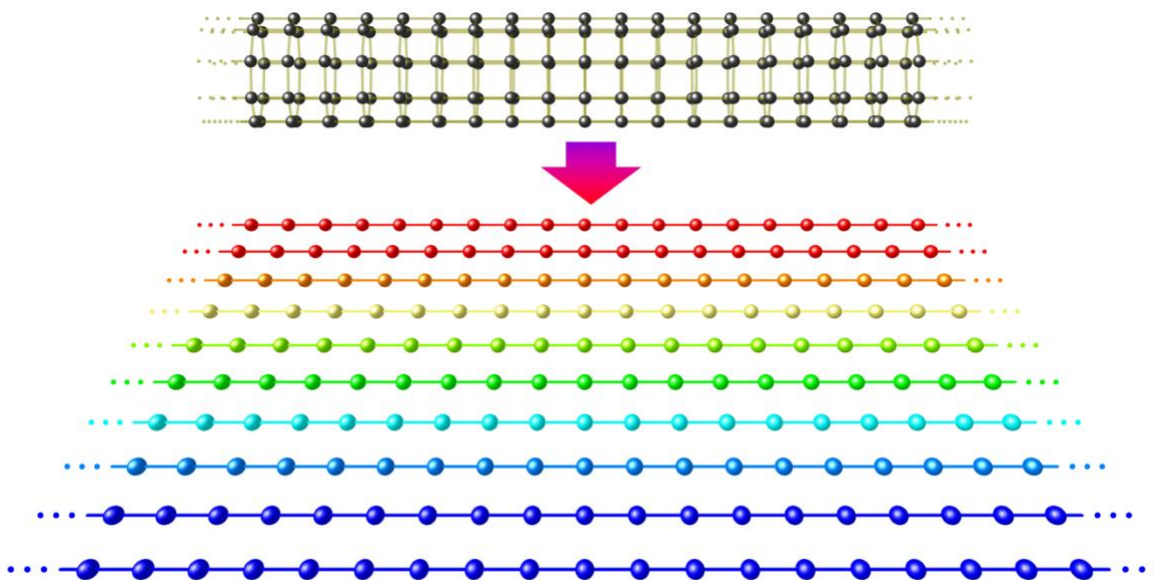


Fig. 3. Esquema de la transformación de un nanotubo en cadenas independientes.

La brecha superconductora del nanotubo se obtiene resolviendo las ecuaciones de BdG. Como se muestra en la parte izquierda de la Figura 4, el procedimiento usual es autoconsistente: primero se propone un valor inicial para la brecha Δ . Con ese valor, se resuelven las ecuaciones de BdG y se obtienen las energías y funciones propias de dichas ecuaciones. A partir de esos resultados, se calcula una nueva Δ , la cual se vuelve a introducir en las ecuaciones y el proceso se repite cíclicamente. El procedimiento termina cuando la diferencia entre las dos brechas consecutivas es suficientemente pequeña, menor que un valor predeterminado.

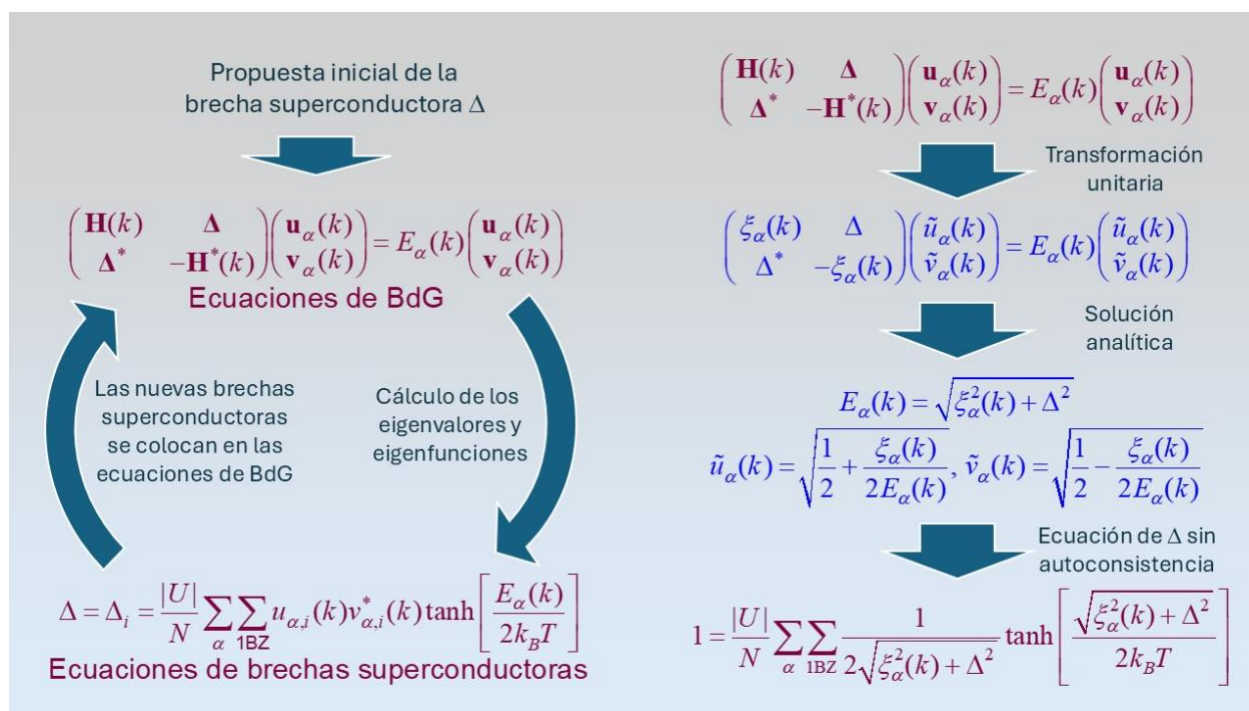


Fig. 4. Solución de las ecuaciones de Bogoliubov-de Gennes (BdG) mediante (Izquierda) el método autoconsistente y (Derecha) el método de matriz por bloques.

Si se aplica la transformación unitaria al nanotubo, como se ilustra en la Figura 3, el problema matemático se simplifica notablemente. La gran matriz de 18×18 asociada a las ecuaciones de BdG se reorganiza en nueve bloques más pequeños de 2×2 , es decir, en lugar de enfrentar una sola matriz grande y complicada, el fenómeno puede analizarse como varios problemas mucho más simples e independientes [7].

Cada uno de estos bloques admite una solución analítica, lo que permite obtener expresiones explícitas para las energías y funciones propias. A partir de ellas, se construye la ecuación que determina la brecha superconductora. Dicha ecuación ya no se resuelve de forma autoconsistente, sino mediante un procedimiento numérico sencillo, como el método de bisección [8], tal como se muestra en la parte derecha de la Figura 4.

En la Figura 5, se compara el tiempo de cómputo necesario para calcular el espectro de la brecha superconductora con 900 puntos, en nanotubos cuya sección transversal va de 3 a 19 átomos. Se muestran dos estrategias: el método autoconsistente tradicional (cuadrados

azules) y el método que utiliza la transformación unitaria para trabajar con matrices por bloques (círculos rojos). La gráfica evidencia que, a medida que aumenta el número de átomos, el tiempo de cálculo crece mucho más rápidamente cuando no se aplica la transformación. En cambio, el enfoque por bloques reduce de forma significativa el costo computacional, permitiendo realizar los mismos cálculos en mucho menos tiempo. Los cálculos se llevaron a cabo en la supercomputadora Mitzli de la UNAM, y los tiempos reportados corresponden al uso equivalente de un solo procesador Intel Xeon E5-2670 v1. Como se observa en los recuadros, la brecha superconductora obtenida por ambos métodos con (izquierda) y sin (derecha) transformación unitaria es la misma, lo cual muestra la exactitud y suficiencia del procedimiento propuesto.

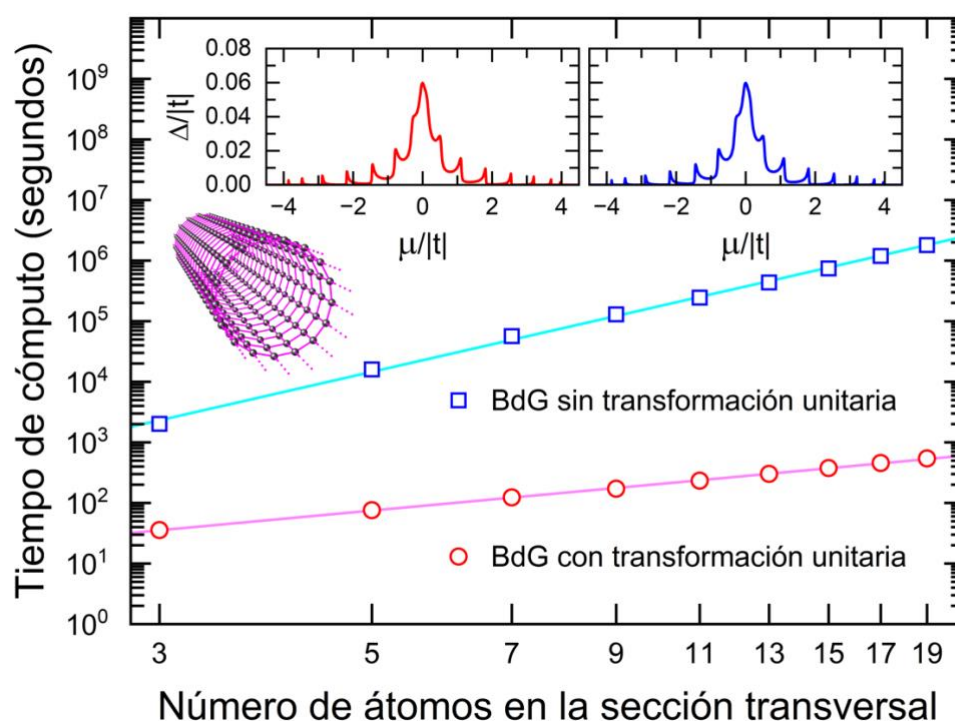


Fig. 5. Comparación del tiempo de cómputo para resolver las ecuaciones de Bogoliubov-de Gennes (BdG) con (círculos rojos) y sin (cuadrados azules) la aplicación de la transformación unitaria, en función del número de átomos en la sección transversal del nanotubo. En los recuadros superiores, se muestran ejemplos del espectro de la brecha superconductora obtenido con cada método (línea roja: con transformación; línea azul: sin transformación), correspondientes al nanotubo ilustrado en el recuadro izquierdo.

La Figura 5 muestra una diferencia notable en los tiempos de cálculo para un nanotubo con 19 átomos en su sección transversal: mientras que el método tradicional (autoconsistente) requería tres semanas de procesamiento continuo, la nueva técnica de matriz por bloques lo resuelve en apenas nueve minutos, usando el mismo procesador. Esta reducción de más de tres mil veces no sólo supone un ahorro de tiempo espectacular, sino que abre la posibilidad de estudiar sistemas mucho más complejos que antes resultaban inabordables.

Conclusiones

La superconductividad ha dejado de ser una curiosidad exclusiva de los laboratorios para convertirse en una tecnología con impacto real en la sociedad. Lo que antes sólo podía observarse en condiciones extremas, hoy forma parte de aplicaciones concretas que mejoran la calidad de vida, tal es el caso de los equipos de resonancia magnética de alta resolución en hospitales.

De cara al futuro, su importancia será aún mayor. La superconductividad es un componente esencial en el desarrollo de reactores de fusión nuclear [9], una posible fuente de energía limpia y prácticamente inagotable. También impulsa avances en transporte, como los trenes de levitación magnética [10], que prometen desplazamientos más rápidos y eficientes. En conjunto, esta tecnología no sólo representa un logro científico, sino una pieza clave en la construcción de sistemas energéticos y de transporte más sostenibles para las próximas décadas.

Para que este futuro tecnológico se concrete, no basta con descubrir nuevos materiales superconductores; también es fundamental desarrollar herramientas matemáticas y computacionales más eficientes que permitan diseñar y estudiar dispositivos reales. En este contexto, el método de matriz por bloques aplicado a las ecuaciones de BdG representa un avance importante.

La idea central es reorganizar el problema mediante transformaciones unitarias que dividen un cálculo grande y complejo en varios más pequeños y manejables. Con ello, se reduce drásticamente el tiempo de cómputo en comparación con el método autoconsistente tradicional. En términos prácticos, esto significa que se pueden analizar nanotubos y otras

nanoestructuras con mayor rapidez y menor costo computacional, facilitando la interpretación de resultados experimentales y la predicción de nuevas propiedades superconductoras [7]. Este tipo de herramientas acerca el diseño cuántico a la aplicación tecnológica, acelerando el desarrollo de dispositivos superconductores más eficientes.

Aunque la técnica de matriz por bloques es una herramienta bien conocida en el álgebra lineal, aplicarla a las ecuaciones de BdG tiene un impacto especial en el estudio de nanotubos superconductores. La combinación de esta técnica con el supercómputo puede contribuir al desarrollo de tecnologías basadas en la superconductividad durante las próximas décadas, en un contexto marcado por el avance de la inteligencia artificial. Esta relación puede visualizarse a través del siguiente ciclo.

La integración del método de matriz por bloques, el cómputo de alto rendimiento y la inteligencia artificial podría formar un ecosistema tecnológico sin precedentes, es decir, el método actúa como el cerebro matemático, simplificando los cálculos cuánticos complejos a problemas manejables, mientras que el supercómputo aporta los músculos, ejecutando esos cálculos simplificados a velocidades vertiginosas y en paralelo, permitiendo explorar simultáneamente miles de candidatos. Por otro lado, la inteligencia artificial cierra el círculo como el gran estratega: aprende de los resultados obtenidos, identifica patrones ocultos en los datos y predice qué compuestos tienen más probabilidades de convertirse en superconductores revolucionarios, así como nuevos dispositivos superconductores. Esta sinergia está transformando radicalmente la investigación. Lo que antes era un proceso lento de prueba y error en el laboratorio, con décadas entre descubrimientos, se ha convertido en un ciclo virtuoso y acelerado.

Agradecimientos

Este trabajo fue apoyado por los proyectos CF-2023-I-830, PAPIIT-IN116625, PAPIIT-IN116126, LANCAD-UNAM-DGTIC-039 y LANCAD-UNAM-DGTIC-182. G.E.L. agradece su estancia posdoctoral realizada gracias al Programa de Becas Posdoctorales en la Universidad Nacional Autónoma de México (UNAM).

Referencias

- [1] M. Tinkham, *Introduction to superconductivity*, 2nd ed., New York, USA: McGraw-Hill, 1996, pp. 1-2.
- [2] O. Navarro and R. Baquero, *Ideas Fundamentales de la Superconductividad*, Ciudad de México, México: Editorial UNAM, 2007.
- [3] T. Akachi, T. Ogasawara, R. Gomez, and C. Wang, "Stability limit field in superconductors. Part 2 multifilamentary composites", *Cryogenics*, vol. 26, no. 6, pp. 358-364, Jun. 1986, doi: [10.1016/0011-2275\(86\)90066-4](https://doi.org/10.1016/0011-2275(86)90066-4)
- [4] M. Noe, *et al.*, "Superconducting high-power cables and lines-development status and technology roadmap", *Supercond. Sci. Technol.* vol. 39, no. 2, p. 023501, Feb. 2026, doi: [10.1088/1361-6668/ae15c2](https://doi.org/10.1088/1361-6668/ae15c2)
- [5] P. G. de Gennes, *Superconductivity of Metals and Alloys*, Boca Raton, FL, USA: Westview Press, 1999, pp. 137-141.
- [6] J.-X. Zhu, *Bogoliubov-de Gennes Method and its Applications*, Cham, Switzerland: Springer Int. Pub., 2016, pp. 37-64.
- [7] G. E. López and C. Wang, "Superconductivity in correlated carbon nanotubes under pressure: A Bogoliubov-de Gennes study", *Physica B: Condensed Matter*, vol. 675, p. 415602, Feb. 2024, doi: [10.1016/j.physb.2023.415602](https://doi.org/10.1016/j.physb.2023.415602)
- [8] R. L. Burden, J. D. Faires and A. M. Burden, *Análisis numérico*, 10th ed. Ciudad de México, México: Cengage Learning, 2017, pp. 36-40.
- [9] H. Liu, S. Gao, W. Hong and F. Liu, "Overview of China's fusion magnet technology based on the superconducting Tokamak strategy", *Cryo*, vol. 2, no. 1, p. 3, Feb. 2026, doi: [10.3390/cryo2010003](https://doi.org/10.3390/cryo2010003)
- [10] J.-S. Wang and S.-Y. Wang, *High temperature superconducting magnetic levitation*, Berlin, Germany: Walter de Gruyter, 2018.