TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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 *(fine-tuning)*	Proceso de ajuste adicional de un modelo preentrenado utilizando datos específicos para adaptarlo a una tarea particular.
Ciclo de entrenamiento *(epoch)*	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.
in silico	Experimentos o análisis realizados a través de una computadora.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

Término	Definición
Instrucción de entrada *(prompt)*	Instrucción que se le da a un modelo para que genere una respuesta.
in vitro	Experimentos dentro de un laboratorio.
Mecanismo de atención	Proceso mediante el cual el modelo pondera la relevancia de otros *tokens* 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 (*tokens*) 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 *(benchmark)*	Conjunto de secuencias o tareas de referencia empleado para evaluar y comparar el desempeño de modelos.
Representación vectorial *(embedding)*	Representación vectorial numérica de una secuencia o residuo de proteína que captura relaciones contextuales y funcionales.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

Término	Definición
Semilla aleatoria *(seed)*	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.
Transformer	Arquitectura base de los modelos de lenguaje que aprende relaciones entre los elementos de una secuencia identificando los más relevantes.
Unidad básica de secuencia *(token)*	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).

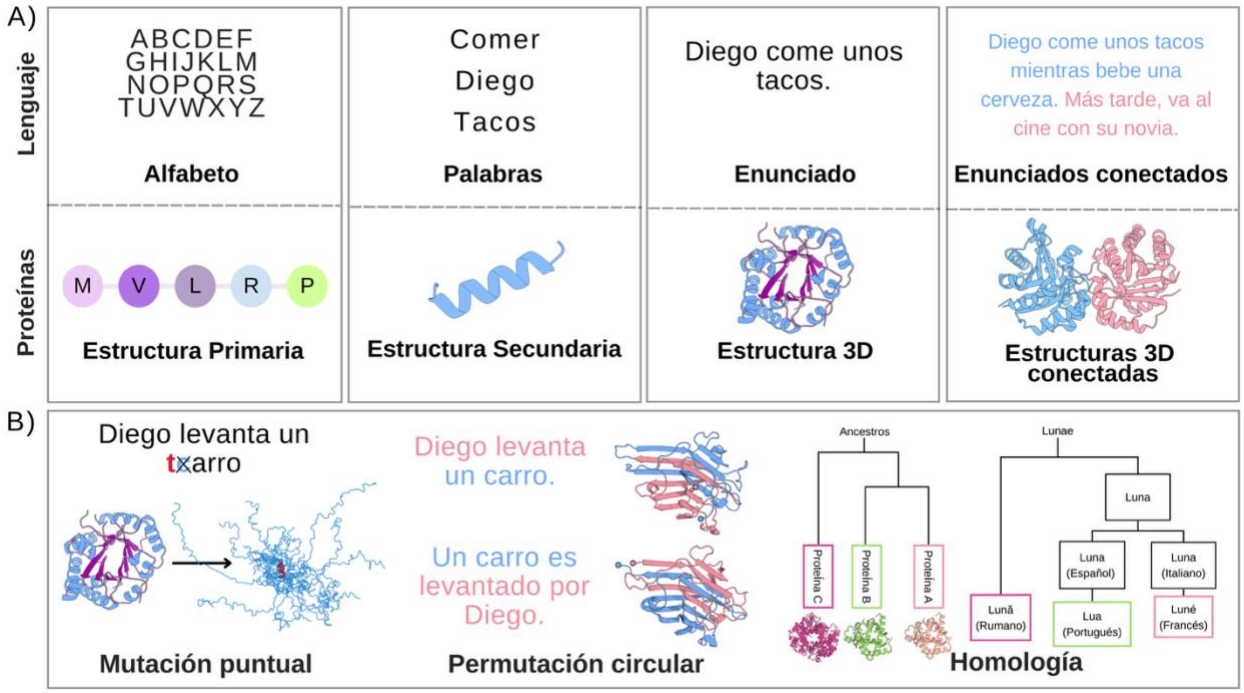

Fig. 1. Analogía entre el lenguaje natural y la organización estructural de las proteínas. A) En la parte superior, se muestra la jerarquía del lenguaje, desde el alfabeto hasta enunciados conectados, ilustrando cómo las unidades básicas se combinan para generar significado en niveles superiores. En 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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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].

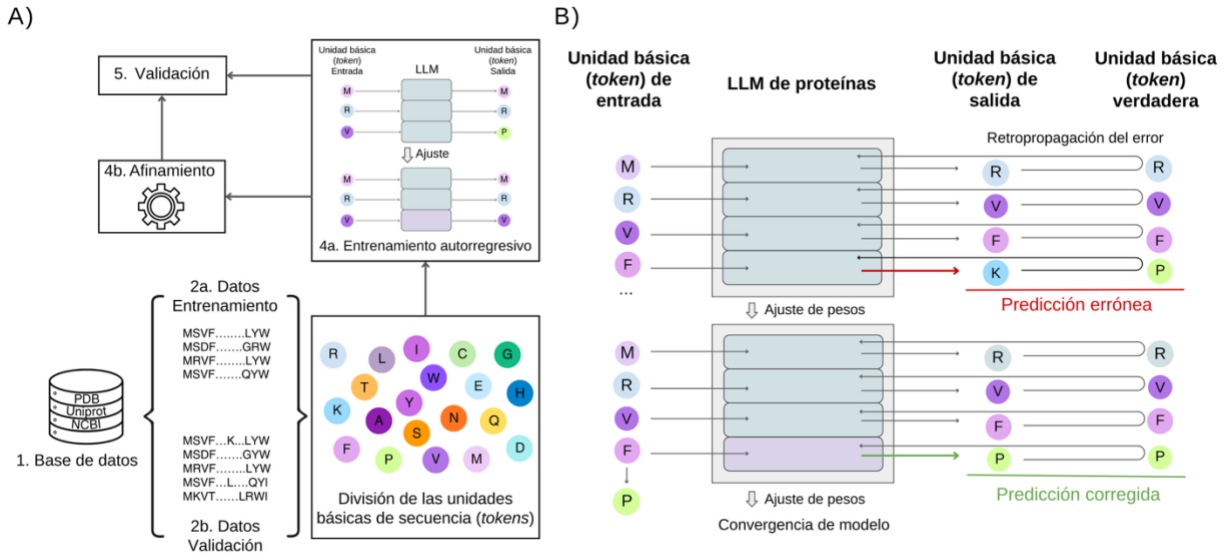

Fig. 2. Representación gráfica del proceso de entrenamiento de un LLM para la generación de secuencias de proteínas. A) Flujo de trabajo para el entrenamiento y validación del modelo. Las secuencias proteicas se obtienen de bases de datos estructurales y de secuencia, se dividen en conjuntos de entrenamiento 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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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-tunning*, 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].

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

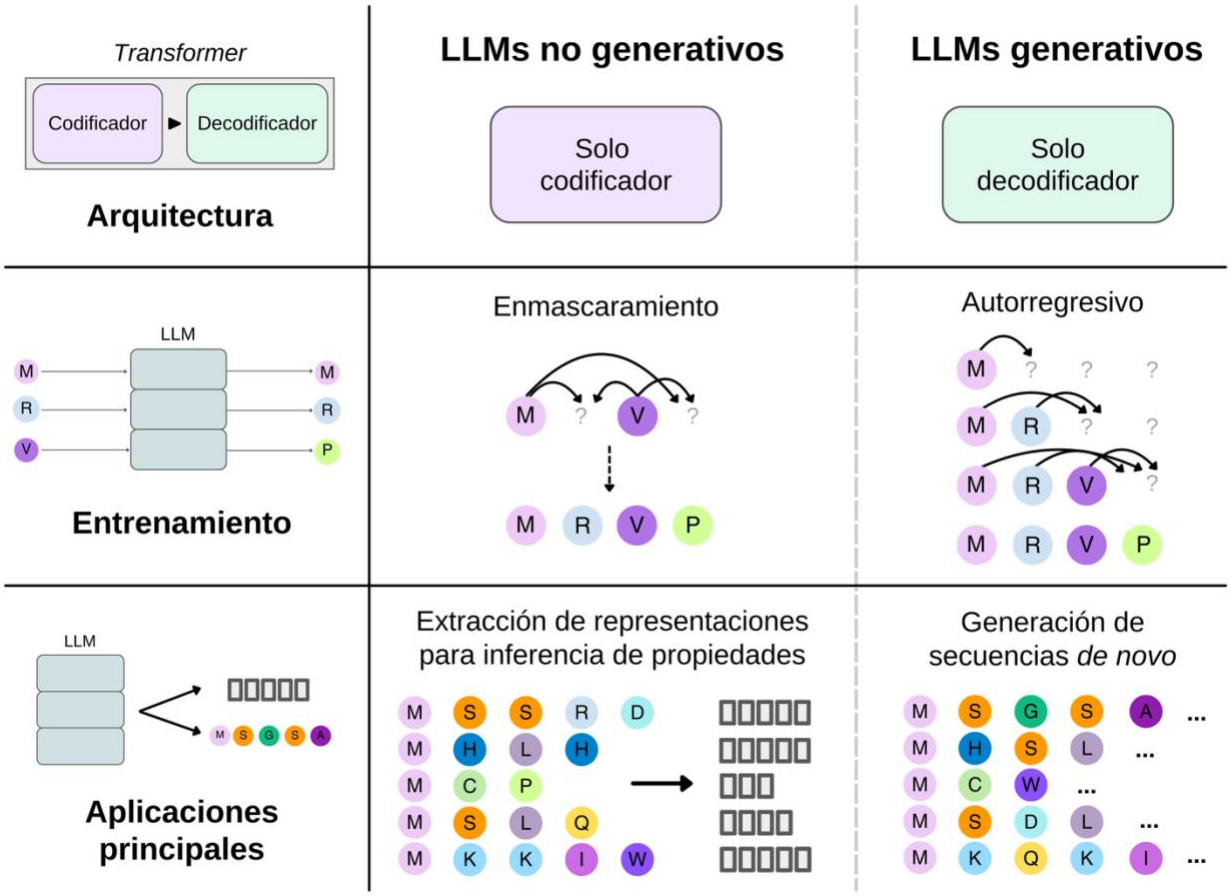

Fig. 3. Arquitecturas basadas en Transformer utilizadas en el campo del diseño de proteínas. La arquitectura Transformer original utiliza un módulo codificador que procesa y genera representaciones de las secuencias de entrada, y un módulo decodificador que toma las representaciones para generar las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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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).

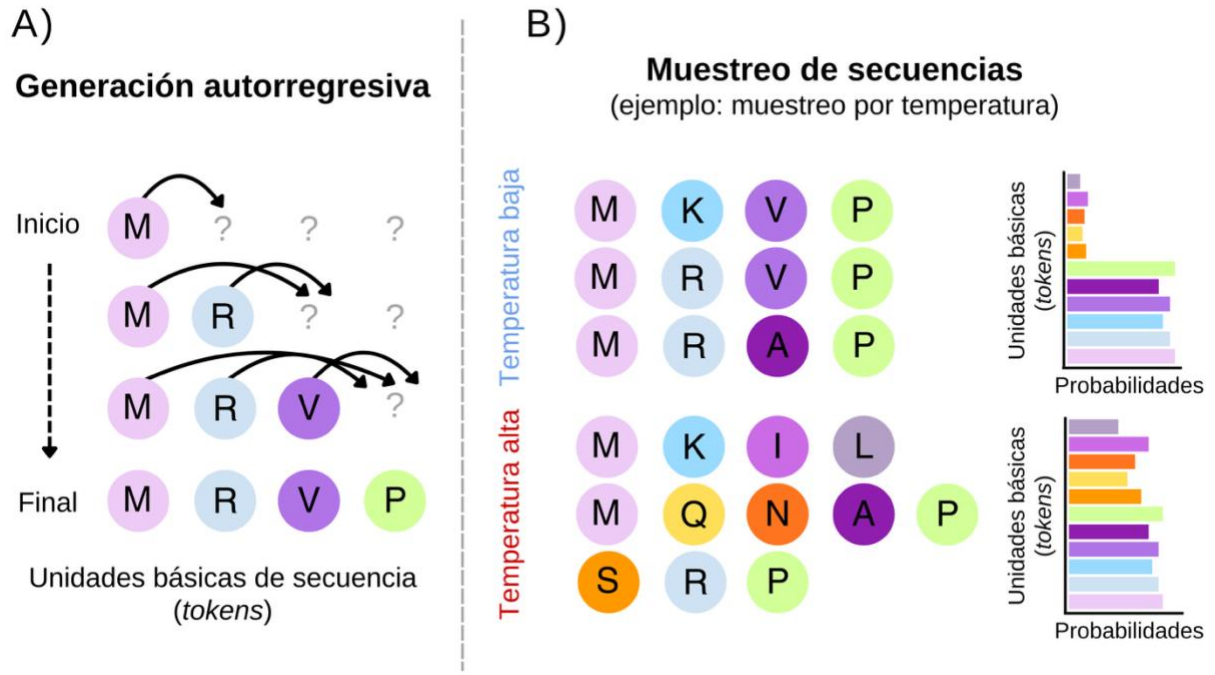

Fig. 4. Generación autorregresiva y estrategias de muestreo para la generación de secuencias proteicas. A) La generación se realiza de manera secuencial, prediciendo iterativamente la siguiente unidad básica de secuencia

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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]

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
ProtGPT2	Decodificador (GPT-2, autorregresivo)	UniRef50	Generación de proteínas *de novo*	[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 *de novo* y predicción de aptitud	[36]
ProGen3	Decodificador (autorregresivo, y de relleno)	Profluent Protein Atlas v1	Generación de proteínas *de novo* 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]

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

LLM	Arquitectura (estrategia de entrenamiento)	Datos de entrenamiento	Aplicación	Referencia
DPLM	Codificador y Decodificador (difusión)	UniRef50	Generación de secuencias *de novo* 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]

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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 *de novo* 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]

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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]

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

[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] Standford 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/gkae1010.

[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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

[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. Brixi *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. Negrevergne, 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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

[55] O. Conchillo-Solé, N. S. de Groot, F. X. Avilés, J. Vendrell, X. Daura, and S. Ventura, "AGGRESCAN: 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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

[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.

TiES Revista de Tecnología e Innovación en Educación Superior
www.ties.unam.mx

[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.