

Instituto Politécnico Nacional



**Unidad Profesional Interdisciplinaria en Ingeniería
y Tecnologías Avanzadas**

Ingeniería Telemática

Dra. Susana Araceli Sánchez Nájera

**Sistema de Reconocimiento de Expresiones
Matemáticas (MER) de cálculo diferencial e integral
con conversión a formato LaTeX**

Reporte parcial 1

Alumno:

Diego Ordoñez Antonio

Asesora:

Dr. Laura Ivoone Garay Jimenez

1	Contenido	
2	Introducción	4
2.1	Propósito	4
2.2	Alcance	4
2.3	Personal involucrado	4
2.4	Definiciones, acrónimos y abreviaturas	5
2.5	Referencias	5
2.6	Resumen	5
3	Descripción general	6
3.1	Perspectiva del producto	6
3.2	Funcionalidad del producto	7
3.3	Características de los usuarios	7
3.4	Restricciones	7
3.5	Suposiciones y dependencias	8
3.6	Evolución previsible del sistema	8
4	Requerimientos específicos	8
4.1	Requerimientos funcionales: Gestión de imágenes	8
4.1.1	Requerimiento funcional 1	9
4.1.2	Requerimiento funcional 2	9
4.1.3	Requerimiento funcional 3	10
4.2	Requerimientos funcionales: Procesamiento y Transcripción	12
4.2.1	Requerimiento funcional 1	12
4.2.2	Requerimiento funcional 2	13
4.3	Requerimientos funcionales: Interfaz de usuario	14
4.3.1	Requerimiento funcional 1	14
4.3.2	Requerimiento funcional 2	15
4.3.3	Requerimiento funcional 3	16
4.4	Requerimientos funcionales: Administración	17
4.4.1	Requerimiento funcional 1	17
4.4.2	Requerimiento funcional 2	18
4.4.3	Requerimiento funcional 3	19
4.5	Requerimientos no funcionales	20
4.5.1	Requerimiento no funcional 1	20

5	Funcionamiento general	21
5.1	Módulo de página web	22
5.1.1	Página web	22
5.1.2	Obtención de imagen	22
5.1.3	Servidor	22
5.2	Módulo de procesamiento de imágenes	23
5.2.1	Preprocesamiento de imágenes	23
5.2.2	Modelo MER	23
5.2.3	Base de datos de etiquetas	23
5.2.4	Post-procesamiento	23
5.3	Módulo de entrenamiento	24
5.3.1	Dataset de imágenes	24
5.3.2	Banco de imágenes preprocesadas	24
5.3.3	Dataset	24
5.3.4	Interpretación del modelo	24
6	Investigación modelo MER	25
6.1	Red Neuronal Convolutiva	25
6.2	Modelo Transformer	26

2 Introducción

En el presente documento de Análisis de Requerimientos tiene como objetivo proporcionar una visión general y detallada de los requerimientos funcionales y no funcionales para el desarrollo y la implementación del servidor para una página web de consulta del modelo MER que tiene como finalidad ser una herramienta auxiliar para la conversión de imágenes de ecuaciones a código en LaTeX y así agilizar el proceso de elaboración de artículos científicos y reportes en el ámbito académico.

Este documento servirá como una guía fundamental para el desarrollo, diseño y evaluación del servidor que aloje al modelo MER. Proporciona un conjunto completo de requerimientos funcionales y no funcionales que guiarán a desarrollador. El análisis de requerimientos es un paso crítico en el proceso de desarrollo, ya que garantiza que todos los aspectos clave del proyecto sean claramente definidos y comprendidos por todas las partes involucradas. Además, establece una base sólida para la planificación y la ejecución del proyecto.

2.1 Propósito

El propósito del documento de Análisis de Requerimientos para el sistema MER es proporcionar una visión detallada y clara de los requerimientos funcionales y no funcionales que guiarán el desarrollo y la implementación de la plataforma web. Este documento se utiliza como una referencia esencial para asegurar que todos los aspectos clave del proyecto estén bien definidos y comprendidos por todas las partes involucradas. Los requerimientos sirven como la base sobre la cual se construirá el servidor, garantizando que el sistema final cumpla con las necesidades.

2.2 Alcance

La página web será una plataforma en línea que permitirá a los usuarios convertir imágenes de ecuaciones matemáticas con tipografía de escritura a código en formato LaTeX. Los usuarios podrán subir imágenes a la página web, visualizar el código de LaTeX y copiar el código perteneciente a la ecuación de cálculo diferencia e integral, no se requerirá que el usuario este registrado previamente.

2.3 Personal involucrado

Nombre	Diego Ordoñez Antonio
Rol	Administrador de proyecto, desarrollador
Categoría profesional	Estudiante de Ingenieria Telemática
Responsabilidades	Administrador, programador
Información de contacto	tonidiego2801@hotmail.com

2.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario no registrado	Persona que ingresa a la plataforma que tiene perfil
Administrador	Persona que gestiona la plataforma y sus procesos
Servidor	Dispositivo físico que aloja la parte funcional del proyecto donde cualquier usuario puede realizar consultas
RF	Requerimiento funcional
RNF	Requerimiento no funcional
MER	Reconocimiento de expresiones matemáticas (Mathematical Expression Recognition)
LaTeX	Lenguaje de marcado para la elaboración de documentos

2.5 Referencias

Título	Referencia
Standard IEEE 830-1998	https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf
Libro Análisis de requerimientos	Análisis de requerimientos, M. del Carmen Gómez, UAM (2011)

2.6 Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

3 Descripción general

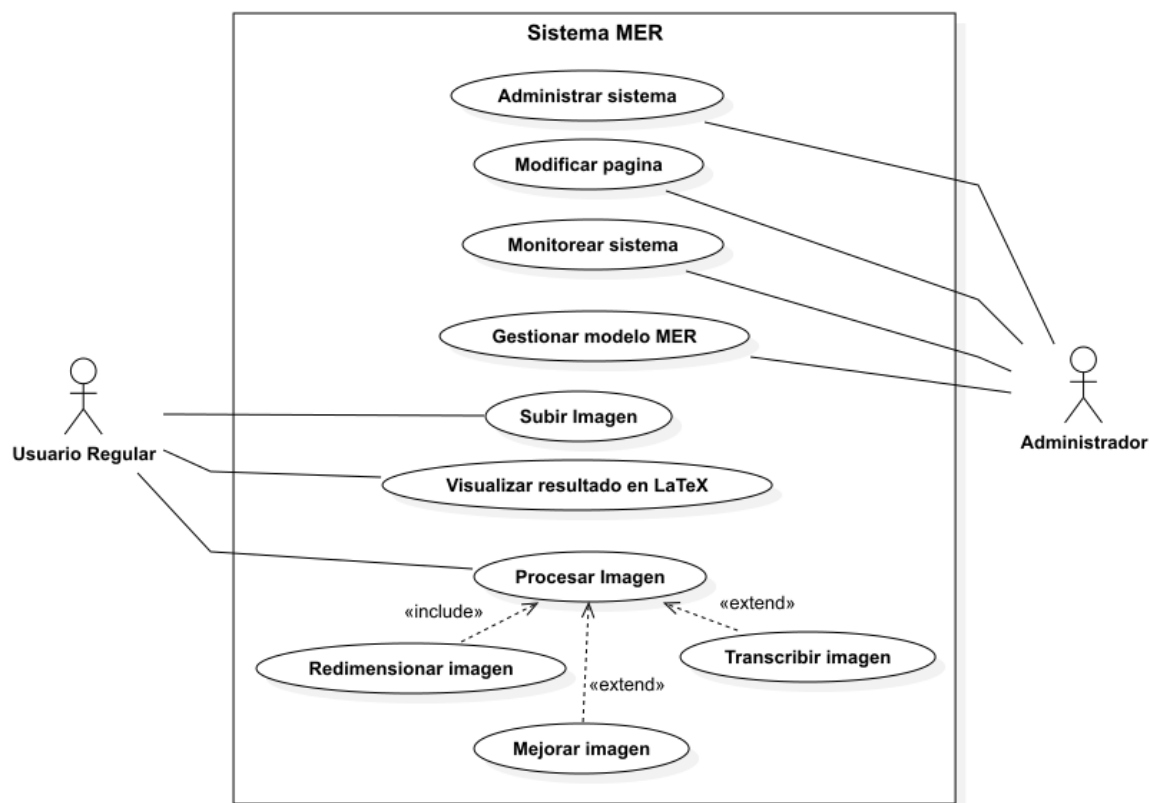
3.1 Perspectiva del producto

El diseño de la página web del proyecto está contemplada para funcionar como una herramienta auxiliar para los usuarios. La plataforma se plantea con un diseño intuitivo y fácil de usar, permitiendo a los usuarios entender la funcionalidad sin una previa inducción, de esta manera el usuario podrá subir imágenes con ecuaciones de tipografía de escritura para poder obtener la transcripción correspondiente a la ecuación en formato de código de LaTeX, posteriormente cuando el usuario suba su imagen dará clic al botón de procesar imagen para que sea enviada al servidor, el servidor procesará la imagen y retornará el código en formato LaTeX, se espera que el servidor responda en un lapso de tiempo no mayor a 30 segundos y que ejecute múltiples peticiones al mismo tiempo es decir, no se tiene contemplado la restricción de velocidad o tiempo de ejecución.

El sistema será capaz de procesar imágenes en formato png, jpg, bmp y svg, ya que son los formatos más comunes, cabe mencionar que no se solicitará una resolución mínima o máxima ya que se contempla que el sistema realizará redimensión de la imagen para poder ser procesada además de realizar un mejoramiento de las características de la imagen como eliminación de ruido, realce de detalles y corrección de perspectiva. No se contará con registro de usuarios ya que el enfoque del proyecto es la realización del modelo MER para la transcripción de ecuaciones, solo el administrador tendrá permiso para modificar la página.

Se planea elegir entre una red neuronal tipo transformador o una red neuronal convolucional para el modelo MER, el dataset aún está por definirse y con respecto a los requerimientos para el entrenamiento de la red neuronal se planea utilizar una tarjeta grafica RTX 3050 de 4GB VRAM con 16 GB de memoria RAM, además del uso de CUDA ya que es un conjunto de herramientas desarrollado por NVIDIA para codificar algoritmos en GPU.

3.2 Funcionalidad del producto



3.3 Características de los usuarios

Tipo de usuario	Usuario no registrado
Formación	Cliente
Habilidades	Visualizar
Actividades	Navegar, subir imágenes y obtener código de LaTeX

Tipo de usuario	Administrador
Formación	Dueño de la plataforma
Habilidades	Administrar plataforma, Navegar
Actividades	Administrar productos, gestionar usuarios

3.4 Restricciones

- Interfaz para ser usada desde un navegador con internet
- Lenguajes y tecnologías en uso por definir.
- El sistema se diseñará según un modelo cliente/servidor.

- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.
- El sistema no está diseñado para realizar transcripciones de ecuaciones con tipografía manuscrita.

3.5 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma.

3.6 Evolución previsible del sistema

- Uso de Python para el desarrollo de la plataforma
- Uso de Pytorch para el desarrollo del sistema MER

4 Requerimientos específicos

A continuación, se muestra el análisis de los requerimientos funcionales y los requerimientos no funcionales de acuerdo con el estándar IEEE 830 para el sistema descrito en la sección anterior.

4.1 Requerimientos funcionales: Gestión de imágenes

A continuación, se muestran los requerimientos funcionales del sistema de acuerdo con la gestión de imágenes.

4.1.1 Requerimiento funcional 1

Identificador: RF1.1		Nombre: Carga de imágenes con expresiones matemáticas	
Tipo: Funcional		Requerimiento que lo utiliza: RF1.3 y RF2.1	¿Crítico? Si
Prioridad de desarrollo: Alta		Documentos de visualización asociados: Diagrama de flujo del proceso de carga de imágenes	
Entrada: <ul style="list-style-type: none">Imagen con ecuaciones matemáticas manuscritas en formato PNG, JPG, BMP o SVG.		Salida: <ul style="list-style-type: none">Imagen cargada en el sistema y lista para preprocesamiento y procesamiento.	
Precondición: <ul style="list-style-type: none">El usuario debe tener una imagen en los formatos admitidos.El sistema debe estar operativo y accesible desde una interfaz web.			
Descripción: El sistema permitirá al usuario subir una imagen que contenga una ecuación matemática manuscrita. Esta imagen puede estar en formato PNG, JPG, BMP o SVG. El sistema verificará el formato y procederá a almacenar temporalmente la imagen en el servidor para su posterior preprocesamiento. Este paso es crucial para que el sistema pueda procesar las ecuaciones y convertirlas en código LaTeX.			
Postcondición: <ul style="list-style-type: none">La imagen ha sido cargada correctamente y está lista para ser procesada.La imagen está disponible para el preprocesamiento (RF1.3) y posterior transcripción a LaTeX (RF2.1).			
Situaciones anormales: <ul style="list-style-type: none">El usuario intenta subir un archivo en un formato no admitido (por ejemplo, PDF, GIF).La imagen no cumple con los criterios básicos de legibilidad (demasiado ruido, desenfoco).Fallo en la conexión del servidor durante la carga de la imagen.			
Criterios de aceptación: <ul style="list-style-type: none">El sistema debe permitir la carga de imágenes en los formatos PNG, JPG, BMP, SVG.Las imágenes deben ser correctamente almacenadas para ser procesadas posteriormente.En caso de formatos no admitidos o problemas durante la carga, el sistema debe mostrar un mensaje de error al usuario explicando el problema.La imagen cargada debe estar disponible para el siguiente paso de preprocesamiento.			

4.1.2 Requerimiento funcional 2

Identificador: RF1.2	Nombre: Soporte de formatos de imagen
-----------------------------	--

Tipo: Funcional	Requerimiento que lo utiliza: RF1.1 y RF1.3	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Especificaciones de formatos de imagen soportados	
Entrada: <ul style="list-style-type: none"> Una imagen en formato PNG, JPG, BMP o SVG 		Salida: <ul style="list-style-type: none"> Imagen aceptada y lista para preprocesamiento.
Precondición: <ul style="list-style-type: none"> El usuario debe cargar una imagen en uno de los formatos admitidos (PNG, JPG, BMP o SVG). El sistema debe ser capaz de reconocer el formato de la imagen cargada. 		
Descripción: El sistema debe ser capaz de recibir y procesar imágenes en los formatos más comunes, que son PNG, JPG, BMP y SVG . Estos formatos son los más utilizados para imágenes digitales y garantizan la compatibilidad y la flexibilidad para los usuarios al momento de subir ecuaciones con tipografía de escritura. El sistema verificará que el formato de la imagen cargada esté entre los admitidos, permitiendo su procesamiento o, en caso contrario, notificará al usuario con un error.		
Postcondición: <ul style="list-style-type: none"> La imagen cargada en cualquiera de los formatos admitidos ha sido aceptada y está lista para el preprocesamiento automático (RF1.3). Si la imagen no está en un formato compatible, el usuario es notificado y se le pide que cargue un archivo en uno de los formatos admitidos. 		
Situaciones anormales: <ul style="list-style-type: none"> El usuario intenta cargar una imagen en un formato no soportado (ej. GIF, TIFF). La imagen cargada tiene una extensión compatible, pero está corrupta o dañada. Fallos del servidor o problemas de red que impidan verificar el formato de la imagen. 		
Criterios de aceptación: <ul style="list-style-type: none"> El sistema debe aceptar imágenes en formato PNG, JPG, BMP y SVG. Si la imagen cargada está en un formato no admitido, el sistema debe mostrar un mensaje de error informativo al usuario. Las imágenes aceptadas deben ser procesadas correctamente en los siguientes pasos del sistema (preprocesamiento y transcripción). 		

4.1.3 Requerimiento funcional 3

Identificador: RF1.3	Nombre: Preprocesamiento automático de imágenes	
Tipo: Funcional	Requerimiento que lo utiliza: RF2.1	¿Crítico? Si

Prioridad de desarrollo:	Documentos de visualización asociados: Diagrama de flujo del preprocesamiento de imágenes	
Entrada: <ul style="list-style-type: none"> Imagen cargada en formato PNG, JPG, BMP o SVG que contiene ecuaciones matemáticas manuscritas. 		Salida: <ul style="list-style-type: none"> Imagen preprocesada lista para ser procesada y transcrita a código LaTeX.
Precondición: <ul style="list-style-type: none"> La imagen ha sido cargada y aceptada en el sistema (RF1.1 y RF1.2). El sistema debe estar operativo con los algoritmos de preprocesamiento implementados. 		
Descripción: El sistema debe realizar de manera automática varias operaciones de preprocesamiento en las imágenes cargadas para mejorar su calidad y facilitar la correcta identificación de ecuaciones matemáticas manuscritas. Estas operaciones incluyen: <ul style="list-style-type: none"> Redimensionamiento automático: Ajuste del tamaño de la imagen para un procesamiento óptimo. Eliminación de ruido: Reducción de imperfecciones en la imagen que puedan interferir con el reconocimiento de los caracteres. Realce de detalles: Mejora del contraste y nitidez de la imagen para resaltar los trazos manuscritos. Corrección de perspectiva: Ajuste de la imagen para corregir inclinaciones o distorsiones que dificulten la lectura de la ecuación. 		
Postcondición: <ul style="list-style-type: none"> La imagen ha sido mejorada y está lista para ser procesada por el modelo de reconocimiento de ecuaciones (MER). La imagen debe estar libre de ruido, correctamente redimensionada y corregida en su perspectiva para garantizar un mejor reconocimiento de los caracteres. 		
Situaciones anormales: <ul style="list-style-type: none"> El sistema no puede corregir una imagen excesivamente borrosa o con demasiadas imperfecciones. Fallo en uno de los algoritmos de preprocesamiento (por ejemplo, la corrección de perspectiva no se aplica correctamente). La imagen es redimensionada incorrectamente, afectando su contenido y dificultando el procesamiento posterior. 		
Criterios de aceptación: <ul style="list-style-type: none"> El sistema debe aplicar correctamente las operaciones de preprocesamiento (redimensionamiento, eliminación de ruido, realce de detalles, corrección de perspectiva) a las imágenes cargadas. La calidad de las imágenes preprocesadas debe ser suficiente para que el modelo MER pueda realizar un reconocimiento preciso. En caso de que la imagen no pueda ser preprocesada correctamente, el sistema debe notificar al usuario y sugerir una nueva imagen. 		

4.2 Requerimientos funcionales: Procesamiento y Transcripción

4.2.1 Requerimiento funcional 1

Identificador: RF2.1	Nombre: Procesamiento de imágenes y conversión en código LaTeX	
Tipo: Funcional	Requerimiento que lo utiliza: RF2.2	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Diagrama de flujo del preprocesamiento de imágenes	
Entrada: <ul style="list-style-type: none"> Imagen preprocesada (RF1.3) que contiene ecuaciones matemáticas con tipografía de escritura 		Salida: <ul style="list-style-type: none"> Código LaTeX que corresponde a la ecuación en la imagen.
Precondición: <ul style="list-style-type: none"> La imagen debe haber sido preprocesada correctamente (RF1.3). El modelo de reconocimiento de ecuaciones (MER) debe estar operativo y entrenado. 		
Descripción: El sistema debe procesar las imágenes preprocesadas para reconocer las ecuaciones que contienen y convertirlas en código LaTeX . Este proceso se llevará a cabo utilizando un modelo de red neuronal (transformador o convolucional) especializado en el reconocimiento de caracteres y símbolos matemáticos. El modelo identificará los trazos en la imagen y los mapeará a su equivalente en LaTeX, produciendo una transcripción precisa de las ecuaciones.		
Postcondición: <ul style="list-style-type: none"> El código LaTeX ha sido generado y está listo para ser mostrado al usuario (RF2.2). Si el sistema no puede reconocer la ecuación correctamente, se notificará al usuario y se le ofrecerán opciones para intentar nuevamente o corregir el error. 		
Situaciones anormales: <ul style="list-style-type: none"> El modelo de reconocimiento no puede identificar algunos símbolos debido a una baja calidad de la imagen o un trazo demasiado ambiguo. Fallo en la conversión de la imagen a código LaTeX. Tiempo de procesamiento demasiado largo, excediendo el límite de 30 segundos (RF2.3). 		
Criterios de aceptación: <ul style="list-style-type: none"> El sistema debe convertir correctamente las ecuaciones manuscritas en código LaTeX, con una tasa de precisión adecuada. El proceso de conversión debe completarse en menos de 30 segundos por imagen (RF2.3). El código LaTeX generado debe estar en formato correcto y ser utilizable para su renderización y edición posterior. 		

4.2.2 Requerimiento funcional 2

Identificador: RF2.2	Nombre: Visualización del resultado en código LaTeX	
Tipo: Funcional	Requerimiento que lo utiliza: RF2.1 y RF3.3	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Prototipo de la interfaz de usuario que muestra el campo donde se visualizará el código LaTeX	
Entrada: <ul style="list-style-type: none"> Código LaTeX generado a partir del procesamiento de la imagen (RF2.1). 		Salida: <ul style="list-style-type: none"> El código LaTeX se muestra en la interfaz gráfica para que el usuario pueda verlo, copiarlo o modificarlo.
Precondición: <ul style="list-style-type: none"> El código LaTeX debe haber sido generado correctamente tras el procesamiento de la imagen (RF2.1). La interfaz debe estar operativa para mostrar el resultado. 		
Descripción: El sistema debe presentar el código LaTeX resultante del procesamiento de la imagen en una sección visible de la interfaz de usuario. El usuario debe poder visualizar el código LaTeX generado para asegurarse de que la transcripción de la ecuación manuscrita es correcta. Además, la interfaz debe permitir que el usuario copie el código para utilizarlo en otros entornos (por ejemplo, editores de texto matemático o plataformas de documentación).		
Postcondición: <ul style="list-style-type: none"> El código LaTeX es visible en la interfaz del usuario. El usuario puede copiar el código generado. En caso de errores en la transcripción, el sistema debe mostrar un mensaje claro al usuario. 		
Situaciones anormales: <ul style="list-style-type: none"> El sistema no muestra el código LaTeX debido a un fallo en el procesamiento o la comunicación entre el servidor y la interfaz. El código LaTeX es incorrecto o incompleto. Problemas de visualización en dispositivos o navegadores específicos. 		
Criterios de aceptación: <ul style="list-style-type: none"> El sistema debe mostrar correctamente el código LaTeX generado. El código LaTeX debe ser visible en un formato que permita su fácil copia o edición. El sistema debe notificar al usuario si hay problemas en la transcripción o si el código LaTeX no puede generarse. 		

4.3 Requerimientos funcionales: Interfaz de usuario

4.3.1 Requerimiento funcional 1

Identificador: RF3.1	Nombre: Interfaz intuitiva y fácil de usar	
Tipo: Funcional	Requerimiento que lo utiliza: RF3.2 y RF3.3	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Prototipo de interfaz que muestra el diseño propuesto y la disposición de los elementos	
Entrada: <ul style="list-style-type: none">Interacción del usuario con los elementos de la interfaz (botones, campos de entrada, etc.).		Salida: <ul style="list-style-type: none">Respuestas claras y precisas a las acciones del usuario (procesar imagen, mostrar resultados, etc.).
Precondición: <ul style="list-style-type: none">La interfaz debe estar completamente implementada y operativa.El usuario debe tener acceso a la plataforma sin necesidad de un registro o autenticación.		
Descripción: <p>La interfaz del sistema debe ser diseñada para ser intuitiva, permitiendo que los usuarios comprendan fácilmente cómo utilizar la herramienta sin necesidad de capacitación previa. Esto implica una disposición clara y lógica de los elementos, un uso consistente de los iconos y etiquetas, y una interacción fluida. Se debe priorizar la accesibilidad y la facilidad de navegación para que cualquier usuario, independientemente de su experiencia técnica, pueda cargar imágenes y obtener resultados sin complicaciones.</p>		
Postcondición: <ul style="list-style-type: none">Los usuarios pueden interactuar con la interfaz sin experimentar confusión o dificultades.La experiencia del usuario es positiva, resultando en una alta tasa de satisfacción.		
Situaciones anormales: <ul style="list-style-type: none">Los usuarios no pueden encontrar la manera de cargar imágenes debido a un diseño confuso.Dificultades técnicas en la interfaz que impiden a los usuarios completar el proceso de transcripción.Mensajes de error que no son claros o que no proporcionan información útil para el usuario.		
Criterios de aceptación: <ul style="list-style-type: none">Los usuarios deben ser capaces de utilizar la interfaz para cargar imágenes y obtener resultados sin necesidad de instrucciones adicionales.		

- La retroalimentación del usuario debe indicar que la interfaz es clara y fácil de navegar.

4.3.2 Requerimiento funcional 2

Identificador: RF3.2	Nombre: Botón de procesamiento de imágenes	
Tipo: Funcional	Requerimiento que lo utiliza: RF3.1 y RF3.3	¿Critico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Prototipo de interfaz que incluya la ubicación y el diseño del botón	
Entrada: <ul style="list-style-type: none"> • Interacción del usuario al hacer clic en el botón para procesar la imagen. 		Salida: <ul style="list-style-type: none"> • El sistema inicia el proceso de transcripción de la imagen y muestra un indicador de carga o procesamiento.
Precondición: <ul style="list-style-type: none"> • La imagen debe haber sido cargada correctamente antes de que el usuario haga clic en el botón de procesamiento. 		
Descripción: El sistema debe incluir un botón claramente visible en la interfaz que permita a los usuarios iniciar el procesamiento de las imágenes que han cargado. Este botón debe ser de un tamaño y color que lo hagan destacar en la interfaz, y debe estar ubicado en una posición lógica que facilite su identificación. Al hacer clic en el botón, el sistema debe comenzar a procesar la imagen y proporcionar retroalimentación al usuario sobre el estado de la operación (por ejemplo, mostrando un indicador de carga).		
Postcondición: <ul style="list-style-type: none"> • El sistema ha comenzado el procesamiento de la imagen tras el clic en el botón. • Se proporciona un indicador de carga o procesamiento mientras la imagen es procesada. 		
Situaciones anormales: <ul style="list-style-type: none"> • El botón no responde al ser clicado debido a un fallo en la interfaz. • El botón es difícil de encontrar o identificar, lo que provoca que los usuarios no sepan cómo continuar. • Problemas de visualización que hacen que el botón no sea visible en ciertos dispositivos o navegadores. 		
Criterios de aceptación: <ul style="list-style-type: none"> • El botón debe ser visible y accesible en todas las resoluciones de pantalla y dispositivos. 		

- Los usuarios deben ser capaces de iniciar el procesamiento de imágenes haciendo clic en el botón sin dificultad.

4.3.3 Requerimiento funcional 3

Identificador: RF3.3	Nombre: Visualización de imagen y transcripción	
Tipo: Funcional	Requerimiento que lo utiliza: RF2.1 y RF3.2	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Prototipo de interfaz que incluya la disposición visual de la imagen cargada y el resultado en LaTeX.	
Entrada: <ul style="list-style-type: none"> • Imagen cargada por el usuario y el código LaTeX generado tras el procesamiento. 		Salida: <ul style="list-style-type: none"> • Imagen cargada visible en la interfaz. • Código LaTeX generado visible en la interfaz, separado de la imagen para fácil visualización.
Precondición: <ul style="list-style-type: none"> • El usuario debe haber subido una imagen y haber iniciado el proceso de transcripción. • El procesamiento de la imagen debe haberse completado correctamente. 		
Descripción: La interfaz del sistema debe mostrar claramente dos elementos esenciales: la imagen que el usuario ha cargado y el resultado del procesamiento en formato LaTeX. Estos elementos deben estar dispuestos de manera lógica y fácilmente visibles para el usuario. La imagen cargada debe ser presentada para confirmar que se subió correctamente, y el código LaTeX debe mostrarse claramente para que el usuario pueda revisar el resultado de la transcripción. El diseño debe ser responsivo, adaptándose a diferentes resoluciones de pantalla.		
Postcondición: <ul style="list-style-type: none"> • La imagen cargada se muestra en la interfaz, permitiendo al usuario verificar que es la imagen correcta. • El código LaTeX generado se muestra en la interfaz, listo para ser copiado o utilizado por el usuario. 		
Situaciones anormales: <ul style="list-style-type: none"> • La imagen cargada no se muestra correctamente debido a problemas de visualización. • El código LaTeX generado no se muestra, o lo hace de manera incorrecta o incompleta. • La interfaz no presenta los resultados correctamente en dispositivos móviles o en diferentes navegadores. 		

Criterios de aceptación:

- La imagen cargada debe ser visible para el usuario inmediatamente después de ser subida.
- El código LaTeX debe mostrarse correctamente en la interfaz después del procesamiento, de forma clara y accesible.

4.4 Requerimientos funcionales: Administración

4.4.1 Requerimiento funcional 1

Identificador: RF4.1	Nombre: Interfaz de administración	
Tipo: Funcional	Requerimiento que lo utiliza: RF4.2	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Especificaciones de seguridad que definan cómo se protegerá la interfaz de administración	
Entrada: <ul style="list-style-type: none"> • Credenciales de administrador (usuario y contraseña u otro método de autenticación). 		Salida: <ul style="list-style-type: none"> • Imagen cargada visible en la interfaz. • Acceso a la interfaz de administración, donde el administrador puede realizar cambios en el sistema.
Precondición: <ul style="list-style-type: none"> • El administrador debe contar con las credenciales adecuadas para acceder a la interfaz de administración. • La autenticación debe haber sido configurada correctamente. 		
Descripción: El sistema debe contar con una interfaz de administración protegida que solo sea accesible para usuarios autorizados, es decir, aquellos que cuenten con las credenciales adecuadas de administrador. Esta interfaz debe permitir al administrador gestionar y modificar configuraciones del sistema, como los parámetros de procesamiento o la estructura de la página web. La autenticación debe ser segura y robusta, utilizando métodos como contraseñas.		
Postcondición: <ul style="list-style-type: none"> • El administrador puede acceder a la interfaz protegida y realizar las modificaciones necesarias. • El sistema está protegido de accesos no autorizados, lo que garantiza la seguridad de las configuraciones y datos. 		
Situaciones anormales:		

<ul style="list-style-type: none"> • Acceso no autorizado debido a vulnerabilidades de seguridad. • Fallos en el proceso de autenticación que impidan al administrador legítimo acceder a la interfaz. • Pérdida de credenciales o mal manejo de la protección de acceso.
Criterios de aceptación: <ul style="list-style-type: none"> • Solo el administrador debe poder acceder a la interfaz de administración con las credenciales correctas. • La autenticación debe funcionar de manera segura, protegiendo el acceso no autorizado. • Durante las pruebas de seguridad, no deben encontrarse vulnerabilidades que permitan accesos indebidos. • El administrador debe poder realizar cambios sin problemas una vez autenticado.

4.4.2 Requerimiento funcional 2

Identificador: RF4.2	Nombre: permisos exclusivos de modificación para el administrador	
Tipo: Funcional	Requerimiento que lo utiliza: RF4.1 y RF4.3	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Especificaciones de seguridad que definan cómo se protegerá la interfaz de administración	
Entrada: <ul style="list-style-type: none"> • Credenciales de administrador (usuario y contraseña u otro método de autenticación). 		Salida: <ul style="list-style-type: none"> • Acceso exclusivo a funciones de edición y configuración del sistema.
Precondición: <ul style="list-style-type: none"> • El administrador debe haber iniciado sesión correctamente a través de la interfaz protegida. • El sistema debe estar configurado para gestionar los roles y permisos adecuados. 		
Descripción: Solo el administrador del sistema debe tener los permisos necesarios para modificar la estructura, configuración y contenido de la página web. Esto incluye la capacidad de realizar cambios en la interfaz, los parámetros de procesamiento y la configuración del servidor. Los usuarios regulares no deben tener acceso a estas funcionalidades, lo que asegura que la plataforma esté protegida contra modificaciones no autorizadas o accidentales. La asignación de permisos debe ser gestionada a través de la interfaz de administración protegida, garantizando que solo las personas con las credenciales correctas puedan realizar cambios.		
Postcondición:		

<ul style="list-style-type: none"> El administrador tiene acceso total a las funciones de modificación del sistema. Los usuarios regulares no pueden acceder a las funciones de modificación.
Situaciones anormales: <ul style="list-style-type: none"> Un usuario no autorizado obtiene acceso a las funciones de modificación. El administrador no puede modificar el sistema debido a un error en la configuración de los permisos. Vulnerabilidades en la protección de permisos permiten cambios no autorizados.
Criterios de aceptación: <ul style="list-style-type: none"> Solo el administrador puede acceder a las funciones de modificación de la página web. Los usuarios regulares no deben poder acceder ni modificar ningún aspecto del sistema.

4.4.3 Requerimiento funcional 3

Identificador: RF4.3	Nombre: Acceso sin registro para usuarios regulares	
Tipo: Funcional	Requerimiento que lo utiliza: RF3.1 y RF3.2	¿Critico? N
Prioridad de desarrollo: Media	Documentos de visualización asociados: Prototipo de interfaz que no incluya funcionalidades de registro o autenticación para usuarios regulares	
Entrada: <ul style="list-style-type: none"> Ninguna, los usuarios no tienen que proporcionar credenciales ni realizar ninguna acción de autenticación. 		Salida: <ul style="list-style-type: none"> Acceso inmediato a las funcionalidades del sistema para cargar y procesar imágenes sin necesidad de iniciar sesión o registrarse.
Precondición: <ul style="list-style-type: none"> Los usuarios regulares deben poder acceder al sistema sin autenticarse ni registrarse. 		
Descripción: El sistema debe estar diseñado de forma que los usuarios regulares puedan utilizarlo sin necesidad de crear cuentas, iniciar sesión o proporcionar cualquier tipo de autenticación. La funcionalidad básica, como cargar imágenes y procesarlas para obtener el código LaTeX, debe estar disponible de manera directa y sin barreras. Esto garantiza que cualquier usuario pueda acceder al		

servicio de forma rápida y sencilla, mejorando la experiencia de usuario y reduciendo la fricción.
Postcondición: <ul style="list-style-type: none"> • Se solicita a un usuario regular que se registre o autentique, lo cual no debería ocurrir. • El sistema rechaza el acceso a usuarios regulares que no han iniciado sesión.
Situaciones anormales: <ul style="list-style-type: none"> • Un usuario no autorizado obtiene acceso a las funciones de modificación. • El administrador no puede modificar el sistema debido a un error en la configuración de los permisos. • Vulnerabilidades en la protección de permisos permiten cambios no autorizados.
Criterios de aceptación: <ul style="list-style-type: none"> • Los usuarios regulares deben poder acceder a todas las funcionalidades sin necesidad de registro o autenticación.

4.5 Requerimientos no funcionales

4.5.1 Requerimiento no funcional 1

Identificador: RNF5.1	Nombre: Interfaz de usuario intuitiva y sin necesidad de capacitación	
Tipo: Funcional	Requerimiento que lo utiliza: RF3.1, RF3.2 y RF3.3	¿Crítico? No
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Prototipos de la interfaz de usuario que muestren el flujo de navegación.	
Entrada: <ul style="list-style-type: none"> • Interacción directa del usuario con la interfaz (clics, toques, navegación). 		Salida: <ul style="list-style-type: none"> • Uso exitoso de las funcionalidades del sistema sin necesidad de instrucciones previas.
Precondición: <ul style="list-style-type: none"> • La interfaz debe estar completamente diseñada y disponible para la interacción del usuario. • Todos los elementos funcionales (botones, menús, etc.) deben estar implementados de manera clara y accesible. 		
Descripción: La interfaz de usuario debe estar diseñada de manera que cualquier persona, sin entrenamiento ni conocimiento previo del sistema, pueda utilizarla de forma efectiva. Todos los elementos interactivos deben ser claramente visibles y		

comprensibles, con etiquetas o íconos que permitan al usuario identificar rápidamente su funcionalidad. La navegación debe ser simple y directa, guiando al usuario hacia las principales funciones del sistema, como cargar imágenes y procesarlas, sin confusión.

Postcondición:

- El usuario debe haber podido realizar tareas básicas, como cargar y procesar una imagen, sin necesidad de asistencia o capacitación adicional.
- El sistema debe haber proporcionado una experiencia clara y fluida, sin causar errores o confusión.

Situaciones anormales:

- El usuario no entiende cómo interactuar con la interfaz debido a un diseño poco claro.
- El usuario necesita ayuda o instrucciones adicionales para usar la interfaz correctamente.
- Elementos importantes de la interfaz no son visibles o no están correctamente etiquetados.

Criterios de aceptación:

- Los usuarios deben poder interactuar con la interfaz y realizar las tareas principales sin recibir instrucción o capacitación.
- Las pruebas de usabilidad deben demostrar que los usuarios pueden navegar por el sistema de manera intuitiva.
- Los elementos interactivos deben estar claramente visibles y etiquetados.
- Los usuarios no deben encontrar dificultades significativas al usar el sistema por primera vez.

5 Funcionamiento general

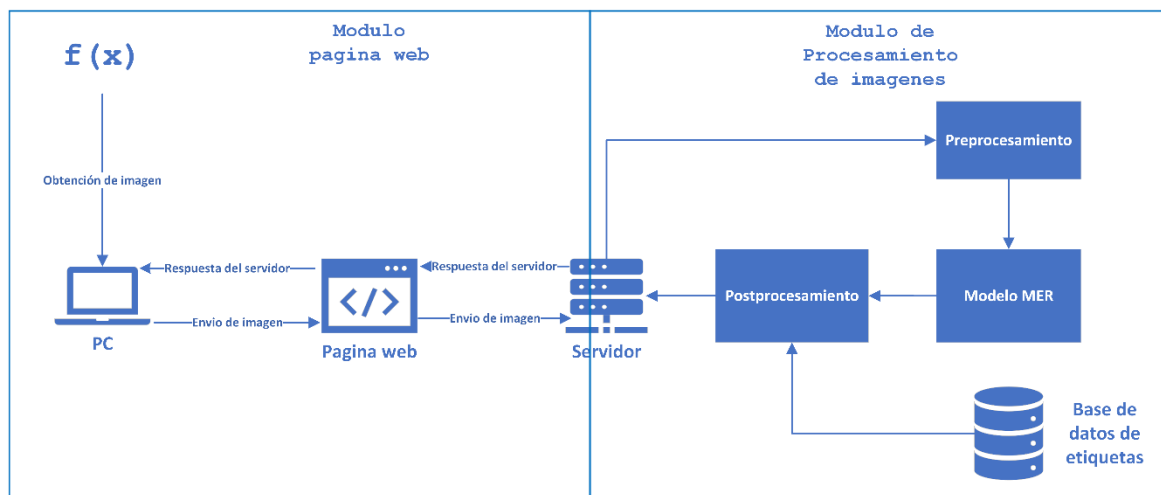


Ilustración 1. Diagrama del sistema propuesto

La propuesta solución consiste en la implementación del sistema, en el cual, el usuario proporcionará imágenes digitales que contengan expresiones matemáticas de cálculo diferencial e integral con tipografía de escritura, esto con el objetivo de poder detectar la estructura de la expresión e identificar los caracteres que contiene la imagen y posteriormente retornar al usuario la expresión matemática en formato LaTeX. El sistema tendrá una arquitectura tipo cliente-servidor, en la parte del cliente el usuario visualizará una página web donde subirá una imagen con una expresión matemática al servidor y a continuación el servidor será el encargado de realizar el procesamiento digital de la imagen para poder quitar el ruido de la imagen, quitar distorsiones y binarizar la imagen, posteriormente el algoritmo MER, realizado con redes neuronales, será el encargado de realizar la detección de la expresión matemática y el servidor responderá al cliente, con el retorno del código en formato LaTeX. A continuación, en la Ilustración 1 se muestra la propuesta de dicho sistema, consiste en los elementos que componen la arquitectura general del proyecto.

5.1 Módulo de página web

5.1.1 Página web

La página web propuesta será la encargada de interactuar con el usuario final, el usuario tendrá la capacidad de subir a la plataforma imágenes en formato JPG, PNG y BMP debido a que son algunos de los formatos más comunes que contengan expresiones matemáticas de acuerdo a los alcances, la página web será la encargada de mostrar la respuesta del servidor después de que la imagen sea procesada, el lenguaje de programación que se usará para desarrollar la página web se definirá en la materia de Proyecto Terminal I.

5.1.2 Obtención de imagen

La imagen que sea ingresada en la página web deberá ser de alguno de los siguientes formatos JPG, PNG y/o BMP debido a que son los formatos más usados en las cámaras digitales o de dispositivos móviles, además, estos formatos tienen mayor flexibilidad al ser manipulados debido a que son imágenes rasterizadas.

5.1.3 Servidor

Será el responsable de recibir la imagen que el usuario haya ingresado desde la página web. Cuando el servidor haya recibido la fotografía este será enviado al módulo de procesamiento de imágenes el cual será el encargado de analizar la imagen y posteriormente dar la respuesta. El servidor enviará la respuesta al usuario, el cual será el código de la ecuación en formato LaTeX, por lo tanto, se propone una arquitectura tipo cliente-servidor. La infraestructura del servidor y el lenguaje de programación se definirán en la materia de Proyecto Terminal I.

5.2 Módulo de procesamiento de imágenes

5.2.1 Preprocesamiento de imágenes

Este módulo será el encargado del preprocesamiento de las imágenes recibidas en el servidor, es decir, se encarga de la eliminación de ruido en las imágenes, corrección de texto en la imagen y realce de detalles. Los posibles métodos para utilizar en la eliminación de ruido son eliminación de ruido Gaussiano y/o ruido Sal y Pimienta. Para la corrección de texto en las imágenes se utilizarán algoritmos para corrección de inclinación de texto y corrección de perspectiva. Para el realce de detalles imágenes se utilizarán técnicas de corrección de color, binarización y detección de bordes usando posiblemente operador laplaciano o el algoritmo de Canny.

5.2.2 Modelo MER

Este es el modelo de reconocimiento de expresiones matemáticas el cual consiste un algoritmo de visión artificial basado en aprendizaje profundo. Este algoritmo será el encargado de reconocer automáticamente patrones y características intrínsecas dentro de las imágenes para identificar expresiones matemáticas de imágenes y asociarlas a un código. La técnica y metodología para entrenar este modelo se elegirán en Proyecto terminal I debido a la gran cantidad de enfoque que se tienen en el aprendizaje profundo, pero se pretende elegir entre redes neuronales convolucionales, redes neuronales tipo transformadores y/o modelos de visión-lenguaje (VL).

5.2.3 Base de datos de etiquetas

En esta base de datos se tendrán las transcripciones de código de LaTeX la cual cumplirá dos funciones, la primera función será en el módulo de procesamiento de imágenes, en la etapa de post-procesamiento, el cual será un codificador donde se adecue los caracteres identificados, se utilizará la base de datos para obtener plantillas para las expresiones matemáticas identificadas por el modelo MER, la segunda aplicación de esta base de datos será en el módulo de entrenamiento, su funcionamiento se explicará en el apartado correspondiente.

5.2.4 Post-procesamiento

En la etapa de post-procesamiento se implementará un codificador, cuando el modelo MER identifique la expresión matemáticas y cada uno de los caracteres intrínsecos, se hará uso de la base de la base de datos de etiquetas para obtener la platilla de código de LaTeX correspondiente al tipo de expresión identificada por el modelo MER y se adecuará el código con los caracteres identificados que posteriormente serán enviados por el servidor.

5.3 Módulo de entrenamiento

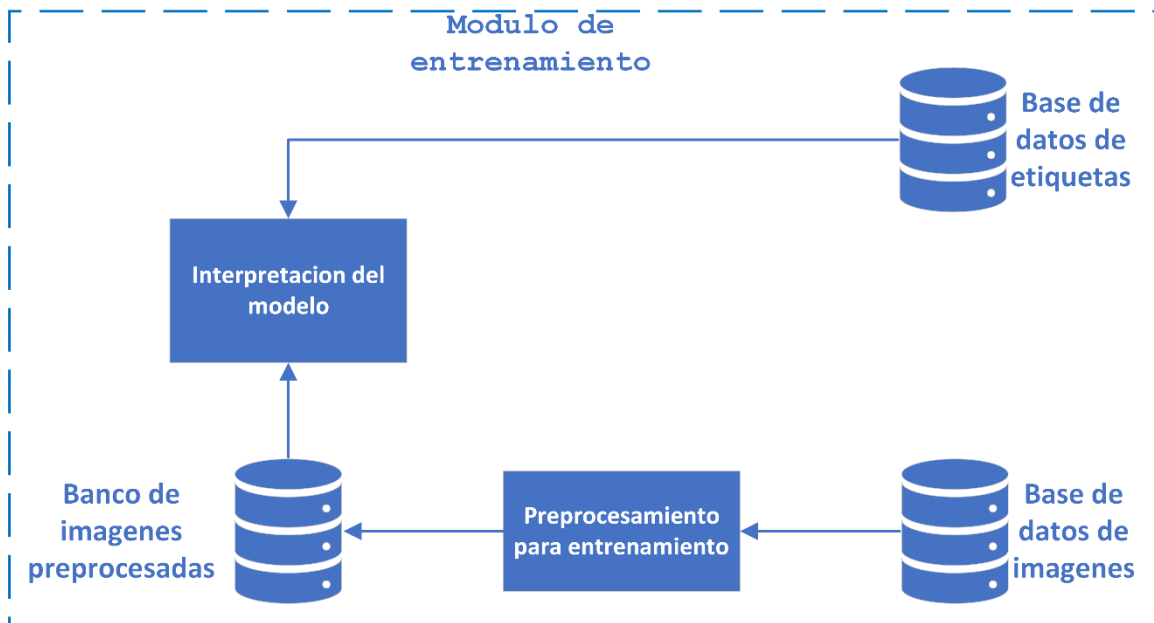


Ilustración 2. Diagrama del módulo de entrenamiento propuesto para el sistema

5.3.1 Dataset de imágenes

Consiste en un dataset que contiene una colección de imágenes con expresiones matemáticas de cálculo diferencial e integral extraídas de algún libro sugerido en la bibliografía de la asignatura, que aún no han sido modificadas o preprocesadas, a este conjunto se le conoce como conjunto de imágenes de entrenamiento.

5.3.2 Banco de imágenes preprocesadas

En este banco de imágenes se tendrán todas las imágenes con condiciones ideales, obtenidas en la etapa de preprocesamiento, para realizar el entrenamiento del algoritmo o modelo MER, es decir, este banco de imágenes será el *dataset* (conjunto organizado de datos que se utiliza para alimentar modelos de aprendizaje) de entrenamiento, la técnica y el lenguaje para el entrenamiento serán definidas en la asignatura Proyecto Terminal I.

5.3.3 Dataset

Est contendrá la transcripción de código en LaTeX de cada una de las expresiones contenidas en el dataset de imágenes, esta base de datos se utilizará para evaluar el entrenamiento del modelo en la etapa de interpretación del modelo y poder realizar ajustes de valores o pesos de la técnica de entrenamiento.

5.3.4 Interpretación del modelo

Este módulo contiene el algoritmo de entrenamiento y la arquitectura del modelo MER, dado que la elección de la técnica y algoritmo del modelo se definen en

Proyecto Terminal I no se pueden dar más especificaciones y solo cabe mencionar que las posibles opciones para el desarrollo del modelo MER son las redes neuronales convolucionales, redes neuronales tipo transformadores y/o modelos Visión-Lenguaje, aunque tampoco se descarta la implementación de alguna otra posible solución.

6 Investigación modelo MER

6.1 Red Neuronal Convolucional

Las redes neuronales convolucionales se distinguen de otras redes neuronales por su mejor desempeño con entradas de señal de imagen, voz o audio. Tienen tres tipos principales de capas, que son:

- Capa convolucional
- Capa de agrupamiento
- Capa totalmente conectada (FC)

La capa convolucional es la primera capa de una red convolucional. Si bien a las capas convolucionales las pueden seguir capas convolucionales adicionales o capas agrupadas, la capa totalmente conectada es la capa final (Observe Ilustración). Con cada capa, la CNN aumenta su complejidad, identificando mayores porciones de la imagen. Las primeras capas se enfocan en características simples, como colores y bordes. A medida que los datos de la imagen avanzan a través de las capas de CNN, se comienzan a reconocer elementos o formas más grandes del objeto, hasta que finalmente se identifica el objeto previsto.

Capa convolucional

La capa convolucional es el pilar central de una CNN, y es donde ocurre la mayor parte del cálculo. Requiere algunos componentes, que son datos de entrada, un filtro y un mapa de características. Supongamos que la entrada será una imagen en color, que se compone de una matriz de píxeles en 3D. Esto significa que la entrada tendrá tres dimensiones (altura, ancho y profundidad) que corresponden a RGB en una imagen. También hay un detector de características, conocido como kernel o filtro, que se moverá a través de los campos receptivos de la imagen, verificando si la característica está presente. Este proceso se conoce como convolución.

Capa de agrupamiento

La agrupación de capas, también conocida como reducción de muestreo, reduce la dimensionalidad y, por lo tanto, el número de parámetros en la entrada. De manera similar a la capa convolucional, la operación de agrupación muestra un filtro en toda la entrada, pero la diferencia es que este filtro no tiene pesos. En cambio, el kernel aplica una función de agregación a los valores dentro del campo receptivo, poblando la matriz de salida. Hay dos tipos principales de agrupación:

- **Agrupación máxima:** a medida que el filtro se mueve a través de la entrada, selecciona el píxel con el valor máximo para enviar a la matriz de salida. Como dato informativo, este enfoque tiende a usarse con más frecuencia en comparación con la agrupación promedio.
- **Agrupación promedio:** a medida que el filtro se mueve a través de la entrada, calcula el valor promedio dentro del campo receptivo para enviarlo a la matriz de salida.

Capa totalmente conectada

El nombre de la capa totalmente conectada es claro. Como se mencionó antes, los valores de píxeles de la imagen de entrada no se conectan de manera directa a la capa de salida en capas parcialmente conectadas. Sin embargo, en la capa totalmente conectada, cada nodo de la capa de salida establece una conexión directa a un nodo en la capa anterior. Esta capa realiza la tarea de clasificación basada en las características extraídas a través de las capas anteriores y sus diferentes filtros. Mientras que las capas convolucionales y agrupadas tienden a usar funciones ReLu, las capas FC suelen aprovechar una función de activación softmax para clasificar las entradas adecuadamente, produciendo una probabilidad de 0 a 1 [15].

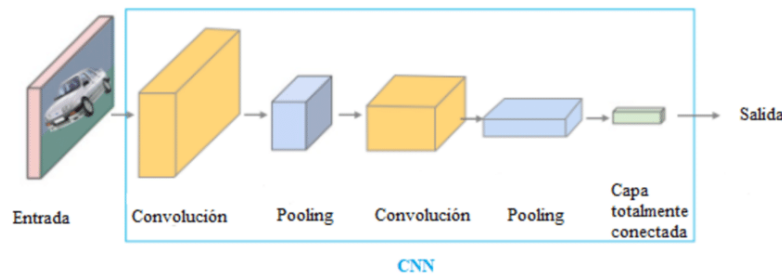


Ilustración 3. Descripción del funcionamiento de una red neuronal convolucional

6.2 Modelo Transformer

Un modelo transformer es una red neuronal que aprende contexto y, por lo tanto, significado mediante el seguimiento de relaciones en datos secuenciales como las palabras de esta oración. Los modelos transformer aplican un conjunto en evolución de técnicas matemáticas, llamadas atención o atención propia, para detectar formas sutiles en que los elementos de datos en una serie se influyen y dependen entre sí. Los transformers se describieron por primera vez en un documento de 2017 de Google. Estos transformers son una de las clases más nuevas y potentes de modelos inventados hasta la fecha. La arquitectura de la red neuronal del transformador tiene varias capas de software que trabajan juntas para generar el resultado final. La siguiente ilustración muestra los componentes de la arquitectura de transformación [16].

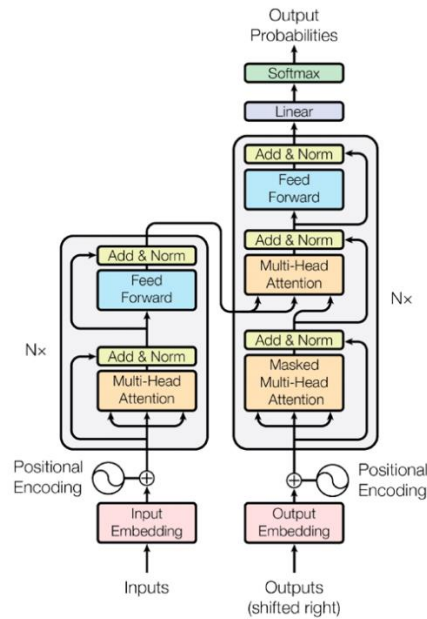


Ilustración 4. Arquitectura de un modelo Transformer

Incrustaciones de entrada

Esta etapa transforma la secuencia de entrada en el formato matemático comprensible para los algoritmos de software. Primero, la secuencia de entrada se divide en una serie de tokens o componentes de secuencia individuales, como palabras en el caso de una oración. Luego, las incrustaciones convierten esta secuencia de tokens en una secuencia vectorial matemática, donde los vectores representan información semántica y sintáctica mediante números aprendidos durante el entrenamiento. Estos vectores pueden imaginarse como coordenadas en un espacio n-dimensional, donde cada palabra se representa por un vector que indica su relación con otras palabras.

Codificación posicional

La codificación posicional es esencial en la arquitectura del transformador para mantener el orden de los tokens en la secuencia de entrada. Agrega información sobre la posición de cada token en la secuencia a sus incrustaciones, asegurando que el modelo comprenda el contexto secuencial.

Bloque transformador

Un modelo de transformador consta de varios bloques de transformadores apilados. Cada bloque tiene dos componentes principales: un mecanismo de autoatención, que pondera la importancia de los tokens dentro de la secuencia, y una red neuronal de retroalimentación por posición. Esto permite al modelo centrarse en partes relevantes de la entrada para realizar predicciones precisas.

Bloques lineales y softmax

Para hacer predicciones específicas, como la próxima palabra en una secuencia, el modelo utiliza un bloque lineal, una capa totalmente conectada que convierte las representaciones internas complejas en puntuaciones para cada token posible. Luego, la función softmax normaliza estas puntuaciones en una distribución de probabilidad, indicando la confianza del modelo en cada posible clase o token [16].