Instituto Politécnico Nacional

ESCUELA SUPERIOR DE CÓMPUTO

Trabajo Terminal

2019-A009

Ingeniería en Sistemas Computacionales

PROTOTIPO DE SISTEMA PARA RECONOCER TEXTO EN IMÁGENES Y TRADUCIRLO A LATEX

PRESENTAN:

Carlos Tonatihu Barrera Pérez
Juan Carlos Garcia Medina
Ian Mendoza Jaimes

DIRECTORES:
Dr. Jorge Cortés Galicia

Ciudad de México, 6 de noviembre de 2019

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Análisis del sistema

1.1. Requerimientos funcionales

Los requerimientos funcionales describen los actividades y comportamientos que tendrá el sistema bajo ciertas condiciones, de igual forma pueden declarar lo que el sistema no debe de hacer.

En esta sección se presentan los requerimientos funcionales que se obtuvieron para el sistema. Dichos requerimientos se encuentran separados de acuerdo a los diferentes módulos que se tienen planeados.

1.1.1. Módulo de usuarios

RF1. Mecanismo de gestión de usuarios. Proporcionar al usuario la posibilidad de creación, consulta y modificación de los datos de su cuenta de usuario. Estas operaciones debe de estar presentes en la aplicación web y de android y solo serán permitidas para usuarios con cuenta verificada.

RF2. Mecanismo de autenticación de usuarios. Proporcionar un mecanismo para el inicio y cierre de sesión de la cuenta del usuario cuya cuenta haya sido verificada en la aplicación web y android.

RF3. Mecanismo de verificación de cuenta. Proporcionar al usuario una forma para verificar que una cuenta creada es valida a través de una verificación basada en el envió de un correo electrónico para la confirmar dicha validez y hacer uso del resto de funcionalidad del sistema. Esta verificación solo se podra realizar a través de la aplicación web.

RF4. Mecanismo de recuperación de contraseñas. Proporcionar al usuario la posibilidad de recuperar la contraseña asociada a su cuenta a través de un envío de correo electrónico con el cual podrá acceder a una interfaz en la aplicación web para recuperar el acceso a su cuenta. La recuperación de contraseñas estará disponible en la aplicación android y web.

RF5. Mecanismo para la comunicación entre la aplicación android y web. El sistema debe de contar con una interfaz para la comunicación entre las dos aplicaciones que se tienen, la forma en que se realizara sera un API REST y el formato para el envió de información sera JSON.

RF6. Mecanismo para la autenticación en el API REST. El sistema debe de brindar una forma de garantizar que la comunicación entre la aplicación web y android es seguro mediante el uso de tokens de autenticación en cada petición y respuesta que se realice.

1.1.2. Módulo de proyectos

RF5. Mecanismo para la gestión de proyectos de LaTeX. Proporcionar al usuario un mecanismo para visualizar, editar, crear o borrar proyectos asociados a su cuenta. La gestión se podrá realizar en la aplicación web y android.

RF6. Permitir descargar el archivo de LaTeX que se haya traducido. Proporcionar al usuario la funcionalidad de generar un archivo LaTeX que contenga la traducción de expresiones matemáticas que se encuentren en un proyecto. La descarga de dicho archivo solo estará disponible en la aplicación web.

RF7. Permitir calificar una traducción realizada. Proporcionar al usuario la funcionalidad de calificar que tan buena fue la traducción de una expresión matemática en una imagen para brindar retroalimentación, dicha calificación podrán ser valores enteros entre 1 y 5. Para poder calificar una traducción se tendrá que hacer uso de la aplicación web.

RF8. Permitir el uso de la cámara del dispositivo android para tomar fotografías. Proporcionar un mecanismo en la aplicación web que permita acceder a la cámara del dispositivo, tomar una fotografía, visualizarla.

RF9. Permitir la visualización del resultado de la traducción. Proporcionar al usuario una interfaz en la cual pueda observar la traducción a LaTeX que se realizó a partir de una imagen. Esto se podrá hacer en la aplicación web y android.

RF10. Permitir añadir al portapapeles alguna traducción seleccionada. Proporcionar al usuario la funcionalidad para copiar el código de una traducción a su portapapeles para su posterior uso. Esta funcionalidad está limitada a solo la aplicación web.

RF11. Mecanismo para el envío de imágenes tomadas por la aplicación android a la aplicación web para su uso. El sistema debe de tener un mecanismo que a través del uso del API REST permita a la aplicación android el enviar una imagen a la aplicación web para que esta sea tratada.

1.1.3. Módulo de análisis

RF12. Mecanismo para el tratamiento de la imagen previo a su análisis. El sistema deberá de realizar un tratamiento al imagen que reciba de la aplicación android para hacer que esta sea más sencilla de trabajar destacando sus partes importantes.

RF13. Mecanismo para el reconocimiento de un conjunto definido de expresiones matemáticas en imágenes. El sistema deberá de reconocer ciertas expresiones en particular con la posibilidad de que expresiones que no se encuentren en este conjunto produzcan resultados no esperados.

1.1.4. Módulo de traducción

RF14. Mecanismo para la traducción a LaTeX de las expresiones matemáticas encontradas en el modulo de análisis. Implementar un algoritmo basado de gramáticas, redes neuronales o arboles de recubrimiento mínimo que permita transformar la salida que proporcione el módulo de análisis a código LaTeXque pueda ser interpretado por un compilador.

1.2. Requerimientos No Funcionales

A continuación se enlistan los requerimientos no funcionales.

RNF1. Usabilidad. La interfaz de usuario debe de ser intuitiva con el objetivo de hacer uso de las funcionalidades del sistema de una forma fácil para el usuario.

RNF2. Seguridad. La seguridad se solventa al realizar un cifrado de las contraseñas de los usuarios así como una verificación de la cuenta del usuario para poder hacer uso del sistema. Por otro lado, en la comunicación que se realiza entre la aplicación móvil y el API REST se utiliza un token para autenticar las peticiones que se realizan, dicho token es único para cada usuario.

RNF3. Escalabilidad. El sistema deberá ser fácilmente escalable y con ello poseer la cualidad de que si se agregan nuevas funcionalidades, estas sean fáciles de acoplar con lo ya desarrollado. Además, de que debe de ser posible aumentar sus capacidades para brindar servicio a más usuarios.

RNF4. Disponibilidad. El sistema debería estar disponible en la mayor parte del tiempo, para lograr esto el uso de un proveedor de hosting es necesario. Entre las posibles opciones están Amazon Web Services, Azure o Google Cloud

1.3. Requerimientos técnicos

A continuación se en listan los requerimientos técnicos para un mejor funcionamiento del sistema:

1.3.1. Aplicación móvil

1.3.1.1. Requerimientos mínimos de software

1. Sistema operativo Android 4.5 o superior.
2. Conexión a internet.

1.3.1.2. Requerimientos mínimos de hardware

1. Memoria RAM: 2 GB.
2. Espacio de almacenamiento de 50 MB.
3. Cámara de al menos 8 Megapixeles.

1.3.2. Aplicación web

1.3.2.1. Requerimientos mínimos de software

1. Cualquiera de las versiones de los siguientes navegadores hasta su versión más reciente.
    - Google Chrome 7
    - Edge 18
    - Internet Explorer 11
    - Firefox 4
    - Safari 5
    - Opera 12.1
    - iOS Safari 13.1
    - Chrome for android 76
    - Firefox para android 68

1.3.2.2. Requerimientos mínimos de hardware

1. Memoria RAM: 2 GB.
2. Espacio de almacenamiento de 100 MB.

1.4. Reglas de Negocio

En esta sección se presentan las reglas de negocio que se necesitan para la elaboración del sistema.

1.4.1. RN-001 Campos obligatorios

Descripción: Aquellos campos que son obligatorios no pueden dejarse vacíos.

Ejemplo: Si se tiene un formulario donde existan los siguientes campos:

- Campo 1
- Campo 2 (obligatorio)
- Campo 3

El usuario puede dejar los campos 1 y 2 vacíos pero el obligatorio no se podrá omitir.

1.4.2. RN-002 Datos correctos

Descripción: Para que los datos sean considerados correctos deben de cumplir con lo establecido en el modelo de información del sistema.

1.4.3. RN-003 Unicidad de identificadores

Descripción: En el conjunto de entidades del sistema, no puede existir elementos con el mismo identificador.

Ejemplo de cumplimiento: Registro de usuario en el cual el correo es el identificador y se puede dar el siguiente caso.

- Usuario1: {nombre=Carlos, correo=carlos_isc7@outlook.com}
- Usuario2: {nombre=Juan, correo=gladwell45@outlook.com"}

Ejemplo de fallo: Registro de usuario en el cual el correo es el identificador y por ende no se puede dar el siguiente caso.

- Usuario1: {nombre=Ian, correo=carlos_isc7@outlook.com}
- Usuario2: {nombre=Juan, correo=carlos_isc7@outlook.com}

1.4.4. RN-004 Calificación proyecto

Descripción: La calificación de un proyecto tendrá un valor entre 1 y 5 resultado del promedio de las traducciones asociadas a dicho proyecto sin decimales que se hará a través de un redondeo.

Sentencia: Sea:

$$C_p = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Donde C_p es la calificación del proyecto, x_i es la calificación de la i traducción y n es el número de traducciones calificadas en el proyecto, deberá actualizarse cada que se agregue o elimine una traducción calificada.

Ejemplo: Si se tiene un proyecto con cinco traducción pero solo tres de ellas tienen calificación se tiene lo siguiente:

- Traducción 1: calificación=4
- Traducción 2: Sin calificación
- Traducción 3: Sin calificación
- Traducción 4: calificación=2
- Traducción 5: calificación=5

La calificación del proyecto sera:

$$C_p = \frac{4+3+5}{3} = 4$$

1.4.5. RN-005 Fecha de modificación

Descripción: La fecha de modificación se actualizará cada que se agregue o modifique una traducción al proyecto o que se modifique el proyecto en si la fecha se actualizará con el valor al momento en que se haga la modificación.

Ejemplo. Si se tiene un proyecto con la fecha 2019-05-03 y se realiza una modificación el día 2019-05-20 ahora la fecha de modificación del proyecto será 2019-05-20.

1.4.6. RN-006 Usuario verificado

Descripción: Para que un usuario pueda acceder al sistema su cuenta debe de estar verificada.

1.4.7. RN-007 Información necesaria para descargar un proyecto

Descripción: Para poder descargar un proyecto es necesario que este cuente con un nombre y al menos una traducción realizada.

1.5. Análisis de Riesgos

Identificar, clasificar y analizar los riesgos potenciales para el presente Trabajo Terminal, nos permite prevenir las posibles amenazas y probables eventos no deseados así como los daños y consecuencias que estos puedan traer al proyecto. La Tabla 1.1 muestra los resultados de nuestro análisis.

Área de impacto	**Nivel de impacto**	**Causas**	**Métodos para contrarrestar el riesgo**
Funcionalidad de la aplicación.	Alto	▪ No obtener un conjunto de entrenamiento lo suficientemente grande para realizar una traducción con pocos errores. ▪ Poca precisión por parte del modelo utilizado.	Investigar sobre posibles nuevos conjuntos de entrenamiento que la comunidad científica libere.

Software	Medio	▪ Cambios drásticos en las herramientas de desarrollo establecidas para el desarrollo del sistema.	Seleccionar tecnologías que sean estables y utilizadas en la industria.
Competitividad de la aplicación	Alto	▪ El lanzamiento de una aplicación similar por parte de una compañía mucho más grande.	Seguir incrementando la precisión así como las capacidades del presente proyecto con el fin de que pueda competir en el mercado.
Confianza del usuario	Alto	▪ Insatisfacción por parte del usuario por factores tales como la poca precisión de la traducción.	Seguir incrementando la precisión. Pedirle retroalimentación directa al usuario final.

Tabla 1.1: Análisis de riesgos.

1.6. Descripción del software

1.6.1. Android

Android es un sistema operativo para móviles que en la actualidad es desarrollado por Google. Esta principalmente pensado para dispositivos con pantalla táctil como smartphones y tablets. Y que en la actualidad junto con iOS son las principales opciones en cuanto a sistemas operativos para teléfonos móviles. Es por esto que en la Tabla 1.2 se hace una comparativa de las principales características que se tomaron en cuenta para la elección de sistema operativo de dispositivos móviles que se usaría en este trabajo.

Características	**Android**	**iOS**
Núcleo	UNIX	UNIX
Lenguajes de desarrollo	Java, C, C++, Kotlin	Swift, C, C++, Objective-C
Ambiente de desarrollo	Android Studio, disponible en Windows, Linux y MacOS	Xcode, solo disponible en MacOS
Complejidad de desarrollo	Medio complejo debido a la gran cantidad de diferencias entre dispositivos y versiones de sistema operativo	Poco compleja debido a que hay poca diversidad de versiones sistemas operativos y dispositivos
Tiempo de desarrollo	Suele tomar 30 %-40 % más que para iOS [1]	Depende de la complejidad del desarrollo
Tiempo de despliegue	Es rápido desplegar una aplicación de Android debido a los test automáticos que se realizan sobre esta para su verificación	Es lento su despliegue debido a que la verificación es manual
Cantidad del mercado	La cantidad de usuarios es mayor, tan solo el primer cuarto del 2017 el 86.1 % de los teléfonos vendidos fueron Android [1]	Su mercado es menor al de Android
Código abierto	El kernel, UI y algunas aplicaciones estándar son de código abierto	El kernel no es de código abierto pero esta basado en Darwin OS que es de código abierto
Ultima versión	Android 10 (Septiembre 3, 2019)	iOS 13 (Septiembre 19, 2019)

Seguridad	Actualizaciones de seguridad bastante regulares. Sin embargo, debido a los fabricantes dichas actualizaciones pueden demorarse en ser aplicadas. También se pueden instalar aplicaciones externas a la Play Store por lo que esto puede generar problemas de seguridad	Pocas actualizaciones, las amenazas de seguridad son pocas debido a que descargar aplicaciones fuera de la App Store es complicado.

Tabla 1.2: Tabla comparativa de sistemas operativos de dispositivos móviles

Android nos brinda un ambiente de desarrollo más flexible que el que presenta iOS debido a las características listadas con anterioridad. Es cierto que el desarrollo puede ser más tardado sin embargo para este proyecto el optar por iOS nos generaría problemas debido a que no solo la plataforma de desarrollo es más cerrada sino que el tiempo de desarrollo seria mayor debido a la curva de aprendizaje de Swift y Objective-C a la cual nos enfrentaríamos.

Finalmente, la cantidad del mercado que tiene Android es significativamente mayor que la que tiene iOS por ende este es un gran punto a considerar. Es por todas estas cuestiones que se opto por desarrollar el trabajo para la plataforma de Android.

1.6.2. Base de datos

Un Sistema Gestor de Bases de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. Para el caso del presente Trabajo Terminal, se decidio utilizar una base de datos relacional. La Tabla 1.3 muestra una comparativa de los tres principales gestores de bases de datos [2].

Características	**MySQL**	**Oracle**	**PostgreSQL**
Modelo de base de datos primario	Relacional	Relacional	Relacional
Modelo de base de datos secundario	Documento	Documento, gráfo, RDF	Documento
Distribución	Código abierto	Comercial	Código abierto
Implementación	C y C++	C y C++	C

Sistemas operativos soportados	FreeBSD, Linux, OS X, Solaris y Windows.	AIX, HP-UX, Linux, OS X, Solaris, Windows y z/OS.	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows.
Soporte de XML	Si	Si	Si
Scripts del lado del servidor	Si	PL/SQL	Funciones definidas por el usuario.
Triggers	Si	Si	Si
Transacciones	ACID	ACID	ACID

Tabla 1.3: Comparación de diferentes gestores de bases de datos.

Para el desarrollo del presente proyecto, se selecciono PostgreSQL como gestor de bases de datos, debido a que es de código abierto y contiene más funcionalidades avanzadas de lo que MySQL soporta.

1.6.3. Framework de desarrollo web

Un framework de desarrollo es un conjunto de utilerías y funciones que permiten de una manera consistente acelerar el proceso de creación de una aplicación web. La Tabla 1.4 muestra una comparativa entre los principales frameworks en la industria [3].

Características	**Django**	**Ruby on Rails**	**Laravel**
Lenguaje soportado	Python	Ruby	PHP
Mercado	1,192 compañías lo utilizan	2,723 compañías lo utilizan	1,032 compañías lo utilizan
Ecosistema	Muchos paquetes disponibles, entre las más importantes se encuentran: Django REST Framework, Django allauth para la autenticación con redes sociales y Celery.	Cientos de gemas disponibles, librerias como Action Mailer y Active Storage, frameworks como Active Job y Action Cable.	Se compone de mas de 15k paquetes. Los más populares son: Cashier, Envoy and Passport, Scout, Socialite, Envoyer, Forge, Horizon, Lumen, entre otros.

Desempeño	Respuestas rápidas de texto plano, actualizaciones de la BD eficientes, excelente en la serialización de JSON.	Respuestas buenas de texto plano, eficiente en las actualizaciones de la DB, eficiente respuesta en JSON.	Respuestas buenas de texto plano, excelente en las actualizaciones de la DB.
Seguridad	Provee métodos contra inyección SQL, ataques XSS, CSRF y *clickjacking*. Excelente en el manejo de autenticación de usuarios y permisos. Actualizaciones constantes.	Provee métodos contra inyección SQL, ataques XSS, CSRF y *clickjacking*.	Es vulnerable a ataques, provee métodos de autenticación y hashing.

Tabla 1.4: Comparación de frameworks de desarrollo web.

Para el desarrollo de la parte web del presente proyecto, se eligió Django como framework de desarrollo debido a que esta enteramente desarrollado en python, lo que nos permite el uso de las mejores librerias de deep learning, las cuales solo estan soportadas en este lenguaje. Además, Django provee de un conjunto muy robusto de métodos para mantener segura la información.

1.7. Delimitación de expresiones matemáticas

Debido a la elección de citarMetodoAtacarProblema, los símbolos y tipo de expresiones matemáticas a reconocer están limitadas por el conjunto de entrenamiento a utilizar:

Es por ello que al utilizar el conjunto de entrenamiento CROHME se realizó un análisis de símbolos existentes en dicho conjunto con ayuda de la herramienta **CROHME Data Extractor** al realizar un conteo de apariciones de cada uno de los símbolos para así generar gráficas de frecuencias clasificando a los tipos de símbolos:

La primer gráfica muestra la frecuencia de los dígitos del sistema arábigo, donde podemos ver que el dígito **1** aparece nueve mil cuatrocientos sesenta veces siendo este el de mayor aparición en esta categoría, mientras que el 7 aparece mil veinticinco veces.

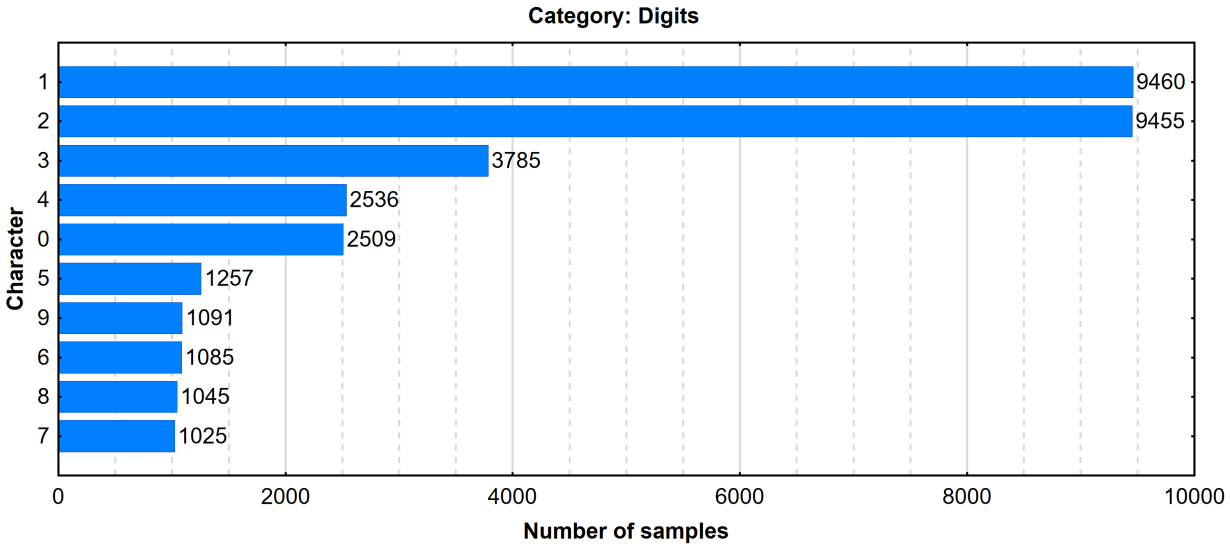

Figura 1.1: Frecuencia de aparición de dígitos del sistema de numeración arábigo

En el caso de los símbolos matemáticos el de mayor aparición es el símbolo **-** con doce mil trecientas veintiocho veces. En esta gráfica podemos observar la aparición de símbolos que soportan expresiones que involucren divisiones o funciones trigonométricas o comparadores, incluso cuantificador de existencia aunque con una muy baja frecuencia de tan solo once apariciones.

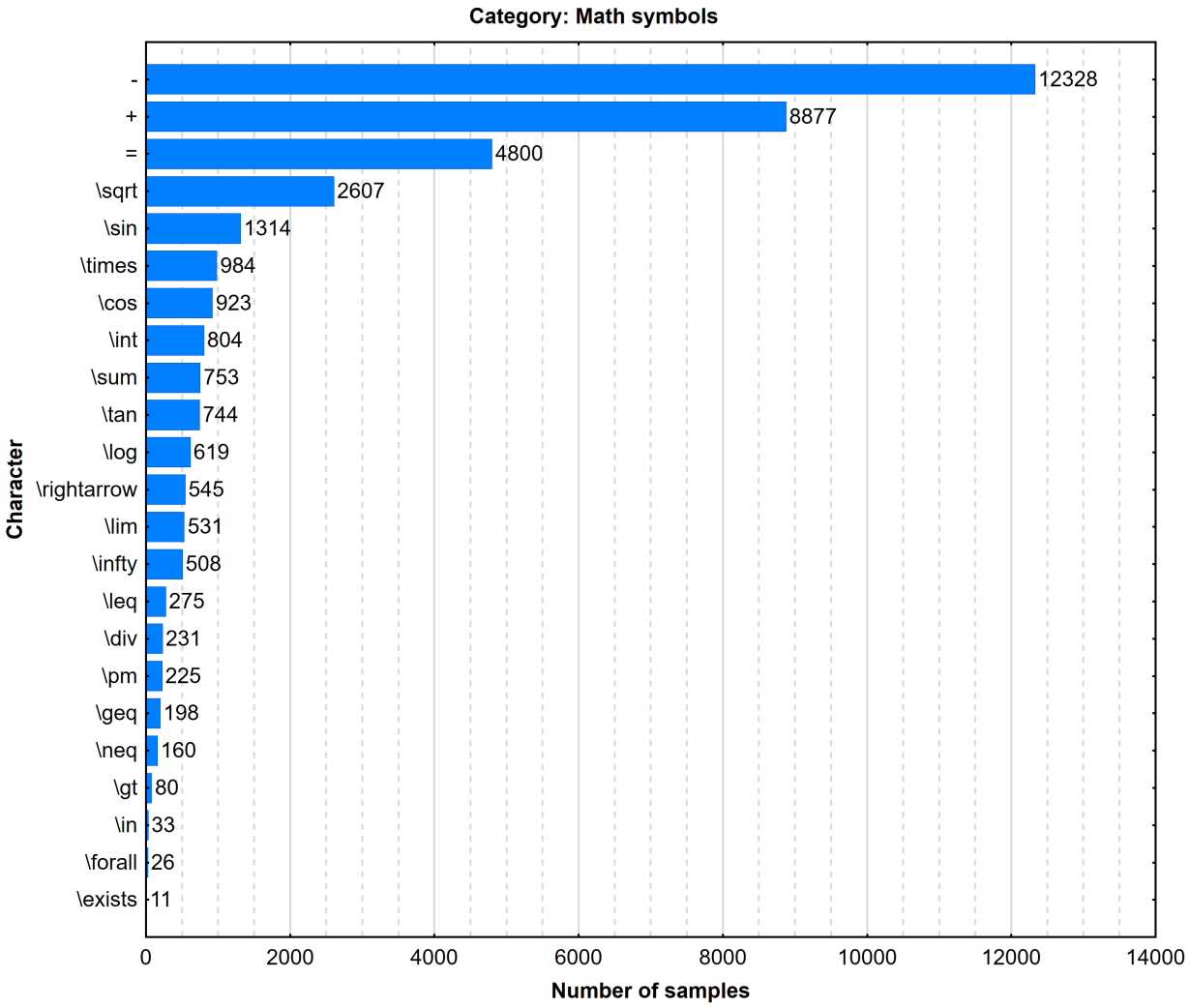

Figura 1.2: Frecuencia de aparición de símbolos matemáticos

En el caso de letras mayúsculas se logró identificar una frecuencia de cuatrocientos sesenta y nueve para el caso de la letra **C** mientras que para la letra **I** de noventa y seis.

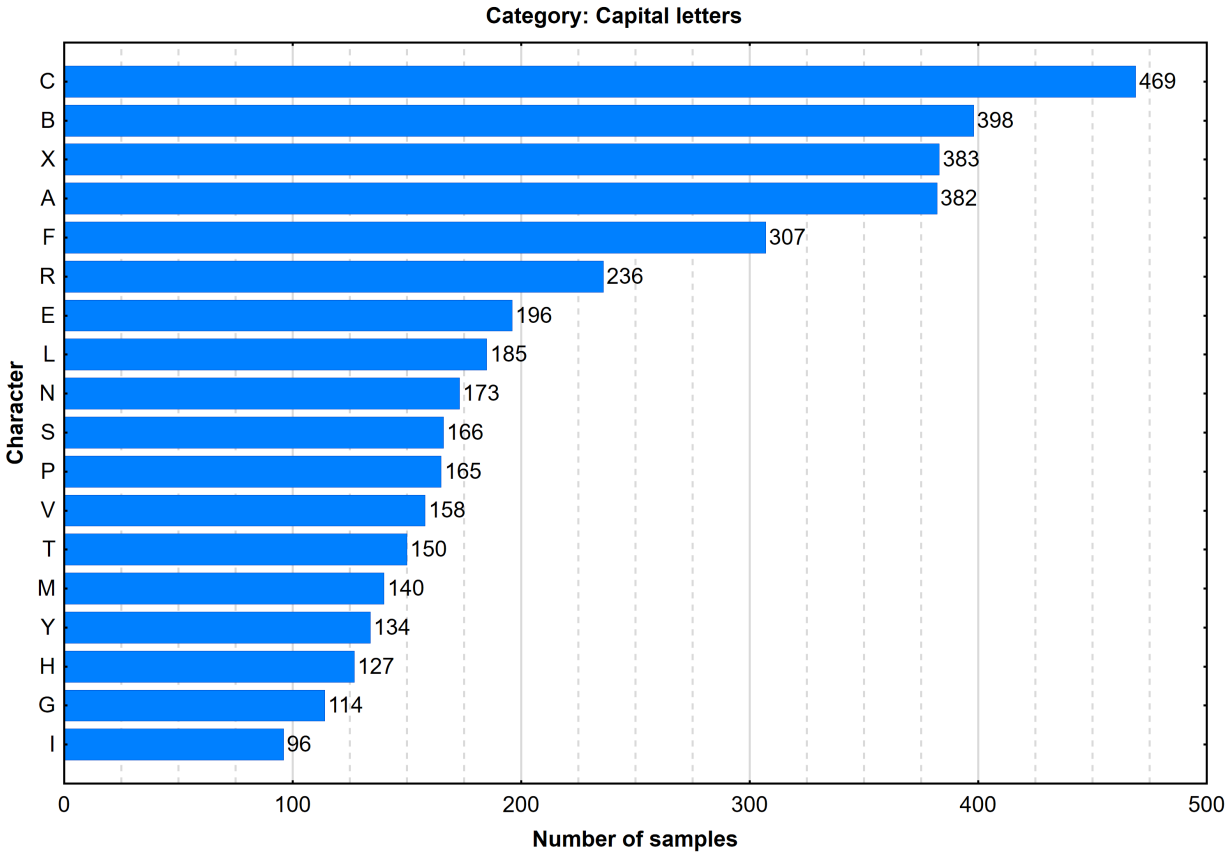

Figura 1.3: Frecuencia de aparición de letras mayúsculas

En el caso de las letras minúsculas como podría esperarse la letra **x** tiene la mayor aparición con ocho mil ciento nueve veces.

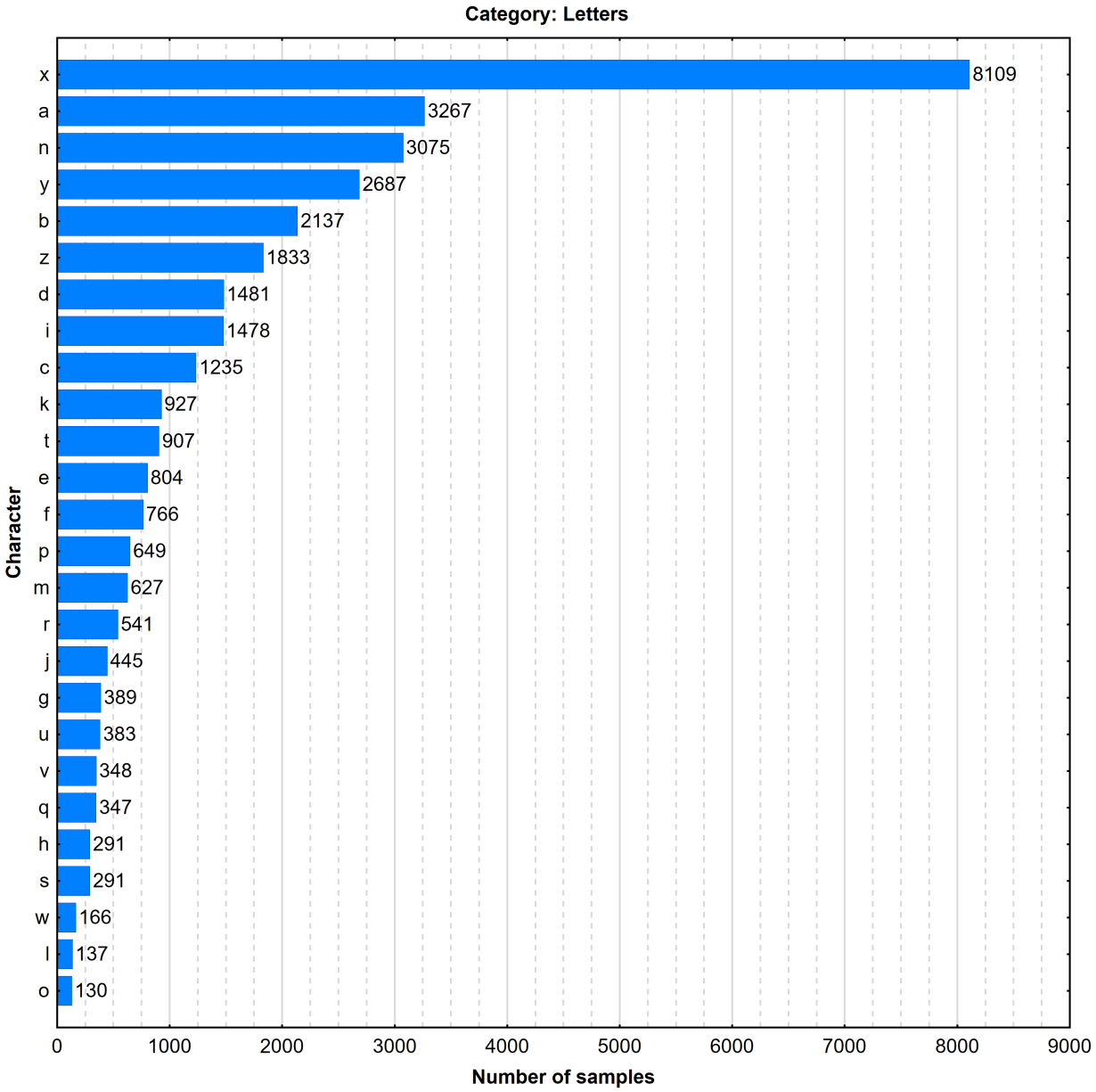

Figura 1.4: Frecuencia de aparición de letras minúsculas

La categoría con menos elementos pertenece a las letras griegas con nueve caracteres, siendo α (alpha) el de mayor aparición con ochocientas diecinueve veces.

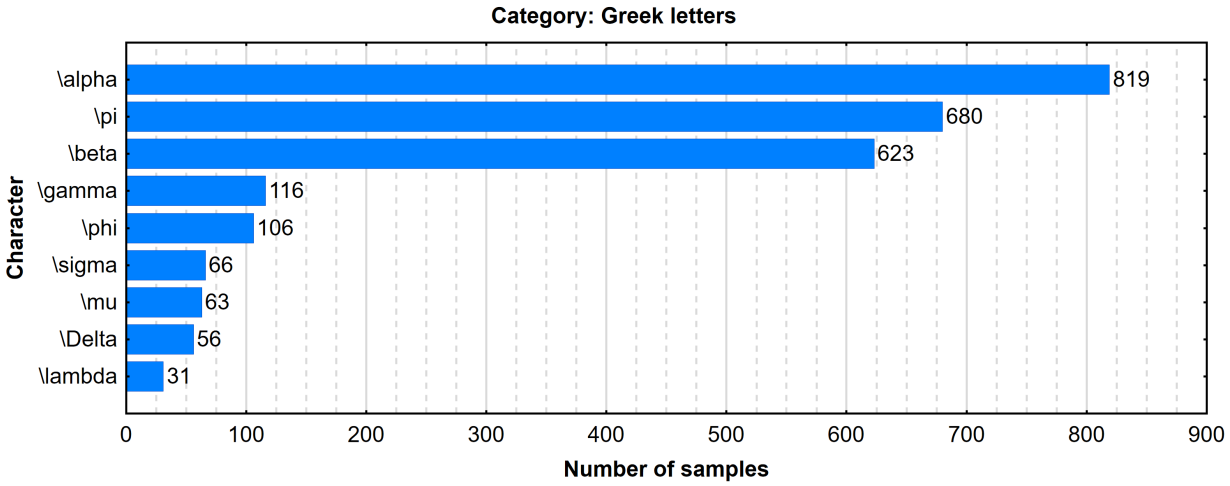

Figura 1.5: Frecuencia de aparición de letras griegas

Otros caracteres que aparecen en el conjunto de entrenamiento se muestran en la figura 1.6 que son también de bastante utilidad como paréntesis, llaves, corchetes, el punto y la coma, que permiten definir jerarquía en las operaciones o números de punto flotante por mencionar dos aplicaciones que muestran la utilidad de los mismos.

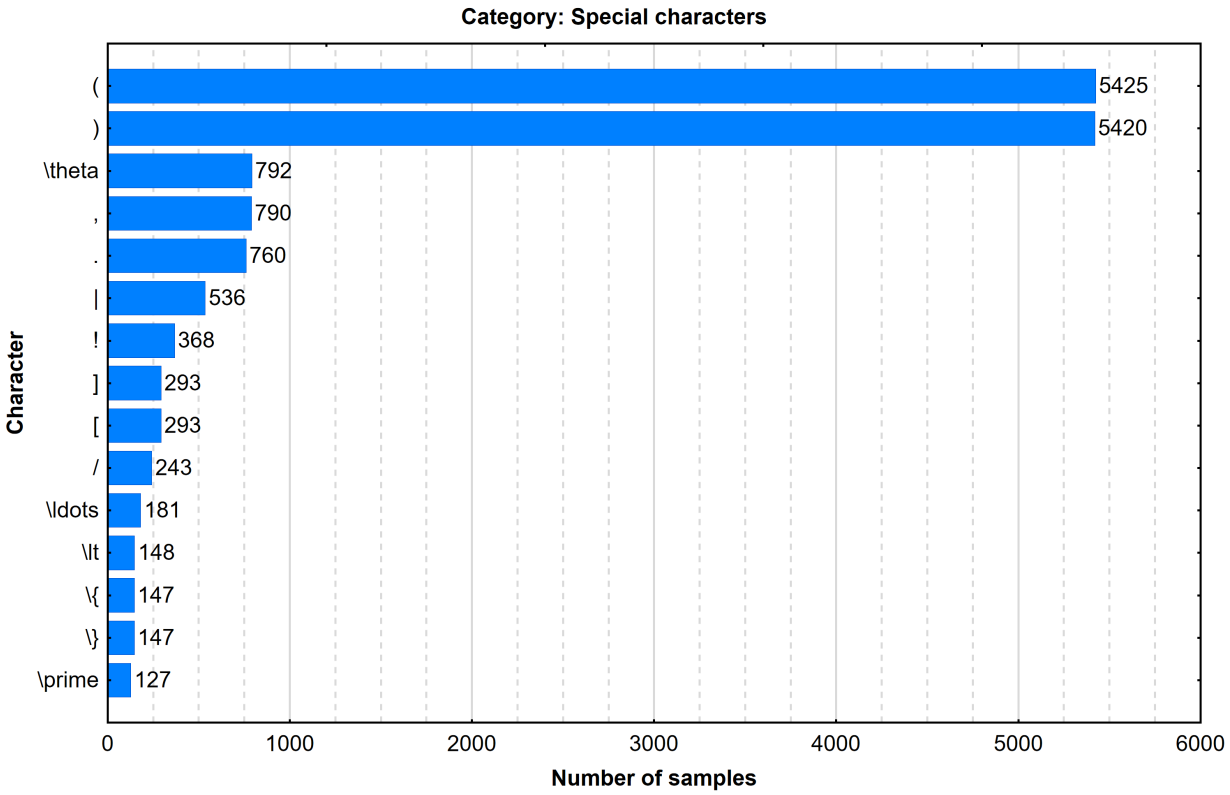

Figura 1.6: Frecuencia de aparición de símbolos especiales

Con estos histogramas se puede determinar qué símbolos se espera puedan ser reconocidos, así como aquellos de los cuales se esperan mejores resultados en función de su frecuencia de aparición en el conjunto de entrenamiento.

1.7.1. Nivel dimensional

Sabiendo que el conjunto de entrenamiento CROHME no es lo suficientemente grande siendo esta una de las limitantes para obtener buenos resultados y teniendo en cuenta la bidimensionalidad que se presenta en las expresiones matemáticas, se propone el reconocimiento dos niveles en ambas direcciones, teniendo esta restricción con un mayor peso en dirección perpendicular al sentido de la escritura de las expresiones, esto basado en los resultados obtenidos por con el mismo conjunto de entrenamiento.

1.8. Traducción de imágenes a LaTeX

El problema de traducir imágenes a LaTeXautomáticamente, es aún un campo de investigación abierto. Dada su naturaleza recursiva miltinivel, las inmensas ambigüedades en la escritura y su fuerte dependencia a un contexto claro, esta tarea difiere en gran medida del reconocimiento óptico de caracteres (OCR) para el cual ya existen sistemas con una precisión suficiente.

Según [4], el reconocimiento de expresiones matemáticas a mano (HMER) comprende dos grandes problemas: reconocimiento de los símbolos y el análisis de su estructura. Existen dos grandes aproximaciones, las cuales son: secuenciales y globales.

Las aproximaciones secuenciales, buscan primero reconocer los símbolos y después realizán el análisis estructural por separado. Uno de los problemas que esta aproximación presenta, es que los errores cometidos en la parte del reconocimiento, serán heredados por el análisis estructural, generando una cadena de malas traducciones. La solución secuencial que mejores resultados ha dado, es el uso de grámaticas predefinidas para llevar a cabo el reconocimiento de la estructura de la expresión.

La segunda alternativa, combina los dos tipos de análisis en uno solo, pues busca ir reconociendo la estructura a la vez que se reconocen los símbolos. Esta aproximación permite generar la segmentación de símbolos utilizando toda la información disponible de la imagen, por lo que este tipo de soluciones parecen ser una mejor opción, sin embargo, este procedimiento es más costoso computacionalmente. Recientes investigaciones, han utilizado las arquitecturas encoder-decoder mencionadas previamente en el Marco Teórico para atacar el problema de forma global como si fuera una tarea de Image Captioning. Los resultados obtenidos por este tipo de aproximación resultan ser menos costos y más precisos.

Las dos propuestas principales, es decir, las gramáticas y la arquitectura encoder-decoder tienen sus limitaciones. En el caso de las gramáticas, podemos ver que los algoritmos para el procesamiento de las mismas incrementan su complejidad exponencialmente entre más expresiones son capaces de reconocer, además de que requieren un conocimiento previo de los símbolos. Mientras que con el avance de los procesadores y GPUs las aproximaciones globales con redes neuronales se vuelven cada vez más plausibles y es por esta razón que los investigadores en el campo del machine learning le han prestado particular atención mejorando así, los resultados obtenidos por cualquier otro método.

En el presente Trabajo Terminal, se ha decidido utilizar la aproximación global propuesta por [4], la cual es llamada Watch, Attend, Parse (WAP) que es una RNN encoder-decoder con un sistema de atención que permite reconocer expresiones matemáticas escritas a mano con una precisión del 46.55 %. En la Figura 1.7 podemos ver un diagrama de esta arquitectura.

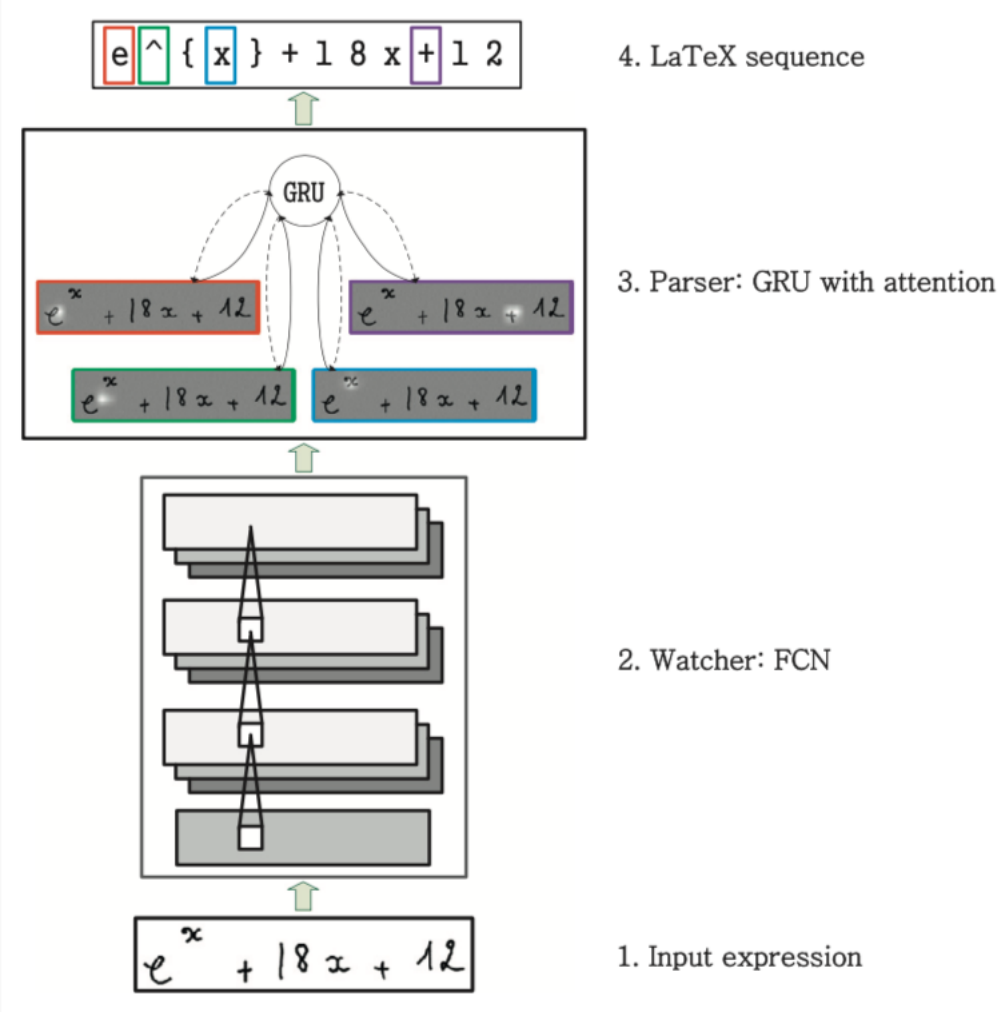

Figura 1.7: Diagrama de la arquitectura Watch, Attend, Parse (WAP).

La red WAP, se compone de dos partes:

1. **Encoder**: El cual es una CNN que permite extraer las características de la imagen y sumarizarlas en un vector de contexto C.

2. **Decoder**: El cual es una LSTM que toma el contexto C y junto con sistema de atención retorna como salida la secuencia de símbolos de LaTeX, un símbolo a la vez.

Bibliografía

[1] D. Development, "10 Major Differences Between Android and iOS App Development." http://ddi-dev.com/blog/programming/10-differences-between-android-and-ios-app-development/, 2018. [Consultado: 2019-09-21]. 9

[2] "System Properties Comparison MySQL vs. Oracle vs. PostgreSQL." https://db-engines.com/en/system/MySQL%3BOracle%3BPostgreSQL, 2019. 10

[3] A. Nesmiyanova, "Ruby on Rails vs Django vs Laravel: The Ultimate Comparison of Popular Web Frameworks." https://steelkiwi.com/blog/ruby-django-laravel-frameworks-comparison, 2019. 11

[4] J. Zhanga, J. Dua, S. Zhanga, D. Liub, Y. Hub, J. Hub, S. Weib, and L. Daia, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, p. 196–206, 2017. 19