



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 Tonatiuh Barrera Pérez

Juan Carlos García Medina

Ian Mendoza Jaimes

DIRECTORES:

Dr. Jorge Cortés Galicia



Ciudad de México, 7 de junio de 2020

Índice general

Índice de figuras	VI
Índice de tablas	IX
1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Estado del arte	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Resultados esperados	4
1.5. Metodología	6
2. Marco teórico	8
2.1. Análisis de Imágenes	8
2.1.1. Preprocesamiento	8
2.1.2. Realce de la imagen	9
2.1.2.1. Transformaciones a nivel de grises básicas	10
2.1.2.2. Negativos de Imagen	11
2.1.2.3. Transformaciones Log	12
2.1.2.4. Transformaciones Power-Law	12
2.1.3. Segmentación de la imagen	12
2.2. Deep Learning	14
2.2.1. Gráficos computacionales	14
2.2.2. Batch Normalization	14
2.2.3. Redes Neuronales Convolucionales	16
2.2.3.1. Pooling	17
2.2.4. Redes Neuronales Recurrentes	18
2.2.4.1. Arquitectura Encoder-Decoder	19
2.2.4.2. Long Short-Term Memory	20
2.2.4.3. Modelos de Lenguaje Condicional	21
2.2.4.4. Sistema de Atención	22
2.2.4.5. Positional Embeddings	22

ÍNDICE GENERAL

2.2.5. Aprendizaje profundo como servicio	22
2.3. Métodos de reconocimiento de expresiones matemáticas	23
2.3.1. Análisis sintáctico dirigido	23
2.3.2. Análisis estructural	24
2.3.3. Image Captioning	25
2.4. Aplicación web	25
2.4.1. Django	26
2.4.1.1. Modelo-Vista-Template	26
2.4.2. API REST	26
2.4.2.1. Application Programming Interface	26
2.4.2.2. Represenational State Transfer	27
2.4.2.3. Autentificación	28
2.5. Aplicación Android	28
2.5.1. Arquitectura Clean	28
2.5.2. Componentes	29
2.5.2.1. Entidades	29
2.5.2.2. Casos de uso	30
2.5.2.3. Adaptadores	30
2.5.2.4. Frameworks y controladores	30
2.5.3. SOLID	30
3. Análisis del sistema	32
3.1. Requerimientos funcionales	32
3.1.1. Módulo de usuarios	32
3.1.2. Módulo de proyectos	33
3.1.3. Módulo de análisis	34
3.1.4. Módulo de traducción	34
3.2. Requerimientos No Funcionales	35
3.3. Requerimientos técnicos	37
3.3.1. Aplicación móvil	37
3.3.1.1. Requerimientos mínimos de software	37
3.3.1.2. Requerimientos mínimos de hardware	37
3.3.2. Aplicación web	37
3.3.2.1. Requerimientos mínimos de software	37
3.3.2.2. Requerimientos mínimos de hardware	38
3.4. Reglas de Negocio	38
3.4.1. RN-001 Campos obligatorios	38
3.4.2. RN-002 Datos correctos	38
3.4.3. RN-003 Unicidad de identificadores	38
3.4.4. RN-004 Calificación proyecto	39
3.4.5. RN-005 Fecha de modificación	39
3.4.6. RN-006 Usuario verificado	40
3.4.7. RN-007 Información necesaria para descargar un proyecto	40

3.5. Análisis de Riesgos	40
3.6. Descripción del software	42
3.6.1. Android	42
3.6.2. Base de datos	43
3.6.3. Framework de desarrollo web	44
3.7. Delimitación de expresiones matemáticas	46
3.7.1. Nivel dimensional	51
3.8. Traducción de imágenes a L ^A T _E X	52
 4. Diseño del sistema	 55
4.1. Arquitectura del sistema	55
4.1.1. Cliente web	55
4.1.2. Cliente android	55
4.1.3. Servidor web	55
4.1.4. Servidor de base de datos	56
4.1.5. Servidor cloud	56
4.2. Modelo de datos del sistema	58
4.2.1. Entidad: Usuario	58
4.2.1.1. Atributos	59
4.2.2. Entidad: Proyecto	59
4.2.2.1. Atributos	59
4.2.3. Entidad: Traducción	60
4.2.3.1. Atributos	60
4.2.4. Entidad: Tipo estudios	61
4.2.4.1. Atributos	61
4.2.5. Entidad: Usuario Token	61
4.2.5.1. Atributos	61
4.3. Aplicación de Android	63
4.3.1. Módulo usuarios	63
4.3.1.1. Diagrama de casos de uso	63
4.3.1.2. A-USR-CU1 Crear cuenta	63
4.3.1.3. A-USR-CU2 Iniciar sesión	66
4.3.1.4. A-USR-CU3 Recuperar contraseña	68
4.3.1.5. A-USR-CU4 Editar perfil	70
4.3.1.6. A-USR-CU5 Cerrar Sesión	72
4.3.2. Módulo proyectos	73
4.3.2.1. Diagrama de casos de uso	73
4.3.2.2. A-PR-CU1 Gestionar proyectos	73
4.3.2.3. A-PR-CU1.1 Visualizar proyecto	75
4.3.2.4. A-PR-CU1.1.2 Editar proyecto	76
4.3.2.5. A-PR-CU1.1.3 Eliminar proyecto	78
4.3.2.6. A-PR-CU1.1.4 Eliminar traducción	80
4.4. Aplicación Web	82

4.4.1. Módulo usuarios	82
4.4.1.1. Diagrama de casos de uso	82
4.4.1.2. W-USR-CU1 Crear cuenta	82
4.4.1.3. W-USR-CU1.1 Verificar cuenta	85
4.4.1.4. W-USR-CU2 Iniciar sesión	86
4.4.1.5. W-USR-CU3 Recuperar contraseña	88
4.4.1.6. W-USR-CU4 Editar perfil	90
4.4.1.7. W-USR-CU5 Cerrar sesión	92
4.4.2. Módulo proyectos	93
4.4.2.1. Diagrama de casos de uso	93
4.4.2.2. W-PR-CU1 Gestión de proyectos	93
4.4.2.3. W-PR-CU1.1 Visualizar Proyecto	95
4.4.2.4. W-PR-CU1.1.1 Editar proyecto	96
4.4.2.5. W-PR-CU1.1.2 Descargar proyecto	98
4.4.2.6. W-PR-CU1.1.3 Eliminar traducción	100
4.4.2.7. W-PR-CU1.1.4 Copiar traducción	102
4.4.2.8. W-PR-CU1.1.5 Eliminar proyecto	103
4.4.2.9. W-PR-CU1.1.6.1 Calificar traducción	105
4.4.2.10. W-PR-CU1.2 Crear proyecto	107
4.5. Mensajes	109
4.6. Interfaces	111
4.6.1. Aplicación en Android	111
4.6.2. Aplicación en Web	118
5. Desarrollo del sistema	121
5.1. Android	121
5.1.1. Arquitectura de la aplicación	121
5.1.1.1. Capa de datos	122
5.1.1.2. Capa de dominio	129
5.1.1.3. Capa de presentación	135
5.2. Web	136
5.2.1. Arquitectura de la aplicación	136
5.2.1.1. Modelo	136
5.2.1.2. Vista	136
5.2.1.3. Template	136
5.3. Conjuntos de entrenamiento	137
5.3.1. CROHME	137
5.3.1.1. Formato del conjunto de datos	137
5.3.1.2. Conversión a imagen	138
5.3.1.3. Generador de secuencia de tokens	141
5.3.2. Harvard 100k	143
5.3.3. Normalización	143
5.4. Desarrollo módulo de análisis de imágenes	145

5.4.1.	Algoritmo de Otsu	146
5.4.2.	Algoritmo de Sauvola	147
5.4.3.	Resultados	147
5.5.	Reconocimiento de expresiones matemáticas	155
5.5.1.	Modelo	155
5.5.1.1.	Encoder	157
5.5.1.2.	Decoder	158
5.5.1.3.	Implementación	160
5.5.1.4.	Entrenamiento	164
5.5.1.5.	Resultados	167
5.5.1.6.	Experimentos Previos	167
6.	Pruebas del sistema	173
6.1.	Pruebas unitarias	173
6.1.1.	Aplicación android	173
6.1.1.1.	Pruebas sobre la clase DateFormater	173
6.1.1.2.	Pruebas sobre la clase RNN002	175
6.1.2.	Aplicación web	179
6.1.2.1.	Pruebas sobre los modelos	180
6.1.2.2.	Pruebas sobre las vistas	181
6.1.2.3.	Pruebas sobre las clases de utilería	182
Bibliografía		185

Índice de figuras

1.1.	Arquitectura del sistema.	6
2.1.	Transformación de nivel de grises para realce de contraste.	10
2.2.	Algunas transformaciones básicas a nivel de grises usadas para realce de imagen.	11
2.3.	Imagen con bloques (Izquierda) y conjunto de segmentos de linea extraídos (Derecha).	13
2.4.	Ejemplo de un perceptrón de una sola capa representado mediante un gráfico computacional, siendo x la entrada, w^1 la matriz de pesos, b^1 la matriz de bias, $u^{(1)}$ y $u^{(2)}$ nodos intermedios en el gráfico y y^1 la salida de la red. La ecuación modelada es $y^1 = f(W^1x + b^1)$	15
2.5.	Matriz de pesos conocida como kernel o filtro aplicada a una imagen.	17
2.6.	Representación gráfica de la operación Max Pooling.	18
2.7.	Ejemplo de una RNN que modela la ecuación $s(t) = f(s^{(t-1)}, \theta)$, donde θ representa los parámetros W y V , f la función no lineal aplicada a $Wx^{(t)}$ y V la matriz de pesos aplicada a $s^{(t-1)}$	19
2.8.	Una compuerta LSTM típica.	20
2.9.	Ejemplo de sistema con módulo de Machine Learning	23
2.10.	a) La primera etapa del método de reconocimiento, se anotan las coordenadas de de cada carácter. b) Se realiza un análisis sintáctico con las gramáticas libres de contexto definidas.	24
2.11.	Utilización de un árbol de recubrimiento mínimo para el reconocimiento de expresiones matemáticas.	24
2.12.	Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.	25
2.13.	Representación del patrón de diseño Modelo-Vista-Template.	27
2.14.	Consumidor y proveedor de servicios comunicándose mediante solicitudes y respuestas REST.	28
2.15.	Arquitectura Clean [1]	29
3.1.	Frecuencia de aparición de dígitos del sistema de numeración arábigo [2]	46
3.2.	Frecuencia de aparición de símbolos matemáticos [2]	47
3.3.	Frecuencia de aparición de letras mayúsculas [2]	48
3.4.	Frecuencia de aparición de letras minúsculas [2]	49

ÍNDICE DE FIGURAS

3.5. Frecuencia de aparición de letras griegas [2]	50
3.6. Frecuencia de aparición de símbolos especiales [2]	51
3.7. Expresión de dos niveles vertical	52
3.8. Diagrama de la arquitectura Watch, Attend, Parse (WAP).	53
4.1. Arquitectura del sistema.	56
4.2. Modelo de datos del sistema.	58
4.3. Diagrama de casos de uso del módulo Usuarios Android	63
4.4. Diagrama de casos de uso del módulo Proyectos Android	73
4.5. Diagrama de casos de uso del módulo Usuarios Web	82
4.6. Diagrama de casos de uso del módulo Proyectos Web	93
4.7. MIUA 1 Iniciar Sesión	111
4.8. MIUA 2 Registro	112
4.9. MIUA 3 Recuperar Contraseña	113
4.10. MIUA 4 Menú de Opciones	114
4.11. MIUA 5 Lista de Proyectos	115
4.12. MIUA 6 Lista de Traducciones	116
4.13. MIUA 7 Editar Perfil	117
4.14. MIUW 1 Iniciar Sesión	118
4.15. MIUW 2 Registro	118
4.16. MIUW 3 Editar Perfil	119
4.17. MIUW 4 Recuperar Contraseña	119
4.18. MIUW 5 Lista de Proyectos	120
4.19. MIUW 6 Lista de Traducciones	120
4.20. MIUW 7 Menú de Opciones	120
5.1. Tres capas que se tienen al utilizar la arquitectura Clean [3]	122
5.2. Capa de datos [4]	124
5.3. Capa de dominio [4]	129
5.4. Capa de presentación [5]	135
5.5. Ejemplo de una imagen del conjunto de entrenamiento Harvard 100k	143
5.6. Imagen de ejemplo del conjunto de entrenamiento generado	145
5.7. Fotografía procesada con algoritmo de Sauvola	145
5.8. Resultados de la aplicación del algoritmo de Otsu y Sauvola	148
5.9. Resultados de la aplicación del algoritmo de Otsu y Sauvola	148
5.10. Resultados de la aplicación del algoritmo de Otsu y Sauvola	149
5.11. Resultados de la aplicación del algoritmo de Otsu y Sauvola	149
5.12. Resultados de la aplicación del algoritmo de Otsu y Sauvola	150
5.13. Resultados de la aplicación del algoritmo de Otsu y Sauvola	150
5.14. Resultados de la aplicación del algoritmo de Otsu y Sauvola	151
5.15. Resultados de la aplicación del algoritmo de Otsu y Sauvola	151
5.16. Resultados de la aplicación del algoritmo de Otsu y Sauvola	152
5.17. Resultados de la aplicación del algoritmo de Otsu y Sauvola	152

ÍNDICE DE FIGURAS

5.18. Resultados de la aplicación del algoritmo de Otsu y Sauvola	153
5.19. Resultados de la aplicación del algoritmo de Otsu y Sauvola	153
5.20. Resultados de la aplicación del algoritmo de Otsu y Sauvola	154
5.21. Resultados de la aplicación del algoritmo de Otsu y Sauvola	154
5.22. Modelo <i>Encoder-Decoder</i> de la red neuronal.	156
5.23. (a) Función de perdida del sistema, el modelo se entreno en varias ocasiones, siempre obteniendo una gráfica parecida. (b) Se muestra la atención obtenida por este modelo, se observa un aprendizaje nulo.	169
5.24. (a) Función de perdida del sistema, el modelo se entreno en varias ocasiones, siempre obteniendo una gráfica parecida. (b) Se muestra la atención obtenida por este modelo, se observa un que el modelo aprendió que a indentificar donde estaba la ecuación, no obstante no aprendio a segmentar los símbolos.	172
6.1. Resultados de las pruebas de la clase DateFormatter	175
6.2. Resultados de las pruebas de la clase RNN002	178
6.3. Reporte del código sometido a pruebas	179
6.4. Resultado de las pruebas de Django	180

Índice de tablas

1.1.	Resumen de productos similares	4
3.1.	Análisis de riesgos.	41
3.2.	Tabla comparativa de sistemas operativos de dispositivos móviles	43
3.3.	Comparación de diferentes gestores de bases de datos.	44
3.4.	Comparación de frameworks de desarrollo web.	45
4.1.	Tabla de los atributos de la entidad usuario	59
4.2.	Tabla de los atributos de la entidad proyecto	60
4.3.	Tabla de los atributos de la entidad traducción	61
4.4.	Tabla de los atributos de la entidad tipo de estudios	61
4.5.	Tabla de los atributos de la entidad usuario token	62
5.1.	Especificación de la CNN. El 'Kernel' esta denotado como <i>número de filtros-dimensiones del filtro</i>	157

Capítulo 1

Introducción

\LaTeX es un sistema de tipografía de alta calidad que incluye características útiles para el diseño de documentos técnicos y científicos. Este software, es de facto un estándar para la comunicación y la publicación de artículos científicos.

A pesar de lo práctico que puede ser \LaTeX , si el documento contiene muchas expresiones matemáticas, puede resultar tedioso para el usuario escribir dichas expresiones. Por lo que este Trabajo Terminal tiene como finalidad brindar una herramienta que permita amenizar la interacción de nuevos usuarios al escribir expresiones matemáticas en \LaTeX mediante el uso de una aplicación móvil y una aplicación web, esta última capaz de reconocer expresiones matemáticas en imágenes que pueden ser tomadas desde el mismo smartphone.

1.1. Planteamiento del problema

En la actualidad no existe un sistema que permita reconocer expresiones matemáticas tomando como entrada una fotografía para su posterior traducción a L^AT_EX, si bien en los últimos años la investigación ha permitido que haya avances en cuanto al reconocimiento y traducción de expresiones matemáticas, al tener un enfoque de investigación se quedan como modelos fuera de lo práctico y los pocos existentes como se muestra en el estado del arte (Tabla 1.1) trabajan con entradas ideales como son expresiones escritas desde dispositivos tales como tabletas digitalizadoras o plumas electrónicas sin considerar entradas como pueden ser fotografías tomadas por un smartphone en las que la resolución varía de dispositivo a dispositivo eliminando un tamaño fijo de la entrada y otros elementos como el ruido que puede incorporarse al momento de la toma de la fotografía que afectan el reconocimiento de las expresiones matemáticas. Además, no cuentan con un control sobre dichas traducciones que se realizan o sobre los usuarios.

El enfoque del presente trabajo terminal consiste en desarrollar un sistema conformado por una aplicación tanto móvil como web que en conjunto permitan tomar fotografías y extraer las expresiones matemáticas para su posterior traducción incluyendo la capacidad de gestionar las traducciones con la posibilidad de la clasificación de las mismas mediante usuarios y proyectos asociados a dichas traducciones.

1.2. Estado del arte

Algunos sistemas similares que se han desarrollado son:

- Mathpix [6]
- MyScript Nebo [7]
- SESHAT [8]
- IDEAL Math Writer [9]

Los cuales se describen en la tabla 1.1:

SOFTWARE	CARACTERISTICAS	PRECIO EN EL MERCADO
Mathpix	Es una aplicación de escritorio en la que puedes usar un comando para tomar una captura de pantalla y convertir el texto capturado a L ^A T _E X. También cuenta con un API de pago	Este producto cuenta con diferentes planes de pago según su uso o el tiempo que decidas pagar lo. Tiene distinto trato para empresas. Un ejemplo de suscripción mensual es \$99 dólares el mes.
MyScript Nebo	Es una aplicación Android que transforma el texto escrito en el dispositivo en texto digital. Incluye soporte para ecuaciones. Es necesario el uso de una pluma digital.	\$189.00 en Google Play Store
Mathpix	Es una aplicación de escritorio en la que puedes usar un comando para tomar una captura de pantalla y convertir el texto capturado a L ^A T _E X. También cuenta con un API de pago	Este producto cuenta con diferentes planes de pago según su uso o el tiempo que decidas pagar lo. Tiene distinto trato para empresas. Un ejemplo de suscripción mensual es \$99 dólares el mes.

SESHAT	<p>Es un proyecto de doctorado de la Universidad Politécnica de Valencia open source.</p> <p>Convierte el texto de imágenes en texto digital y en formato LATEX. Soporta ecuaciones. Necesita ser instalado mediante terminal en Linux.</p>	No es una aplicación comercial
--------	---	--------------------------------

Tabla 1.1: Resumen de productos similares

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un sistema que reconozca un conjunto delimitado de tipos de expresiones matemáticas en una imagen dada y las traduzca a un formato que un compilador de LATEX pueda procesar, el conjunto de entrenamiento se encuentra delimitado en el capítulo de análisis.

1.3.2. Objetivos específicos

1. Desarrollar una aplicación móvil que permita tomar fotografías y que pueda conectarse a una aplicación web para su posterior procesamiento.
2. Desarrollar un módulo de análisis de imágenes para el reconocimiento de las expresiones matemáticas.
3. Desarrollar un módulo de traducción a LATEX.
4. Desarrollar la interfaz que conecte el módulo de análisis de imágenes alojado en el servidor con la aplicación móvil y web.

1.4. Resultados esperados

Podemos separar el sistema en dos bloques principales, el primero el lado del cliente el cual se compondrá de la aplicación móvil y de la interfaz web con las cuales el usuario podrá interactuar. Por otro lado, se tiene la parte del servidor web que se conectará con la aplicación móvil, en

el servidor se encontrará la parte de análisis de imágenes junto con el módulo de traducción. Finalmente se tiene el módulo de gestión de usuarios. Esto se puede apreciar en la Figura 1.1.

El sistema se compone de cinco módulos:

1. Módulo de análisis de imágenes. Este módulo se encargará de procesar las imágenes e identificar un conjunto delimitado de tipos de expresiones matemáticas.
2. Módulo de traducción. Este módulo tomará como entrada lo obtenido en la etapa de análisis de imágenes y regresará el respectivo código de L^AT_EX que represente las expresiones matemáticas reconocidas.
3. Módulo de gestión de usuarios. En este módulo se hará la gestión de la información de los usuarios de la aplicación, lo que implica tener un control de su información y de los archivos que generen al usar el sistema. Esta gestión de usuarios estará presente tanto en la aplicación móvil como en el servidor web.
4. Aplicación móvil. Este módulo consiste en el desarrollo de la aplicación móvil que el usuario final llevará en su Smartphone y con la cual podrá tomar fotos que cumplan ciertos requisitos para así obtener el código en L^AT_EX a través de la comunicación con la aplicación web.
5. Servidor web. El servidor web hará uso de los módulos de análisis de imágenes, de traducción y de gestión de usuarios por lo que deberá de mantener una comunicación entre ellos y la aplicación móvil, así como con la base de datos. Además, es el punto que permite visualizar el resultado final de todo el procesamiento de imágenes a través de una interfaz web.

Los productos esperados del Trabajo Terminal son:

6. Código fuente del trabajo desarrollado (todos los módulos).
7. Documentación técnica del sistema.

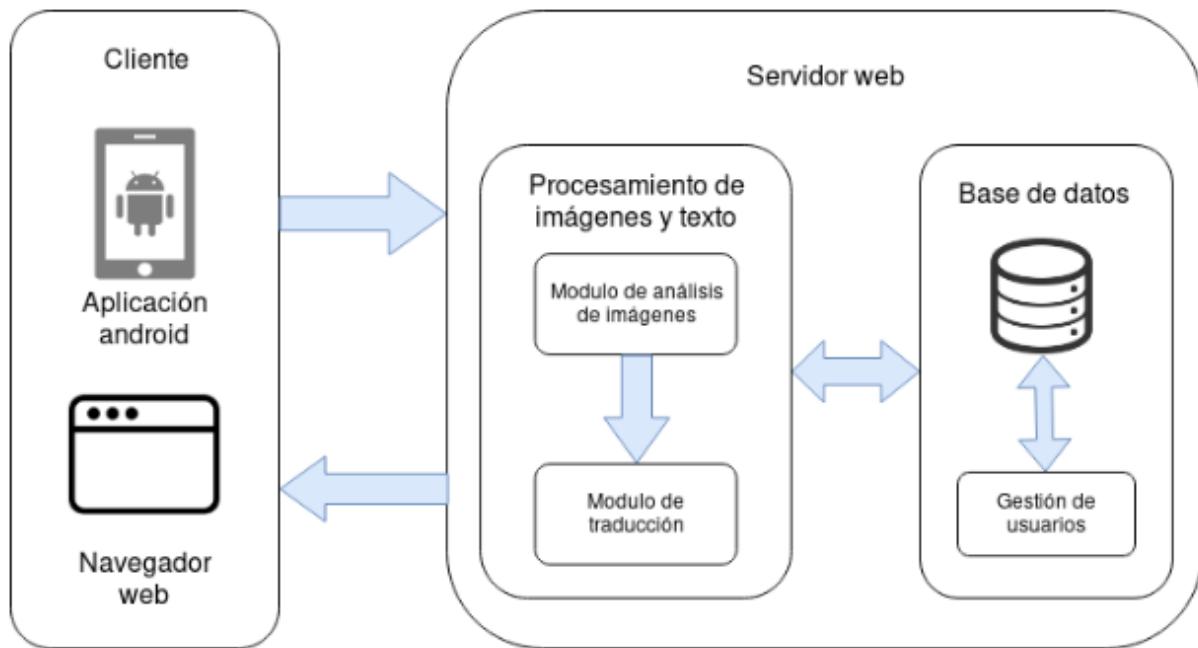


Figura 1.1: Arquitectura del sistema.

1.5. Metodología

Para el desarrollo del proyecto se eligió la metodología incremental debido a que permite dividir el proyecto en pequeños incrementos [10]. Este modelo nos permite flexibilidad a la hora de decidir que requerimientos funcionales se trabajaran primero, es decir, primero se trabajan aquellos con mayor prioridad en los primeros incrementos del sistema para después trabajar el resto o realizar un análisis más profundo a dichos requerimientos de incrementos futuros [11]. De esta forma, se plantean los siguientes incrementos con el objetivo de que al final de cada uno de estos se obtenga la funcionalidad requerida y validar que lo que se trabajo sea correcto para que al final de todos los incrementos se tenga el sistema completo.

- Se desarrollará la aplicación móvil la cual su principal funcionalidad es la obtención de las imágenes y su comunicación con el servidor web.
- Se desarrollará la aplicación web ya que es lo que nos servirá para conectar con la aplicación móvil con el resto de los módulos que se desarrollaran en un futuro.
- Se hará la integración entre la aplicación móvil y la aplicación web.
- Se detallaran a fondo requerimientos si es necesario, se realizara el análisis y diseño del módulo de análisis de imágenes, construcción del módulo y finalmente se harán pruebas sobre este.

- El módulo de traducción va de la mano con el módulo anterior, al tener esto listo y con sus respectivas validaciones aprobadas se realizará el desarrollo y con ello realizar pruebas sobre todos los componentes que se tengan hasta este punto.
- Este incremento final tendrá como objetivo el integrar todo lo desarrollado hasta este punto.

Capítulo 2

Marco teórico

Con el desarrollo del sistema producto del presente trabajo terminal se involucran ciertos conceptos provenientes en su mayoría de ramas de ciencias de la computación y en general en alusión a la Inteligencia Artificial, por lo que es conveniente dar contexto sobre los elementos necesarios para el desarrollo del trabajo terminal.

2.1. Análisis de Imágenes

Una imagen puede definirse como una función bidimensional. $f(x,y)$, donde x y y son coordenadas espaciales (plano) y la amplitud de f en cualquier par de coordenadas (x,y) es llamada *intensity* o *gray level* (nivel de grises) de la imagen en ese punto.

Cuando x , y y los valores de amplitud de f son todas cantidades discretas finitas, la imagen se denomina imagen digital. El campo de análisis de imagen se refiere a procesar imágenes por medio de una computadora digital. Una imagen digital está compuesta por número de elementos finitos, de los cuales cada uno tiene una posición particular y valor. Estos elementos son referidos como **picture elements**, **image elements**, **pels** y **pixels**. Pixel es el término más ampliamente usado para denotar los elementos de una imagen digital.

El análisis de imágenes comprende un conjunto de operaciones sobre una o varias imágenes con el propósito de obtener una imagen con mayor realce o para extraer características útiles, es un tipo de dispensación de señales en el que la entrada es una imagen y la salida puede ser otra imagen o características asociadas a la imagen, algunos de los pasos generales se describen a continuación:

2.1.1. Preprocesamiento

Preprocesamiento es un nombre común para operaciones con imágenes al más bajo nivel de abstracción. Tanto entrada como salida son imágenes de intensidad. Estas imágenes tienen el mismo tipo de datos que la original, con una imagen de intensidad usualmente representada por una

matriz de valores de función de imagen (Brillo) El objetivo de preprocesar es la mejora de los datos de la imagen que borre distorsiones o realce características importantes para procesamiento posterior, incluso las transformaciones geométricas de las imágenes e.g (rotación, escalamiento y traslación) son también clasificadas como métodos de preprocesamiento, ya que técnicas similares son utilizadas [12].

2.1.2. Realce de la imagen

El objetivo principal de realce de imagen es también procesar una imagen dada tal que el resultado sea mas ajustable que la imagen original para aplicaciones específicas. Por ejemplo para la remoción de ruido.

Acentúa o afina características de la imagen como ejes, límites o contraste para hacer un despliegue gráfico mas útil para el análisis.

El realce no incrementa o decrementa el contenido de la información inherente de los datos pero sí incrementa el rango dinámico de las características elegidas de tal modo que puedan ser detectadas fácilmente.

Proveé mejoréentrada para otras técnicas avanzadas de procesamiento automatizadas de imágenes.

Los enfoques de realce de imagen se dividen en categorías amplias: métodos de dominio espacial y métodos de dominio de frecuencia. El término *dominio espacial* se refiere al plano mismo de la imagen y las aproximaciones en esta categoría son basadas en manipulación directa en una imagen. Técnicas de procesamiento de *dominio de frecuencia* están basadas en modificar la transformada de Fourier de una imagen.

El término *dominio espacial* se refiere al agregado de píxeles que componen una imagen. Los métodos de dominio espacial son procesos que operan directamente en estos píxeles. Los procesos de dominio espacial serán denotados por la expresión

$$g(x,y) = T[f(x,y)] \quad (2.1)$$

donde $f(x,y)$ es la imagen de entrada, $g(x,y)$ es la imagen procesada y T es un operador sobre f , definido sobre alguna vecindad de (x,y) . Además, T puede operar sobre un conjunto de imágenes de entrada, como llevar a cabo la suma pixel-by-pixel de K imágenes para reducción de ruido.

La principal aproximación para definir una vecindad de un punto (x,y) es usar una área de una sub-imagen cuadrada o rectangular centrada en (x,y) , como se muestra en

El centro de la subimagen es movido de pixel a pixel iniciando por ejemplo en la esquina superior izquierda. El operador T es aplicado en cada posición (x,y) para producir la salida g en esa posición. El proceso utiliza solo los píxeles en el área de la imagen expandida por la vecindad. A pesar de que otras vecindades le dan forma, como aproximaciones a un círculo son usadas

en ocasiones, arreglos cuadrados y rectangulares son por mucho los más predominantes por la facilidad de implementación.

La forma mas simple de T es cuando la vecindad es de tamaño 1×1 (un solo pixel). En este caso, g depende solo del valor de f en (x, y) y T se convierte en una *función de transformación a nivel de grises* (también llamada *intensity o mapeo*) de la forma

$$s = T(r) \quad (2.2)$$

donde por simplicidad de notación, r y s son variables denotando respectivamente el nivel de gris de $f(x, y)$ y $g(x, y)$ en cualquier punto (x, y) . Por ejemplo, si $T(r)$ tiene la forma mostrada en 2.1(a), el efecto de esta transformación sería producir una imagen de mas alto contraste que la original al oscurecer los niveles debajo de m y darle brillo a los niveles arriba de m en la imagen original. En esta técnica conocida como contrast stretching, los valores de r por debajo de m son comprimidos por la función de transformación en un rango estrecho de s hacia el negro. El efecto opuesto da lugar para valores de r por encima de m . En el caso limitante mostrado en 2.1(b), $T(r)$ produce una imagen two-level (binaria). Un mapeo de esta forma es llamado una función de *thresholding*.

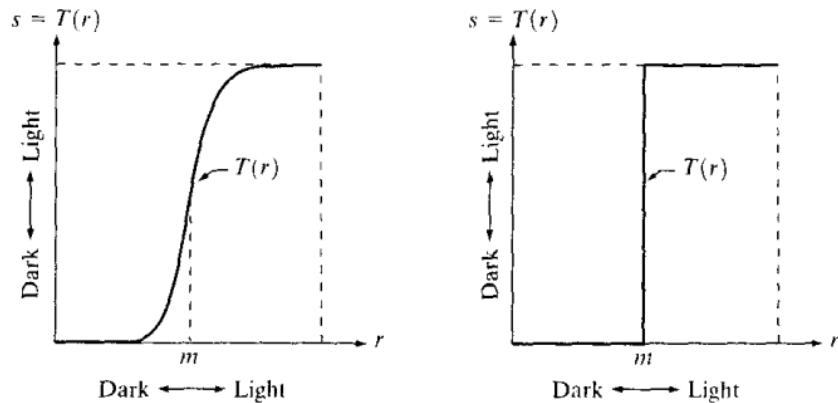


Figura 2.1: Transformación de nivel de grises para realce de contraste.

2.1.2.1. Transformaciones a nivel de grises básicas

Las transformaciones a nivel de grises son las más básicas en el proceso de realce de imágenes. Los valores de los píxeles antes y después de ser procesados se denotan por r y s respectivamente. Como se mencionó anteriormente, estos valores son relacionados por 2.2, como se trabaja con cantidades digitales, los valores de las funciones de transformación típicamente son almacenadas en un arreglo unidimensional y los mapeos de r y s son implementados via por búsquedas de tabla (lookup tables). Para un entorno de 8 bits, una tabla lookup que contiene los valores de T tendrá 256 entradas.

Como introducción a las transformaciones a nivel de grises, considera la figura 2.2 la cual muestra tres tipos básicos de funciones usadas frecuentemente para realce de imágenes: lineal (transformaciones negative e identity), logarítmicas (transformaciones log e inverse-log) y power-law (transformaciones de potencia n th y raíz n th). La función identidad es el caso trivial en el cual las intensidades de salida son idénticas a la intensidades de entrada.

2.1.2.2. Negativos de Imagen

El negativo de una imagen con niveles de gris en el rango $[0, L - 1]$ es obtenido al usar la transformación negativa mostrada en 2.2, la cual es dado por la expresión

$$s = L - 1 - r \quad (2.3)$$

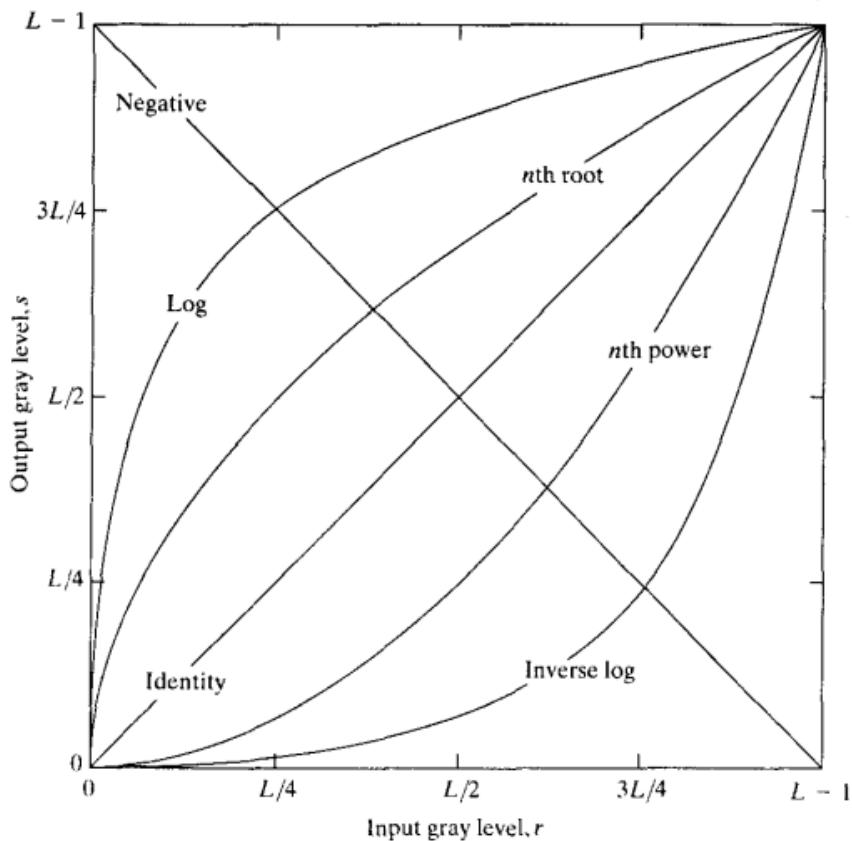


Figura 2.2: Algunas transformaciones básicas a nivel de grises usadas para realce de imagen.

Invertir los niveles de intensidad de una imagen en esta forma produce el equivalente del negativo de fotografía. Este tipo de procesamiento es particularmente ajustado para el realce de detalles de blanco y negro embebidos en regiones oscuras de una imagen, especialmente cuando las áreas negras dominan en tamaño.

2.1.2.3. Transformaciones Log

La forma general de la transformación log que se muestra en la figura 2.2 es

$$s = c \log(1 + r) \quad (2.4)$$

donde c es una constante y se asume que $r \geq 0$. La forma de la curva log en 2.2 muestra que esta transformación mapea un rango estrecho de valores con bajos niveles de gris en la imagen de entrada a un rango más amplio de niveles de salida. Lo opuesto es verdadero de valores más altos de niveles de entrada. Este tipo de transformaciones se usa para expandir los valores de pixeles oscuros en una imagen mientras se comprimen los valores de nivel más altos. Lo opuesto es verdadero de la transformación inversa de log.

Cualquier curva que tiene la forma general de funciones log mostradas en 2.2 completaría esta expansión/compresión de niveles de gris de una imagen. Las transformaciones log tienen la importante característica de que comprime el rango dinámico de imágenes con grandes variaciones en valores de pixeles. Una clásica ilustración de una aplicación en la cual los valores de pixel tienen un gran rango dinámico es el espectro de Fourier, no es inusual encontrar valores de espectro entre 0 y 10^6 o más grandes.

2.1.2.4. Transformaciones Power-Law

Las transformaciones power-law tienen la forma básica

$$s = cr^\gamma \quad (2.5)$$

donde c y γ son constantes positivas. En ocasiones 2.5 se escribe como $c(r + \varepsilon)^\gamma$ para considerar una compensación (esto es, una salida medible cuando la entrada es cero). Sin embargo, las compensaciones típicamente son un problema de calibración y como resultado se ignoran normalmente en la ecuación 2.5. Las gráficas de r vs s para distintos valores de γ se muestran en

2.1.3. Segmentación de la imagen

El término segmentación utilizada en el contexto de análisis de imágenes se refiere a la partición de una imagen en un conjunto de regiones que la cubren. El objetivo en muchas de las tareas es que para las regiones se representen áreas significativas de la imagen, como áreas urbanas, fronteras o bosques de una imagen satelital. En otras tareas de análisis, las regiones pueden ser conjuntos de bordes de pixeles agrupados en estructuras como segmentos de líneas y segmentos de arcos circulares en imágenes de objetos industriales en 3D. Las regiones pueden también estar definidas como grupos de pixeles teniendo ambos un borde y una forma particular como un círculo o una elipse o polígono. Cuando las regiones de interés no cubren la imagen completa, aún se requiere el proceso de segmentación en regiones de fondo para ignorarse. [12]

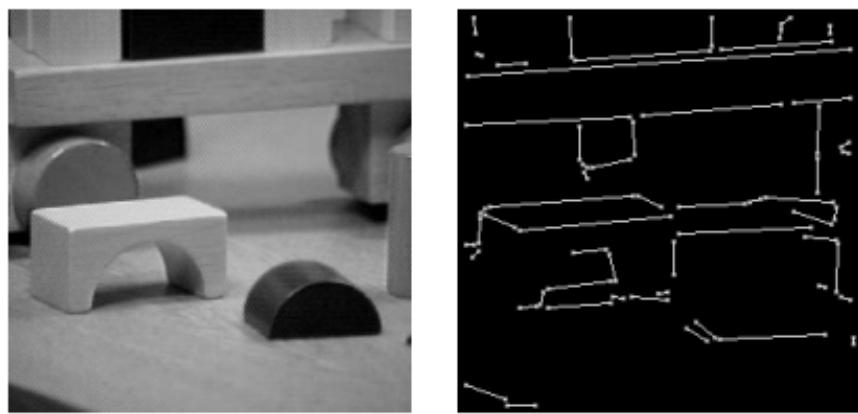


Figura 2.3: Imagen con bloques (Izquierda) y conjunto de segmentos de linea extraídos (Derecha).

2.2. Deep Learning

El aprendizaje profundo o *deep learning* es una rama del aprendizaje automático (*machine learning* en inglés) que intenta modelar abstracciones de alto nivel a través de complejas arquitecturas computacionales que admiten transformaciones no lineales. El deep learning es la evolución de las ya conocidas redes neuronales las cuales experimentaban problemas de desvanecimiento del gradiente si se usaban demasiadas capas. Con las nuevas técnicas propuestas en el deep learning se logró evitar este problema y así, poder entrenar arquitecturas de millones de parámetros.

El perceptrón multicapa (MLP) es el modelo más básico y aún así uno de los más útiles en el campo de las redes neuronales. Su principal objetivo es tratar de aproximar una función f^* . En el caso del *aprendizaje supervisado*, la función f^* toma la forma $y^* = f^*(x)$ de donde x es un parámetro de entrada que puede ser desde un simple número hasta un tensor y y^* es la salida de la función. Entonces, una red neuronal que quiera aproximar f^* definirá una función de mapeo de la forma $y = f(x, \theta)$ y aprenderá los parámetros θ (usualmente conocidos como W y b) que dan como resultado la mejor aproximación de tal forma que $y \approx y^*$.

Las redes neuronales profundas son llamadas redes porque pueden representarse mediante la composición de varias funciones de mapeo. Es decir, un MLP que quiera aproximar una función puede expresarse como $y^n = f^n(y^{n-1}, \theta^n)$ con $y^1 = f^1(x, \theta^1)$ y n siendo la profundidad de la red.

2.2.1. Gráficos computacionales

Para describir con formalidad a las redes neuronales es preciso utilizar una notación que pueda ser expresada a través de gráficos. Según [13] se puede indicar cada nodo del gráfico como una variable que puede ser un tensor para no perder generalidad.

Para terminar con la definición de estos gráficos, es necesario introducir el concepto de operación, la cual simplemente es una función entre dos o más nodos del gráfico. Sin perder generalidad, se dice que una operación retorna únicamente una variable, es decir, un solo nodo.

En la Figura 2.4 se puede ver un ejemplo de un perceptrón de una sola capa que es representado mediante un gráfico computacional. Gracias a esta definición, se puede extender fácilmente el gráfico mostrado en 2.4 para modelar a un perceptrón de n capas.

2.2.2. Batch Normalization

Cada capa de una red neuronal tiene entradas con su correspondiente distribución la cual es afectada durante el entrenamiento por la aleatoriedad en la inicialización de parámetros y en la aleatoriedad de los datos de entrada. Los efectos de esta aleatoriedad en la distribución de las entradas de la red hacia las capas internas en el momento de entrenamiento es descrita como un cambio en la covarianza, el cual provoca inestabilidad en la red.

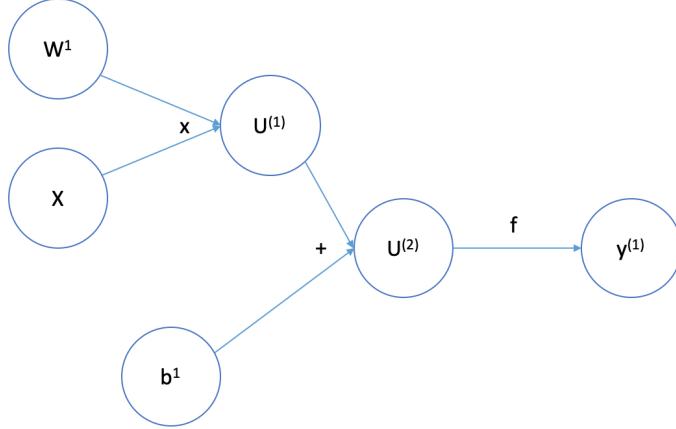


Figura 2.4: Ejemplo de un perceptrón de una sola capa representado mediante un gráfo computacional, siendo x la entrada, w^1 la matriz de pesos, b^1 la matriz de bias, $u^{(1)}$ y $u^{(2)}$ nodos intermedios en el gráfo y y^1 la salida de la red. La ecuación modelada es $y^1 = f(W^1x + b^1)$.

La técnica de *Batch Normalization* es utilizada para incrementar la estabilidad, la velocidad y el desempeño de una red neuronal. Esta técnica fue introducida en el 2005 por [14].

En una red neuronal, la técnica de Batch Normalization consiste en una etapa de normalización de las medias y varianzas de las entradas de una capa de la red. Esta operación esta definida de la siguiente manera:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.6)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2.7)$$

done B denota algún mini batch de tamaño m . Para una capa con una entrada $x \in \mathbb{R}^n$ entonces:

$$\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_B^{(k)2} + \epsilon}} \quad (2.8)$$

done $k \in [1, n]$, $i \in [1, m]$, $\mu_B^{(k)}$ y $\sigma_B^{(k)2}$ son la media y varianza por cada dimensión respectivamente; ϵ es añadido como una constante arbitraria y pequeña para asegurar la estabilidad numérica del denominador.

Finalmente, la salida de la capa es:

$$y_i^{(k)} = \gamma^{(k)} \hat{x}_i^{(k)} + \beta^{(k)} \quad (2.9)$$

de donde los parámetros $\gamma^{(k)}$ y $\beta^{(k)}$ serán aprendidos durante el entrenamiento.

2.2.3. Redes Neuronales Convolucionales

Las redes neuronales convolucionales también llamadas convolutivas (CNN), son una arquitectura especial de redes neuronales que permite el procesamiento de datos que tengan una estructura de grilla. Algunos ejemplos de esta estructura son: una serie de tiempo es una grilla 1D o una imagen, la cual puede ser vista como una grilla 2D de pixeles. Se puede decir, que una red convolutiva no es más que una red neuronal que utiliza la operación de convolución en lugar de una multiplicación tensorial normal [13].

La operación de convolución está definida de la siguiente manera:

$$y(t) = x(t) * w(t) \quad (2.10)$$

$$y(t) = \int_{-\infty}^{\infty} x(\tau)w(t - \tau)d\tau \quad (2.11)$$

En terminología de las CNN x es la entrada a la red, w es llamado el **kernel** o **filtro** y y el mapa de características.

Propiamente en los sistemas computacionales no se pueden tener funciones continuas, estas tienen que ser discretas, por lo tanto la operación de convolución que una CNN utiliza es:

$$y(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)w(t - \tau) \quad (2.12)$$

Naturalmente, si lo que se está trabajando es una imagen, tenemos que aplicar la operación de convolución en dos ejes, es decir:

$$y(i, j) = \sum_m \sum_n x(i, j)w(i - m, j - n) \quad (2.13)$$

Si se toma el gráfico computacional del ejemplo de la Figura 2.4 el cual modela a un perceptrón de una sola capa y se cambia su operación de multiplicación matricial por la operación de convolución, tenemos una capa convolucional:

$$y^1 = f(W^1 * x + b^1) \quad (2.14)$$

La manera en la que se entrena estas variantes de MLP es con el mismo algoritmo de optimización que un perceptrón.

En la Figura 2.5 se puede ver un ejemplo de cómo una imagen es procesada por la etapa de convolución en una CNN.

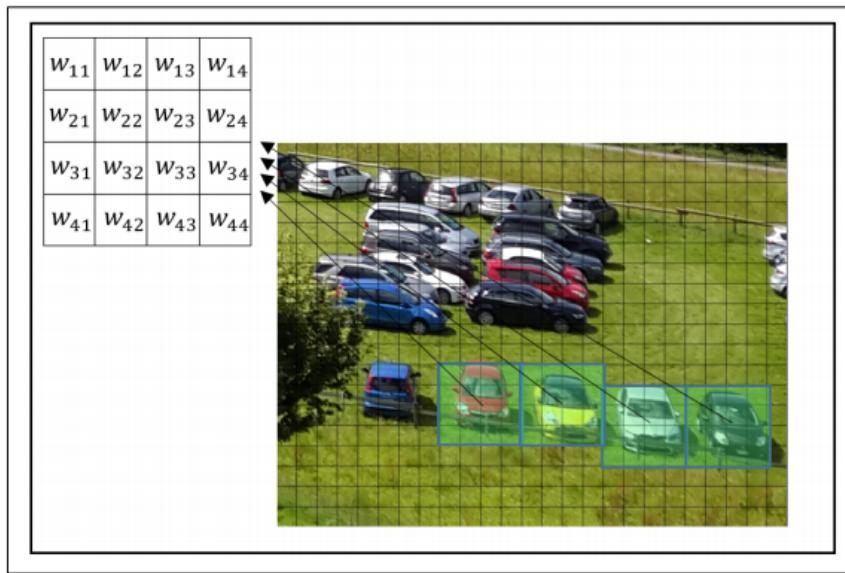


Figura 2.5: Matriz de pesos conocida como kernel o filtro aplicada a una imagen.

2.2.3.1. Pooling

Es una de las operaciones adicionales que se implementan en las CNN. En general se puede establecer que una típica capa convolucional tiene tres partes:

1. Operación de convolución: Es donde utiliza una entrada y la convoluciona con un filtro w .
2. Función de activación: A veces es llamada fase de detección. Consiste en aplicar una función f al resultado de la primera etapa para modificar la salida y volverla no lineal.
3. Etapa de Pooling: Es una manera de modificar aún más la salida de modo que se incremente la significancia estadística de los datos extraídos.

La operación de pooling permite que la red sea tolerante a traslaciones y rotaciones leves de la imagen. Por ejemplo, se puede aplicar la función *max pooling* a cada uno de los valores de $f(W * h)$ para obtener una generalización de los datos más importantes observados por la etapa de detección.

Una manera efectiva de reducir el tamaño de los datos que la red tiene que procesar a través de sus capas, es utilizar la operación pooling para remuestrear. Esto se logra definiendo una ventana a la que se le aplicará la operación. Esta técnica también permite que algunas CNN puedan procesar imágenes de longitud variable.

Existen diversas operaciones de pooling, las más comunes son *Max Pooling* y *Average Pooling*. En la Figura 2.6 se puede apreciar una representación gráfica de la operación de Max Pooling sobre alguna matriz.

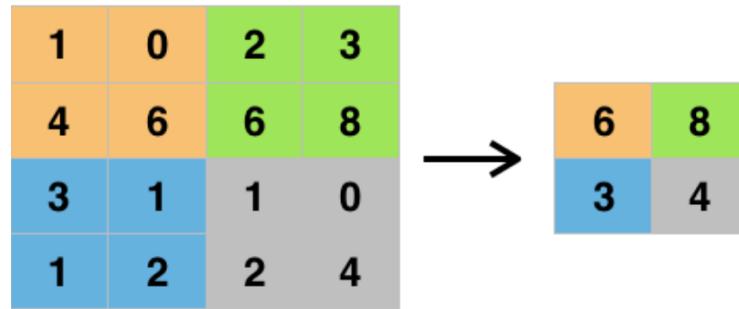


Figura 2.6: Representación gráfica de la operación Max Pooling.

2.2.4. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNNs) son una familia de redes neuronales especializadas en procesar secuencias de datos. Es decir, pueden procesar entradas de la forma: $x^{(1)}, x^{(2)}, \dots, x^{(t)}$. Esta particular arquitectura, permite modelar secuencias de datos de longitud variable que serían imprácticas para cualquier otra arquitectura.

Las RNNs modelan sistemas dinámicos, es decir, que cada instante t tenemos un sistema $s(t)$ con un estado diferente. Para que los siguientes estados del sistema tengan información de los estados pasados, las RNNs tienen al menos una conexión que es recurrente, es decir, que depende de un estado anterior o en algunos casos de estados futuros.

Una de sus principales ventajas es la compartición de parámetros. Esto quiere decir que para diferentes estados del sistema, podemos aplicar los mismos parámetros θ , permitiendo así, entrenar solo un conjunto de parámetros para todo el sistema en vez de un conjunto particular para cada estado.

En la Figura 2.7 se puede apreciar un ejemplo de una RNN que cuya ecuación es: $s(t) = f(s^{(t-1)}, \theta)$. Como se han definido las redes neuronales en términos de gráficos, es posible utilizar el algoritmo de back-propagation para calcular los gradientes de cada estado. Normalmente cuando se aplica back-propagation a una RNN se le conoce como back-propagation through time (BPTT).

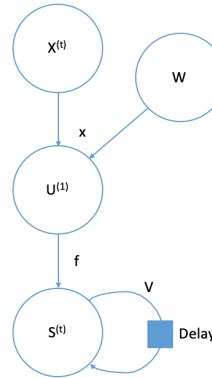


Figura 2.7: Ejemplo de una RNN que modela la ecuación $s(t) = f(s^{(t-1)}, \theta)$, donde θ representa los parámetros W y V , f la función no lineal aplicada a $Wx^{(t)}$ y V la matriz de pesos aplicada a $s^{(t-1)}$.

La manera que tiene una RNN para poder procesar una secuencia de longitud variable y brindar información respecto de ella, es a través de los estados $s(t)$ ya que estos si tienen una longitud fija. De este modo se puede decir que funcionan muy parecido a una CNN, ya que sumarizan las características de la secuencia hasta el momento t . Para realizar alguna predicción final, normalmente se utiliza alguna otra capa que efectue una función f , justo como lo hacen las CNNs.

2.2.4.1. Arquitectura Encoder-Decoder

Esta arquitectura de RNN se ha vuelto muy popular pues permite a una red procesar secuencias de longitud variable y obtener como resultado secuencias de longitud variable. Este tipo de problemas son conocidos como problemas sequence-to-sequence. Algunos ejemplos de problemas de este estilo son el reconocimiento del habla en tiempo real o el reconocimiento de la escritura.

La arquitectura se compone de dos redes recurrentes, la primera es denominada el *encoder*, el cual es una red que 'codifica' la entrada $x^{(t)}$. Si lo que se quiere es codificar una serie de tiempo, normalmente se usará una RNN y como salida será la señal $s(t)$. En este tipo de redes a la señal $s(t)$ se le conoce como el contexto C .

La segunda red que compone a la arquitectura es el *decoder*, el cual toma como entrada el contexto C generado previamente por el encoder y lo utiliza para generar la secuencia de salida. El decoder es en general una RNN.

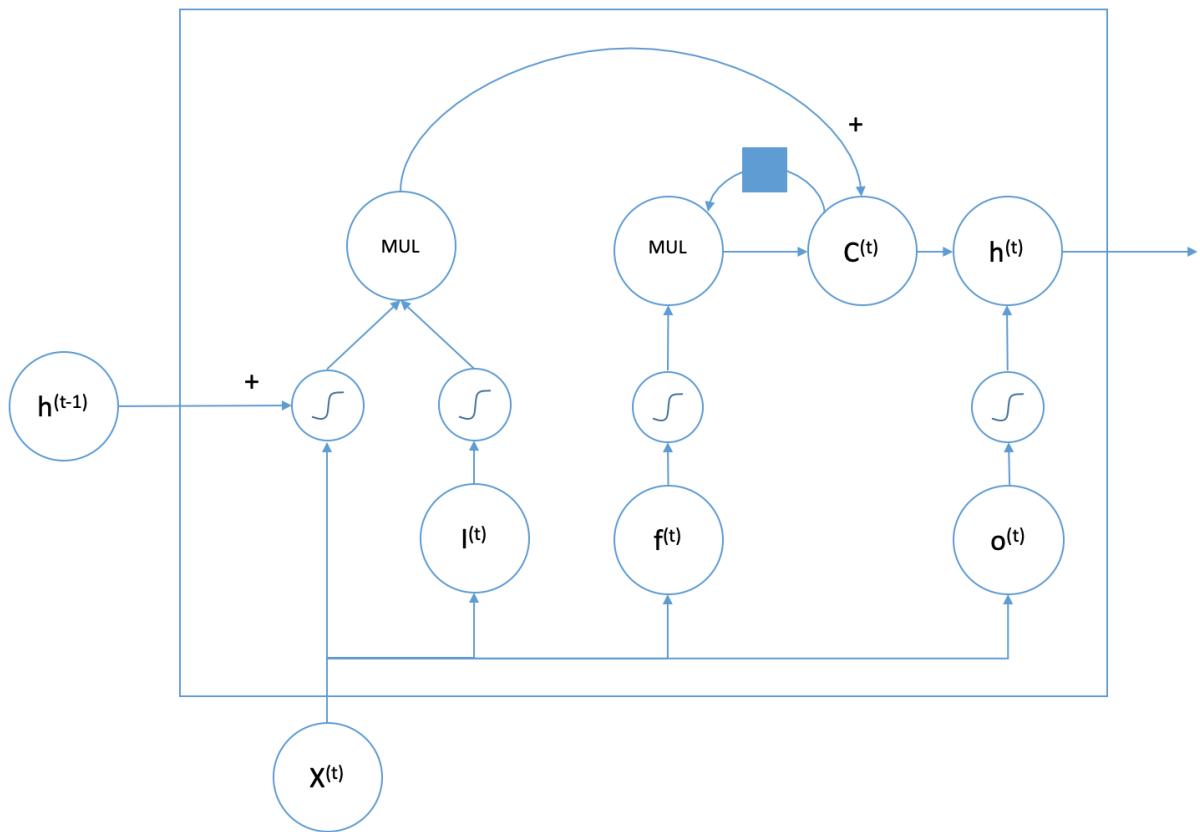


Figura 2.8: Una compuerta LSTM típica.

2.2.4.2. Long Short-Term Memory

Hasta ahora, las arquitecturas más exitosas para procesar secuencias son las llamadas Gated RNN. Estas redes permiten aprender secuencias muy largas o que requieran memorizar mucha información. El problema que existía, era que el gradiente tendía desaparecer conforme la red se volviera más y más profunda. Las Gated RNN son la solución más exitosa a este problema.

La Gated RNN más popular es la Long Short-Term Memory (LSTM). La idea de esta red, es crear ciclos entre los mismos nodos pero que estén condicionados con el contexto, de modo que la red pueda decidir cuando olvidar la información previa.

La LSTM se puede ver como una compuerta lógica que puede ser implementada en otra RNN para crear una arquitectura más robusta. En la Figura 2.8 se muestra un ejemplo de la compuerta LSTM. La LSTM implementa diferentes puertas las cuales son: **input gate** $g^{(t)}$, **forget gate** $f^{(t)}$ y la **output gate** $o^{(t)}$. Todas estas puertas ayudan a la LSTM a controlar en qué momento es oportuno olvidar la información previamente encontrada en los anteriores estados.

Las ecuaciones que describen a una LSTM típicamente son:

$$f_i^{(t)} = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{t-1} \right) \quad (2.15)$$

$$s_i^{(t)} = f_i^{(t)} s_i^{t-1} + g_i^{(t)} \sigma \left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{t-1} \right) \quad (2.16)$$

$$g_i^{(t)} = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{t-1} \right) \quad (2.17)$$

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \quad (2.18)$$

$$q_i^{(t)} = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{t-1} \right) \quad (2.19)$$

de donde $s^{(t)}$ es el estado interno de la RNN, los parámetros b, W, U son transformaciones lineales que serán aprendidas, $x^{(t)}$ son las entradas a la red.

2.2.4.3. Modelos de Lenguaje Condicional

Un modelo de lenguaje define la distribución de probabilidad sobre una serie de tokens en un lenguaje natural. Dependiendo de como el modelo es diseñado, un token puede representar una letra, una palabra o incluso un solo bit. Cuando se utilizan modelos de deep learning para resolver este problema, usualmente este modelo es llamado modelo de lenguaje neural.

Los modelos de lenguaje neurales tienen ventajas tales como:

- Resuelven el problema de dimensionalidad presentado por otros modelos.
- Son capaces de reconocer la similitud entre dos palabras sin perder la generalidad de codificarlas de una forma distinta.
- Alcanzan una buena comprensión estadística entre una palabra y su contexto.

En estos modelos, las palabras tienden a ser representadas por un identificador, el cual usualmente es un número, no obstante, una práctica muy común y que ha demostrado ser muy útil son los *word embeddings*. Estos concisten en codificar las palabras no a través de un solo número, sino a través de un conjunto de números decimales.

Un modelo de lenguaje condicional, es entonces, un modelo de lenguaje que toma en cuenta las palabras que han sido generadas previamente. Es decir:

$$P(w_1, \dots, w_m) = \prod_{i=1}^m P(w_i | w_1, \dots, w_{i-1}) \quad (2.20)$$

2.2.4.4. Sistema de Atención

El sistema de atención fue introducido por [15] y es en esencia un promedio con pesos. El resultado de un sistema de atención es un vector de contexto c_t el cual es obtenido al realizar el promedio entre las anotaciones a_t y multiplicarlas por unos pesos α_t , los cuales usualmente están en el rango $[0, 1]$. El modelo de atención es capaz de aprender en que parte del texto enfocarse en cada instante de tiempo t por lo que ha supuesto un impresionante avance en los modelos de lenguaje natural.

2.2.4.5. Positional Embeddings

Debido a que en un modelo de lenguaje una palabra puede representar cosas diferentes dependiendo de la posición que ocupe en la secuencia, es en ocasiones necesario brindar a la red de más información sobre la ubicación de cada token. Los *Positional Embeddings* fueron originalmente propuestos por [16]. La técnica conciste en sumar un vector *pe* a los *words embeddings* de cada token. Los autores del artículo proponen al vector *pe* de tal forma que:

$$pe_t^{(i)} = f(t)^{(i)} = \begin{cases} \sin(\omega_k t), & \text{si } i = 2k \\ \cos(\omega_k t), & \text{si } i = 2k + 1 \end{cases} \quad (2.21)$$

con

$$\omega_k = \frac{1}{10000^{2k/d}} \quad (2.22)$$

donde d es la dimensión de los *word embeddings*.

2.2.5. Aprendizaje profundo como servicio

El entrenamiento de redes neuronales profundas, conocidas como aprendizaje profundo es en la actualidad altamente complejo computacionalmente. Requiere un sistema con la combinación correcta de software, drivers, memoria, red y recursos de almacenamiento, por factores como estos los proveedores de cloud computing como Amazon Web Services, Microsoft Azure o Google Cloud ofrecen actualmente servicios de Machine Learning proveyendo API's como pueden ser de entrenamiento de modelos o de visualización de datos y generación de estadísticas, permitiendo que desarrolladores y científicos de datos se enfoquen mas en tareas como el entrenamiento de modelos, análisis de datos, etc.

Los modelos de aprendizaje profundo requieren de un proceso experimental e iterativo, requiriendo cientos e incluso miles de ejecuciones que requieren de un amplio poder de cómputo para encontrar la combinación correcta de las configuraciones e hiper-parámetros de la red neuronal. Esto puede tomar semanas o incluso meses y este tipo de servicios garantiza una disponibilidad en todo momento y en conjunto permite que un sistema completo esté dividido en módulos consiguiendo que el mantenimiento y detección de fallos sean tareas más sencillas.

En la figura 2.9 se puede observar la arquitectura de un sistema dividido en módulos y que hace uso de herramientas de Machine Learning para dar respuesta a otros módulos del sistema.

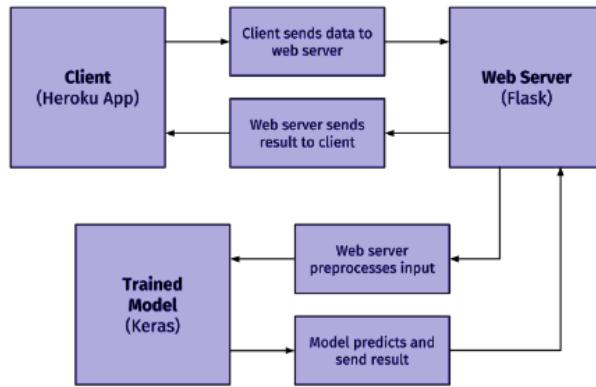


Figura 2.9: Ejemplo de sistema con módulo de Machine Learning

2.3. Métodos de reconocimiento de expresiones matemáticas

2.3.1. Análisis sintáctico dirigido

Los lenguajes libres de contexto, son aquellos que tienen una notación recursiva natural llamada *gramática libre de contexto*. Por ende, un lenguaje libre de contexto es todo aquel que puede ser representado por una gramática libre de contexto.

Una gramática libre de contexto G , queda definida de la siguiente manera:

$$G = (V, T, P, S) \quad (2.23)$$

de donde, V es el conjunto de variables, T el conjunto de símbolos terminales, P el conjunto de producciones y S el punto de inicio [17].

El lenguaje matemático, es decir, las expresiones matemáticas son un lenguaje libre de contexto. Esto quiere decir que es posible definir una gramática libre de contexto para definir a las expresiones matemáticas. Esta es una práctica muy común en la teoría de compiladores.

Sabiendo esto, una aproximación a la resolución del problema de reconocer expresiones matemáticas en imágenes es la que describe [18]. El método consiste en dos etapas:

- Reconocimiento de caracteres: Utilizar algún método de reconocimiento de patrones para localizar los caracteres y anotar sus coordenadas.
- Análisis sintáctico dirigido: Utilizar la etapa previa para determinar la jerarquía, basándose en un conjunto de reglas definidas (las gramáticas libres de contexto).

En la Figura 2.10, se puede ver un ejemplo del proceso de reconocimiento propuesto por este método.

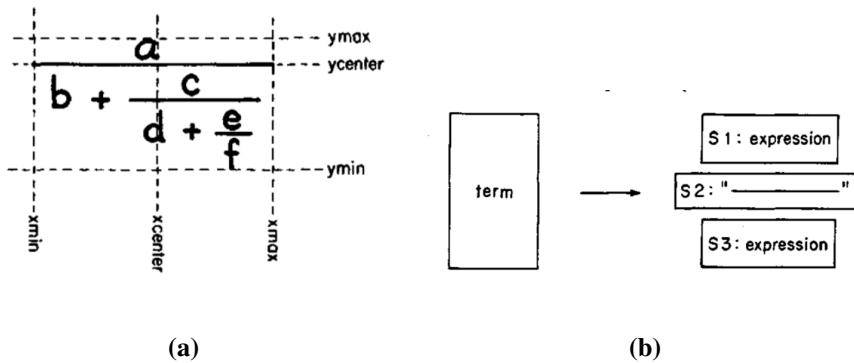


Figura 2.10: a) La primera etapa del método de reconocimiento, se anotan las coordenadas de cada carácter. b) Se realiza un análisis sintáctico con las gramáticas libres de contexto definidas.

2.3.2. Análisis estructural

Este método coincide con el primero en que debe de utilizar una técnica de reconocimiento de patrones para etiquetar primero los símbolos. Las etiquetas que utiliza tienen que ver son su posición en la imagen, así como su tamaño.

La diferencia en este método, radica en su segunda etapa. No se realizará un análisis con gramáticas, en su lugar se intentara deducir la estructura jerárquica con algún otro método. El artículo [19] propone utilizar el algoritmo de Kruskal para obtener un árbol de recubrimiento mínimo por niveles, de este modo se puede saber la estructura de la expresión matemática reconociendo el gráfico resultante. La Figura 2.11 muestra un ejemplo de como funciona este método.

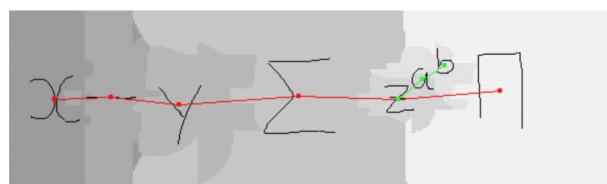


Figura 2.11: Utilización de un árbol de recubrimiento mínimo para el reconocimiento de expresiones matemáticas.

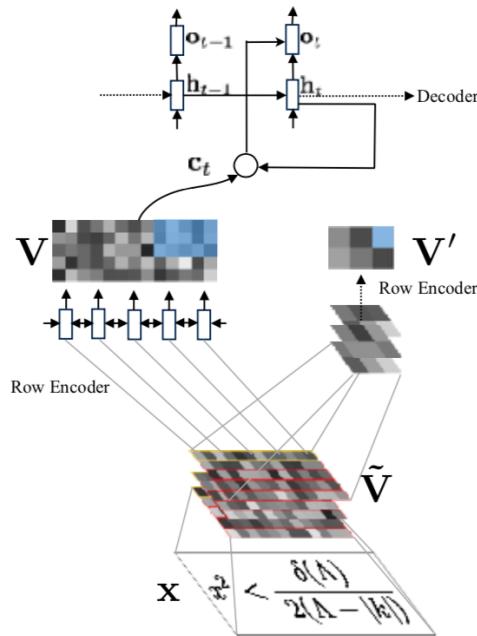


Figura 2.12: Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.

2.3.3. Image Captioning

Es una rama emergente del *deep learning* que ha ido ganando atención en los últimos años. Este campo es un punto intermedio entre la visión por computadora y el procesamiento del lenguaje natural. El actual estado del arte en image captioning tiene una aproximación similar a los modelos sequence-to-sequence, los cuales utilizan una arquitectura Encoder-Decoder [20].

Si se trata al problema de reconocer expresiones matemáticas en imágenes como un problema sequence-to-sequence de image captioning, se puede emplear una arquitectura de Encoding-Decoding para hacer la conversión de la imagen a LaTeX de manera directa. Esto permite a la red manejar imágenes de longitud variable y reconocer los símbolos a la vez que va reconociendo las expresiones.

La arquitectura que se está utilizando actualmente para solucionar este problema se muestra en la Figura 2.12 [20][21][22].

2.4. Aplicación web

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador [23].

2.4.1. Django

Django es un framework de desarrollo web completamente desarrollado en Python. Permite de una manera rápida poder implementar una aplicación web en el lenguaje Python. Las siguientes, son de las principales características de Django:

- **Rápido:** Tiene como filosofía ayudar a los desarrolladores a crear aplicaciones en el menor tiempo posible.
- **Completo:** Incluye cientos de librerías que permiten ahorrar tiempo y automatizar tareas.
- **Seguro:** Es una de las principales características de Django ya que incluye soluciones a los principales ataques que puede sufrir una aplicación web.
- **Escalable:** Con el patrón de diseño de Django es posible incrementar o decrementar la capacidad de un sitio.
- **Versátil:** Es utilizado por muchas empresas y organizaciones a lo largo del mundo para crear diferentes tipos de proyectos.

2.4.1.1. Modelo-Vista-Template

El Modelo-Vista-Template (MTV) es el patrón de diseño que Django implementa. Este patrón es una modificación al conocido Modelo-Vista-Controlador (MVC). La diferencia radica en que Django se encarga de hacer la parte del controlador, por ende el desarrollador solamente tiene que preocuparse por implementar la lógica de negocio y de como mostrará los datos. En la Figura 2.13, se puede ver una representación del patrón MTV. En el patrón de diseño MTV [24]:

- **Modelo:** La capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- **Template:** La capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.
- **Vista:** La capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelos y las plantillas.

2.4.2. API REST

2.4.2.1. Application Programming Interface

Una Interfaz de Programación de Aplicaciones o API es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones.

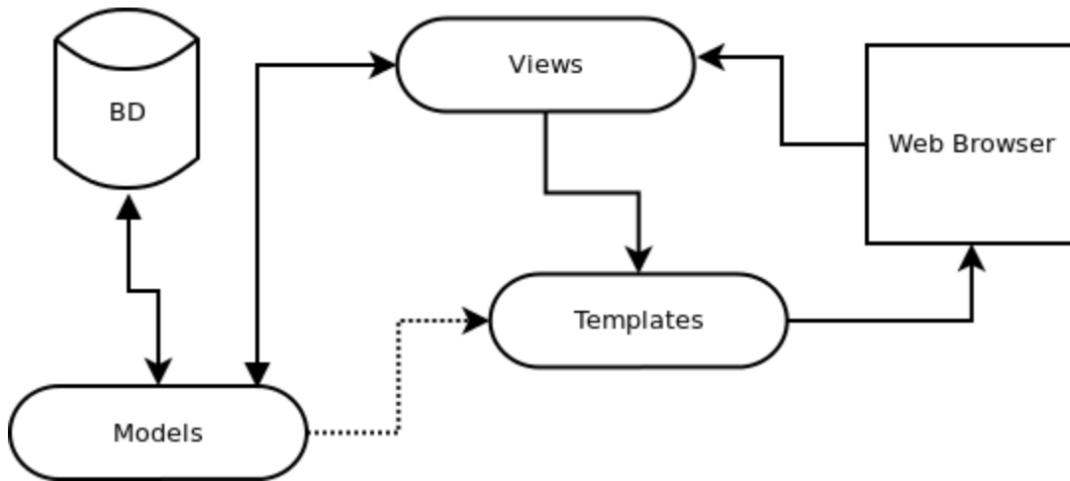


Figura 2.13: Representación del patrón de diseño Modelo-Vista-Template.

Las API permiten que sus servicios se comuniquen con otros, sin necesidad de saber como están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero [25].

2.4.2.2. Represenational State Transfer

Representational State Transfer (REST) es un estilo de arquitectura basado en un conjunto de principios que describen como recursos interconectados son definidos y direccionados. Estos principios fueron descritos en el año 2000 por Roy Fielding como parte de obtención de su doctorado.

Es importante hacer énfasis en que REST es un **estilo de software de arquitectura** y no un conjunto de estandares. Como resultado, dichas aplicaciones o arquitecturas son en ocasiones referidas como aplicaciones RESTful o REST-style [26].

Una aplicación o arquitectura considerada RESTful o REST-style es caracterizada por:

- La funcionalidad y estado son divididas en recursos distribuidos.
- Cada recurso es únicamente direccionable usando un conjunto uniforme y mínimo de comandos (típicamente usando comandos HTTP de GET, POST, PUT o DELETE).
- El protocolo es cliente/servidor, stateless, por capas y soporta almacenamiento cache.

La figura 2.14 ilustra el uso de REST para servicios Web.



Figura 2.14: Consumidor y proveedor de servicios comunicandose mediante solicitudes y respuestas REST.

2.4.2.3. Autentificación

Para hablar de autentificación es necesario primero entender la diferencia entre identificación y autentificación, por un lado la identificación es la capacidad de identificar de forma exclusiva a un usuario de un sistema o una aplicación que se está ejecutando, mientras que la autentificación es la capacidad de demostrar que un usuario o una aplicación es quien dicha persona o aplicación asegura ser [27].

Para el caso de una REST API es en muchas ocasiones necesario que se lleve a cabo la autentificación para permitir o denegar el acceso a recursos de un servidor por ejemplo de acuerdo a los permisos concedidos en función de las credenciales de autentificación para así asegurar que los datos sean visibles únicamente a aquellos que proporcionen las credenciales adecuadas y dispongan de los permisos necesarios.

2.5. Aplicación Android

2.5.1. Arquitectura Clean

El escribir código de buena calidad puede resultar difícil por lo cual es necesario seguir ciertas reglas para lograr esto, es por eso que se utiliza la arquitectura Clean en android.

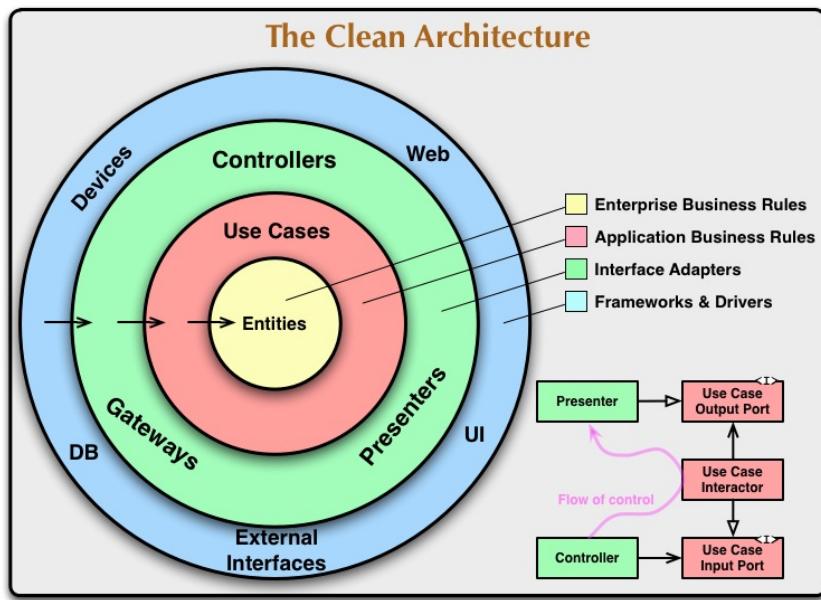


Figura 2.15: Arquitectura Clean [1]

La arquitectura Clean como se muestra en la figura 2.15 fue descrita por Robert C. Martin en 2012 y al utilizarla se puede llegar a tener las siguientes características en los sistemas [1]:

- Independientes de frameworks. El sistema no depende de las librerías o frameworks que se utilicen por lo que estas son fácilmente reemplazables.
- Que se puedan probar. Las reglas de negocio se pueden probar sin base de datos, interfaz de usuario, servidor web u otro elemento externo.
- Independientes de la interfaz de usuario. La interfaz de usuario puede cambiar fácilmente.
- Independientes de la base de datos. Las reglas de negocio no están ligadas a la fuente de datos por lo cual esta puede cambiar.
- Independencia de cualquier agente externo. Las reglas de negocio no dependen de otras capas, por lo cual se vuelve la parte más importante de la arquitectura.

La forma en la que se estructura la arquitectura permite seguir la regla de dependencia en la cual capas internas no deben conocer nada sobre las capas externas.

2.5.2. Componentes

2.5.2.1. Entidades

Son objetos de negocio de la aplicación, por lo cual en esta capa se tienen las reglas de negocio y también se tienen DTOs.

2.5.2.2. Casos de uso

En esta capa se ejecutan las reglas de negocio del sistema, se implementan todos los casos de uso que se tienen. Un caso de uso tiene que recibir datos estructurados y devolver datos estructurados por lo cual esta capa no se debe de ver afectada por capas superiores.

2.5.2.3. Adaptadores

La información que viene de los casos de uso y de las entidades se transforma en esta capa a algo que pueda ser entendido por la capa externa la cual puede ser una conexión a base de datos, una interfaz de usuario o un servidor web.

2.5.2.4. Frameworks y controladores

Esta capa externa está compuesta por frameworks y herramientas tales como la base de datos, una interfaz de usuario, un cliente HTTP o un framework web.

2.5.3. SOLID

SOLID es un acrónimo que hace referencia a los cinco principios de la programación orientada a objetos, este acrónimo fue descrito por Robert C. Martin el seguir estos conceptos permite tener software que sea escalable y fácil de mantener, estos principios están ligados con la alta cohesión y bajo acoplamiento en el software. [28]

S-Responsabilidad simple Cada clase debe de tener una sola responsabilidad. De no seguir esto puede generar el problema de que alguna clase tenga comportamiento que nada tiene que ver con ella debido a que dicho comportamiento no se aisló en otra clase diferente. [29]

O-Abierto/Cerrado Las clases, métodos y módulos deben de ser abiertos a la extensión pero cerrados en su modificación. Esto implica no reescribir código que ya se tiene si no crear nuevo código que haga uso de lo que ya se tiene desarrollado, una forma de hacer esto es hacer uso de la herencia o utilizar interfaces.[29]

L-Sustitución Liskov Las subclases nunca deben de romper la definición de la clase padre. Cuando se utiliza una clase y existen clases que heredan de dicha clase debe de ser posible el utilizar una de estas clases en lugar de la clase padre.[29]

I-Segregación de la interfaz Si algún método de una interfaz no es utilizado no se debe de obligar a tener que implementarlo. Eso indica que las interfaces deben de ser bastante específicas para no tener métodos innecesarios por lo cual es preferible tener muchas interfaces con pocos métodos a pocas interfaces con pocos métodos que no se utilicen.[29]

D-Inversión de dependencias Los módulos de alto nivel no deben de depender de los de bajo nivel. Ambos deben depender de abstracciones, a su vez, las abstracciones no dependen de los detalles sino al contrario. Con esto se logra que las clases no estén totalmente acopladas debido a que en caso contrario es difícil de mantener. [29]

Capítulo 3

Análisis del sistema

3.1. 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.

3.1.1. 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 deben 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 envío de un correo electrónico para la confirmar dicha validez y hacer uso del resto de funcionalidad del sistema. Esta verificación solo se podrá 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 envío 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.

3.1.2. Módulo de proyectos

RF5. Mecanismo para la gestión de proyectos de L^AT_EX. 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 L^AT_EX que se haya traducido. Proporcionar al usuario la funcionalidad de generar un archivo L^AT_EX 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 L^AT_EX 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.

3.1.3. 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.

3.1.4. Módulo de traducción

RF14. Mecanismo para la traducción a L^AT_EX 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 L^AT_EXque pueda ser interpretado por un compilador.

3.2. Requerimientos No Funcionales

Requerimiento No Funcional	Descripción	Métricas asociadas
RNF1 Seguridad	<p>La seguridad se incrementa 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.</p>	<p>Al utilizar el framework de desarrollo Django se cubren los aspectos de seguridad como vulnerabilidades tales como SQL injection, CSRF, CSS, Clickjacking y Session Hijacking, ataques JWT son manejados por autenticación por acceso limitado con el uso de oauth2 que define 4 tipos de autorización:</p> <ul style="list-style-type: none"> ■ Código de autorización: usado con aplicaciones del lado del servidor ■ Implícito: utilizado con aplicaciones móviles o aplicaciones web (aplicaciones que se ejecutan en el dispositivo del usuario). ■ Credenciales de contraseña del propietario del recurso: utilizado con aplicaciones confiables, como aquellas pertenecientes al servicio ■ Credenciales del cliente: Usadas con el acceso API de aplicaciones.

3.2 Requerimientos No Funcionales

RNF2 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.	<p>La aplicación android y web fueron desarrollados utilizando material design que respalda con métricas de Actitud y Comportamiento de User Experience (UX) tales como:</p> <ul style="list-style-type: none"> ■ Abandonment Rate ■ Usability ■ Appearance ■ credibility
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.	La escalabilidad puede llevarse a cabo mediante los proveedores de servicios de Cloud que implementan el denominado Auto Scaling que es un sub-sistema propio de cada servicio de Cloud con la función de ajustar automáticamente la capacidad para mantener un desempeño predecible y estable, sin mencionar que también se dispone de medidores de recursos gráfico.
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	Actualmente los proveedores de servicios de Cloud ofrecen una disponibilidad que se mide como "Monthly Uptime Percentage" que se calcula al sustraer del 100% el porcentaje de minutos durante el mes se encuentra en estado Region No Disponible" garantizando un porcentaje mayor al 99 % mensual pero menor al 99.9 %.

3.3. Requerimientos técnicos

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

3.3.1. Aplicación móvil

3.3.1.1. Requerimientos mínimos de software

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

3.3.1.2. Requerimientos mínimos de hardware

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

3.3.2. Aplicación web

3.3.2.1. 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

3.3.2.2. Requerimientos mínimos de hardware

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

3.4. Reglas de Negocio

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

3.4.1. 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.

3.4.2. 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.

3.4.3. RN-003 Unidad 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}

3.4.4. 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:

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

3.4.5. 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.

3.4.6. RN-006 Usuario verificado

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

3.4.7. 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.

3.5. 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 3.1 muestra los resultados de nuestro análisis.

Área de impacto	Nivel de impacto	Causas	Métodos para contrarrestar el riesgo
Funcionalidad de la aplicación.	Alto	<ul style="list-style-type: none"> ■ 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	<ul style="list-style-type: none"> ■ 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	<ul style="list-style-type: none"> ■ 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	<ul style="list-style-type: none"> ■ 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 3.1: Análisis de riesgos.

3.6. Descripción del software

3.6.1. 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 3.2 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 [30]	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 [30]	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 3.2: 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 sería 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 optó por desarrollar el trabajo para la plataforma de Android.

3.6.2. 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 decidió utilizar una base de datos relacional. La Tabla 3.3 muestra una comparativa de los tres principales gestores de bases de datos [31].

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 3.3: 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.

3.6.3. 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 3.4 muestra una comparativa entre los principales frameworks en la industria [32].

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, librerías 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 <i>clickjacking</i> . Excelente en el manejo de autenticación de usuarios y permisos. Actualizaciones constantes.	Provee métodos contra inyección SQL, ataques XSS, CSRF y <i>clickjacking</i> .	Es vulnerable a ataques, provee métodos de autenticación y hashing.

Tabla 3.4: 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 librerías 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.

3.7. Delimitación de expresiones matemáticas

Debido a la elección del WAP [22] para atacar el problema de reconocimiento y traducción de las expresiones, 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 [33] se tomó un análisis de símbolos existentes en dicho conjunto de entrenamiento realizado con ayuda de la herramienta **CROHME Data Extractor** [2] el conteo de apariciones de símbolos arroja las siguientes gráficas de frecuencias clasificando 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.

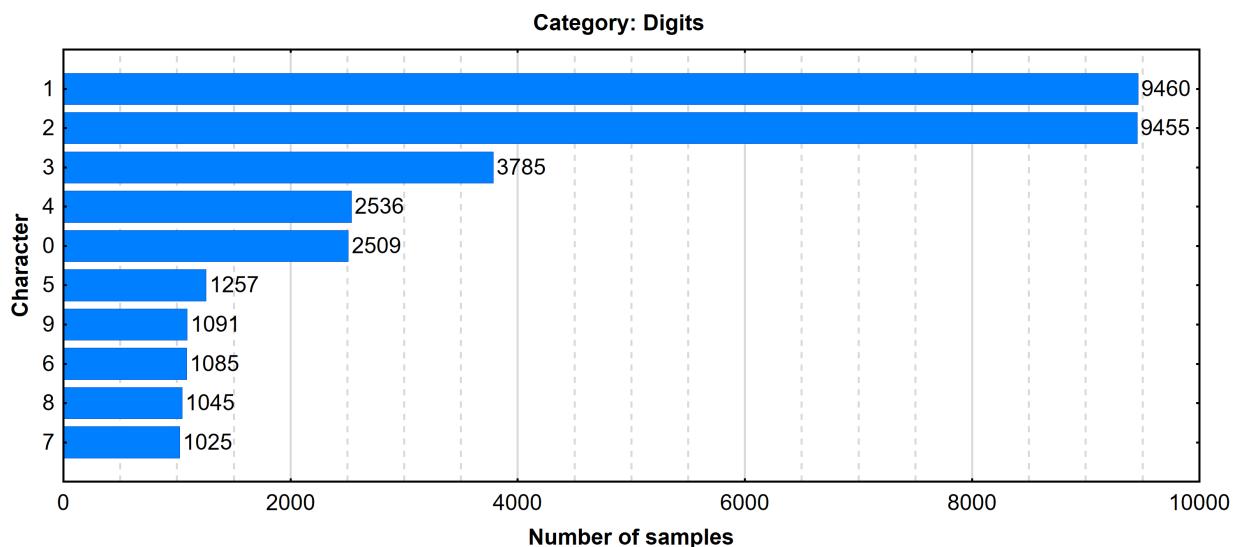


Figura 3.1: Frecuencia de aparición de dígitos del sistema de numeración arábigo [2]

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

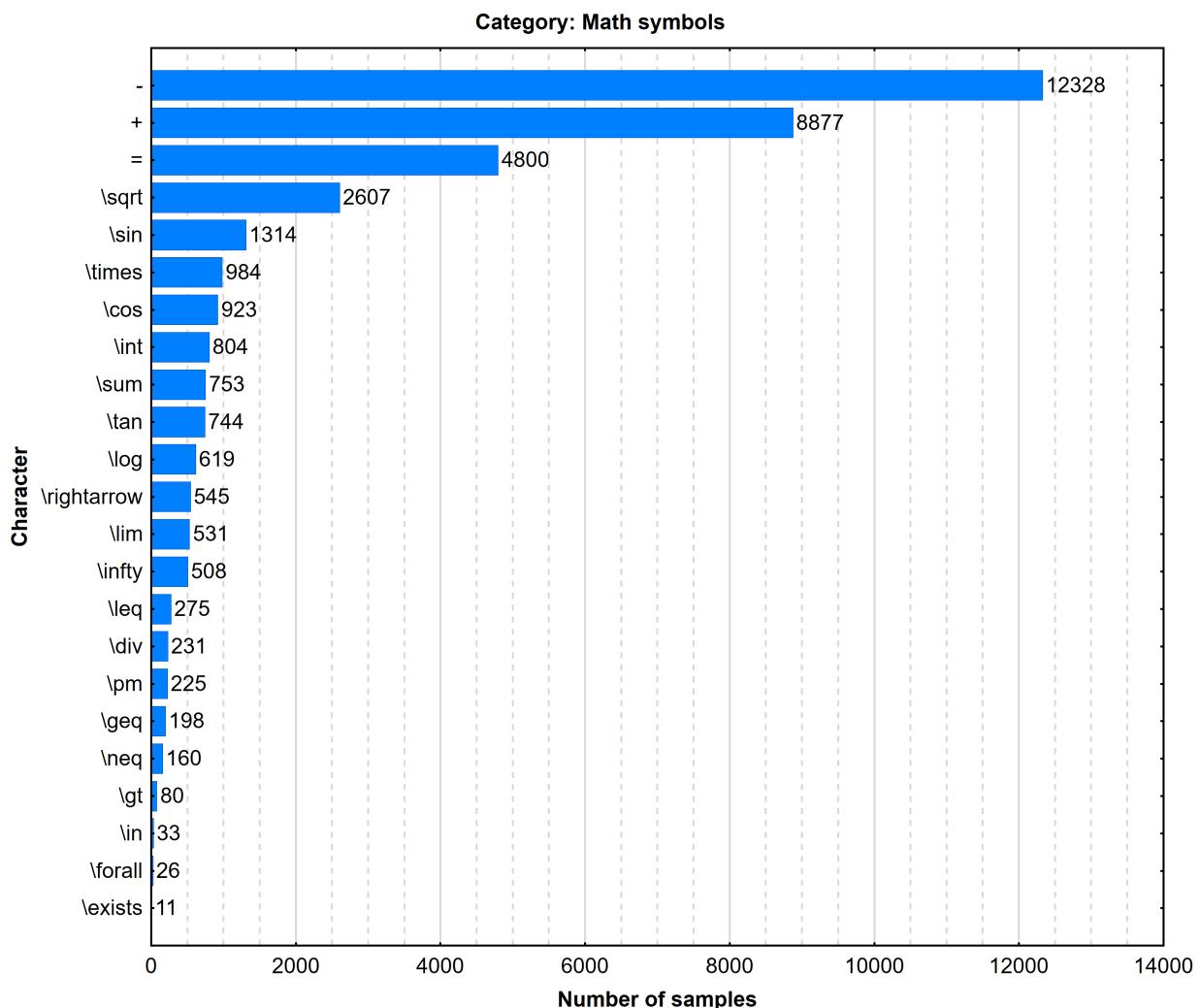


Figura 3.2: Frecuencia de aparición de símbolos matemáticos [2]

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.

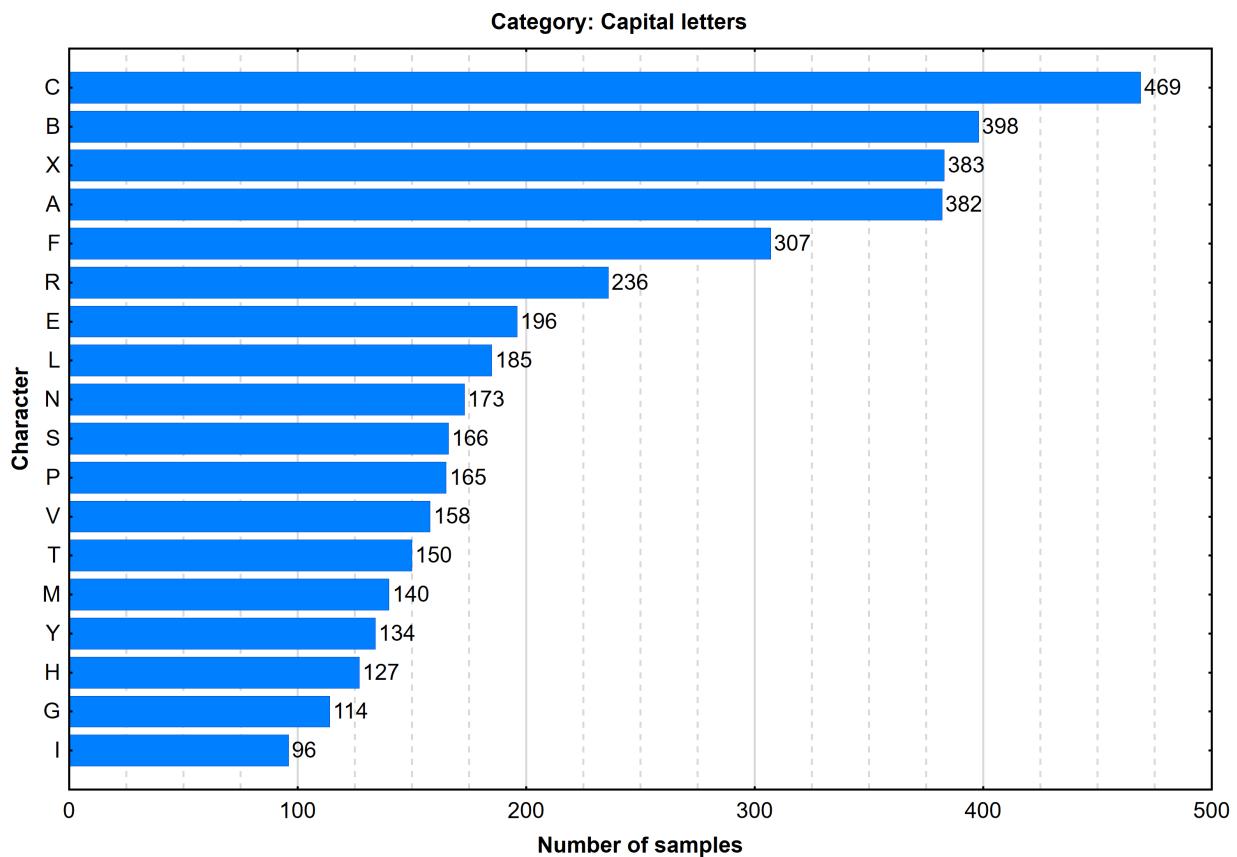


Figura 3.3: Frecuencia de aparición de letras mayúsculas [2]

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

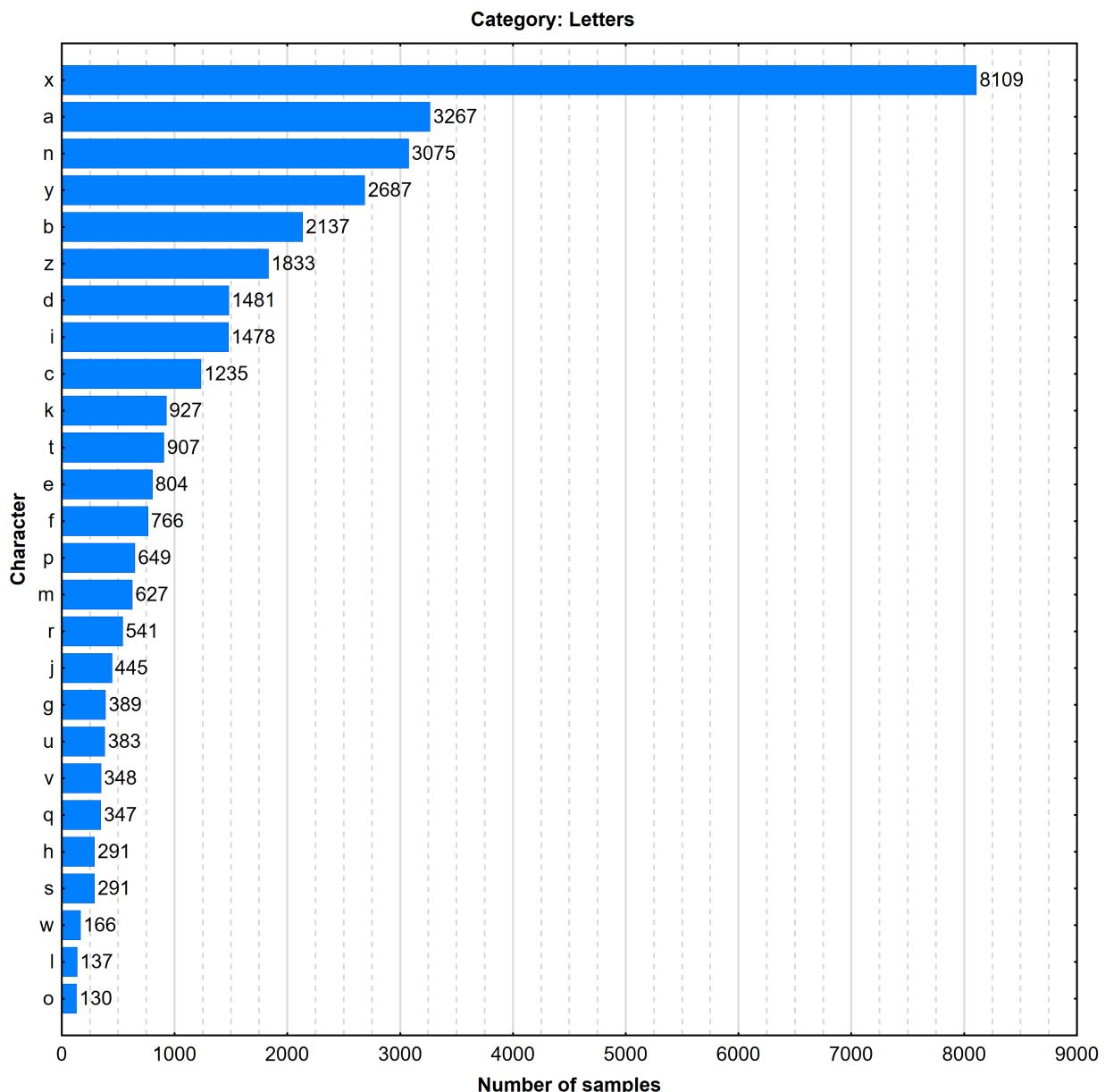


Figura 3.4: Frecuencia de aparición de letras minúsculas [2]

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

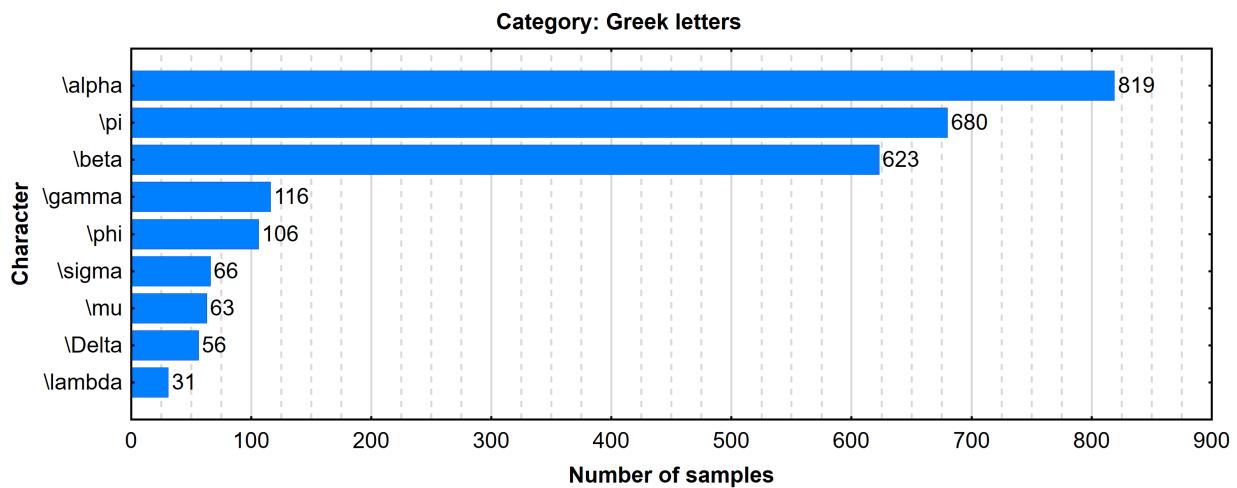


Figura 3.5: Frecuencia de aparición de letras griegas [2]

Otros caracteres que aparecen en el conjunto de entrenamiento se muestran en la figura 3.6 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.

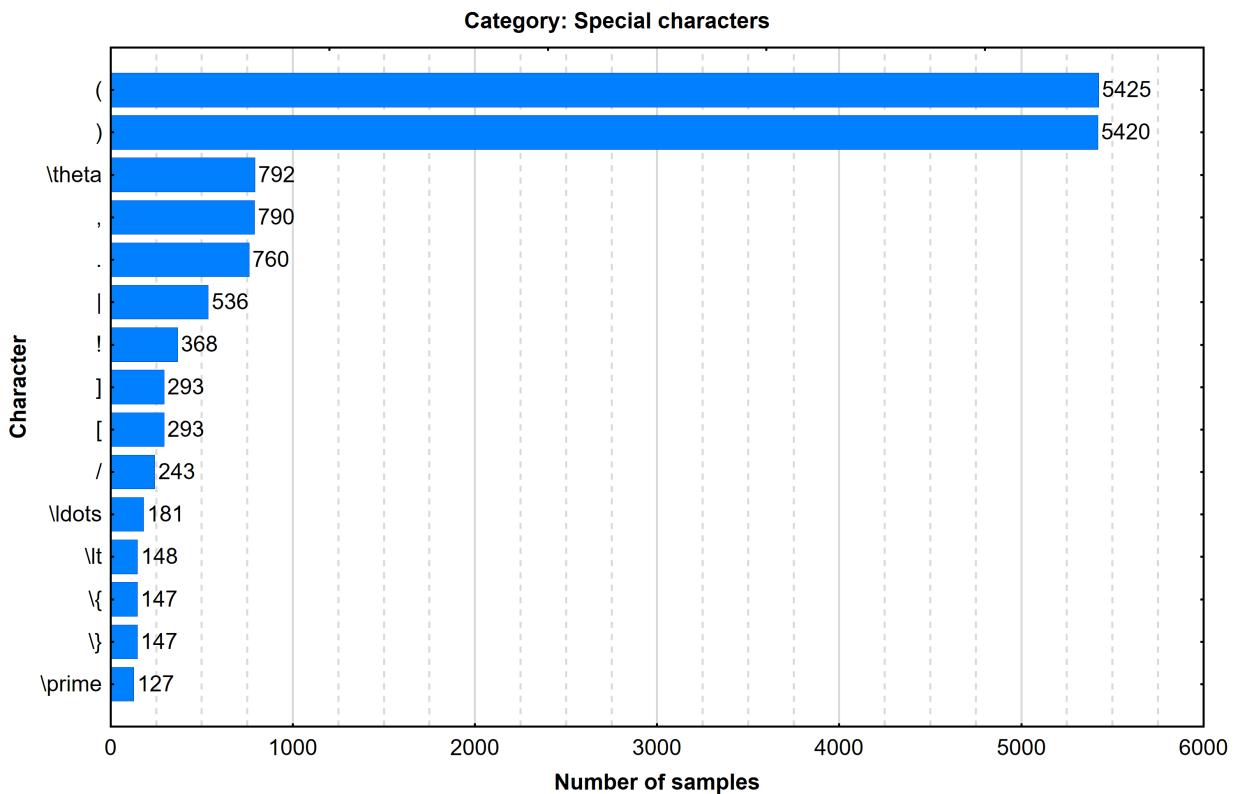


Figura 3.6: Frecuencia de aparición de símbolos especiales [2]

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.

3.7.1. 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 [22] con el mismo conjunto de entrenamiento. La figura 3.7 muestra un ejemplo del conjunto de entrenamiento CROHME.

$$\begin{array}{c} pl^t \\ \hline 1 - (1 - p)l^t \end{array}$$

Figura 3.7: Expresión de dos niveles vertical

3.8. Traducción de imágenes a LATEX

El problema de traducir imágenes a LATEX automáticamente, es aún un campo de investigación abierto. Dada su naturaleza recursiva multivel, 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 [22], 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 realizarán 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 gramáticas 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 [22], 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 3.8 podemos ver un diagrama de esta arquitectura.

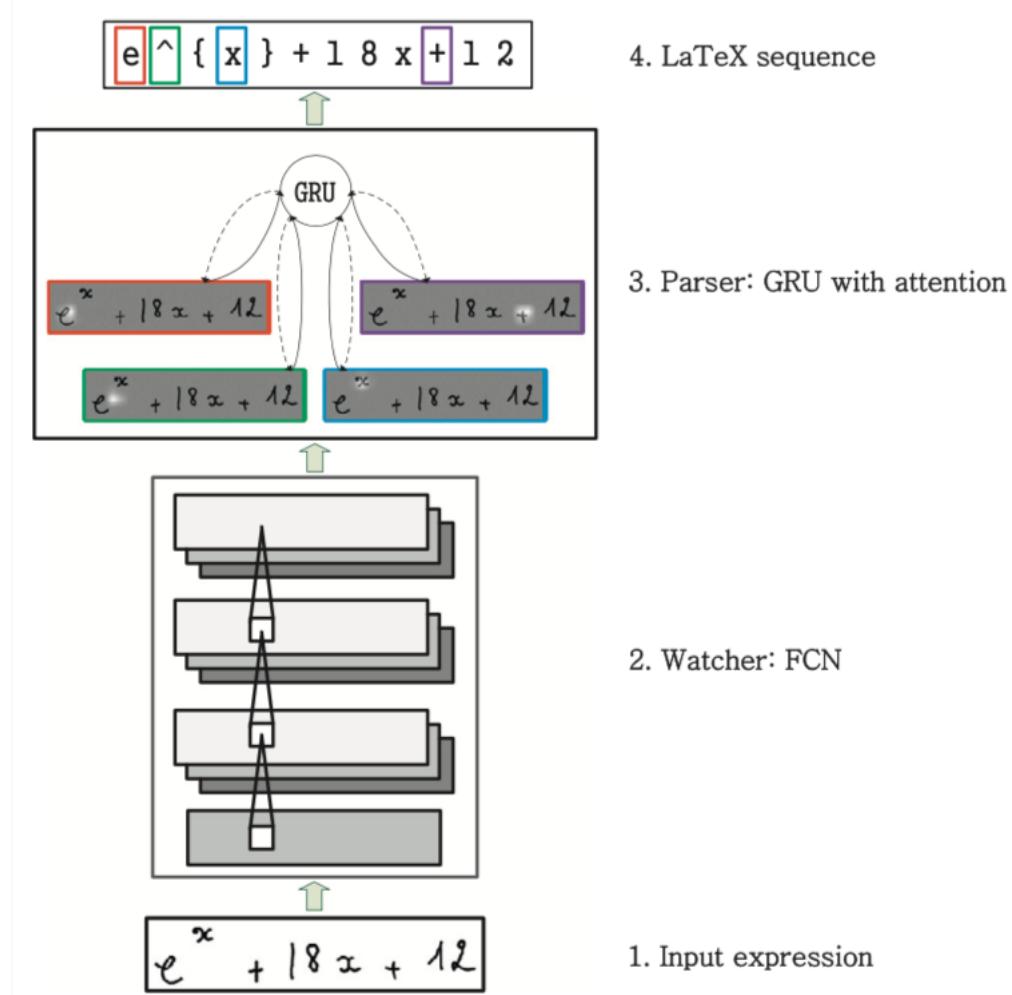


Figura 3.8: Diagrama de la arquitectura Watch, Attend, Parse (WAP).

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 L^AT_EX, un símbolo a la vez.

Capítulo 4

Diseño del sistema

En este capítulo se detallara en el como se desarrollara el sistema, incluyendo una descripción de la arquitectura del sistema, la explicación del modelo de datos con el cual se trabajara. Se detallan los casos de uso que indican el como interactua el usuario con cada modulo del sistema, así como las interfaces respectivas de la aplicación web y android.

4.1. Arquitectura del sistema

En la figura 4.1 se muestra el diagrama de componentes que ilustra el como esta compuesta la arquitectura del sistema con cada uno de sus respectivos módulos.

4.1.1. Cliente web

En el cliente web se tienen las páginas web desarrolladas con el motor de templates de Django, además, para mejorar la interacción que tiene el usuario con el sistema se incluyen recursos como lo son scripts de Javascript para el control de acciones y los archivos de estilos CSS.

4.1.2. Cliente android

En la parte de Android se tiene un cliente REST para llevar a cabo las peticiones y controlar las respuestas del servidor web, se tiene las actividades y plantillas XML para la interfaz de usuario junto con los casos de uso que controlan la lógica del sistema.

4.1.3. Servidor web

En el servidor web es donde se tendrán los componentes para conectar el resto de módulos entre si, es decir, por un lado tenemos el presentador de vistas de Django para controlar al cliente

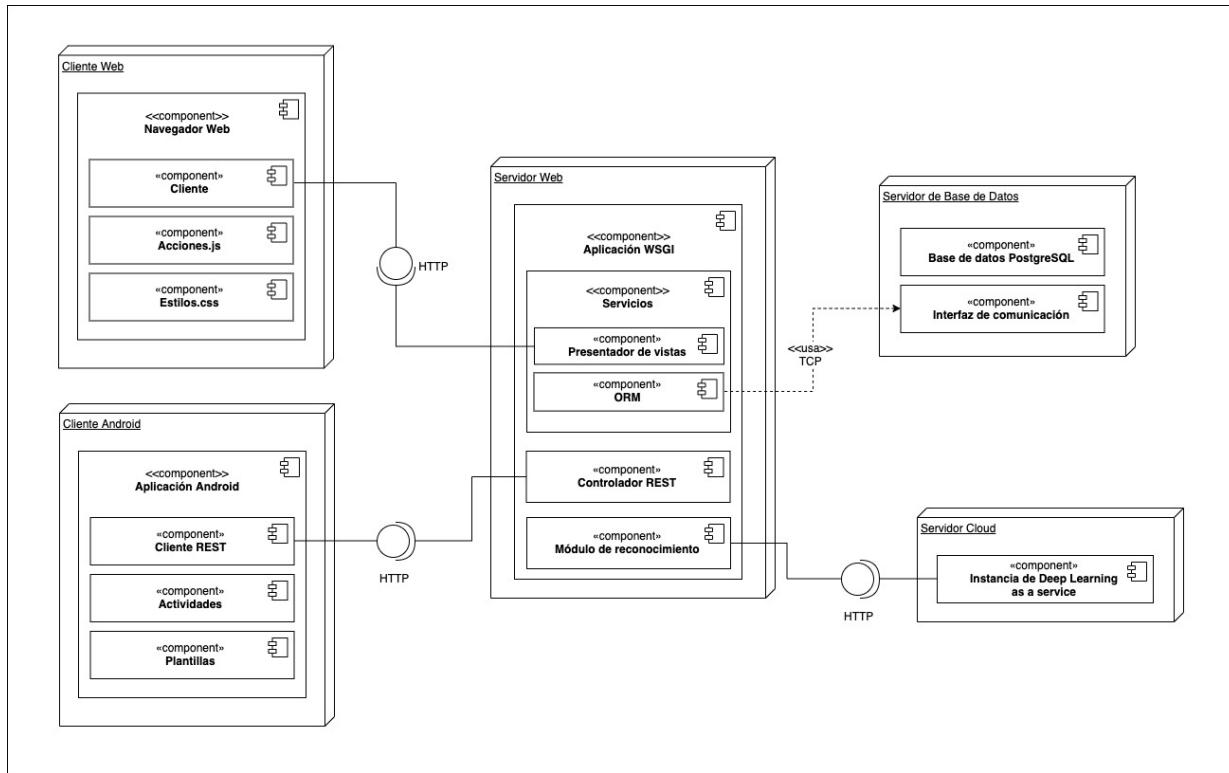


Figura 4.1: Arquitectura del sistema.

web, para la interacción con el servidor web se utiliza directamente el ORM de Django para evitar lidiar con querys y seguir con un desarrollo orientado a objetos, se tienen los serializadores para la interacción con el cliente REST de a Android que también hace uso de los modelos de Django y que en conjunto controlan las peticiones y resultados del API REST, por último se tiene el módulo de reconocimiento que de forma más precisa funciona como un cliente que consume y realiza peticiones al servidor web con el modelo para el reconocimiento y traducción de expresiones matemáticas.

4.1.4. Servidor de base de datos

El sistema gestor de base de datos se encontrará en su propio servidor para separarlo de del servidor web, se desarrollará en postgreSQL y con ello permitir que el sistema este mejor modularizado y tener una independencia en capa complemento de la arquitectura.

4.1.5. Servidor cloud

Servidor cloud es en dónde se realizará la tarea de realizar la traducción a L^AT_EX mediante el uso de redes neuronales cómo se estableció con anterioridad, el utilizar este tipo de servicio que

proporciona Amazon, Google o Microsoft permite evitar lidiar con problemas de infraestructura y despliegue. Además hace que el entrenamiento sea más sencillo debido a los recursos que nos proporcionan.

4.2. Modelo de datos del sistema

En la figura 4.2 se aprecia el modelo de datos del sistema y se describen las entidades que lo conforman, así como los respectivos atributos de cada una de estas entidades.

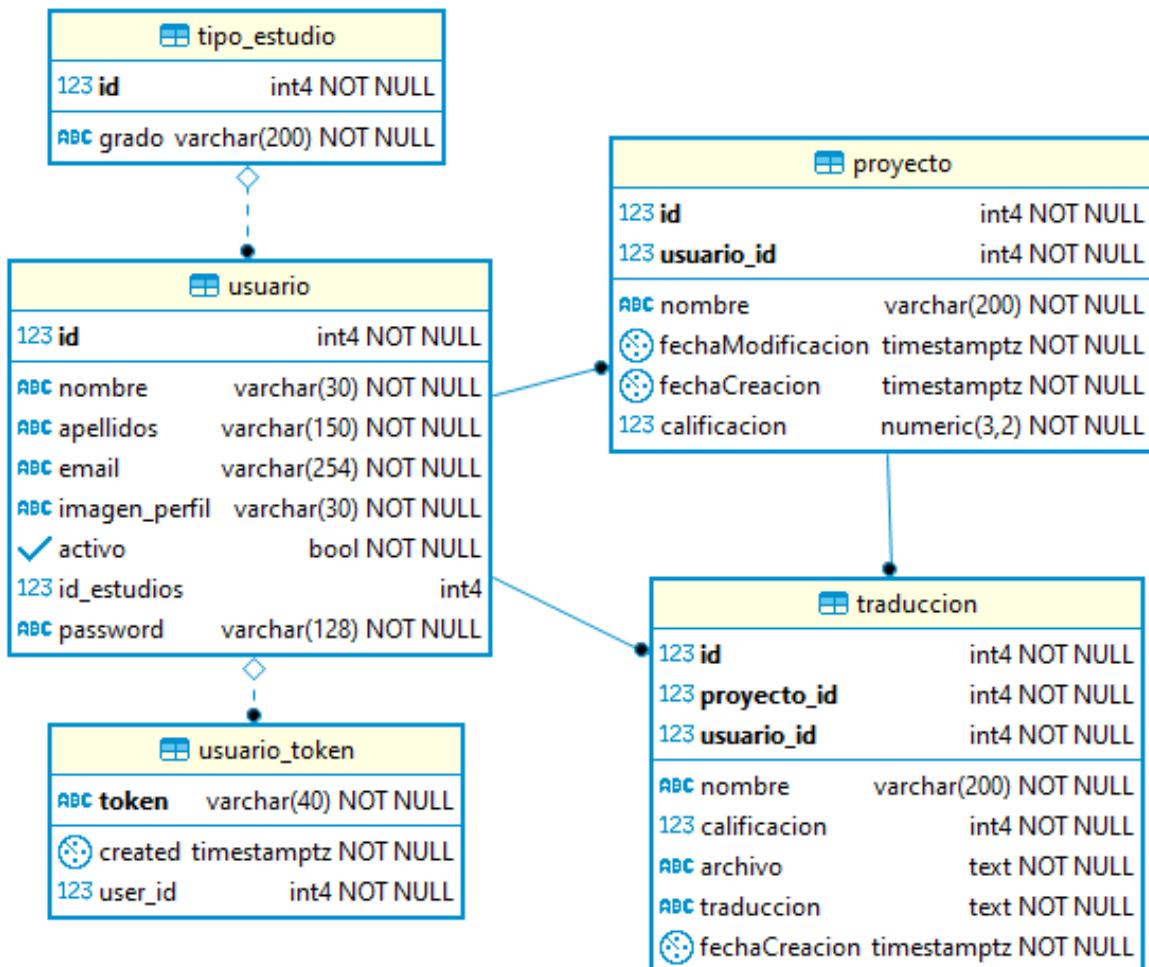


Figura 4.2: Modelo de datos del sistema.

4.2.1. Entidad: Usuario

Se refiere a la cuenta que un usuario puede tener, es la forma en la que accede al sistema.

4.2.1.1. Atributos

Nombre	Descripción	Tipo	Requerido	Único
id	Llave primaria autoincrementable del usuario	integer	Sí	Sí
nombre	Nombre que tiene un usuario	varchar(30)	Sí	No
apellidos	Apellidos de un usuario	varchar(150)	Sí	No
email	Correo electrónico asociado a un usuario	varchar(254)	Sí	Sí
imagen_perfil	Ruta relativa de la ubicación de la imagen de perfil del usuario	varchar(30)	Sí	No
activo	Bandera que nos permite saber si un usuario ha verificado su cuenta	boolean	Sí	No
id_estudios	Identificador para relacionar al usuario con el grado de estudio que tiene del catalogo de tipos de estudios	integer	No	No
password	Contraseña cifrada del usuario	varchar(128)	Sí	No

Tabla 4.1: Tabla de los atributos de la entidad usuario

4.2.2. Entidad: Proyecto

Se refiere a los proyectos que están asociados a un usuario, un proyecto esta compuesto por varias traducciones.

4.2.2.1. Atributos

Nombre	Descripción	Tipo	Requerido	Único
id	Identificador autoincrementable del proyecto, forma parte de la llave primaria	integer	Sí	Sí

usuario_id	Identificador del usuario que tiene el proyecto, forma parte de la llave primaria	integer	Sí	No
nombre	Nombre del proyecto	varchar(200)	Sí	No
fechaModificacion	Fecha que se actualiza cada vez que se modifica el proyecto	timestamp	Sí	No
fechaCreacion	Fecha en la cual se crea el proyecto	timestamp	Sí	No
calificacion	Calificación que se calcula a través de las calificaciones de las traducciones asociadas con el proyecto, puede contener decimales	numeric	Sí	No

Tabla 4.2: Tabla de los atributos de la entidad proyecto

4.2.3. Entidad: Traducción

Hace referencia a la traducción de L^AT_EX que forma parte de un proyecto y que pertenece a un usuario. Esta traducción es el resultado de analizar una imagen.

4.2.3.1. Atributos

Nombre	Descripción	Tipo	Requerido	Único
id	Identificador autoincrementable de la traducción, forma parte de la llave primaria	integer	Sí	Sí
proyecto_id	Identificador del proyecto al cual pertenece la traducción, forma parte de la llave primaria	integer	Sí	No
usuario_id	Identificador del usuario, forma parte de la llave primaria	integer	Sí	No

nombre	Nombre que se le da a la traducción	varchar(200)	Sí	No
calificacion	Calificación que se le da a la traducción para retroalimentación del usuario y del sistema	integer	Sí	No
fechaCreacion	Fecha en la cual se crea la traducción	timestamp	Sí	No

Tabla 4.3: Tabla de los atributos de la entidad traducción

4.2.4. Entidad: Tipo estudios

Hace referencia a los distintos grados de estudios que tiene un usuario.

4.2.4.1. Atributos

Nombre	Descripción	Tipo	Requerido	Único
id	Llave primaria autoincrementable	integer	Sí	Sí
grado	Nombre del grado de estudio	varchar(200)	Sí	No

Tabla 4.4: Tabla de los atributos de la entidad tipo de estudios

4.2.5. Entidad: Usuario Token

Se refiere al token de autenticación que tiene un usuario al momento de crear una cuenta y que su utiliza en la comunicación entre la aplicación web y android.

4.2.5.1. Atributos

Nombre	Descripción	Tipo	Requerido	Único

token	Llave primaria y token de autenticación del usuario para la API REST	varchar(40)	Sí	Sí
created	Fecha de generación del token	timestamp	Sí	No
user_id	Identificador del usuario al que pertenece el token	integer	Sí	No

Tabla 4.5: Tabla de los atributos de la entidad usuario token

4.3. Aplicación de Android

4.3.1. Módulo usuarios

4.3.1.1. Diagrama de casos de uso

En la figura 4.3 se observa el diagrama de casos de uso correspondiente a este módulo.

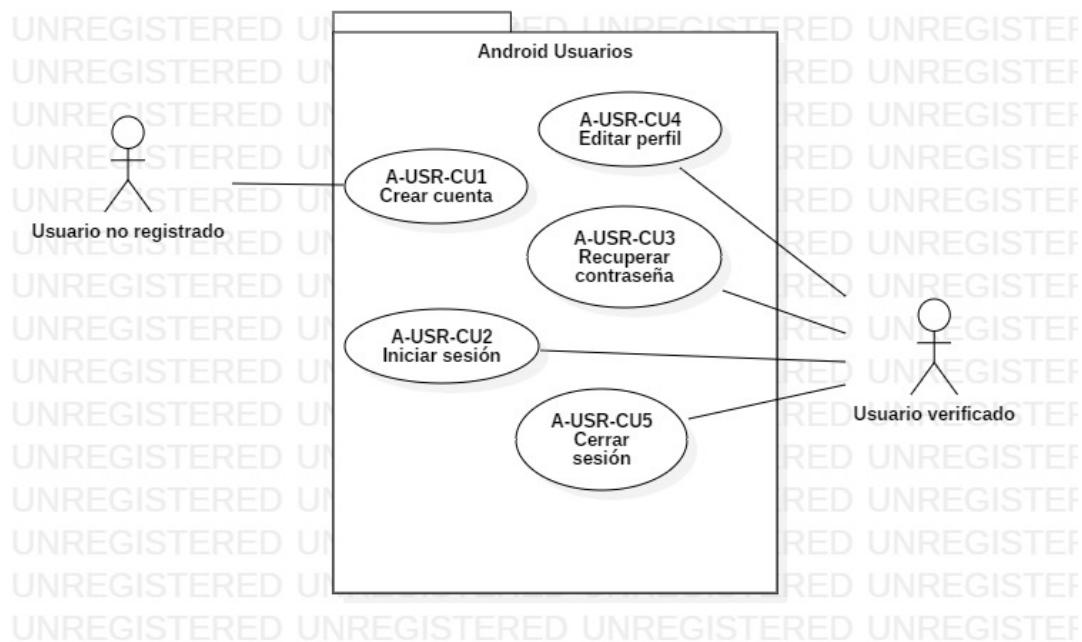


Figura 4.3: Diagrama de casos de uso del módulo Usuarios Android

4.3.1.2. A-USR-CU1 Crear cuenta

Para que un usuario no registrado pueda acceder a la funcionalidad del sistema debe de crear una cuenta utilizando información como su correo electrónico y una contraseña para su posterior autenticación.

Nombre del Caso de Uso	A-USR-CU1 Crear cuenta
Objetivo	Permitir al actor el crear una cuenta
Actores	Usuario no registrado
Disparador	El actor requiere crear una cuenta para ingresar al sistema

Entradas	<ul style="list-style-type: none"> ■ Correo electrónico ■ Contraseña ■ Confirmación de contraseña ■ Nombre
Salidas	<ul style="list-style-type: none"> ■ Correo electrónico para la verificación de la cuenta ■ MSG5 Verifique su cuenta
Precondiciones	Ninguna
Postcondiciones	<ul style="list-style-type: none"> ■ El actor creara una cuenta no verificada
Prioridad	Alta
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-003 Unicidad de identificadores

Trayectoria Principal

1. [Actor] Presiona el botón *Crear cuenta* de la pantalla **MIUA 1 Iniciar Sesión**.
2. [Sistema] Muestra la pantalla **MIUA 2 Registro**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Crear cuenta*.
5. [Sistema] Verifica la información introducida según la regla de negocio RN-001 y RN-002. [Trayectoria alternativa A]
6. [Sistema] Valida la información introducida según la regla de negocio RN-003. [Trayectoria alternativa B]

7. [Sistema] Persiste la información del usuario con estado no verificada.
8. [Sistema] Envía un correo de verificación.
9. [Sistema] Muestra el mensaje **MSG5 Verifique su cuenta** en la pantalla **MIUA 2 Registro**.

Fin del caso de uso.

Trayectoria Alternativa A Datos no validos.

- A1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUA 2 Registro**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El correo electrónico ya se encuentra en uso.

- B1. [Sistema] Muestra el mensaje **MSG4 Correo electronico ya registrado** en la pantalla **MIUA 2 Registro**.

Fin de la trayectoria alternativa.

4.3.1.3. A-USR-CU2 Iniciar sesión

Para poder hacer uso toda la funcionalidad de la aplicación es necesario que el usuario inicie sesión.

Nombre del Caso de Uso	A-USR-CU2 Iniciar sesión
Objetivo	Permitir a un usuario verificado el ingreso al sistema
Actores	Usuario validado
Disparador	El actor requiere iniciar sesión
Entradas	<ul style="list-style-type: none"> ■ Correo ■ Contraseña
Salidas	Ninguna
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de tener una cuenta verificada
Postcondiciones	<ul style="list-style-type: none"> ■ El actor pude hacer uso del resto de funcionalidades de la aplicación
Prioridad	Media
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-006 Usuario verificado

Trayectoria Principal

1. [Actor] Ingresa a la aplicación.
2. [Sistema] Muestra la pantalla **MIUA 1 Iniciar Sesión**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Ingresar*.
5. [Sistema] Valida la información según la regla de negocio RN-001 y RN-002. [Trayectoria alternativa A]

6. [Sistema] Autentica al usuario según la regla de negocio RN-006. [Trayectoria alternativa B]
7. [Sistema] Muestra la pantalla **MIUA 5 Lista de Proyectos.**

Fin del caso de uso.

Trayectoria Alternativa A La información ingresada por el actor no es valida.

- A1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUA 1 Iniciar Sesión.**

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El usuario no ha verificado su cuenta.

- B1. [Sistema] Muestra el mensaje **MSG2 Cuenta no verificada** en la pantalla **MIUA 1 Iniciar Sesión.**

Fin de la trayectoria alternativa.

4.3.1.4. A-USR-CU3 Recuperar contraseña

En caso de que el usuario no recuerde su contraseña se le brinda un mecanismo para poder recuperar la contraseña y con ello poder ingresar a la aplicación

Nombre del Caso de Uso	A-USR-CU3 Recuperar contraseña
Objetivo	Permitir al actor el recuperar su contraseña
Actores	Usuario verificado
Disparador	El actor requiere recuperar su contraseña
Entradas	<ul style="list-style-type: none"> ■ Correo electrónico.
Salidas	<ul style="list-style-type: none"> ■ Correo electrónico para la recuperación de la contraseña ■ MSG6 Envio de correo de recuperación
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de tener una cuenta verificada
Postcondiciones	<ul style="list-style-type: none"> ■ El actor recupera su contraseña
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-006 Usuario verificado

Trayectoria Principal

1. [Actor] Presiona el enlace *Recuperar contraseña* de la **MIUA 1 Iniciar Sesión**.
2. [Sistema] Muestra la pantalla **MIUA 3 Recuperar Contraseña**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Enviar correo de recuperación*.

5. [Sistema] Valida la información introducida según la regla de negocio RN-001 y RN-002. [Trayectoria alternativa A].
6. [Sistema] Verifica la información introducida según la regla de negocio RN-006. [Trayectoria alternativa B].
7. [Sistema] Envía un correo de recuperación.
8. [Sistema] Muestra el mensaje **MSG6 Envío de correo de recuperación** en la pantalla **MIUA 3 Recuperar Contraseña**.

Fin del caso de uso.

Trayectoria Alternativa A La información ingresada por el actor no es valida.

- A1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUA 3 Recuperar Contraseña**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El correo electrónico no se encuentra registrado.

- A1. [Sistema] Muestra el mensaje **MSG3 Correo electronico no registrado** en la pantalla **MIUA 3 Recuperar Contraseña**.

Fin de la trayectoria alternativa.

4.3.1.5. A-USR-CU4 Editar perfil

La posibilidad de modificar los datos de una cuenta es indispensable para un usuario por lo que se proporciona un mecanismo para realizar esta tarea.

Nombre del Caso de Uso	A-USR-CU4 Editar perfil
Objetivo	Permitir al actor editar los datos de su cuenta
Actores	Usuario verificado
Disparador	El actor requiere modificar su información
Entradas	<ul style="list-style-type: none"> ■ Nombre ■ Contraseña ■ Confirmación de contraseña ■ Grado de estudio
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión
Postcondiciones	<ul style="list-style-type: none"> ■ El actor modifica la información de su cuenta
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos

Trayectoria Principal

1. [Actor] Presiona el la opción *Cuenta* del menú **MIUA 4 Menú de Opciones**.
2. [Sistema] Muestra la pantalla **MIUA 7 Editar Perfil**.
3. [Actor] Modifica la información.

4. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación.**
5. [Actor] Presiona el botón *Sí*. [Trayectoria Alternativa A]
6. [Sistema] Valida los datos según la regla de negocio RN-001 y RN-002
7. [Sistema] Persiste la información modificada.
8. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUA 7 Editar Perfil**.

Fin del caso de uso.

Trayectoria Alternativa A Cancela la operación.

- A1. [Sistema] Muestra la pantalla **MIUA 7 Editar Perfil**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B Datos incorrectos.

- B1. [Sistema] Muestra el mensaje **MSG11 Datos no validos** en la pantalla **MIUA 7 Editar Perfil**.

Fin de la trayectoria alternativa.

4.3.1.6. A-USR-CU5 Cerrar Sesión

Este caso de uso permite al usuario cerrar su sesión en la aplicación.

Nombre del Caso de Uso	A-USR-CU5 Cerrar sesión
Objetivo	Cerrar sesión en la aplicación
Actores	Usuario verificado
Disparador	El actor requiere cerrar su sesión
Entradas	Ninguna.
Salidas	Ninguna.
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe haber iniciado sesión
Postcondiciones	<ul style="list-style-type: none"> ■ La sesión del actor se cierra
Prioridad	Baja
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Presiona el botón *Cerrar Sesión* del menú **MIUA 4 Menú de Opciones**.
2. [Sistema] Cierra la sesión del actor en el servidor.
3. [Sistema] Muestra el mensaje **MSG11 Confirmar operación**.
4. [Actor] Presiona el botón *Sí*. [Trayectoria alternativa A]
5. [Sistema] Finaliza la sesión del usuario.
6. [Sistema] Muestra la pantalla **MIUA 1 Iniciar Sesión**.

Fin del caso de uso.

Trayectoria Alternativa A Cancelar la operación.

- A1. [Sistema] Muestra el menú **MIUA 4 Menú de Opciones**.

Fin de la trayectoria alternativa.

4.3.2. Módulo proyectos

4.3.2.1. Diagrama de casos de uso

En la figura 4.4 se observa el diagrama de casos de uso correspondiente a este módulo.

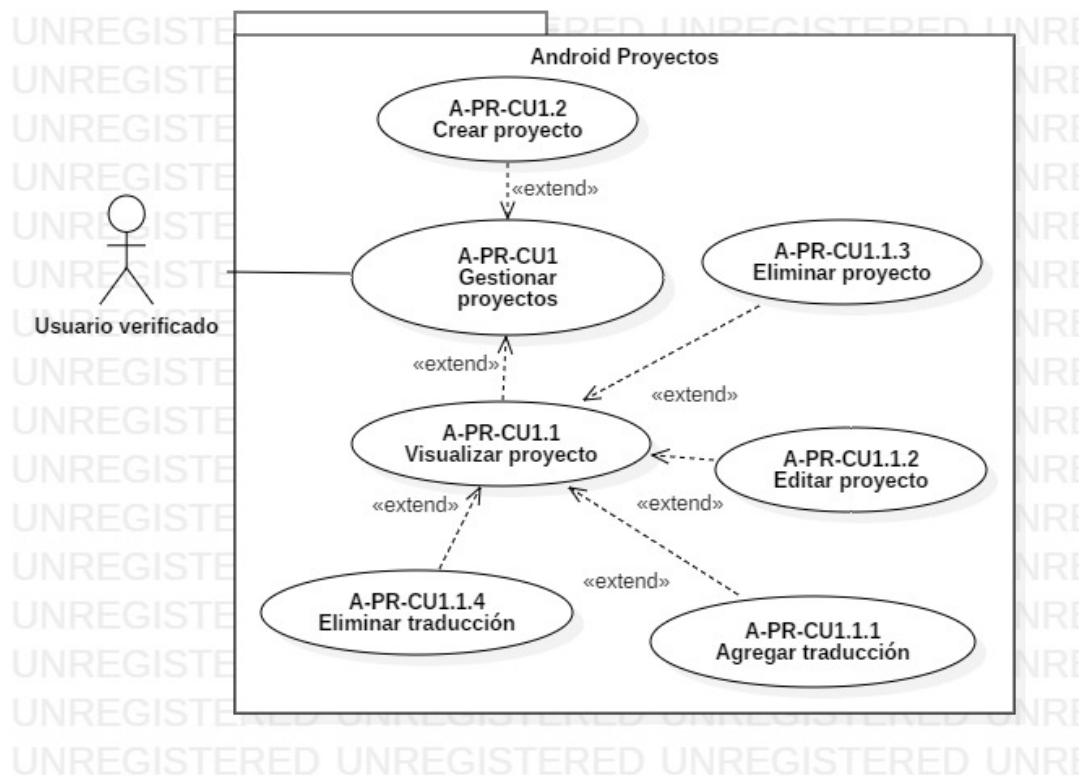


Figura 4.4: Diagrama de casos de uso del módulo Proyectos Android

4.3.2.2. A-PR-CU1 Gestionar proyectos

Se tiene un punto de partida para la gestión de los proyectos en la cual se muestran todos los proyectos que se han creado y la posibilidad de crear nuevos.

Nombre del Caso de Uso	A-PR-CU1 Gestionar proyectos
Objetivo	Permitir al actor el gestionar sus proyectos y crear nuevos
Actores	Usuario verificado
Disparador	El actor requiere trabajar con sus proyectos realizados
Entradas	Ninguna
Salidas	Ninguna

Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión
Postcondiciones	<ul style="list-style-type: none"> ■ El actor pude gestionar sus proyectos o crear uno nuevo
Prioridad	Alta
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Ingresa a la aplicación después de haber iniciado sesión.
2. [Sistema] Recupera todos los proyectos asociados con el actor ordenados del más reciente al más antiguo. [Trayectoria alternativa A]
3. [Sistema] Muestra la pantalla **MIUA 5 Lista de Proyectos** con los proyectos recuperados.

Fin del caso de uso.

Trayectoria Alternativa A No se obtuvo ningún proyecto

- A1. [Sistema] Muestra el mensaje **MSG7 No existen proyectos para mostrar** en la pantalla **MIUA 5 Lista de Proyectos**.

Fin de la trayectoria alternativa.

4.3.2.3. A-PR-CU1.1 Visualizar proyecto

El mostrar información específica sobre los proyectos así como las traducciones que lo componen es algo importante para el usuario.

Nombre del Caso de Uso	A-PR-CU1.1 Visualizar proyecto
Objetivo	Permitir al actor el ver información de algún proyecto en particular
Actores	Usuario verificado
Disparador	El actor quiere gestionar un proyecto en específico
Entradas	Ninguna
Salidas	Ninguna
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión ■ Deben de existir proyectos que visualizar
Postcondiciones	<ul style="list-style-type: none"> ■ El actor puede gestionar un proyecto en específico
Prioridad	Alta
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Selecciona alguno de los proyectos mostrados en la pantalla **MIUA 5 Lista de Proyectos**.
2. [Sistema] Recupera la información asociada al proyecto seleccionado. [Trayectoria alternativa A]
3. [Sistema] Muestra la pantalla **MIUA 6 Lista de Traducciones** con la información recuperada.

Fin del caso de uso.

Trayectoria Alternativa A No se pudo recuperar la información del proyecto.

- A1. [Sistema] Muestra el mensaje **MSG8 No se puede mostrar el proyecto** en la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.3.2.4. A-PR-CU1.1.2 Editar proyecto

Un proyecto tiene asociado un nombre por lo que se debe de modificar de acuerdo a las necesidades que tenga el usuario.

Nombre del Caso de Uso	A-PR-CU1.1.2 Editar proyecto
Objetivo	Permitir al actor editar algún proyecto en particular
Actores	Usuario verificado
Disparador	El actor requiere modificar un proyecto en específico
Entradas	<ul style="list-style-type: none"> ■ Nombre del proyecto
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión ■ Deben de existir proyectos que editar
Postcondiciones	<ul style="list-style-type: none"> ■ Se modifica la información del proyecto
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ ■ RN-005 Fecha de modificación

Trayectoria Principal

1. [Actor] Presiona el ícono *Opciones* de la pantalla **MIUA 6 Lista de Traducciones**.
2. [Sistema] Muestra la opción de *Editar Proyecto*.
3. [Actor] Presiona la opción *Editar Proyecto*.

4. [Sistema] Muestra el mensaje **MSG12 Ingrese el nombre.**
5. [Actor] Ingresa la información requerida.
6. [Actor] Presiona el botón *Aceptar*. [Trayectoria Alternativa A]
7. [Sistema] Verifica la información introducida según la regla de negocio RN-001 y RN-002. [Trayectoria Alternativa B]
8. [Sistema] Calcula la fecha de modificación del proyecto en el sistema según la regla de negocio RN-005.
9. [Sistema] Modifica la información almacenada. [Trayectoria Alternativa C]
10. [Sistema] Muestra el mensaje **MSG8 Operación exitosa** en la pantalla **MIUA 5 Lista de Traducciones**.

Fin del caso de uso.

Trayectoria Alternativa A Cancela la operación.

- A1. [Sistema] Muestra la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B Datos no validos.

- B1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa C No se puede modificar el proyecto.

- C1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.3.2.5. A-PR-CU1.1.3 Eliminar proyecto

Si un proyecto ya no es requerido el actor se puede deshacer de él al eliminarlo y una vez eliminado ya no se podrá recuperar.

Nombre del Caso de Uso	A-PR-CU1.1.3 Eliminar proyecto
Objetivo	Permitir al actor eliminar algún proyecto en particular
Actores	Usuario verificado
Disparador	El actor requiere eliminar un proyecto en específico
Entradas	Ninguna
Salidas	<ul style="list-style-type: none"> ▪ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ▪ El actor debe de haber iniciado sesión ▪ Deben de existir proyectos que eliminar
Postcondiciones	<ul style="list-style-type: none"> ▪ El actor elimina un proyecto en específico.
Prioridad	Baja.
Reglas de Negocio	Ninguna.

Trayectoria Principal

1. [Actor] Presiona el ícono *Opciones* de la pantalla **MIUA 6 Lista de Traducciones**.
2. [Sistema] Muestra la opción de *Eliminar Proyecto*.
3. [Actor] Presiona la opción *Eliminar Proyecto*.
4. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación**.
5. [Actor] Presiona el botón *Sí*. [Trayectoria Alternativa A]
6. [Sistema] Elimina las traducciones asociadas al proyecto del almacenamiento. [Trayectoria Alternativa B]
7. [Sistema] Elimina el proyecto del almacenamiento. [Trayectoria Alternativa B]
8. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUA 5 Lista de Proyectos**.

Fin del caso de uso.

Trayectoria Alternativa A Cancela la operación.

A1. [Sistema] Muestra la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B No se puede eliminar el proyecto o las traducciones.

B1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.3.2.6. A-PR-CU1.1.4 Eliminar traducción

Un proyecto puede tener asociadas varias traducciones por lo que es posible eliminar alguna traducción que no se requiera.

Nombre del Caso de Uso	A-PR-CU1.1.4 Eliminar traducción
Objetivo	Permitir al actor eliminar alguna traducción en particular de algún proyecto
Actores	Usuario verificado
Disparador	El actor requiere eliminar una traducción en específico
Entradas	Ninguna
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión ■ Deben de existir proyectos que eliminar ■ Deben de existir traducciones asociadas a algún proyecto
Postcondiciones	<ul style="list-style-type: none"> ■ Se elimina una traducción
Prioridad	Baja
Reglas de Negocio	Ninguna.

Trayectoria Principal

1. [Actor] Presiona el ícono *Eliminar traducción* de la pantalla **MIUA 6 Lista de Traducciones**.
2. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación**.
3. [Actor] Presiona el botón *Sí*. [Trayectoria Alternativa A].
4. [Sistema] Elimina la traducción del proyecto. [Trayectoria Alternativa B].
5. [Sistema] Muestra el mensaje **MSG8 Operación exitosa** en la pantalla **MIUA 5 Lista de Traducciones**.

Fin del caso de uso.

Trayectoria Alternativa A Cancelar la operación.

A1. [Sistema] Muestra la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B No se puede eliminar la traducción.

B1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUA 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4. Aplicación Web

4.4.1. Módulo usuarios

4.4.1.1. Diagrama de casos de uso

En la figura 4.5 se observa el diagrama de casos de uso correspondiente a este módulo.

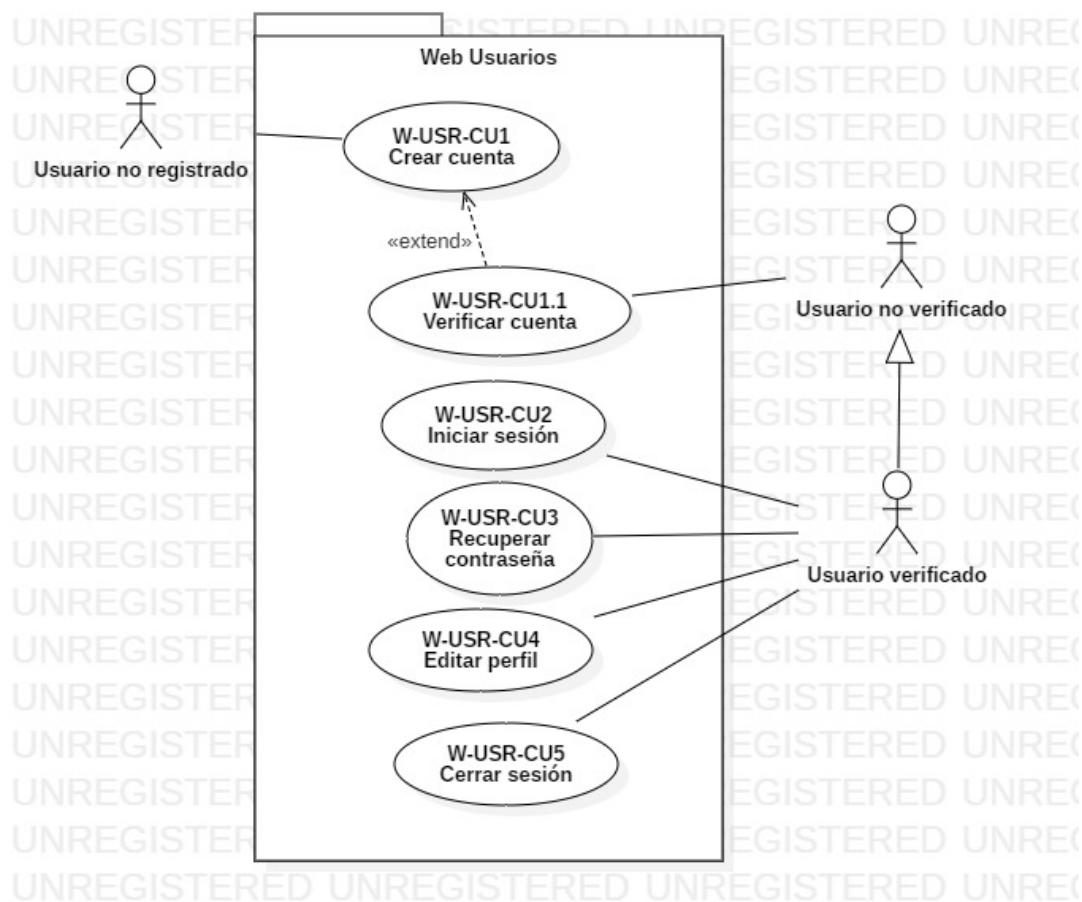


Figura 4.5: Diagrama de casos de uso del módulo Usuarios Web

4.4.1.2. W-USR-CU1 Crear cuenta

Para que un usuario no registrado pueda acceder a la funcionalidad del sistema debe de crear una cuenta utilizando información como su correo electrónico y una contraseña para su posterior autenticación.

Nombre del Caso de Uso	W-USR-CU1 Crear cuenta
Objetivo	Permitir a un usuario no registrado el crear una cuenta
Actores	Usuario no registrado
Disparador	El actor requiere crear una cuenta para ingresar al sistema
Entradas	<ul style="list-style-type: none"> ■ Correo electrónico ■ Contraseña ■ Confirmación de contraseña ■ Nombre
Salidas	<ul style="list-style-type: none"> ■ Correo electrónico para la verificación de la cuenta ■ MSG5 Verifique su cuenta
Precondiciones	Ninguna.
Postcondiciones	<ul style="list-style-type: none"> ■ El actor creara una cuenta no verificada.
Prioridad	Alta.
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-003 Unicidad de identificadores

Trayectoria Principal

1. [Actor] Presiona el botón *Crear cuenta* de la pantalla **MIUW 1 Iniciar Sesión**.
2. [Sistema] Muestra la pantalla **MIUW 2 Registro**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Crear cuenta*.
5. [Sistema] Verifica la información introducida según la regla de negocio RN-001 y RN-002..
[Trayectoria alternativa A]

6. [Sistema] Valida la información introducida. [Trayectoria alternativa B]
7. [Sistema] Persiste la información del usuario con estado cuenta no verificada.
8. [Sistema] Envía un correo de verificación.
9. [Sistema] Muestra el mensaje **MSG5 Verifique su cuenta** en la pantalla **MIUW 1 Iniciar Sesión**.

Fin del caso de uso.

Trayectoria Alternativa A Datos no validos.

- A1. [Sistema] Muestra el mensaje **MSG2 Datos no validos** en la pantalla **MIUW 2 Registro**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El correo electrónico ya se encuentra en uso.

- B1. [Sistema] Muestra el mensaje **MSG4 Correo electrónico ya registrado** en la pantalla **MIUW 2 Registro**.

Fin de la trayectoria alternativa.

4.4.1.3. W-USR-CU1.1 Verificar cuenta

Este caso de uso permite al usuario verificar su cuenta en el sistema.

Nombre del Caso de Uso	W-USR-CU1.1 Verificar cuenta
Objetivo	Completar el registro de un usuario mediante la verificación de su cuenta
Actores	Usuario no verificado
Entradas	Ninguna
Salidas	Ninguna
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus no verificado
Postcondiciones	La cuenta del actor será registrada con el estatus de verificada
Prioridad	Alta.
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Sistema] Verifica el código de autenticación.
2. [Sistema] Cambia el estatus del actor como *Verificado*.
3. [Sistema] Muestra el mensaje MS9 Operación exitosa en la pantalla **MIUW 1 Iniciar Sesión**.

Fin del caso de uso.

4.4.1.4. W-USR-CU2 Iniciar sesión

Para poder hacer uso toda la funcionalidad de la aplicación es necesario que el usuario inicie sesión.

Nombre del Caso de Uso	W-USR-CU2 Iniciar sesión
Objetivo	Permitir a un usuario validado y con cuenta validada el ingreso al sistema
Actores	Usuario verificado
Disparador	El actor requiere iniciar sesión
Entradas	<ul style="list-style-type: none"> ■ Correo ■ Contraseña
Salidas	Ninguna.
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de tener una cuenta verificada
Postcondiciones	<ul style="list-style-type: none"> ■ El actor puede hacer uso del resto de funcionalidades del sistema
Prioridad	Media
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-006 Usuario verificado

Trayectoria Principal

1. [Actor] Ingresa al sitio web.
2. [Sistema] Muestra la pantalla **MIUW 1 Iniciar Sesión**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Iniciar sesión*.

5. [Sistema] Valida la información según la regla de negocio RN-001 y RN-002. [Trayectoria alternativa A]
6. [Sistema] Autentica al usuario según la regla de negocio RN-006. [Trayectoria alternativa B]
7. [Sistema] Muestra la pantalla **MIUW 5 Lista de Proyectos.**

Fin del caso de uso.

Trayectoria Alternativa A La información ingresada por el actor no es valida.

- A1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUW 1 Iniciar Sesión.**

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El usuario no ha verificado su cuenta.

- B1. [Sistema] Muestra el mensaje **MSG2 Cuenta no verificada** en la pantalla **MIUW 1 Iniciar Sesión.**

Fin de la trayectoria alternativa.

4.4.1.5. W-USR-CU3 Recuperar contraseña

En caso de que el usuario no recuerde su contraseña se le brinda un mecanismo para poder recuperar la contraseña y con ello poder ingresar a la aplicación

Nombre del Caso de Uso	W-USR-CU3 Recuperar contraseña
Objetivo	Permitir al usuario verificado el recuperar su contraseña.
Actores	Usuario verificado
Disparador	El actor requiere recuperar su contraseña
Entradas	<ul style="list-style-type: none"> ■ Correo del actor con el que se registro
Salidas	<ul style="list-style-type: none"> ■ Correo electrónico para la recuperación de la contraseña ■ MSG6 Envio de correo de recuperación
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de tener una cuenta verificada
Postcondiciones	<ul style="list-style-type: none"> ■ El actor recupera su contraseña
Prioridad	Baja.
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-006 Usuario verificado

Trayectoria Principal

1. [Actor] Presiona el enlace *Recuperar contraseña* de la **MIUW 1 Iniciar Sesión**.
2. [Sistema] Muestra la pantalla **MIUW 4 Recuperar Contraseña**.
3. [Actor] Ingresa la información solicitada en pantalla.
4. [Actor] Presiona el botón *Enviar correo de recuperación*.

5. [Sistema] Valida la información introducida según la regla de negocio RN-001 y RN-002. [Trayectoria alternativa A].
6. [Sistema] Verifica que el correo se encuentre registrado según la regla de negocio RN-006. [Trayectoria alternativa B].
7. [Sistema] Envía un correo de recuperación.
8. [Sistema] Muestra el mensaje **MSG6 Envío de correo de recuperación** en la pantalla **MIUW 4 Recuperar Contraseña**.

Fin del caso de uso.

Trayectoria Alternativa A Datos incorrectos.

- A1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUW 4 Recuperar Contraseña**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B El correo electrónico introducido no se encuentra registrado.

- B1. [Sistema] Muestra el mensaje **MSG3 Correo electronico no registrado** en la pantalla **MIUW 4 Recuperar Contraseña**.

Fin de la trayectoria alternativa.

4.4.1.6. W-USR-CU4 Editar perfil

La posibilidad de modificar los datos de una cuenta es indispensable para un usuario por lo que se proporciona un mecanismo para realizar esta tarea.

Nombre del Caso de Uso	W-USR-CU4 Editar perfil
Objetivo	Permitir al actor editar los datos de su cuenta.
Actores	Usuario verificado
Disparador	El actor requiere modificar su información
Entradas	<ul style="list-style-type: none"> ■ Nombre ■ Contraseña ■ Confirmación de contraseña ■ Grado de estudio
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión
Postcondiciones	<ul style="list-style-type: none"> ■ El actor modifica la información de su cuenta
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos

Trayectoria Principal

1. [Actor] Presiona el la opción *Cuenta* del menú **MIUW 7 Menú de Opciones**.
2. [Sistema] Muestra la pantalla **MIUW 3 Editar Perfil**.
3. [Actor] Modifica la información.

4. [Actor] Presiona el botón guardar.
5. [Sistema] Valida la información según la regla de negocio RN-001 y RN-002. [Trayectoria Alternativa A]
6. [Sistema] Persiste la información modificada.
7. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 3 Editar Perfil**.

Fin del caso de uso.

Trayectoria Alternativa A Datos incorrectos.

- A1. [Sistema] Muestra el mensaje **MSG11 Datos no validos** en la pantalla **MIUW 3 Editar Perfil**.

Fin de la trayectoria alternativa.

4.4.1.7. W-USR-CU5 Cerrar sesión

Este caso de uso permite al usuario final cerrar su sesión en la aplicación.

Nombre del Caso de Uso	W-USR-CU5 Cerrar sesión
Objetivo	Cerrar sesión en la aplicación
Actores	Usuario verificado
Disparador	El actor requiere cerrar su sesión
Entradas	Ninguna.
Salidas	Ninguna.
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus de verificado ■ El actor debe haber iniciado sesión
Postcondiciones	Se cierra la sesión
Prioridad	Media
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Solicita cerrar sesión oprimiendo el botón *Cerrar Sesión* del menú **MIUW 7 Menú de Opciones**.
2. [Sistema] Cierra la sesión del actor en el servidor.
3. [Sistema] Muestra la pantalla **MIUW 1 Iniciar Sesión**.

Fin del caso de uso.

4.4.2. Módulo proyectos

4.4.2.1. Diagrama de casos de uso

En la figura 4.6 se observa el diagrama de casos de uso correspondiente a este módulo.

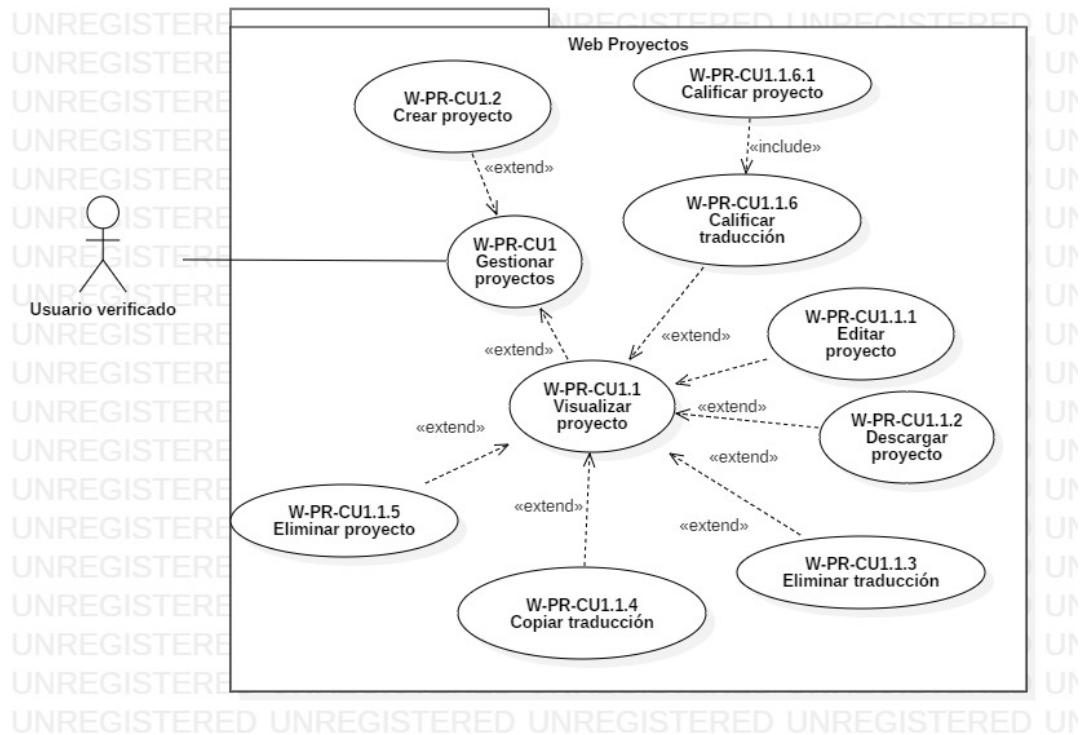


Figura 4.6: Diagrama de casos de uso del módulo Proyectos Web

4.4.2.2. W-PR-CU1 Gestión de proyectos

Se tiene un punto de partida para la gestión de los proyectos en la cual se muestran todos los proyectos que se han creado y la posibilidad de crear nuevos.

Nombre del Caso de Uso	W-PR-CU1 Gestión de proyectos
Objetivo	Permitir al actor el gestionar sus proyectos y crear nuevos
Actores	Usuario verificado
Disparador	El actor requiere trabajar con sus proyectos realizados
Entradas	Ninguna
Salidas	Ninguna

Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión
Postcondiciones	<ul style="list-style-type: none"> ■ El actor pude gestionar sus proyectos o crear uno nuevo
Prioridad	Alta
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Ingresa a la aplicación después de haber iniciado sesión.
2. [Sistema] Recupera todos los proyectos asociados con el actor. [Trayectoria alternativa A]
3. [Sistema] Muestra la pantalla **MIUW 5 Lista de Proyectos** con los proyectos recuperados.

Fin del caso de uso.

Trayectoria Alternativa A No se obtuvo ningún proyecto

- A1. [Sistema] Muestra el mensaje **MSG7 No existen proyectos para mostrar** en la pantalla **MIUW 5 Lista de Proyectos**.

Fin de la trayectoria alternativa.

4.4.2.3. W-PR-CU1.1 Visualizar Proyecto

El mostrar información específica sobre los proyectos así como las traducciones que lo componen es algo importante para el usuario.

Nombre del Caso de Uso	W-PR-CU1.1 Visualizar Proyecto
Objetivo	Permitir al actor el ver información de algún proyecto en particular
Actores	Usuario verificado
Disparador	El actor quiere gestionar un proyecto en específico
Entradas	Ninguna
Salidas	Ninguna
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe de haber iniciado sesión ■ Deben de existir proyectos que visualizar
Postcondiciones	<ul style="list-style-type: none"> ■ El actor puede gestionar un proyecto en específico
Prioridad	Alta
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Selecciona alguno de los proyectos mostrados en la pantalla **MIUW 5 Lista de Proyectos**.
2. [Sistema] Recupera la información asociada al proyecto seleccionado. [Trayectoria alternativa A]
3. [Sistema] Muestra la pantalla **MIUW 6 Lista de Traducciones** con la información recuperada.

Fin del caso de uso.

Trayectoria Alternativa A No se pudo recuperar la información del proyecto.

- A1. [Sistema] Muestra el mensaje **MSG8 No se puede mostrar el proyecto** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.4. W-PR-CU1.1.1 Editar proyecto

Este caso de uso permite al actor cambiar el nombre de algún proyecto previamente creado por él.

Nombre del Caso de Uso	W-PR-CU1.1.1 Editar proyecto
Objetivo	Cambiar la información de algún proyecto
Actores	Usuario verificado
Disparador	El actor requiere cambiar la información de algún proyecto.
Entradas	<ul style="list-style-type: none"> ■ Nombre del proyecto
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus de validado ■ El usuario debe haber iniciado sesión ■ El proyecto debe de pertenecer al actor
Postcondiciones	Los datos del proyecto se modifican
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos ■ RN-005 Fecha de modificación

Trayectoria Principal

1. [Actor] Presiona el botón *Editar* en la pantalla **MIUW 6 Lista de Traducciones**.
2. [Sistema] Muestra un campo de texto con el actual nombre del proyecto en la pantalla **MIUW 6 Lista de Traducciones**.
3. [Actor] Ingresa el nuevo nombre del proyecto.

4. [Actor] Presiona el botón aceptar *Aceptar* de la pantalla **MIUW 6 Lista de Traducciones**.
[Trayectoria alternativa A]
5. [Sistema] Verifica los datos como se especifica en las reglas de negocio RN-001 y RN-002.
[Trayectoria alternativa B]
6. [Sistema] Calcula la fecha de modificación del proyecto en el sistema según la regla de negocio RN-005.
7. [Sistema] Persiste los cambios realizados.
8. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 6 Lista de Traducciones** y actualiza la información mostrada.

Fin del caso de uso.

Trayectoria Alternativa A El actor cancela la operación.

- A1. [Actor] Solicita cancelar la operación oprimiendo el botón *Cancelar* en la pantalla **MIUW 6 Lista de Traducciones**.
- A2. [Sistema] Muestra la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B Datos no validos.

- B1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUW 6 Lista de Traducciones**.
- B2. [Actor] Continua con el paso 3 de la Trayectoria Principal.

Fin de la trayectoria alternativa.

4.4.2.5. W-PR-CU1.1.2 Descargar proyecto

El poder obtener un archivo de LaTeX para su uso es útil para el usuario por lo que se le brinda esta funcionalidad.

Nombre del Caso de Uso	W-PR-CU1.1.2 Descargar proyecto
Objetivo	Generar un archivo de LaTeX para su descarga
Actores	Usuario verificado
Disparador	El actor requiere descargar las traducciones de un proyecto
Entradas	Ninguna
Salidas	Archivo de LaTeX
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe haber iniciado sesión ■ Deben de existir traducciones para descargar el proyecto
Postcondiciones	<ul style="list-style-type: none"> ■ Se genera un archivo LaTeX
Prioridad	Media
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-007 Información necesaria para descargar un proyecto

Trayectoria Principal

1. [Actor] Presiona el botón *Descargar* de la pantalla **MIUW 6 Lista de Traducciones**.
2. [Sistema] Recupera la información necesaria para realizar esta operación según la regla de negocio RN-007. [Trayectoria alternativa A]
3. [Sistema] Genera un archivo LaTeX con las traducciones del proyecto seleccionado. [Trayectoria alternativa B]
4. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin del caso de uso.

Trayectoria Alternativa B Falta información para realizar la operación

- A1. [Sistema] Muestra el mensaje **MSG13 Falta información** en la pantalla **MIUW 6 Lista de Traducciones**.

Trayectoria Alternativa B No se pudo realizar la operación.

- B1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.6. W-PR-CU1.1.3 Eliminar traducción

Este caso de uso permite al usuario final eliminar una traducción previamente solicitada por él.

Nombre del Caso de Uso	W-PR-CU1.1.3 Eliminar traducción
Objetivo	Eliminar una traducción del usuario
Actores	Usuario verificado
Disparador	El actor requiere eliminar una traducción
Entradas	Ninguna
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus de validado ■ El actor debe haber iniciado sesión ■ La traducción debe de existir en el sistema ■ La traducción debe de pertenecer al actor
Postcondiciones	<ul style="list-style-type: none"> ■ Al aceptar la eliminación de una traducción, toda su información correspondiente será eliminada del sistema
Prioridad	Baja
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Presiona el botón *Eliminar* de algún registro mostrado en la pantalla **MIUW 6 Lista de Traducciones**.
2. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación**.
3. [Actor] Acepta la eliminación oprimiendo el botón *Aceptar*. [Trayectoria alternativa A]
4. [Sistema] Verifica que la traducción pertenezca al usuario.
5. [Sistema] Elimina toda la información correspondiente a la traducción.

6. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 6 Lista de Traducciones** indicando que la operación se ha realizado exitosamente y actualiza la lista de traducciones.

Fin del caso de uso.

Trayectoria Alternativa A El actor cancela la operación.

A1. [Actor] Solicita cancelar la operación oprimiendo el botón *Cancelar*.

A2. [Sistema] Muestra la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.7. W-PR-CU1.1.4 Copiar traducción

Una traducción que se realice por el sistema puede ser copiada sin necesidad de descargar todo el proyecto sobre el que se trabaja.

Nombre del Caso de Uso	W-PR-CU1.1.4 Copiar traducción
Objetivo	Permitir copiar una traducción realizada
Actores	Usuario verificado
Disparador	El actor requiere copiar una traducción
Entradas	Ninguna
Salidas	Copia de la traducción
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe haber iniciado sesión ■ Deben existir traducciones que copiar
Postcondiciones	<ul style="list-style-type: none"> ■ Se copia la traducción
Prioridad	Media
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Presiona el botón *Copiar* de la pantalla **MIUW 6 Lista de Traducciones**.
2. [Sistema] Recupera la traducción.
3. [Sistema] Copia la traducción. [Trayectoria alternativa A]
4. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin del caso de uso.

Trayectoria Alternativa A No se pudo realizar la operación.

- A1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.8. W-PR-CU1.1.5 Eliminar proyecto

Este caso de uso permite al usuario final eliminar un proyecto previamente creado por él.

Nombre del Caso de Uso	W-PR-CU1.1.5 Eliminar proyecto
Objetivo	Eliminar un proyecto del usuario
Actores	Usuario validado
Disparador	El actor requiere eliminar un proyecto
Entradas	Ninguna
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus de verificado ■ El actor debe haber iniciado sesión ■ El proyecto debe de existir en el sistema ■ El proyecto debe de pertenecer al actor
Postcondiciones	<ul style="list-style-type: none"> ■ Al aceptar la eliminación del proyecto, todas sus traducciones asociadas serán eliminadas del sistema
Prioridad	Media
Reglas de Negocio	Ninguna

Trayectoria Principal

1. [Actor] Presiona el botón *Eliminar Proyecto* de la pantalla **MIUW 6 Lista de Traducciones**.
2. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación**.
3. [Actor] Presiona el botón *Aceptar*. [Trayectoria alternativa A]
4. [Sistema] Verifica que el proyecto pertenezca al usuario.
5. [Sistema] Elimina toda la información correspondiente al proyecto.
6. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 5 Lista de Proyectos** y actualiza la lista de proyectos.

Fin del caso de uso.

Trayectoria Alternativa A El actor cancela la operación.

A1. [Actor] Solicita cancelar la operación oprimiendo el botón *Cancelar*.

A2. [Sistema] Muestra la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.9. W-PR-CU1.1.6.1 Calificar traducción

Una traducción que se realice por el sistema se puede calificar para apoyar el algoritmo de traducción en un futuro, así como proporcionar retroalimentación al usuario de que tan útil es el proyecto que se trabaja.

Nombre del Caso de Uso	W-PR-CU1.1.6.1 Calificar traducción
Objetivo	Permitir la calificación de una traducción
Actores	Usuario verificado
Disparador	El actor requiere calificar una traducción
Entradas	<ul style="list-style-type: none"> ▪ Calificación
Salidas	<ul style="list-style-type: none"> ▪ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ▪ El actor debe haber iniciado sesión ▪ Deben existir traducciones que calificar
Postcondiciones	<ul style="list-style-type: none"> ▪ Se califica la traducción ▪ Se calcula la calificación del proyecto
Prioridad	Baja
Reglas de Negocio	<ul style="list-style-type: none"> ▪ RN-004 Calificación proyecto ▪ RN-005 Fecha de modificación

Trayectoria Principal

1. [Actor] Presiona el número de estrellas como calificación
2. [Sistema] Muestra el mensaje **MSG11 Confirmación de operación** en la pantalla **MIUW 6 Lista de Traducciones**.
3. [Actor] Presiona el botón *Sí*. [Trayectoria alternativa A]

4. [Sistema] Calcula la calificación del proyecto según la regla de negocio RN-004.
5. [Sistema] Calcula la fecha de modificación según la regla de negocio RN-005.
6. [Sistema] Persiste la información [Trayectoria alternativa B]
7. [Sistema] Muestra el mensaje **MSG9 Operación exitosa** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin del caso de uso.

Trayectoria Alternativa A Se cancela la operación

- A1. [Actor] Presiona el botón *No*.
- A2. [Sistema] Muestra la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B No se pudo realizar la operación.

- A1. [Sistema] Muestra el mensaje **MSG10 Operación fallida** en la pantalla **MIUW 6 Lista de Traducciones**.

Fin de la trayectoria alternativa.

4.4.2.10. W-PR-CU1.2 Crear proyecto

Este caso de uso permite al usuario final crear un nuevo proyecto para almacenar sus traducciones.

Nombre del Caso de Uso	W-PR-CU1.2 Crear proyecto
Objetivo	Crear un nuevo proyecto en el cual se puedan almacenar traducciones
Actores	Usuario verificado
Disparador	El actor requiere crear un nuevo proyecto
Entradas	<ul style="list-style-type: none"> ■ Nombre del proyecto
Salidas	<ul style="list-style-type: none"> ■ MSG9 Operación exitosa
Precondiciones	<ul style="list-style-type: none"> ■ El actor debe estar registrado en el sistema con un estatus de verificado. ■ El usuario debe haber iniciado sesión.
Postcondiciones	Ninguna.
Prioridad	Media.
Reglas de Negocio	<ul style="list-style-type: none"> ■ RN-001 Campos obligatorios ■ RN-002 Datos correctos

Trayectoria Principal

1. [Actor] Presiona el botón *Crear Proyecto* en la pantalla **MIUW 5 Lista de Proyectos**.
2. [Sistema] Muestra el mensaje **MSG12 Ingrese nombre** en la pantalla **MIUW 5 Lista de Proyectos** solicitando el nombre del nuevo proyecto.
3. [Actor] Ingresa el nombre del nuevo proyecto.
4. [Actor] Presiona el botón *Aceptar* en el mensaje **MSG12 Ingrese nombre**. [Trayectoria alternativa A]

5. [Sistema] Verifica la información según la regla de negocio RN-001 y RN-002.[Trayectoria alternativa B]
6. [Sistema] Persiste el nuevo proyecto en el sistema.
7. [Sistema] Muestra el mensaje **Operación exitosa** en la pantalla **MIUW 5 Lista de Proyectos** y actualiza la lista de proyectos.

Fin del caso de uso.

Trayectoria Alternativa A El actor cancela la operación.

- A1. [Actor] Solicita cancelar la operación oprimiendo el botón *Cancelar* en el mensaje **MSG12 Ingrese nombre**.
- A2. [Sistema] Muestra la pantalla **MIUW 5 Lista de Proyectos**.

Fin de la trayectoria alternativa.

Trayectoria Alternativa B Datos no validos.

- B1. [Sistema] Muestra el mensaje **MSG1 Datos no validos** en la pantalla **MIUW 5 Lista de Proyectos**.
- B2. [Actor] Continua con el paso 3 de la Trayectoria Principal.

Fin de la trayectoria alternativa.

4.5. Mensajes

En esta sección se muestran los mensajes que se despliegan en pantalla para los diferentes casos de uso que se tienen.

MSG1. MSG1 Datos no validos

Tipo Error

Redacción Los datos introducidos no son validos.

MSG2. MSG2 Cuenta no verificada

Tipo Error

Redacción La cuenta aun no ha sido verificada.

MSG3. MSG3 Correo electrónico no registrado

Tipo Error

Redacción No existe una cuenta con el correo electrónico introducido.

MSG4. MSG4 Correo electrónico ya registrado

Tipo Error

Redacción El correo electrónico ya se ha utilizado.

MSG5. MSG5 Verifique su cuenta

Tipo Informativo

Redacción Verifique su cuenta a través del correo electrónico que se le ha mandado.

MSG6. MSG6 Envio de correo de recuperación

Tipo Informativo

Redacción Se ha enviado un correo electrónico para la recuperación de su contraseña.

MSG7. MSG7 No existen proyectos para mostrar

Tipo Informativo

Redacción No hay proyectos que se puedan mostrar.

MSG8. MSG8 No se puede mostrar el proyecto

Tipo Error

Redacción En este momento no se puede mostrar el proyecto.

MSG9. MSG9 Operación exitosa

Tipo Informativo

Redacción La operación se ha realizado con éxito.

MSG10. MSG10 Operación fallida

Tipo Error

Redacción La operación no se pudo realizar.

MSG11. MSG11 Confirmación de operación

Tipo Confirmación

Redacción

¿Está seguro de <OPERACIÓN> <ELEMENTO> <NOMBRE_ELEMENTO>?

Parametros

- <OPERACIÓN>: Operación a realizar que se debe de confirmar.
- <ELEMENTO>: Elemento sobre el cual se realizara la operación.
- <NOMBRE_ELEMENTO>: Nombre del elemento sobre el que se trabaja.

Ejemplo

- ¿Está seguro de eliminar el proyecto tesis final?

MSG12. MSG12 Ingrese el nombre

Tipo Confirmación

Redacción Ingrese el nombre del proyecto.

MSG13. MSG13 Falta información

Tipo Error

Redacción Falta información necesaria para realizar la operación.

4.6. Interfaces

4.6.1. Aplicación en Android

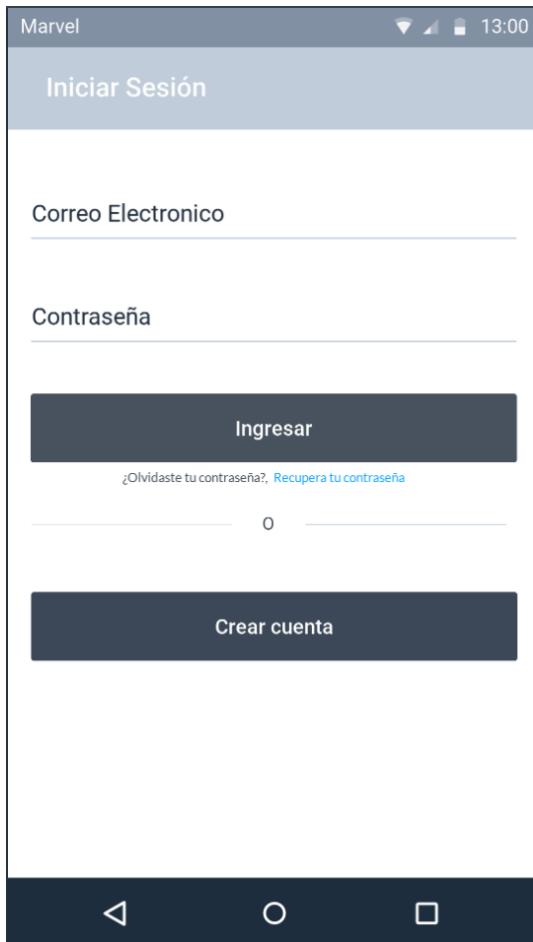


Figura 4.7: MIUA 1 Iniciar Sesión



Figura 4.8: MIUA 2 Registro



Figura 4.9: MIUA 3 Recuperar Contraseña

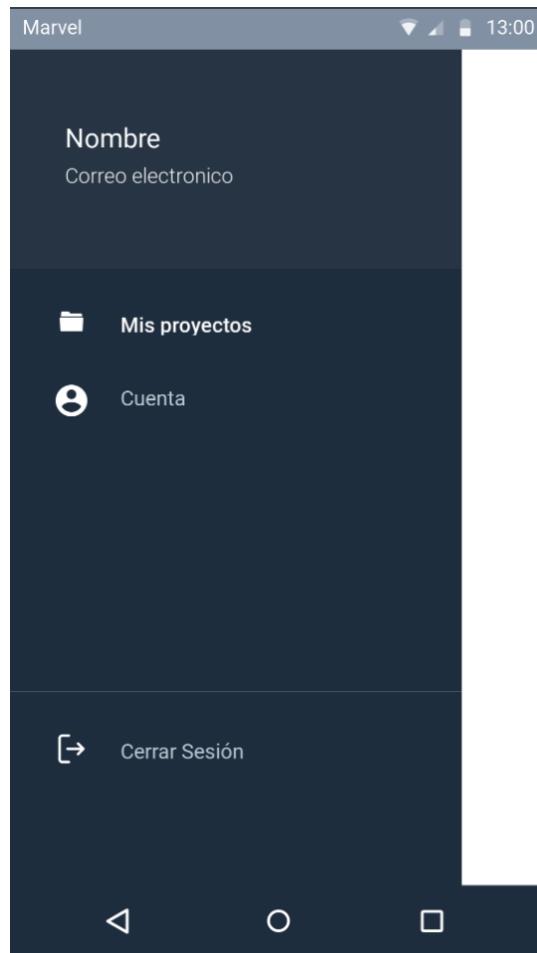


Figura 4.10: MIUA 4 Menú de Opciones



Figura 4.11: MIUA 5 Lista de Proyectos

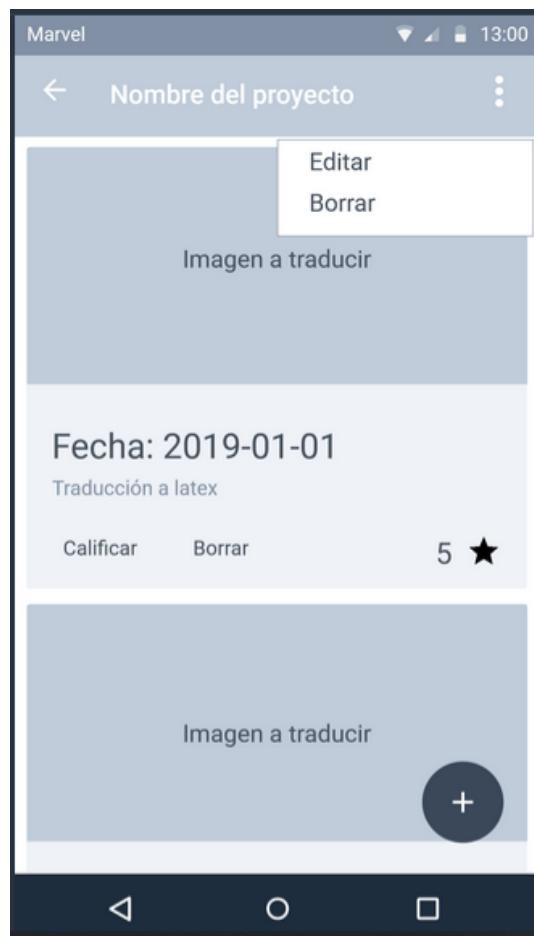


Figura 4.12: MIUA 6 Lista de Traducciones

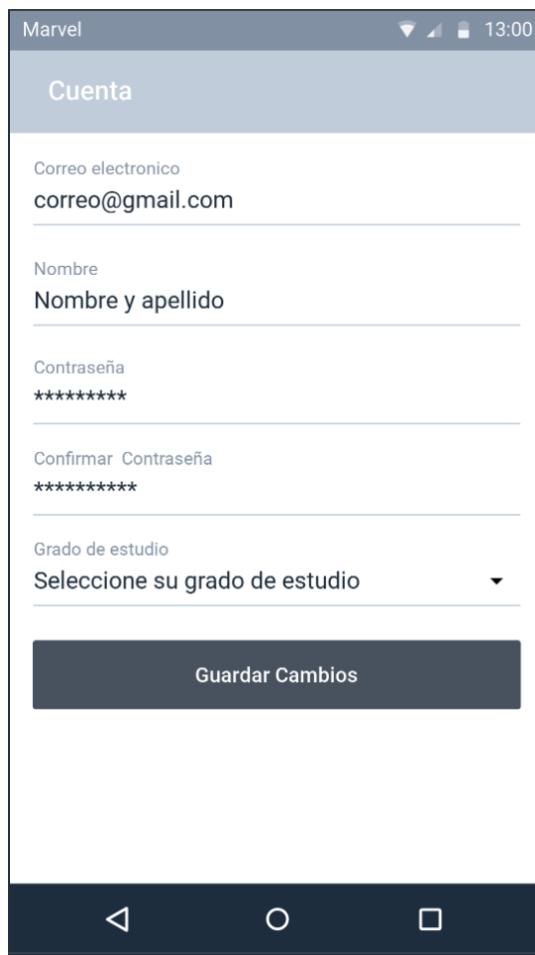


Figura 4.13: MIUA 7 Editar Perfil

4.6.2. Aplicación en Web



The image shows the login screen of the MIUW 1 application. At the top, there is a header bar with three dots and a search bar labeled "Nombre de la app". Below the header is a dark blue navigation bar with the text "Inicio de Sesión". The main form area has two input fields: "Correo" and "Contraseña". Below the password field is a link "Olvidaste tu contraseña, recuperala aquí.". A large dark blue button labeled "Iniciar sesión" is centered. Below it is a horizontal separator line with the text "o" in the center. At the bottom is another dark blue button labeled "Regístrate".

Figura 4.14: MIUW 1 Iniciar Sesión



The image shows the registration screen of the MIUW 2 application. At the top, there is a header bar with three dots and a search bar labeled "Nombre de la app". Below the header is a dark blue navigation bar with the text "Registro". The main form area has four input fields: "Nombre", "Correo", "Contraseña", and "Confirmar contraseña". Below these fields is a large dark blue button labeled "Registrarse". Below it is a horizontal separator line with the text "o" in the center. At the bottom is another dark blue button labeled "Iniciar sesión".

Figura 4.15: MIUW 2 Registro

This screenshot shows the 'Editar Perfil' (Edit Profile) page of the MIUW 3 application. At the top, there is a header bar with three dots on the left and a search bar on the right. Below the header, the text 'Nombre de la app' is displayed. On the right side of the header, there are links for 'Mis proyectos' and 'Cuenta'. A large circular placeholder image with a blue 'X' is centered above a form field labeled 'Correo electrónico'. The form includes input fields for 'Nombre', 'Contraseña', and 'Confirmar contraseña', followed by a dropdown menu for 'Grado de estudios'.

Figura 4.16: MIUW 3 Editar Perfil

This screenshot shows the 'Recuperar Contraseña' (Forgot Password) page of the MIUW 4 application. At the top, there is a header bar with three dots on the left and a search bar on the right. Below the header, the text 'Nombre de la app' is displayed. In the center of the page, the text 'Recuperar contraseña' is shown above a single input field labeled 'Correo de recuperación'. Below the input field is a dark blue button labeled 'Recuperar Contraseña'.

Figura 4.17: MIUW 4 Recuperar Contraseña

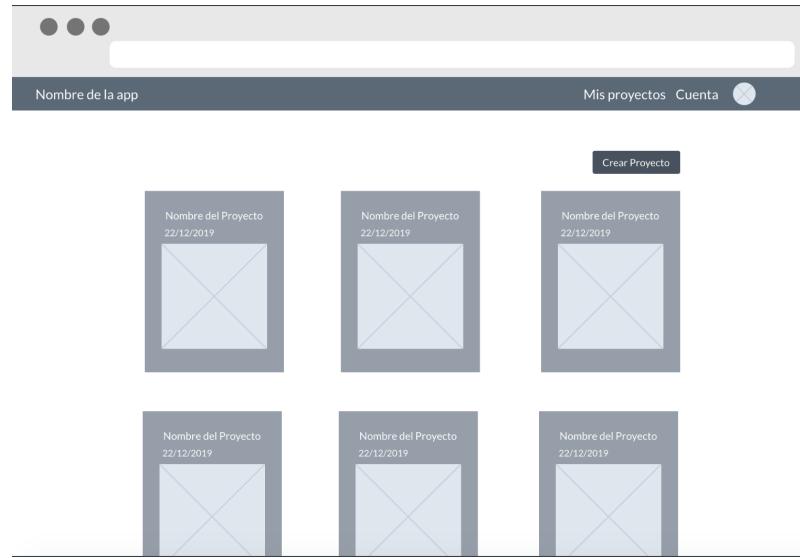


Figura 4.18: MIUW 5 Lista de Proyectos

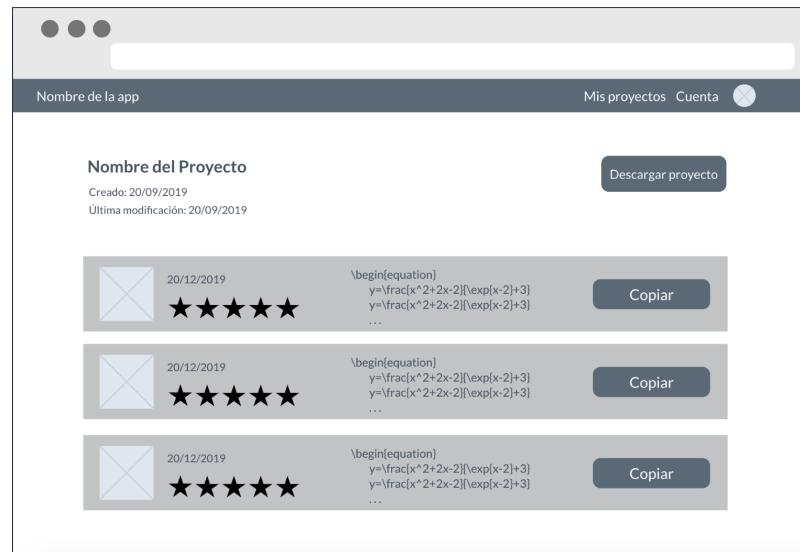


Figura 4.19: MIUW 6 Lista de Traducciones



Figura 4.20: MIUW 7 Menú de Opciones

Capítulo 5

Desarrollo del sistema

A continuación, se explica el desarrollo del sistema, dicha explicación se encuentra dividida en la sección del desarrollo de la aplicación móvil y por otra parte se encuentra el desarrollo de la aplicación web.

5.1. Android

Esta sección tiene objetivo presentar las principales características en el desarrollo de la aplicación para Android.

5.1.1. Arquitectura de la aplicación

Para el desarrollo de la aplicación se implemento la arquitectura Clean, la cual como ya se ha mencionado antes se ha mencionado se ha vuelto muy popular en el desarrollo de aplicaciones móviles para android debido a que es una solución que produce sistemas que presentan las siguientes características.

- Escalables, por lo que se pueden agregar más funcionalidades de forma sencilla.
- Presentan modularidad.
- Presentan independencia en cuanto a frameworks, interfaz de usuario y bases de datos.
- El proyecto es más fácil de mantener por lo que es más sencillo hacer cambios.

Al utilizar esta arquitectura el proyecto queda separado en tres capas como se observa en la figura 5.1 con lo cual cada una de ellas tiene su propósito definido.

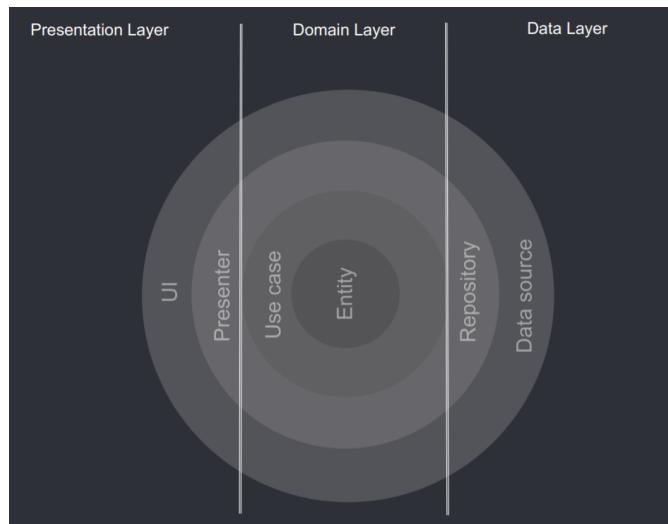


Figura 5.1: Tres capas que se tienen al utilizar la arquitectura Clean [3]

5.1.1.1. Capa de datos

La información que se utiliza en el resto de capas proviene de esta capa. Esta capa a su vez se encuentra dividida en la capa de repositorio y en la capa de fuente de datos.

Capa de repositorio En esta capa se utiliza el patrón de repositorio como se muestra en la figura 5.2. Gracias a este patrón se puede tener acceso a diferentes fuentes de datos que se encuentran en la capa más baja de nuestra arquitectura, esto nos permite un acceso a los datos de forma transparente para el usuario bajo las condiciones que se presenten.

La forma de utilizar este patrón en la aplicación desarrollada crear una clase en la cual se hace uso de la interfaz que se tiene para el acceso a la fuente de datos. En el siguiente código se puede apreciar el como se crea una instancia de APIService que es nuestra interfaz para fuente de datos.

Después, en nuestro método findAllProyectosByUser se recupera la información necesaria para mandarla a las capas superiores.

```

1 public class ProjectRepositoryImpl implements ProjectRepository {
2     private APIService service = ServiceGenerator.createService(APIService.
3         class);
4     private static final String TAG = ProjectRepositoryImpl.class .
5         getCanonicalName();
6
7     @Override
8     public MutableLiveData<BusinessResult<ProyectoModel>>
9         findAllProyectosByUser(Integer id, String key) {

```

```

7     MutableLiveData<BusinessResult<ProyectoModel>>
8     proyectoDataMutableLiveData = new MutableLiveData<>();
9
10    try {
11        service.getProyectosByUsuario(id, key).enqueue(new Callback<List<
12            ProyectoData>>() {
13            @Override
14            public void onResponse(Call<List<ProyectoData>> call,
15            Response<List<ProyectoData>> response) {
16                BusinessResult<ProyectoModel> model = new BusinessResult
17                <>();
18                List<ProyectoModel> modelos = new ArrayList<>();
19                if (response.isSuccessful()) {
20                    for (ProyectoData data : response.body()) {
21                        ProyectoModel proyectoModel = new ProyectoModel()
22                        ;
23                        proyectoModel.setRate(data.getCalificacion());
24                        proyectoModel.setId(data.getId());
25                        proyectoModel.setName(data.getNombre());
26                        proyectoModel.setTextDate(data.getFecha());
27                        modelos.add(proyectoModel);
28                    }
29                    model.setResults(modelos);
30                    model.setCode(ResultCodes.SUCCESS);
31                }
32                proyectoDataMutableLiveData.setValue(model);
33            }
34            @Override
35            public void onFailure(Call<List<ProyectoData>> call,
36            Throwable t) {
37                BusinessResult<ProyectoModel> model = new BusinessResult
38                <>();
39                proyectoDataMutableLiveData.setValue(model);
40            }
41        });
42    }

```

```

35     } catch (NetworkOnMainThreadException e) {
36         Log.e(TAG, "findAllProyectosByUser ", e);
37     }
38     return proyectoDataMutableLiveData;
39 }
40 }
```

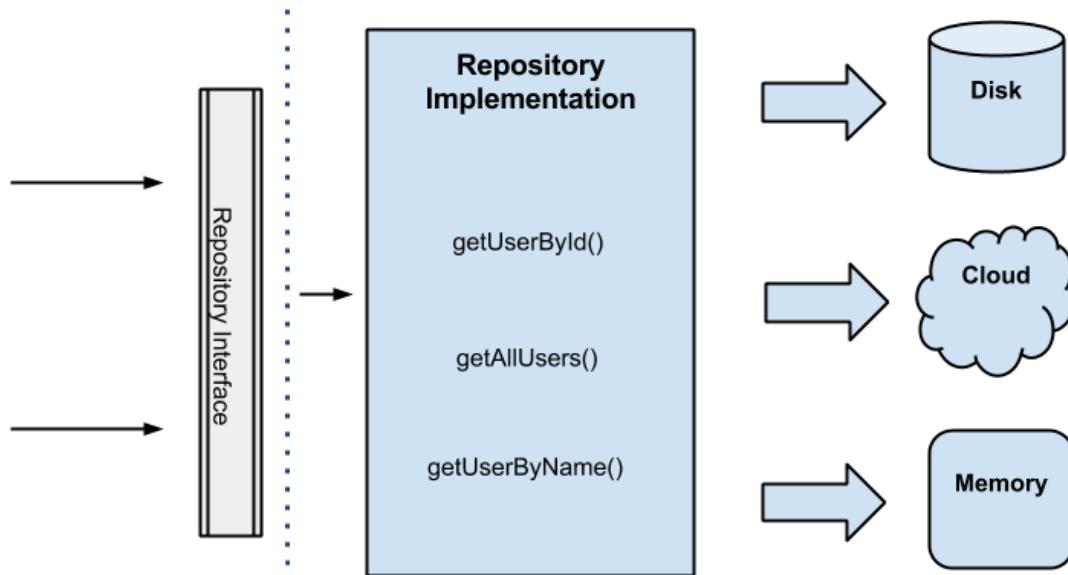


Figura 5.2: Capa de datos [4]

Capa de fuente de datos En este trabajo, la fuente de datos que se tiene es un API REST, sin embargo si se requiere acceder a información que se persista en el teléfono se puede agregar otra fuente de datos. Se utiliza retrofit para poder realizar la comunicación con el API REST.

La forma de utilizar retrofit es crear una interfaz con todos los métodos para recuperar o enviar información al API REST, en esta interfaz cada método tiene la URL a la cual se realizará la petición con alguno de los métodos que tiene HTTP, se tienen los parámetros que se envían y cada método nos regresa una llamada asíncrona que se trabaja en la capa de repositorio. Esto se puede apreciar en el siguiente código.

```

1 public interface APIService {
2     // Crear proyectoModel
3     @POST("/proyectos")
4     Call<ProyectoData> createProyecto(@Body ProyectoData proyectoModel,
5                                         @Header("Authorization") String token);
```

```

5   // Edicion de un proyectoModel
6   @PUT("/ proyectos/{ idProyecto }")
7   Call<ProyectoData> editProyecto(@Path("idProyecto") Integer idProyecto ,
8   @Body ProyectoData proyectoModel , @Header("Authorization") String token);
9   // Elimina un proyecto
10  @DELETE("/ proyectos/{ idProyecto }")
11  Call<ProyectoData> deleteProyecto(@Path("idProyecto") Integer idProyecto ,
12  @Header("Authorization") String key);
13  // Obtiene las traducciones asociadas a un proyecto
14  @GET("/ proyectos/{ idProyecto }/ traducciones")
15  Call<List<TraduccionData>> getTraduccionesByProyecto(@Path("idProyecto") Integer idProyecto , @Header("Authorization") String key);
16  // Creacion de una traduccionModel
17  @POST("/ traducciones")
18  Call<TraduccionData> createTraduccion(@Body TraduccionData traduccionData );
19  // Edicion de un traduccionModel
20  @PUT("/ traducciones/{ idTraducion }")
21  Call<TraduccionData> editTraduccion(@Path("idTraducion") Integer idTraducion , @Body TraduccionData traduccionData );
22  // Elimina una traduccion
23  @DELETE("/ traducciones/{ idTraducion }")
24  Call<TraduccionData> deleteTraduccion(@Path("idTraducion") Integer idTraducion , @Header("Authorization") String token);
25  // Manda a crear un usuarioData
26  @POST("/ usuarios")
27  Call<UsuarioData> createUsuario(@Body UsuarioData usuarioData);
28  // Para hacer login
29  @POST("/ users/login")
30  Call<UsuarioData> loginUsuario(@Body UsuarioData usuarioData);
31  // Para hacer la recuperacion de contra
32  @POST("/ usuarios/recuperar")
33  Call<UsuarioData> recuperarUsuario(@Body UsuarioData usuarioData);
34  // Para editar usuarioData
35  @PUT("/ users/{ idUsuario }")

```

```

34     Call<UsuarioData> editUsuario(@Path("idUsuario") Integer id, @Body
35     UsuarioData userData, @Header("Authorization") String key);
36     // Obtiene los proyectos asociados a un usuario
37     @GET("/usuarios/{idUsuario}/proyectos")
38     Call<List<ProyectoData>> getProyectosByUsuario(@Path("idUsuario") Integer
39     idUsuario, @Header("Authorization") String key);
}

```

Para poder hacer uso de esta interfaz se tiene que configurar bajo ciertas características específicas como lo son la URL a la cual hará peticiones, el logger que se utilizará para poder observar las peticiones que se realizan y brindar una retroalimentación a la hora de hacer pruebas y por último el parser que se utilizará para trabajar y pasar de clases a datos que el API REST entienda y pueda utilizar, en este caso se utilizó el formato JSON. La definición de estas características se tiene en el siguiente código.

```

1 public class ServiceGenerator {
2     private static final String BASE_URL = "http://10.100.72.207:8000/";
3     private static Retrofit.Builder builder = new Retrofit.Builder().baseUrl(
4         BASE_URL)
5             .addConverterFactory(GsonConverterFactory.create());
6
7     private static Retrofit retrofit = builder.build();
8     private static OkHttpClient.Builder httpClient = new OkHttpClient.Builder();
9
10    private static HttpLoggingInterceptor loggingInterceptor = new
11        HttpLoggingInterceptor();
12
13    public static <S> S createService(Class<S> serviceClass) {
14        if (!httpClient.interceptors().contains(loggingInterceptor)) {
15            loggingInterceptor.level(HttpLoggingInterceptor.Level.BODY);
16            httpClient.addInterceptor(loggingInterceptor);
17            builder.client(httpClient.build());
18            retrofit = builder.build();
19        }
20        return retrofit.create(serviceClass);
21    }
22}

```

Finalmente, en esta capa se tienen clases Java que después se mapean a objetos JSON y viceversa, para realizar esto se crea un POJO con los atributos que se necesitan además de agregar anotaciones de retrofit para que el parser pude hacer la conversión necesaria. Un ejemplo de esto es en la siguiente clase de java.

```
1 public class UsuarioData {
2     @SerializedName("id")
3     private Integer id;
4     @SerializedName("nombre")
5     private String nombre;
6     @SerializedName("apellido")
7     private String apellidos;
8     @SerializedName("username")
9     private String email;
10    @SerializedName("password")
11    private String password;
12    @SerializedName("responseCode")
13    private Integer responseCode;
14    @SerializedName("keyAuth")
15    private String keyAuth;
16    @SerializedName("currentPassword")
17    private String currentPassword;
18
19    public String getEmail() {
20        return email;
21    }
22    public void setEmail(String email) {
23        this.email = email;
24    }
25    public String getPassword() {
26        return password;
27    }
28    public void setPassword(String password) {
29        this.password = password;
30    }
31    public Integer getId() {
```

```
32         return id;
33     }
34     public void setId(Integer id) {
35         this.id = id;
36     }
37     public String getNombre() {
38         return nombre;
39     }
40     public void setNombre(String nombre) {
41         this.nombre = nombre;
42     }
43     public String getApellidos() {
44         return apellidos;
45     }
46     public void setApellidos(String apellidos) {
47         this.apellidos = apellidos;
48     }
49     public Integer getResponseCode() {
50         return responseCode;
51     }
52     public void setResponseCode(Integer responseCode) {
53         this.responseCode = responseCode;
54     }
55     public String getKeyAuth() {
56         return keyAuth;
57     }
58     public void setKeyAuth(String keyAuth) {
59         this.keyAuth = keyAuth;
60     }
61     public String getCurrentPassword() {
62         return currentPassword;
63     }
64     public void setCurrentPassword(String currentPassword) {
65         this.currentPassword = currentPassword;
66     }
```

5.1.1.2. Capa de dominio

En esta capa es la intermediaria entre las otras dos capas que se tienen, es donde se encuentran los casos de uso también conocidos como interactors como se muestra en la figura 5.3 en ellos la lógica del negocio es ejecutada es por esto que es el núcleo de la aplicación.

Es importante mencionar que esta capa, al ser la encargada del negocio es donde se hacen validaciones en la información y dicha información se adapta para que sea trabajada en la capa de presentación o en la de datos

Además de contener los casos de uso en esta capa se encuentran las entidades y se hace uso de los repositorios para acceder a la información proporcionada por la capa de datos.

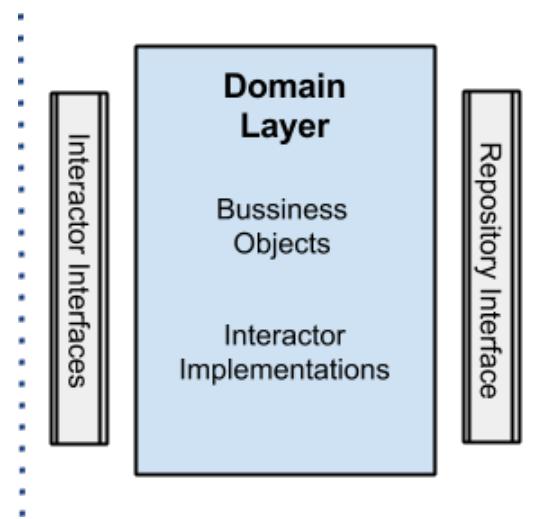


Figura 5.3: Capa de dominio [4]

Para tener un control sobre posibles errores en la capa de presentación o en la capa de datos se utilizan códigos de resultados al igual que una clase que contiene el resultado que se puede presentar, así como la información que se le regresa a la capa de presentación. Se hace uso de genéricos para poder reutilizar esta clase en toda la aplicación y no duplicar código. La clase es la siguiente.

```

1 public class BusinessResult<T> {
2     private Integer code = ResultCodes.ERROR;
3     private T result;
4     private List<T> results;
5
6     public Integer getCode() {

```

```

7         return code;
8     }
9
10    public void setCode(Integer code) {
11        this.code = code;
12    }
13
14    public T getResult() {
15        return result;
16    }
17
18    public void setResult(T result) {
19        this.result = result;
20    }
21
22    public List<T> getResults() {
23        return results;
24    }
25
26    public void setResults(List<T> results) {
27        this.results = results;
28    }
29}

```

La forma en la que se utiliza esta clase en un caso de uso se presenta en el siguiente código que permite iniciar sesión.

```

1 public class UserInteractorImpl implements UserInteractor {
2     public static final String TAG = UserInteractorImpl.class.
3         getCanonicalName();
4
5     private UserRepository repository;
6
7     public UserInteractorImpl() {
8         repository = new UserRepositoryImpl();
9     }
10
11    @Override

```

```

10    public MutableLiveData<BusinessResult<UsuarioModel>> logIn(UsuarioModel
11        usuarioModel) {
12            BusinessResult<UsuarioModel> resultado = new BusinessResult<>();
13            MutableLiveData<BusinessResult<UsuarioModel>> mutableLiveData = new
14            MutableLiveData<>();
15            usuarioModel.setValidPassword(RN002.isValidPassword(usuarioModel.
16                getPassword()));
17            usuarioModel.setValidEmail(RN002.isValidEmail(usuarioModel.getEmail()));
18            if (usuarioModel.getValidEmail() && usuarioModel.getValidPassword())
19            {
20                UsuarioData usuarioData = new UsuarioData();
21                usuarioData.setEmail(usuarioModel.getEmail());
22                usuarioData.setPassword(usuarioModel.getPassword());
23                mutableLiveData = repository.login(usuarioData);
24            } else {
25                resultado.setCode(ResultCodes.RN002);
26                resultado.setResult(usuarioModel);
27                mutableLiveData.setValue(resultado);
28            }
29        }
30
31        return mutableLiveData;
32    }
33}

```

A su vez el caso de uso utiliza sus propios clases de java para presentar información al usuario en la capa de presentación así como controlar posibles errores en la información que ingrese el usuario los campos de los formularios, un ejemplo de este tipo de clases es el siguiente.

```

1 public class UsuarioModel {
2     private Integer id;
3     private String email;
4     private String password;
5     private String keyAuth;
6     private String name;
7     private String secondPassword;
8     private String lastname;

```

```
9     private String currentPassword;
10    private Boolean validLastName = false;
11    private Boolean validName = false;
12    private Boolean validSecondPassword = false;
13    private Boolean validEmail = false;
14    private Boolean validPassword = false;
15    private Boolean validCurrentPassword = false;
16
17    public String getEmail() {
18        return email;
19    }
20
21    public void setEmail(String email) {
22        this.email = email;
23    }
24
25    public String getPassword() {
26        return password;
27    }
28
29    public void setPassword(String password) {
30        this.password = password;
31    }
32
33    public Boolean getValidEmail() {
34        return validEmail;
35    }
36
37    public void setValidEmail(Boolean validEmail) {
38        this.validEmail = validEmail;
39    }
40
41    public Boolean getValidPassword() {
42        return validPassword;
43    }
```

```
44
45     public void setValidPassword(Boolean validPassword) {
46         this.validPassword = validPassword;
47     }
48
49     public Integer getId() {
50         return id;
51     }
52
53     public void setId(Integer id) {
54         this.id = id;
55     }
56
57     public String getKeyAuth() {
58         return keyAuth;
59     }
60
61     public void setKeyAuth(String keyAuth) {
62         this.keyAuth = keyAuth;
63     }
64
65     public String getName() {
66         return name;
67     }
68
69     public void setName(String name) {
70         this.name = name;
71     }
72
73     public String getSecondPassword() {
74         return secondPassword;
75     }
76
77     public void setSecondPassword(String secondPassword) {
78         this.secondPassword = secondPassword;
```

```
79     }
80
81     public String getLastname() {
82         return lastname;
83     }
84
85     public void setLastname(String lastname) {
86         this.lastname = lastname;
87     }
88
89     public Boolean getValidLastName() {
90         return validLastName;
91     }
92
93     public void setValidLastName(Boolean validLastName) {
94         this.validLastName = validLastName;
95     }
96
97     public Boolean getValidName() {
98         return validName;
99     }
100
101    public void setValidName(Boolean validName) {
102        this.validName = validName;
103    }
104
105    public Boolean getValidSecondPassword() {
106        return validSecondPassword;
107    }
108
109    public void setValidSecondPassword(Boolean validSecondPassword) {
110        this.validSecondPassword = validSecondPassword;
111    }
112
113    public String getCurrentPassword() {
```

```

114     return currentPassword;
115 }
116
117 public void setCurrentPassword(String currentPassword) {
118     this.currentPassword = currentPassword;
119 }
120
121 public Boolean getValidCurrentPassword() {
122     return validCurrentPassword;
123 }
124
125 public void setValidCurrentPassword(Boolean validCurrentPassword) {
126     this.validCurrentPassword = validCurrentPassword;
127 }
128 }
```

5.1.1.3. Capa de presentación

En esta capa como se muestra en la figura 5.4 se trabaja con la lógica relacionada a las interfaces que se tienen en la aplicación, es decir a actividades, fragmentos y archivos XML.

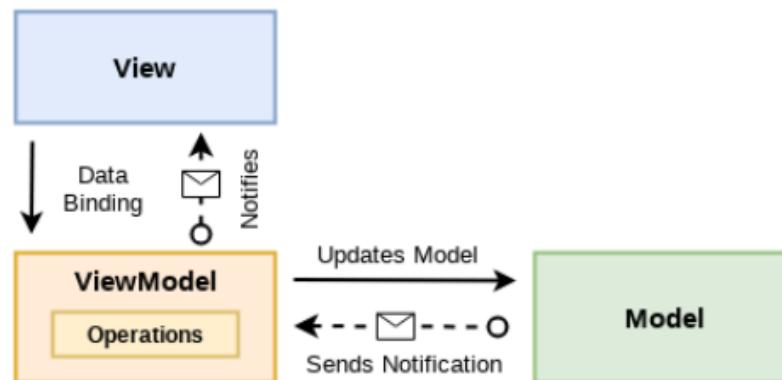


Figura 5.4: Capa de presentación [5]

En esta capa se pueden trabajar con patrones como MVC y MVP pero en este caso se utiliza el patrón MVVM cada uno con una función en particular. [5]

- **Modelo** Se encarga de representar la información que sera presentada en la vista.

- **Vista** Compuesta en este caso por las actividades y fragmentos de la aplicación, su tarea es mostrar la información, hacen uso de los viewmodels para poder realizar cambios en la interfaz.
- **ViewModel** El ViewModel sera el encargado de ejecutar los casos de uso o interactors con el objetivo de actualizar la vista de acuerdo a la información que presente el modelo.

5.2. Web

Esta sección tiene como objetivo presentar las principales características en el desarrollo de la aplicación Web.

5.2.1. Arquitectura de la aplicación

Para el desarrollo de la aplicación se implemento el patrón de diseño Modelo Vista Template (MTV), que como se menciono previamente, es el patrón que Django utiliza.

5.2.1.1. Modelo

En esta capa se maneja todo el acceso a los datos de la aplicación. Django provee de un ORM que permite controlar una base de datos (PostgreSQL para este proyecto). De este modo, todas las tablas que componen la base de datos, están declaradas en esta capa.

5.2.1.2. Vista

En esta capa se maneja la lógica del negocio. Se implementan las validaciones necesarias y se decide que datos deben de ser mostrados al usuario sin indicar como deben de ser presentados a diferencia del tradicional MVC. Esta capa debe de ser vista como un puente entre el Modelo y los Templates.

5.2.1.3. Template

Esta es la capa de presentación. En ella se maneja la forma en la que serán mostrados los datos de la aplicación.

5.3. Conjuntos de entrenamiento

5.3.1. CROHME

El conjunto de datos de entrenamiento para el desarrollo de la red es el denominado **CROHME** por sus siglas en inglés: **Competition on Recognition of Online Handwritten Mathematical Expressions** publicado por los organizadores de la competencia internacional CROHME; para el caso del conjunto de entrenamiento fue posible reunir 7169 elementos que de acuerdo a investigación previa [22], son relativamente pocos para esperar una buena precisión, los elementos en el conjunto son archivos de tipo INKML.

5.3.1.1. Formato del conjunto de datos

Como ya se mencionó, los elementos del conjunto son archivos de tipo INKML (Ink Markup Language) y que principalmente se compone de tres partes:

- Ink: Un conjunto de trazos definidos por puntos.
- Nivel de símbolo Ground-Truth: La segmentación e información de etiqueta de cada símbolo de la expresión.
- Ground-truth: La estructura MATHML de la expresión.

La información de *Ground-Truth* tanto de nivel de símbolo como de la expresión matemática fueron ingresadas manualmente por los colaboradores de la generación del conjunto, además, alguna información general es agregada en el archivo:

- Los canales (en este caso X, Y)
- La información del escritor (Identificación, entrega, edad, Mano dominante, género, etc), si está disponible.
- Ground-Truth en L^AT_EX para fácilmente renderizarlo.

A continuación se muestra un ejemplo de un archivo del dataset que representa la expresión 2^{-1} :

```

1 <ink xmlns="http://www.w3.org/2003/InkML">
2 <traceFormat>
3 <channel name="X" type="decimal"/>
4 <channel name="Y" type="decimal"/>
5 </traceFormat>
```

```

6 <annotation type="truth">$2^{-1}$</annotation>
7 <annotation type="UI">2011_IVC_CIEL_F696_E6</annotation>
8 <annotation type="copyright">LUNAM/IRCCyN</annotation>
9 <annotation type="writer">CIEL696</annotation>
10 <annotationXML type="truth" encoding="Content-MathML">
11   <math xmlns='http://www.w3.org/1998/Math/MathML'>
12     <msup>
13       <mn xml:id="2_1">2</mn>
14     <mrow>
15       <mo xml:id="-_1">-</mo>
16       <mn xml:id="1_1">1</mn>
17     </mrow>
18   </msup>
19 </math>
20 </annotationXML>
21 <trace id="0">
22 3849 2989, 3849 2989, 3847 2979, 3853 2964, 3868 2949, 3875 2949, 3887 2962,
23   3887 2974, 3880 2991, 3862 3011, 3822 3042, 3827 3052, 3839 3061, 3898
24   3049
25 </trace>
26 ...
27 <trace id="2">
28 4009 2905, 4009 2905, 4027 2900, 4049 2887, 4040 2947, 4040 2958
29 </trace>
30 <traceGroup xml:id="3">
31 ...
32 </traceGroup>
33 </ink>
```

Sin embargo, para el propósito del trabajo terminal, el conjunto de datos no es útil en este formato, por lo que se tenía que transformar la información de los trazos en imágenes.

5.3.1.2. Conversión a imagen

Con la información que contiene el archivo INKML es posible generar una imagen con los puntos de los trazos en negro y fondo blanco. El primer reto fue identificar las etiquetas que contenían a los trazos, para ello se utilizó la biblioteca de Python `xml.etree` y una función de

terceros para poder utilizar dichos trazos posteriormente:

```

1 traces_data = []
2
3     tree = ET.parse(inkml_file_abs_path)
4     root = tree.getroot()
5     doc_namespace = "{http://www.w3.org/2003/InkML}"
6
7     'Stores traces_all with their corresponding id'
8     traces_all = [{ 'id': trace_tag.get('id'),
9                     'coords': [[round(float(axis_coord)) if float(axis_coord).
10                         is_integer() else round(float(axis_coord) * 10000) \
11                             for axis_coord in coord[1:].split(' ')] if coord.
12                             startswith(' ') \
13                             else [round(float(axis_coord)) if float(axis_coord).
14                                 is_integer() else round(float(axis_coord) * 10000) \
15                                     for axis_coord in coord.split(' ')]] \
16                             for coord in (trace_tag.text).replace('\n', '').split(',') ]
17             } \
18             for trace_tag in root.findall(doc_namespace + 'trace')]
```

Una vez obtenidos los trazos y con ayuda de matplotlib fueron separados como puntos x,y y utilizados en la función plot de matplotlib para posteriormente guardarla como imagen.

```

1 def inkml2img(input_path, output_path):
2     traces = get_traces_data(input_path)
3     plt.gca().invert_yaxis()
4     plt.gca().set_aspect('equal', adjustable='box')
5     plt.axes().get_xaxis().set_visible(False)
6     plt.axes().get_yaxis().set_visible(False)
7     plt.axes().spines['top'].set_visible(False)
8     plt.axes().spines['right'].set_visible(False)
9     plt.axes().spines['bottom'].set_visible(False)
10    plt.axes().spines['left'].set_visible(False)
11    for elem in traces:
12        ls = elem['trace_group']
13        for subls in ls:
```

```

14         data = np.array(subls)
15         x,y=zip(*data)
16         plt.plot(x,y,linewidth=2,c='black')
17         plt.savefig(output_path, bbox_inches='tight', dpi=100)
18         plt.gcf().clear()

```

Este procedimiento tenía que realizarse por cada uno de los elementos del conjunto de entrenamiento, además de también extraer la expresión matemática encerrada entre las etiquetas <annotation></annotation> con el atributo **type**, para ello nuevamente se utilizó la biblioteca de python xml.etree para acceder a los nodos del árbol directamente:

```

1 def inkml2tag(self, inkml_path):
2     tree = ET.parse(inkml_path)
3     root = tree.getroot()
4     prefix = "{http://www.w3.org/2003/InkML}"
5     GT_tag = [GT for GT in root.findall(prefix + 'annotation') if GT.attrib
6 == {'type': 'truth'}]
7     if GT_tag is None or len(GT_tag) == 0:
8         return ""
9     if GT_tag[0] is None or GT_tag[0].text is None:
10        return ""
11     return GT_tag[0].text

```

Con estos subscripts fue posible desarrollar una biblioteca que permite acceder a la carpeta con el conjunto de entrenamiento en formato inkml y guardarlos como imagen en otra carpeta junto con un archivo CSV contenido la ruta relativa de la imagen y la expresión en L^AT_EX correspondiente separados por coma.

```

1 converted_expressions/200924-1312-148.i.png, \left( \{ a + 7 \} \right)
2 converted_expressions/200926-1550-27.i.png, \mbox{ G }
3 converted_expressions/TrainData2_0_sub_71.i.png, \frac{\sin B + \sin C}{\cos B
+ \cos C}
4 converted_expressions/200923-1251-164.i.png, \zeta^{\phi}
5 converted_expressions/200923-1553-186.i.png, \sqrt{ \{ \mbox{ N } - \mbox{ P } \} }
6 converted_expressions/formulaire012-equation003.i.png, $4 \pi d^2$
7 converted_expressions/formulaire012-equation036.i.png, $a_{ns-3} + a_{n-1}s_2 +
a_{n-2}s_1 + 3a_{n-3} = 0$
8 converted_expressions/79_carlos.i.png, $p = \sqrt{a^2 + b^2 - 2ab \cos A}$

```

```

9 converted_expressions/200923-1553-313.i.png, \left( \sum{ \mbox{S} } \right) \
right )
10 converted_expressions/formulaire010-equation073.i.png, $x^7-x^6-x^4-x^2-1$
```

5.3.1.3. Generador de secuencia de tokens

El archivo CSV generado con lo descrito anteriormente no es suficiente para cargarlo en TensorFlow, la expresión matemática en L^AT_EX debía ser expresada como una secuencia numérica, por lo que se necesitaba identificar a cada símbolo con un número único y conformar a la secuencia, de modo que la estructura del archivo CSV pasaría de tener la forma **RUTA_IMAGEN, EXPRESION_LATEX** a tener la forma **RUTA_IMAGEN, SECUENCIA_NUMÉRICA**, teniendo así una nueva representación del conjunto de entrenamiento conformada por una carpeta con las imágenes y un archivo CSV previamente descrito para poder cargarse en TensorFlow.

Para esto se desarrolló gracias a lex en Python un script para especificar los tokens tomando como base los símbolos especificados en la sección 3.7.

```

1 tokens = [
2     #'corta',
3     #'larga',
4     'alpha',
5     'pi',
6     'beta',
7     'gamma',
8     'lambda',
9     'sigma',
10    'mu',
```



```

1 def tokenizeDataset(self, file):
2
3     fileTokenized = open("tokenized_test.csv", "w")
4
5     myMap = dict()
6     for line in file:
7
8         self.lexer.input(",".join(line.split(",")[1:]))
9         #print(line.split(",")[1])
```

```

10     #print(repr(line.split(",") [1]))
11     token = self.lexer.token()
12     arr = []
13     while token is not None:
14         print(token)
15         myMap[token.type]= token.value #fileMap.write(token.type+","+token.
16         value+"\n")
17         arr.append(Rules.tokens.index(token.type) + 1 ) #+1 means shift one
due to 0 is reserved
18         token = self.lexer.token()
19         fileTokenized.write(line.split(",") [0] + "," + toString(arr) + "\n")
fileTokenized.close()

1 converted_expressions/200924-1312-148.i.png,64 33 41 74 47 58 42 65 34
2 converted_expressions/200926-1550-27.i.png,41 115 42
3 converted_expressions/TrainData2_0_sub_71.i.png,12 41 14 100 47 14 99 42 41
   16 100 47 16 99 42
4 converted_expressions/200923-1251-164.i.png,41 41 78 42 42 62 41 9 42
5 converted_expressions/200923-1553-186.i.png,13 41 41 107 42 46 41 109 42
   42 42
6 converted_expressions/formulaire012-equation003.i.png,55 2 79 62 53
7 converted_expressions/formulaire012-equation036.i.png,74 61 75 95 61 54 47 74
   61 41 75 46 52 42 95 61 53 47 74 61 41 75 46 53 42 95 61 52 47 54 74 61
   41 75 46 54 42 48 51
8 converted_expressions/79_carlos.i.png,86 48 13 41 74 62 53 47 77 62 53 46 53
   74 77 16 102 42
9 converted_expressions/200923-1553-313.i.png,64 33 18 41 41 108 42 42 65 34
10 converted_expressions/formulaire010-equation073.i.png,73 62 58 46 73 62 57 46
    73 62 55 46 73 62 53 46 52

```

Es importante destacar que se debe también guardar este mapeo único y no alterar el orden, por lo que de requerir agregar nuevos símbolos al conjunto es necesario hacerlo al final de los ya existentes, ya que la red entrenada devuelve secuencias numéricas que deben mapearse para tener la correspondiente expresión en LATEX.

5.3.2. Harvard 100k

El segundo conjunto de entrenamiento utilizado fue el provisto por [36]. Los investigadores que realizaron el paper liberaron el conjunto de entrenamiento que recabaron, el cual cuenta con alrededor de cien mil expresiones matemáticas escritas a computadora.

Este conjunto de entrenamiento puede descargarse libremente en la página provista por el artículo. Consiste en un conjunto de imágenes png sin preprocessar y sus respectivos resultados esperados (Ground-Truth). Así mismo, los autores del artículo proveen scripts de normalización y de preprocessado para el tratamiento de las imágenes. Estos scripts se pueden encontrar en el github del artículo.

Se procedió a preprocessar las imágenes con los scripts provistos por los investigadores de Harvard y se obtuvo un conjunto de entrenamiento con 78000 imágenes. Debido a que en el presente Trabajo Terminal las estructuras matriciales no están contempladas, se descartaron todas las imágenes que contuvieran matrices, arreglos o listas, así como aquellas que produjeron errores tras ser procesadas. El conjunto final contiene 73000 imágenes de distintos tamaños los cuales son: (200,50), (240,40), (280,40), (360,60), (160,40), (360,50), (120,50), (320,50), (400,50), (360,10), (360,40), (200,40), (320,40), (280,50) y (240,50). En la Figura 5.5 se puede ver un ejemplo de una imagen del conjunto de entrenamiento de Harvard.

$$d(a.b) = (da).b + a.(db)$$

Figura 5.5: Ejemplo de una imagen del conjunto de entrenamiento Harvard 100k

5.3.3. Normalización

Entre los scripts provistos por los investigadores de Harvard, se encuentra un normalizador. Este código, realiza una serie de transformaciones seguras con el fin de estandarizar ciertas inconsistencias con las expresiones en LATEX. Un ejemplo podría ser las distintas formas que se tienen para expresar un exponente: $a\hat{b}$ y $a \{ b \}$ producirían el mismo resultado. Para mejorar el rendimiento de la red, ambos conjuntos de entrenamiento fueron preprocessados con el script de normalizado.

Cabe mencionar que ambos conjuntos de entrenamiento fueron unificados mediante la combinación de su Ground-Truth, por lo que es posible utilizarlos indistintamente. Para unificarlos se procesó el conjunto CROHME para convertir sus tokens en tokens del conjunto de Harvard. El código usado se muestra a continuación.

```

1 # get the tokens
2 tokens_file = open('latex_vocab.txt', 'r')
3 tokens = dict()
4 count = 0

```

```
5
6 for line in tokens_file:
7     count += 1
8     tokens[line.strip()] = count
9
10 tokens_file.close()
11
12 expressions_file = open('expressions.txt', 'r')
13 expressions_norm_file = open('expressions.norm.lst', 'r')
14 expressions_norm = expressions_norm_file.read().split('\n')
15 tokenized = open('tokenized.csv', 'w')
16 MAXLENGTH = 103
17 i = 0
18
19 for line in expressions_file:
20
21     if 'Object' in expressions_norm[i] or '\\div' in expressions_norm[i]:
22         i += 1
23         continue
24
25     expressions_split = line.split('$')
26     expression = expressions_norm[i].split(' ')
27     sequence = '1000'
28
29     for e in expression:
30         sequence += ' ' + str(tokens[e])
31
32     sequence += ' 1001'
33
34     for x in range(0, MAXLENGTH - len(expression)):
35         sequence += ' 0'
36
37     tokenized.write(expressions_split[0] + ',' + sequence + '\n')
38     i += 1
```

5.4. Desarrollo módulo de análisis de imágenes

Las imágenes generadas para el conjunto de entrenamiento se formaron insertando los puntos en fondo blanco, teniendo como resultado una imagen binaria (contenido en negro y fondo blanco) por lo que al considerar que las fotografías tomadas por un dispositivo móvil no iba a generar por sí mismo un formato semejante fue necesario desarrollar un módulo para esta tarea.

$$\int e^{x^2} x^3 dx$$

Figura 5.6: Imagen de ejemplo del conjunto de entrenamiento generado

El proceso consiste en convertir el área de la expresión matemática en negro y el resto en fondo blanco, para ello se consideraron las técnicas mencionadas en y específicamente se probaron los algoritmos de Otsu y Sauvola [34].

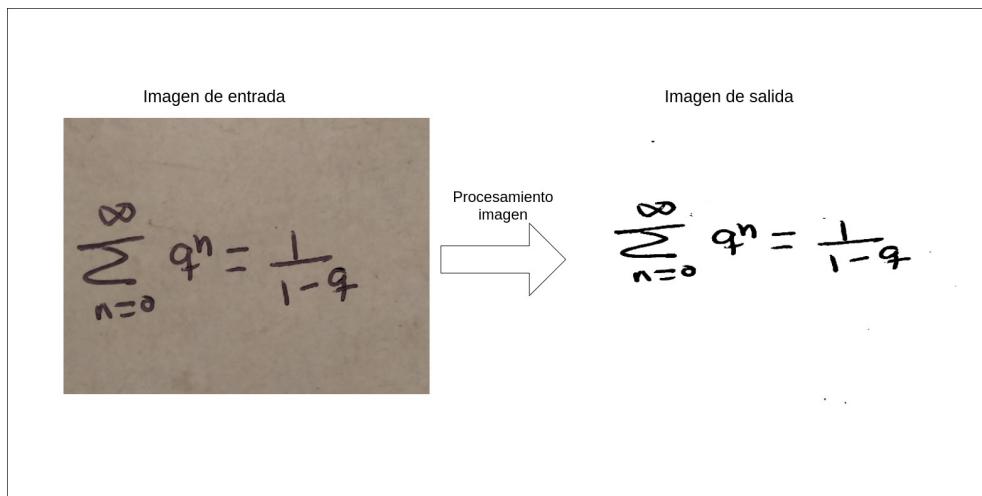


Figura 5.7: Fotografía procesada con algoritmo de Sauvola

De acuerdo a lo mencionado en el marco teórico existen técnicas para binarizar una imagen que calculan el umbral de manera automática, dos de estos algoritmos son el algoritmo de Otsu y el algoritmo de Sauvola, a continuación se describen brevemente y se dan detalles de la implementación o uso según el caso.

5.4.1. Algoritmo de Otsu

El método de Otsu evita tener que elegir un valor de umbral y lo determina automáticamente.

Consideré una imagen con solos dos valores distintos (imagen bimodal), donde el histograma consistiría solo de dos picos. Un buen valor de umbral sería a la mitad de estos dos valores. Similarmente el método de Otsu determina un valor de umbral óptimo global del histograma de la imagen.

Al tratarse de una imagen bimodal, el algoritmo intenta encontrar un valor de umbral (t) que minimice la varianza ponderada dentro de la clase, dada por la relación:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (5.1)$$

donde:

$$\begin{aligned} q_1(t) &= \sum_{i=1}^t P(i) \wedge q_2(t) = \sum_{i=t+1}^I P(i) \\ \mu_1(t) &= \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \wedge \mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)} \\ \sigma_1^2(t) &= \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \wedge \sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \end{aligned} \quad (5.2)$$

```

1 # Filtro Gaussiano con ventana (5,5)
2 blur = cv.GaussianBlur(img,(5,5),0)
3 # Busca el histograma y su funcion de distribucion acumulada
4 hist = cv.calcHist([blur],[0],None,[256],[0,256])
5 hist_norm = hist.ravel()/hist.sum()
6 Q = hist_norm.cumsum()
7 bins = np.arange(256)
8 fn_min = np.inf
9 thresh = -1
10 for i in xrange(1,256):
11     p1,p2 = np.hsplit(hist_norm,[i]) # probabilidades
12     q1,q2 = Q[i],Q[255]-Q[i] # suma de clases
13     if q1 < 1.e-6 or q2 < 1.e-6:
14         continue
15     b1,b2 = np.hsplit(bins,[i]) # pesos
16     # encuentra medias y varianzas
17     m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2
18     v1,v2 = np.sum(((b1-m1)**2)*p1)/q1,np.sum(((b2-m2)**2)*p2)/q2
19     # calcula la funcion de minimizacion

```

```
20     fn = v1*q1 + v2*q2
21     if fn < fn_min:
22         fn_min = fn
23         thresh = i
```

5.4.2. Algoritmo de Sauvola

El algoritmo de Sauvola tiene una implementación más extensa por lo que se dispuso de la utilización de la implementación contenida en **skimage** dentro del paquete filters.

```
1 from skimage import morphology
2 import numpy as np
3 import skimage
4 import os, sys

1 def processBinarization(self, algorithm = ImageAlgorithm.SAUVOLA):
2     if algorithm == ImageAlgorithm.SAUVOLA:
3         image = skimage.io.imread(fname=self.pathImg, as_gray=True)
4         thresh_sauvola = threshold_sauvola(image, window_size=51)
5         self.binary_sauvola = image > thresh_sauvola
```

5.4.3. Resultados

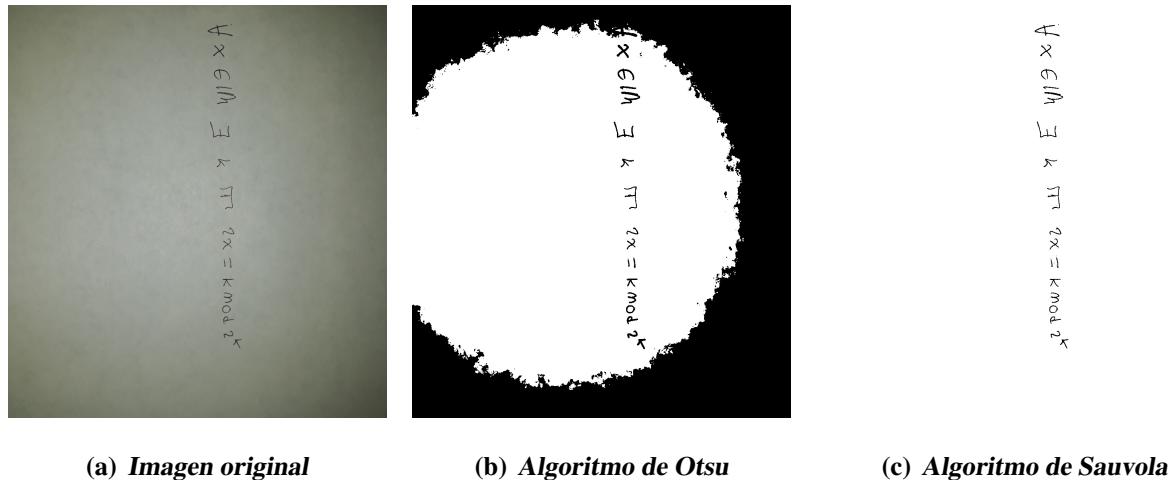


Figura 5.8: Resultados de la aplicación del algoritmo de Otsu y Sauvola



Figura 5.9: Resultados de la aplicación del algoritmo de Otsu y Sauvola

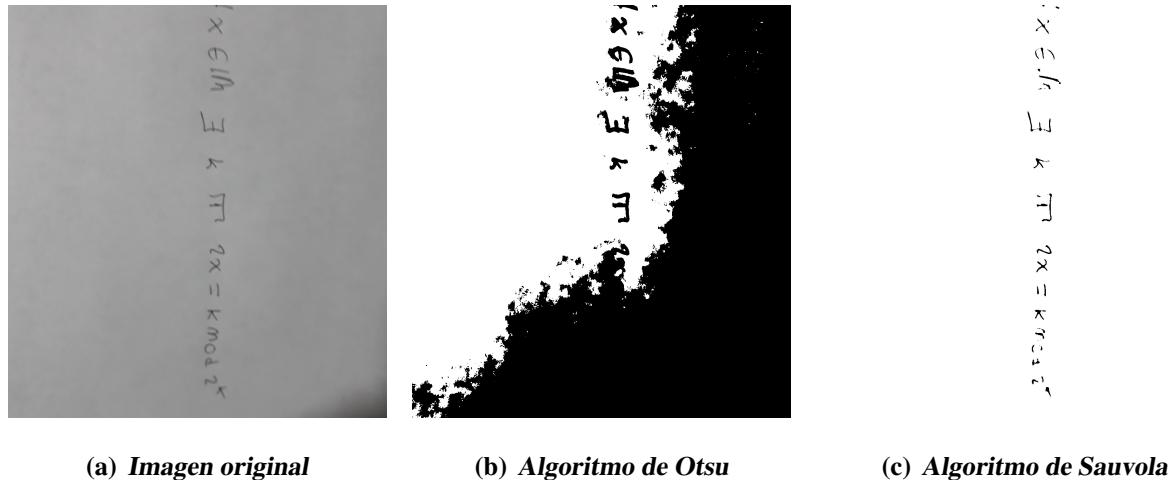
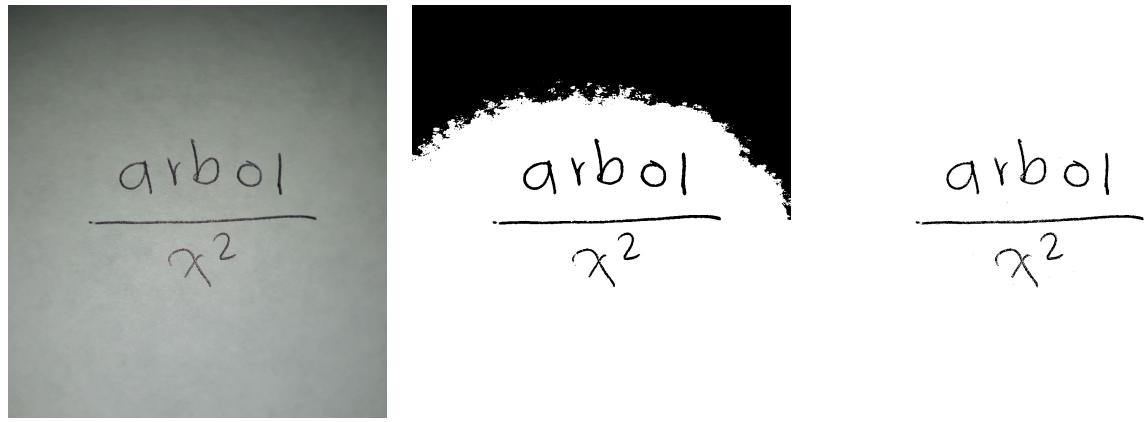


Figura 5.10: Resultados de la aplicación del algoritmo de Otsu y Sauvola



Figura 5.11: Resultados de la aplicación del algoritmo de Otsu y Sauvola

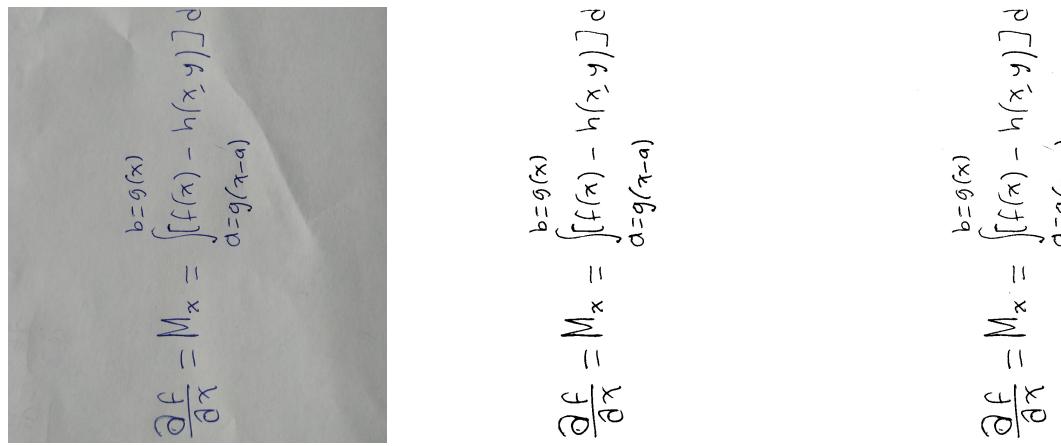


(a) *Imagen original*

(b) *Algoritmo de Otsu*

(c) *Algoritmo de Sauvola*

Figura 5.12: Resultados de la aplicación del algoritmo de Otsu y Sauvola



(a) *Imagen original*

(b) *Algoritmo de Otsu*

(c) *Algoritmo de Sauvola*

Figura 5.13: Resultados de la aplicación del algoritmo de Otsu y Sauvola

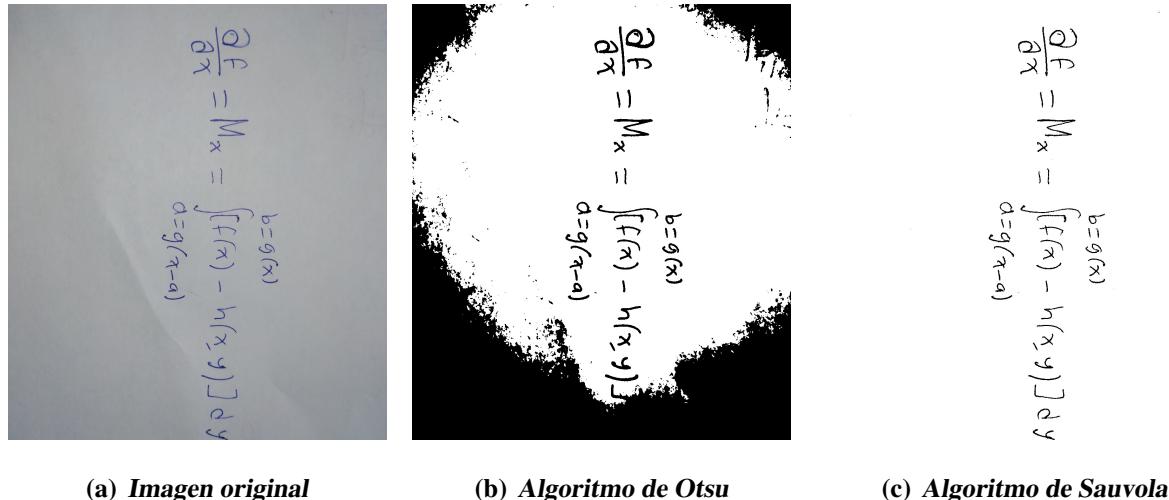


Figura 5.14: Resultados de la aplicación del algoritmo de Otsu y Sauvola

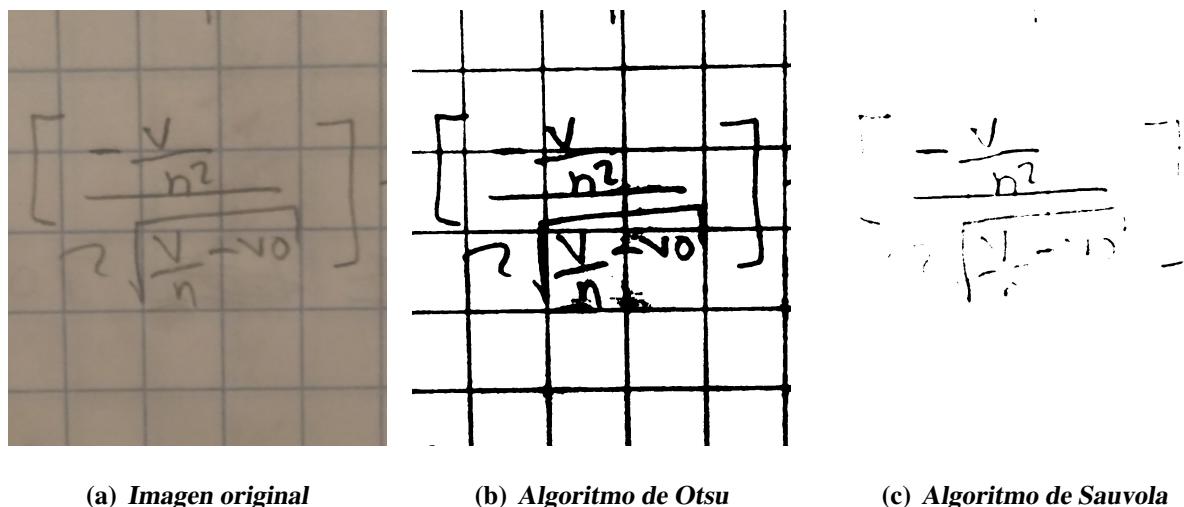
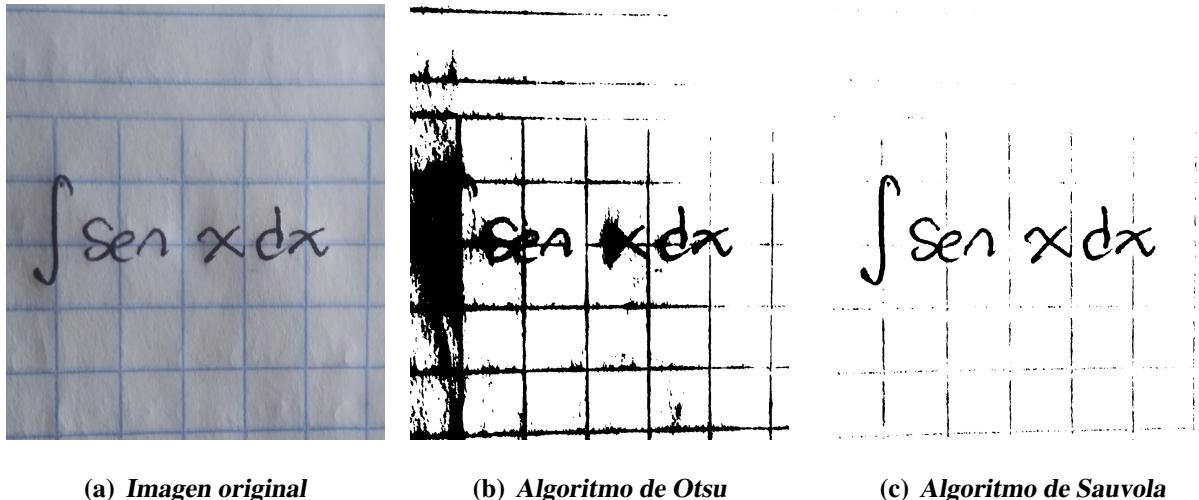
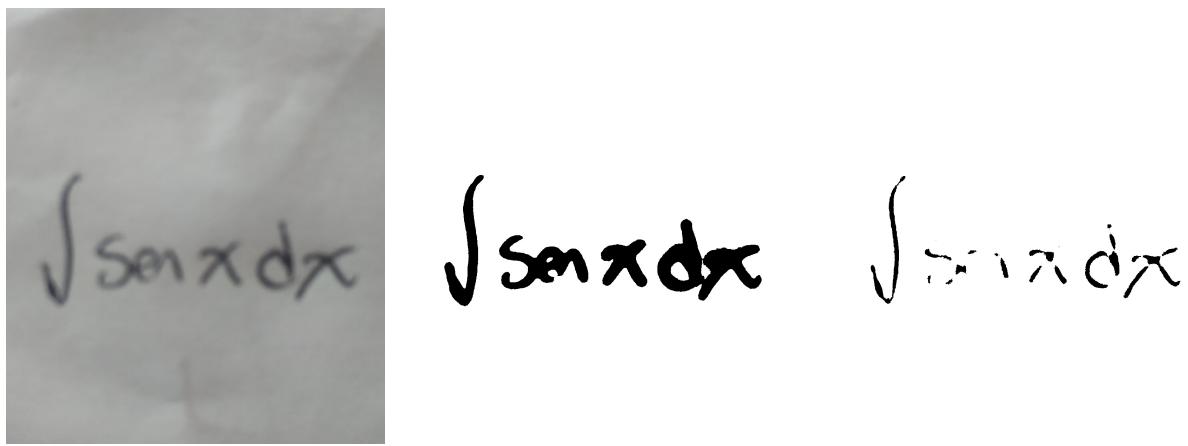


Figura 5.15: Resultados de la aplicación del algoritmo de Otsu y Sauvola



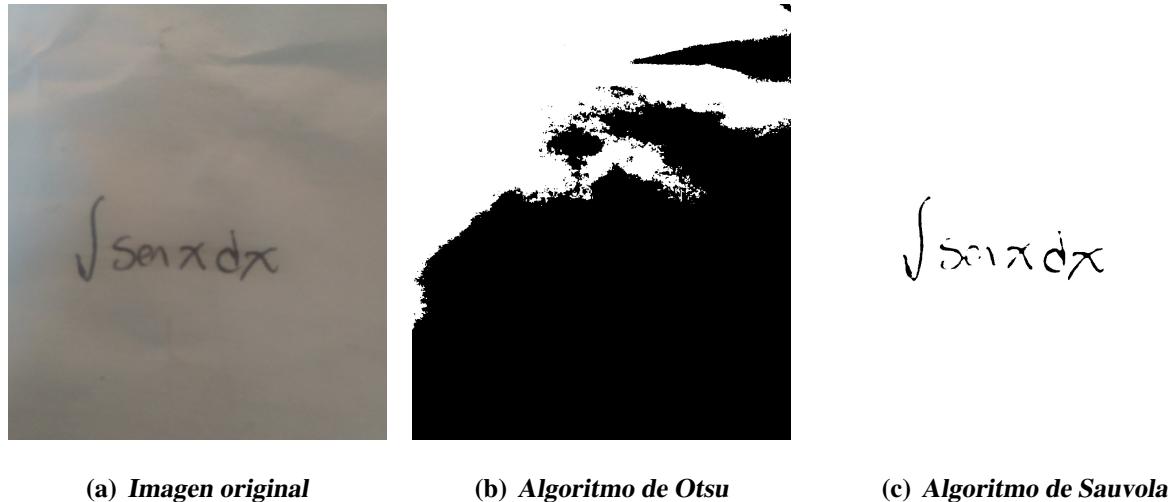
(a) *Imagen original* (b) *Algoritmo de Otsu* (c) *Algoritmo de Sauvola*

Figura 5.16: Resultados de la aplicación del algoritmo de Otsu y Sauvola



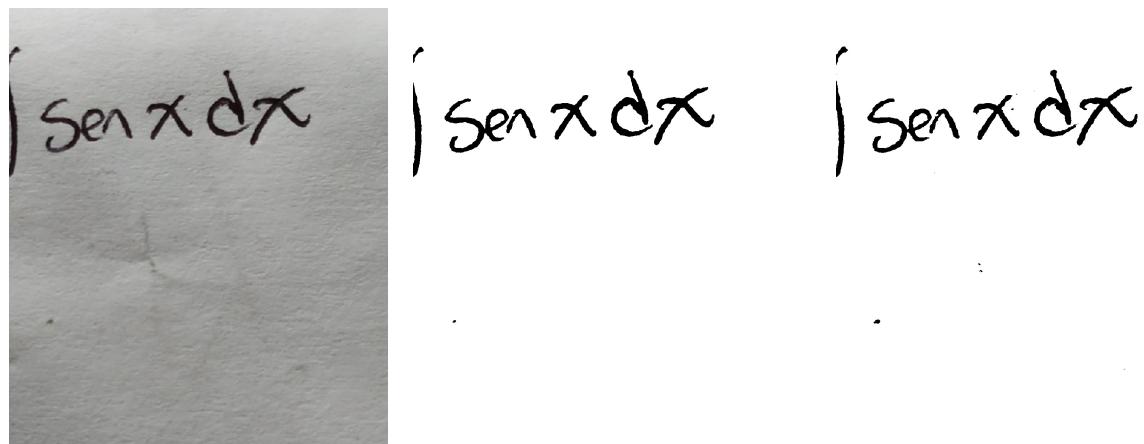
(a) *Imagen original* (b) *Algoritmo de Otsu* (c) *Algoritmo de Sauvola*

Figura 5.17: Resultados de la aplicación del algoritmo de Otsu y Sauvola



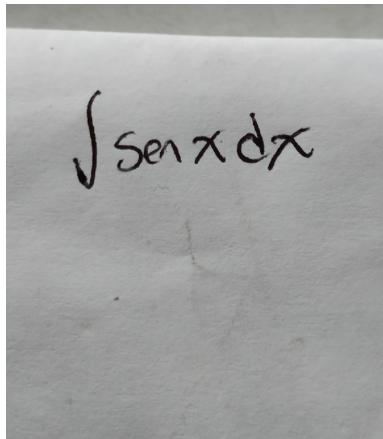
(a) *Imagen original* (b) *Algoritmo de Otsu* (c) *Algoritmo de Sauvola*

Figura 5.18: Resultados de la aplicación del algoritmo de Otsu y Sauvola



(a) *Imagen original* (b) *Algoritmo de Otsu* (c) *Algoritmo de Sauvola*

Figura 5.19: Resultados de la aplicación del algoritmo de Otsu y Sauvola



(a) *Imagen original*

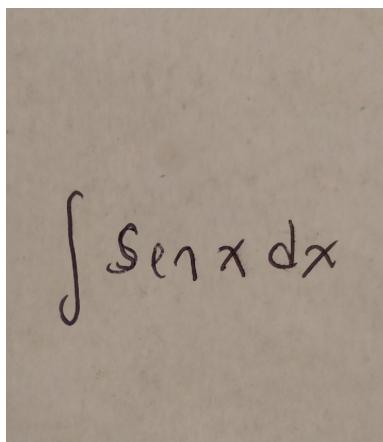
$$\int \sin x dx$$

(b) *Algoritmo de Otsu*

$$\int \sin x dx$$

(c) *Algoritmo de Sauvola*

Figura 5.20: Resultados de la aplicación del algoritmo de Otsu y Sauvola



(a) *Imagen original*

$$\int \sin x dx$$

(b) *Algoritmo de Otsu*

$$\int \sin x dx$$

(c) *Algoritmo de Sauvola*

Figura 5.21: Resultados de la aplicación del algoritmo de Otsu y Sauvola

5.5. Reconocimiento de expresiones matemáticas

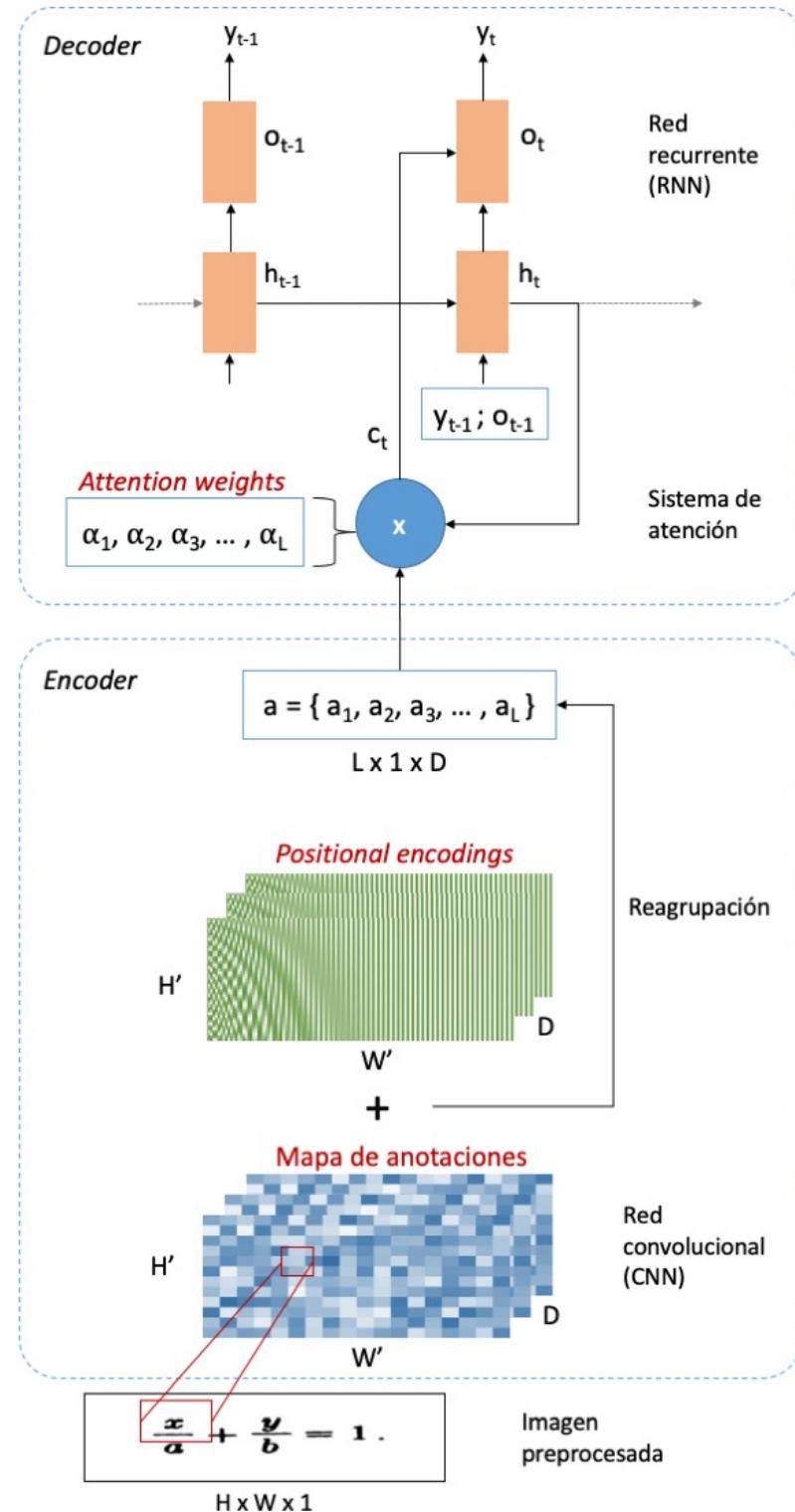
El problema de reconocer expresiones matemáticas en imágenes conciste en encontrar una función capaz de convertir una imagen preprocesada en una secuencia de caracteres que describa en su totalidad la expresión incluida en la imagen. La imagen de entrada x es preprocesada para obtener una imagen en escala de grises de tal forma que $x \in \mathbb{R}^{H \times W \times 1}$, siendo H y W la altura y el ancho de la imagen respectivamente. La secuencia de caracteres objetivo y es de la forma y_1, y_2, \dots, y_k siendo k la longitud de la secuencia y y_i es un carácter válido de L^AT_EX para el presente Trabajo Terminal.

Para resolver el problema de reconocer expresiones matemáticas, existen aproximaciones secuenciales o globales. De acuerdo con [35] ambos métodos implementados de una forma convencional presentan las siguientes limitaciones: 1) La difícil segmentación de símbolos, 2) el análisis estructural es comúnmente basado en gramáticas libres de contexto, lo cual requiere un conocimiento previo de las expresiones a reconocer para diseñar una gramática, 3) la complejidad de los algoritmos de parseo se incrementa con el tamaño de la gramática usada.

Recientemente el modelo de *Encoder-Decoder* con un sistema de atención comenzó a cobrar relevancia por presentar excelentes resultados en campos del machine learning como el procesamiento del lenguaje natural, segmentación de imágenes e *Image Captioning*. Esta aproximación es completamente opuesta a las soluciones convencionales debido a que es una solución entrenable *end-to-end*, es decir, el modelo puede entrenarse como un todo y no por separado; su capacidad de predicción depende completamente del conjunto de entrenamiento por lo que para mejorar el modelo solo se necesita incrementar la cantidad de datos disponibles sin hacer cambios a la arquitectura de la red; la segmentación de símbolos puede hacerse automáticamente a través de la atención. Por las razones expuestas, se decidió utilizar este modelo de deep learning en el presente Trabajo Terminal.

5.5.1. Modelo

Como se mencionó anteriormente la arquitectura de la red neuronal es del tipo *Encoder-Decoder*, en la Figura 5.22 se muestra un esquema de la red neuronal implementada. La traducción de imágenes se realiza de la siguiente manera: Primero se extraen las características de la imagen en forma de grilla utilizando una red convolucional (CNN), luego se procedera a utilizar una versión de dos dimensiones de los *positional encodings* mostrados en el marco teórico y finalmente se procede a reagrupar las filas de la grilla en una sola, esta es la fase de codificación.


 Figura 5.22: Modelo *Encoder-Decoder* de la red neuronal.

Número de capa	Capa Convolucional				Capa de Max Pooling			
	Kernel	Strides	Padding	BN	Size	Strides	Padding	
1	64-(3,3)	(1,1)	(1,1)	-	(2,2)	(2,2)	(2,2)	
2	128-(3,3)	(1,1)	(1,1)	-	(2,2)	(2,2)	(0,0)	
3	256-(3,3)	(1,1)	(1,1)	si			-	
4	256-(3,3)	(1,1)	(1,1)	-	(2,1)	(2,1)	(0,0)	
5	512-(3,3)	(1,1)	(1,1)	si	(1,2)	(1,2)	(0,0)	
6	512-(3,3)	(1,1)	(0,0)	si			-	

Tabla 5.1: Especificación de la CNN. El 'Kernel' esta denotado como *número de filtros-(dimensiones del filtro)*.

Para la decodificación, las características codificadas serán usadas por una red neuronal recurrent RNN con un sistema de atención. El *Decoder* implementa un modelo de lenguaje condicional sobre un vocabulario λ , el cual esta dado por el conjunto de entrenamiento y que será expuesto más adelante.

5.5.1.1. Encoder

La extracción de características de la imagen de entrada x se realiza con una CNN mult capa con *max pooling* y *batch normalization* la cual es ahora un estandar y está completamente inspirada en [36]. La Tabla 5.1 sumariza la arquitectura de la red utilizada. Las características extraídas están almacenadas en vector $v \in \mathbb{R}^{H' \times W' \times D}$, siendo H' y W' las nuevas altura y anchura respectivamente y D el tamaño del último filtro de la red, $D = 512$ en este caso.

Se procede a calcular los *positional encodings*. Existen dos formas de hacer esto, la primera es con parámetros aprendidos y la segunda es con valores fijos. El artículo [36] utiliza una RNN bidireccional con el vector v como entrada para aprender estos positional encodings. No obstante, acorde con los inventores de esta técnica [16], los positional encodings fijos producen resultados muy similares a los aprendidos. Así, se optó por utilizar positional encodings fijos y se utilizó la generalización a dos dimensiones propuesta por [37]. El vector de positional encodings $pe \in \mathbb{R}^{H' \times W' \times D}$ se obtiene con las siguientes ecuaciones:

$$pe(w, h, 2i) = \sin\left(\frac{w}{10000^{\frac{4i}{D}}}\right) \quad (5.3)$$

$$pe(w, h, 2i+1) = \cos\left(\frac{w}{10000^{\frac{4i}{D}}}\right) \quad (5.4)$$

$$pe(w, h, 2j+D/2) = \sin\left(\frac{h}{10000^{\frac{4j}{D}}}\right) \quad (5.5)$$

$$pe(w, h, 2j+1+D/2) = \cos\left(\frac{h}{10000^{\frac{4j}{D}}}\right) \quad (5.6)$$

Con esto, se puede obtener el vector de anotaciones a como sigue:

$$a = v + pe \quad (5.7)$$

Por último, se reagrupa el vector de anotaciones a de tal forma que $a \in \mathbb{R}^{LxD}$, donde $L = H'xW'$, así el vector final de salida del *Encoder* es $a = \{a_1, a_2, \dots, a_L\}$ con $a_i \in \mathbb{R}^D$.

5.5.1.2. Decoder

Las secuencias y_t se generan a partir de un modelo de lenguaje condicional que utiliza las anotaciones a obtenidas en el Encoder. La salida del decoder es la probabilidad de obtener el siguiente token dada la historia de tokens previamente generados y el vector de contexto obtenido mediante la atención.

El modelo de lenguaje condicional es como sigue:

$$p(y_t | y_1, \dots, y_{t-1}, a) = \text{softmax}(W^{out} o_t) \quad (5.8)$$

con

$$o_t = \tanh(W^c[h_t; c_t]) \quad (5.9)$$

$$h_t = LSTM(h_{t-1}, [y'_{t-1}; o_{t-1}]) \quad (5.10)$$

$$y'_t = E y_{t-1} \quad (5.11)$$

donde W^{out} , W^c y E son parámetros que serán aprendidos. La matriz E , es también conocida en la literatura como la matriz de *embeddings*. El vector h_t es utilizado para sumarizar toda la historia de decodificación y el vector de contexto c_t es usado para capturar la información de contexto de la grilla de características de la imagen x . Los corchetes $[;]$, indican una concatenación entre los vectores dentro de ellos.

Para los estados iniciales h_0 y o_0 se utilizó un perceptrón multicapa (MLP) que aprendiera los mejores valores iniciales. Las ecuaciones que describen h_0 y o_0 respectivamente son:

$$h_0 = \tanh(W_{h_0}a + b_{h_0}) \quad (5.12)$$

$$o_0 = \tanh(W_{o_0}a + b_{o_0}) \quad (5.13)$$

En cada instante t , el vector de contexto es calculado. Dado que la mayoría de anotaciones a no contienen información útil para decidir el mejor candidato y_t , el modelo debe de saber cuáles anotaciones atender. Esta tarea, es delegada a un sistema de atención como el propuesto por [15]:

$$e_t = u(h_t, a) \quad (5.14)$$

$$\alpha_t = \text{softmax}(e_t) \quad (5.15)$$

$$c_t = \phi(\alpha_t, a) \quad (5.16)$$

ϕ y u son funciones que pueden ser escogidas libremente, en este caso se utilizaron las mismas que en [36]:

$$e_{it} = \beta^T \tanh(W_h h_{i-1} + W_v a) \quad (5.17)$$

$$c_t = \sum_i^L \alpha_{it} a_i \quad (5.18)$$

donde W_h y W_v son parámetros que serán aprendidos. Con esto el modelo de atención puede saber que parte de las anotaciones es importante en el instante t . Una consecuencia de esto, es que la segmentación de símbolos es delegada también a este sistema.

Finalmente, se decidió modificar la atención para implementar la técnica del *Coverage Vector* como en [22]. El coverage vector F se incorporó para dar más información al sistema de atención sobre el historial de la misma atención, de este modo, es posible mejorar la cobertura global del sistema lo cual ayuda a reducir los errores de símbolos no parseados. El vector de coverage se calcula como sigue:

$$\eta_t = \sum_l^{t-1} \alpha_l \quad (5.19)$$

$$F = Q * \eta_t \quad (5.20)$$

Donde η_t es la suma de todos los pesos de las atenciones pasadas y Q es una matriz de pesos que serán aprendidos. De este modo, la Ecuación 5.17 se modifica por:

$$e_{it} = \beta^T \tanh(W_h h_{i-1} + W_v a + W_f F) \quad (5.21)$$

de donde W_f es una matriz de pesos que serán aprendidos.

5.5.1.3. Implementación

El lenguaje de programación utilizado para desarrollar la red fue Python 3 junto con el framework para deep learning TensorFlow 2. Las unidades usadas en el estado h_t de la LSTM son 512 y 80 para la dimensión de *embedding* para los tokens 80. El algoritmo de optimización utilizado fue Adam con la Crosentropia Categórica como función de perdida.

El código que implementa la red es el siguiente:

```

1 def get_positional_encoding_2d(height, width, d_model):
2
3     positional_encodings = np.zeros((height, width, d_model))
4     d_model = int(d_model / 2)
5     h_vector = np.arange(height)
6     w_vector = np.expand_dims(np.arange(width), axis=1)
7
8     div_term = np.arange(d_model) // 2
9     div_term = np.exp((-2 * div_term / d_model) * np.log(10000.0))
10
11    positional_encodings[:, :, 0:d_model:2] = np.sin((positional_encodings[:, :, 0:d_model:2] + div_term[0::2]) * w_vector)
12    positional_encodings[:, :, 1:d_model:2] = np.cos((positional_encodings[:, :, 1:d_model:2] + div_term[1::2]) * w_vector)
13    positional_encodings[:, :, d_model::2] = np.sin((positional_encodings[:, :, d_model::2] + div_term[0::2]) * h_vector[:, np.newaxis, np.newaxis])
14    positional_encodings[:, :, d_model+1::2] = np.cos((positional_encodings[:, :, d_model+1::2] + div_term[0::2]) * h_vector[:, np.newaxis, np.newaxis])
15
16    return positional_encodings
17
18
19 class Encoder(tf.keras.Model):
20
21     def __init__(self):
22         super(Encoder, self).__init__()
23
24         self.pooling1 = tf.keras.layers.MaxPool2D(pool_size=(2,2), strides
```

```
25         =(2,2), padding='same')
26         self.pooling2 = tf.keras.layers.MaxPool2D(pool_size=(2,2), strides
27         =(2,2), padding='same')
28         self.pooling3 = tf.keras.layers.MaxPool2D(pool_size=(2,1), strides
29         =(2,1), padding='same')
30         self.pooling4 = tf.keras.layers.MaxPool2D(pool_size=(1,2), strides
31         =(1,2), padding='same')
32
33         self.conv1 = tf.keras.layers.Conv2D(64, (3,3), (1,1), 'same',
34         activation='relu')
35         self.conv2 = tf.keras.layers.Conv2D(128, (3,3), (1,1), 'same',
36         activation='relu')
37         self.conv3 = tf.keras.layers.Conv2D(256, (3,3), (1,1), 'same',
38         activation='relu')
39         self.conv4 = tf.keras.layers.Conv2D(256, (3,3), (1,1), 'same',
40         activation='relu')
41         self.conv5 = tf.keras.layers.Conv2D(512, (3,3), (1,1), 'same',
42         activation='relu')
43         self.conv6 = tf.keras.layers.Conv2D(512, (3,3), (1,1), 'valid',
44         activation='relu')
45
46         self.batch_normalization1 = tf.keras.layers.BatchNormalization()
47         self.batch_normalization2 = tf.keras.layers.BatchNormalization()
48         self.batch_normalization3 = tf.keras.layers.BatchNormalization()
49
50     def call(self, x, training=False):
51
52         y = self.conv1(x)
53         y = self.pooling1(y)
54         y = self.conv2(y)
55         y = self.pooling2(y)
56         y = self.conv3(y)
57         y = self.batch_normalization1(y, training)
58         y = self.conv4(y)
59         y = self.pooling3(y)
```

```

50     y = self.conv5(y)
51     y = self.batch_normalization2(y, training)
52     y = self.pooling4(y)
53     y = self.conv6(y)
54     y = self.batch_normalization3(y, training)
55
56     y += get_positional_encoding_2d(y.shape[1], y.shape[2], y.shape[3])
57
58     y = tf.reshape(y, (y.shape[0], -1, y.shape[3]))
59
60     return y

```

```

1 class InitialHidden(tf.keras.Model):
2
3     def __init__(self, units):
4         super(InitialHidden, self).__init__()
5
6         self.fc = tf.keras.layers.Dense(units, activation='tanh')
7
8     def __call__(self, x):
9
10        x = tf.math.reduce_mean(x, axis=1)
11
12        return self.fc(x)
13
14
15 class BahdanauAttention(tf.keras.layers.Layer):
16     def __init__(self, units):
17         super(BahdanauAttention, self).__init__()
18
19         self.W1 = tf.keras.layers.Dense(units)
20         self.W2 = tf.keras.layers.Dense(units)
21         self.V = tf.keras.layers.Dense(1)
22
23         self.Q = tf.keras.layers.Conv1D(filters=16, kernel_size=64, padding="same",
24                                       use_bias=False)

```

```

23     self.Uf = tf.keras.layers.Dense(units , use_bias=False)
24
25
26 def call(self , features , hidden , B):
27     # features (Encoder output) shape == (batch_size , L, 512)
28
29     # hidden shape == (batch_size , hidden_size)
30     # hidden_with_time_axis shape == (batch_size , 1, hidden_size)
31     hidden_with_time_axis = tf.expand_dims(hidden , 1)
32
33     F = self.Q(B)
34
35     # score shape == (batch_size , L, hidden_size)
36     score = tf.nn.tanh(self.W1(features) + self.W2(hidden_with_time_axis) +
37                         self.Uf(F))
38
39     # attention_weights shape == (batch_size , L, 1)
40     # you get 1 at the last axis because you are applying score to self.V
41     attention_weights = tf.nn.softmax(self.V(score) , axis=1)
42
43     # context_vector shape after sum == (batch_size , hidden_size)
44     context_vector = attention_weights * features
45     context_vector = tf.reduce_sum(context_vector , axis=1)
46
47
48
49 class Decoder(tf.keras.Model):
50
51     def __init__(self , units , embedding_dim , vocab_size):
52         super(Decoder , self).__init__()
53
54         self.Wout = tf.keras.layers.Dense(vocab_size , use_bias=False)
55         self.o_t = tf.keras.layers.Dense(units , use_bias=False , activation='tanh')
56
57

```

```

56     self.lstm = tf.keras.layers.LSTM(units, return_state=True)
57     self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
58
59     self.attention = BahdanauAttention(units)
60
61
62 def __call__(self, features, hidden, previous_y, previous_out):
63
64     # embedding_previous_y shape == (batch_size, embedding_dim)
65     embedding_previous_y = self.embedding(previous_y)
66
67     # previous_out shape == (batch_size, units)
68     # lstm_input shape == (batch_size, embedding_dim + units)
69     lstm_input = tf.concat([embedding_previous_y, previous_out], axis=1)
70     lstm_input = tf.expand_dims(lstm_input, axis=1)
71
72     # hidden is a list of hidden state and carry state respectively
73     # each element has a shape == (batch_size, units)
74     # actually output and state are the same tensors
75     output, state, carry = self.lstm(lstm_input, initial_state=hidden)
76
77     # context_vector shape == (batch_size, hidden_size)
78     # attention_weights == (batch_size, L, 1)
79     context_vector, attention_weights = self.attention(features, state)
80
81     # calculating the output
82     out = self.o_t(tf.concat([output, context_vector], axis=1))
83     output = self.Wout(out)
84
85     return output, out, attention_weights, [state, carry]

```

5.5.1.4. Entrenamiento

La red fue entranada en el entorno Colaboratory de Google utilizando una GPU, el proceso de entrenamiento es de alrededor de 30 horas.

Gracias al uso del framework Tensorflow 2, el entrenamiento de la red se puede simplificar mucho. El cálculo de los gradientes de la función de perdida se puede automatizar así como la actualización de los pesos, no obstante, dada la arquitectura de la red, no es posible utilizar los métodos *compile* y *fit* implementados en Keras para Tensorflow 2. Por lo tanto, el ciclo de entrenamiento tuvo que ser diseñado acorde con las necesidades de la arquitectura propuesta.

El código del ciclo de entrenamiento es el siguiente:

```

1 @tf.function
2 def train_step(img_tensor, target):
3     loss = 0
4     loss_counter = 0
5     loss_before = tf.constant(0.0)
6
7     with tf.GradientTape() as tape:
8         features = encoder(img_tensor, True)
9         tf.debugging.assert_all_finite(features, 'the features have exploded')
10
11     hidden = initial_state(features)
12     carry = initial_carry(features)
13     out = initial_out(features)
14     dec_input = tf.constant([BEGIN] * target.shape[0])
15
16     hidden_and_carry = [hidden, carry]
17
18     B = tf.zeros((features.shape[0], features.shape[1], 1))
19
20     for i in range(1, target.shape[1]):
21         # passing the features through the decoder
22         predictions, out, attention_weights, hidden_and_carry = decoder(
23             features, hidden_and_carry, dec_input, out, B)
24         B += attention_weights
25
26         real = get_num_token_end(target[:, i])
27
28         loss += loss_function(real, predictions)
29         tf.debugging.assert_all_finite(loss, 'the loss has exploded')
```

```
29
30         # is_done = tf.where(tf.less(real, end_tensor), is_done, end_tensor
31     )
32
33     if loss != loss_before:
34         loss_counter += 1
35         loss_before = loss
36
37     # using teacher forcing
38     dec_input = real
39
40     total_loss = (loss / int(target.shape[1]))
41
42 trainable_variables = encoder.trainable_variables + initial_state.
43 trainable_variables + initial_carry.trainable_variables + initial_out.
44 trainable_variables + decoder.trainable_variables
45
46 gradients = tape.gradient(loss, trainable_variables)
47
48 for g in gradients:
49     tf.debugging.assert_all_finite(g, 'the gradients have exploded')
50
51 optimizer.apply_gradients(zip(gradients, trainable_variables))
52
53 train_loss(loss)
54
55 # train_accuracy(y_train, predictions)
56
57 return loss, total_loss, loss_counter
58
59
60 start = time.time()
61 total_loss = 0
62 total_val_loss = 0
63
64 for (batch, (img_tensor, target)) in enumerate(dataset):
```

```

61 # img_tensor = noise(img_tensor, training=True)
62 batch_loss, t_loss, loss_counter = train_step(img_tensor, target)
63 total_loss += t_loss
64
65 print ('Epoch {} Batch {} Loss {:.6f}'.format(
66     epoch, batch, batch_loss.numpy() / loss_counter))

```

5.5.1.5. Resultados

5.5.1.6. Experimentos Previos

Antes de encontrar la configuración adecuada para la red neuronal, se llevaron a cabo varios experimentos. A continuación, se presentan los más relevantes.

Variante con GRU

Se utilizó una GRU como en [38] en lugar de una LSTM en el *Decoder*. No existe un consenso respecto a cual tipo de RNN basadas en compuertas es mejor, por lo que lo más recomendable es experimentar con ambas. En este caso particular, la GRU no logró abstraer las características del problema. El código fuente del *Decoder* con GRU se muestra a continuación:

```

1 class RNN_Decoder(tf.keras.Model):
2     def __init__(self, embedding_dim, units, vocab_size):
3         super(RNN_Decoder, self).__init__()
4         self.units = units
5
6         self.embedding = tf.keras.layers.Embedding(vocab_size, embedding_dim)
7
8         self.gru = tf.keras.layers.GRU(self.units,
9                                         return_sequences=True,
10                                        return_state=True,
11                                        recurrent_initializer='glorot_uniform')
12
13         self.fc1 = tf.keras.layers.Dense(self.units, kernel_regularizer=tf.keras.
14                                         regularizers.l1_l2(l1=0.01, l2=0.01))
15         self.fc2 = tf.keras.layers.Dense(vocab_size, kernel_regularizer=tf.keras.
16                                         regularizers.l1_l2(l1=0.01, l2=0.01))

```

```

16     self.attention = BahdanauAttention(self.units)

17

18 def call(self, x, features, hidden):
19     # defining attention as a separate model
20     context_vector, attention_weights = self.attention(features, hidden)

21

22     # x shape after passing through embedding == (batch_size, 1,
23     embedding_dim)
24     x = self.embedding(x)

25

26     # x shape after concatenation == (batch_size, 1, embedding_dim +
27     hidden_size)
28     x = tf.concat([tf.expand_dims(context_vector, 1), x], axis=-1)

29

30     # passing the concatenated vector to the GRU
31     output, state = self.gru(x, initial_state=hidden)

32

33     # shape == (batch_size, max_length, hidden_size)
34     x = self.fc1(output)

35

36     # x shape == (batch_size * max_length, hidden_size)
37     x = tf.reshape(x, (-1, x.shape[2]))

38

39     # output shape == (batch_size * max_length, vocab)
40     x = self.fc2(x)

41

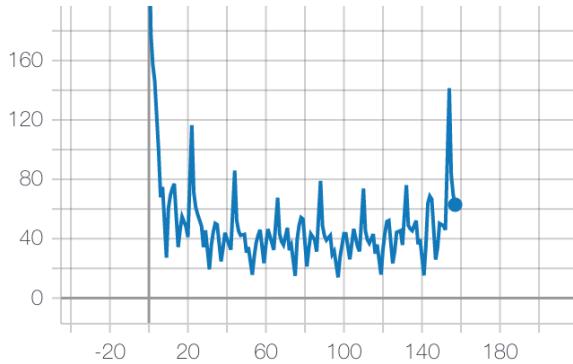
42     return x, state, attention_weights

43

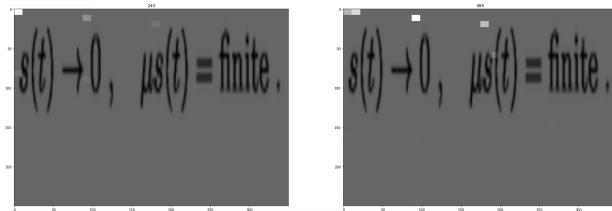
44     def reset_state(self, batch_size):
45         return tf.zeros((batch_size, self.units))

```

Una muestra del resultado del entrenamiento se muestra en la Figura 5.23, en (a) vemos que la función de perdida se estanca, demostrando que no es capaz de modelar el problema en su totalidad mientras que en (b), la atención no mostró aprendizaje alguno.



(a) Función de perdida más característica del modelo.



(b) Atención del sistema.

Figura 5.23: (a) Función de perdida del sistema, el modelo se entreno en varias ocasiones, siempre obteniendo una gráfica parecida. (b) Se muestra la atención obtenida por este modelo, se observa un aprendizaje nulo.

Variante con Dropout

Acorde con [13], el *dropout* es una técnica utilizada para regularizar redes neuronales, es decir, para que la probabilidad de que un modelo se sobreentrene disminuya. Consiste en aplicar un muestreo a las salidas de alguna capa en un modelo con la finalidad de desactivar un porcentaje de neuronas. Suponiendo a μ como el vector de muestreo y a $p(\mu)$ como la distribución de probabilidad que modela si la neurona se desactiva o no, podemos aplicar dropout a una capa de alguna red neuronal que modele una distribución $p(y|x)$ de la siguiente manera:

$$P' = p(\mu)p(y|x, \mu) \quad (5.22)$$

De este modo, algunas neuronas [36] son "desconectadas" haciendo que cada neurona aprenda sobre la información proveída en el entrenamiento de manera independiente. Esto produce también, un efecto de regularización.

La técnica del dropout es recomendada en la mayoría de los modelos de deep learning, sin embargo en el caso del presente Trabajo Terminal no tuvo ningún efecto. Esto es debido posiblemente a que el modelo no es suficientemente profundo para mostrar los efectos del dropout. En

la Figura 5.24 se pueden ver los resultados de generales de este modelo.

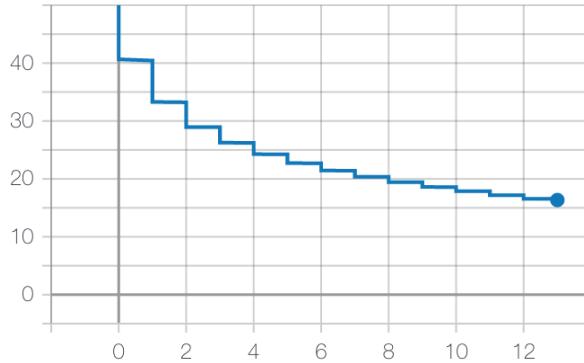
```
1 class CNN_Layer(tf.keras.layers.Layer):  
2  
3     def __init__(self, num_filters, kernel_size, padding,  
4                  apply_batch_normalization=False, apply_dropout=False, dropout_rate=0.5):  
5         super(CNN_Layer, self).__init__()  
6  
7         self.conv = tf.keras.layers.Conv2D(filters=num_filters, kernel_size=  
8                                           kernel_size,  
9                                           strides=(1,1), padding=padding,  
10                                         use_bias=False)  
11  
12         self.batch_normalization = tf.keras.layers.BatchNormalization()  
13         self.dropout = tf.keras.layers.Dropout(rate=dropout_rate)  
14         self.apply_batch_normalization = apply_batch_normalization  
15         self.apply_dropout = apply_dropout  
16  
17     def __call__(self, x, training=False):  
18  
19         y = self.conv(x)  
20         if self.apply_batch_normalization:  
21             y = self.batch_normalization(y, training)  
22         y = tf.nn.relu(y)  
23         if self.apply_dropout:  
24             y = self.dropout(y, training)  
25  
26     return y  
27  
28 class CNN_Encoder(tf.keras.Model):  
29  
30     def __init__(self, embedding_dim):  
31         super(CNN_Encoder, self).__init__()
```

```

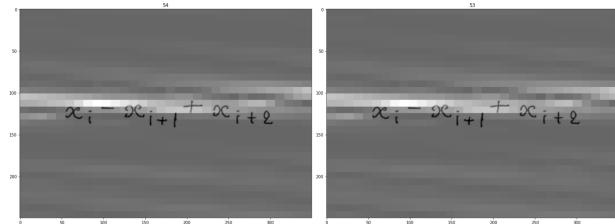
31     self.pooling1 = tf.keras.layers.MaxPool2D(pool_size=(2,2), strides
32         =(2,2))
33     self.pooling2 = tf.keras.layers.MaxPool2D(pool_size=(2,2), strides
34         =(2,2), padding='same')
35     self.pooling3 = tf.keras.layers.MaxPool2D(pool_size=(2,1), strides
36         =(2,1), padding='same')
37     self.pooling4 = tf.keras.layers.MaxPool2D(pool_size=(1,2), strides
38         =(1,2), padding='same')
39
40         self.conv1 = CNN_Layer(64, (3,3), 'valid', apply_dropout=True,
41         dropout_rate=0.2)
42         self.conv2 = CNN_Layer(128, (3,3), 'valid')
43         self.conv3 = CNN_Layer(256, (3,3), 'valid', apply_batch_normalization
44         =True)
45         self.conv4 = CNN_Layer(256, (3,3), 'valid', apply_dropout=True)
46         self.conv5 = CNN_Layer(512, (3,3), 'valid', apply_batch_normalization
47         =True, apply_dropout=True)
48         self.conv6 = CNN_Layer(512, (3,3), 'same', apply_batch_normalization=
49         True)
50
51     def call(self, x, training=False):
52
53         y = self.conv1(x, training)
54         y = self.pooling1(y)
55         y = self.conv2(y, training)
56         y = self.pooling2(y)
57         y = self.conv3(y, training)
58         y = self.conv4(y, training)
59         y = self.pooling3(y)
60         y = self.conv5(y, training)
61         y = self.pooling4(y)
62         y = self.conv6(y, training)
63
64         y += get_positional_encoding_2d(y.shape[1], y.shape[2], y.shape[3])
65         y = tf.reshape(y, (y.shape[0], -1, y.shape[3]))

```

58

59 **return** y

(a) Función de perdida más característica del modelo.



(b) Atención del sistema.

Figura 5.24: (a) Función de perdida del sistema, el modelo se entreno en varias ocasiones, siempre obteniendo una gráfica parecida. (b) Se muestra la atención obtenida por este modelo, se observa un que el modelo aprendió que a indentificar donde estaba la ecuación, no obstante no aprendio a segmentar los símbolos.

Capítulo 6

Pruebas del sistema

En este capítulo se detallan las diversas pruebas realizadas a los módulos desarrollados con el fin de verificar y validar su correcto funcionamiento.

6.1. Pruebas unitarias

6.1.1. Aplicación android

Para realizar pruebas unitarias en android se utilizo la biblioteca JUnit la cual es utilizada para realizar pruebas unitarias en Java.

JUnit se utilizo sobre aquellas clases que se utilizaron para trabajar partes en específico de la lógica del negocio que no involucraban utilitarias de android, es decir, que solamente utilizan Java.

6.1.1.1. Pruebas sobre la clase DateFormater

La clase DateFormater se utiliza en la aplicación para obtener fechas y presentarlas en un formato correcto en pantalla, así como pasar de una cadena de texto a un objeto de tipo Date, por lo cual las pruebas realizadas sobre esta clase fueron tres.

1. En la primera se valida que la conversión de una cadena de texto a un objeto de tipo Date se haga correctamente siempre y cuando se trate de una cadena valida.
2. En la segunda prueba se valida que se pueda pasar de un tipo date a una cadena de texto y que el valor que se obtenga sea el mismo.
3. En la ultima prueba se valida que se lance una excepción si el valor a convertir a una cadena de texto no es valido.

El código utilizado en cada una de las pruebas es el siguiente.

```
1 package com.equipo.superttapp;
2
3 import com.equipo.superttapp.util.DateFormater;
4
5 import org.junit.Assert;
6 import org.junit.Test;
7
8 import java.util.Calendar;
9 import java.util.Date;
10
11 public class DateFormaterTest {
12     @Test
13     public void convertStringToDate_validString_shouldParseToDate() {
14         Calendar cal = Calendar.getInstance();
15         cal.set(2020, 0, 1, 0, 0, 0);
16         cal.setTimeInMillis(1577858400000L);
17         Date date = cal.getTime();
18         Assert.assertEquals("Probando que se obtenga un objeto date con la
19             fecha en string", date.getTime(), DateFormater.convertStringToDate("01/01/2020").getTime());
20     }
21
22     @Test
23     public void convertDateToString_validDate_shouldParseToString() {
24         Calendar cal = Calendar.getInstance();
25         cal.set(2020, 0, 1);
26         Date date = cal.getTime();
27         Assert.assertEquals("Probando que no se obtenga null despues de
28             realizar un cast", "01/01/2020", DateFormater.convertDateToString(date));
29     }
30
31     @Test(expected = NullPointerException.class)
32     public void convertDateToString_NoValidDate_shouldThrowException() {
33         DateFormater.convertDateToString(null);
34     }
```

33 }

En el código anterior, cada uno de los métodos anotados con @Test es una prueba a ejecutar, la forma en la que se valida que la ejecución sea correcta es con la linea de código correspondiente a Assert.assertEquals, la cual compara dos valores y los compara, en caso de ser iguales la prueba es exitosa en caso contrario la prueba ha fallado.

En la ultima prueba la anotación @Test tiene como parámetro una excepción a esperar, si la excepción se dispara la prueba es correcta, por otro lado, si no se da el caso la prueba ha fallado.

La correcta ejecución de las pruebas se puede apreciar en la figura ??

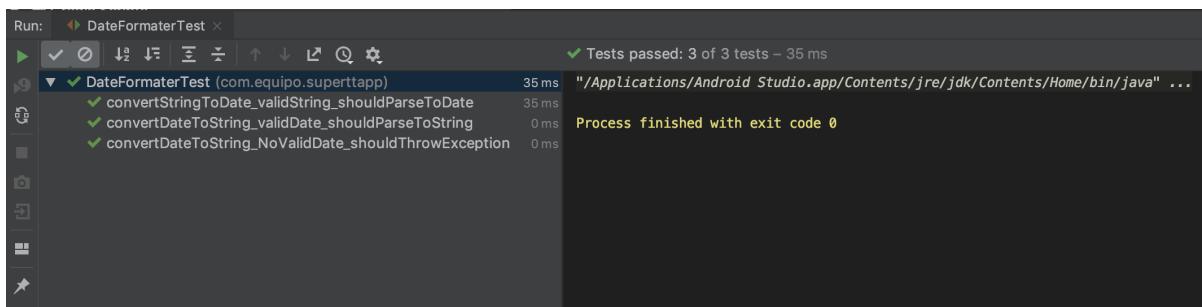


Figura 6.1: Resultados de las pruebas de la clase DateFormater

6.1.1.2. Pruebas sobre la clase RNN002

La clase RNN002 se utiliza en la aplicación para validar el formato de diversos campos que se tienen en los diferentes formularios que incluye la aplicación, es por esto que las pruebas que se realizaron en este caso fueron las siguientes.

1. En la primera prueba se verifica que para un email valido, el resultado sea true.
2. En la siguiente prueba, para un email no valido, el resultado debe de ser false.
3. En la siguiente prueba, para una contraseña valida, el resultado debe de ser true.
4. En la siguiente prueba, para una contraseña no valida, el resultado debe de ser false.
5. En la siguiente prueba, para dos contraseñas validas, el resultado debe de ser true.
6. En la siguiente prueba, para dos contraseñas, una de ellas no valida, el resultado debe de ser false.
7. En la siguiente prueba, para un nombre valido, el resultado debe de ser true.
8. En la siguiente prueba, para un nombre no valido, el resultado debe de ser false.
9. En la siguiente prueba, para un apellido valido, el resultado debe de ser true.

10. En la siguiente prueba, para un apellido no valido, el resultado debe de ser false.
11. En la siguiente prueba, para un nombre de proyecto valido, el resultado debe de ser true.
12. En la ultima prueba, para un nombre de proyecto no valido, el resultado debe de ser false.

El código utilizado en cada una de las pruebas es el siguiente.

```
1 package com.equipo.superttapp;  
2  
3 import com.equipo.superttapp.util.RN002;  
4  
5 import org.junit.Assert;  
6 import org.junit.Test;  
7  
8 public class RN002Test {  
9     @Test  
10    public void isEmailValid_validEmail_shouldReturnTrue() {  
11        Assert.assertTrue("Prrobando que se regrese true en una email valido"  
12            , RN002.isEmailValid("carlostonatihu@gmail.com"));  
13    }  
14  
15    @Test  
16    public void isEmailValid_invalidEmail_shouldReturnFalse() {  
17        Assert.assertFalse("Prrobando que se regrese false en una email no  
18            valido", RN002.isEmailValid("carlostonatihumail.com"));  
19    }  
20  
21    @Test  
22    public void isPassordValid_validPassword_shouldReturnTrue() {  
23        Assert.assertTrue("Debe retornar true ya que la contrasena es valida"  
24            , RN002.isPasswordValid("madremiawilly"));  
25    }  
26  
27    @Test  
28    public void isPassordValid_invalidPassword_shouldReturnFalse() {  
29        Assert.assertFalse("Debe retornar false ya que la contrasena es  
30            invalida", RN002.isPasswordValid(""));
```

```
27    }
28
29    @Test
30    public void isSecondPassordValid_validPassword_shouldReturnTrue() {
31        Assert.assertTrue("Debe retornar true ya que las dos contrasenas son
32        validas", RN002.isSecondPasswordValid("madremiawilly", "madremiawilly"));
33    }
34
35    @Test
36    public void isSecondPassordValid_invalidPassword_shouldReturnFalse() {
37        Assert.assertFalse("Debe retornar true ya que las dos contrasenas son
38        validas", RN002.isSecondPasswordValid("madremiawilly", null));
39    }
40
41    @Test
42    public void isNameValid_validName_shouldReturnTrue() {
43        Assert.assertTrue("Debe retornar true ya que el nombre es valido",
44        RN002.isNameValid("Carlos Tonatiuh"));
45    }
46
47    @Test
48    public void isNameValid_invalidName_shouldReturnFalse() {
49        Assert.assertFalse("Debe retornar false ya que el nombre no es valido",
50        RN002.isNameValid(" "));
51    }
52
53    @Test
54    public void isLastNameValid_validLastName_shouldReturnTrue() {
55        Assert.assertTrue("Debe retornar true ya que el apellido es valido",
56        RN002.isLastnameValid("Barrera"));
57    }
58
59    @Test
60    public void isLastNameValid_invalidLastName_shouldReturnFalse() {
61        Assert.assertFalse("Debe retornar false ya que el apellido no es
```

```

    valido", RN002.isNameValid(null));
57 }
58
59     @Test
60     public void isProyectoNombreValid_validProyectoNombre_shouldReturnTrue()
61     {
62         Assert.assertTrue("Debe retornar true ya que el nombre del proyecto
63             es valido", RN002.isProyectoNombreValid("Proyecto 1"));
64     }
65
66     @Test
67     public void isProyectoNombreValid_invalidProyectoNombre_shouldReturnFalse
68     () {
69         Assert.assertTrue("Debe retornar false ya que el nombre del proyecto
70             no es valido", RN002.isProyectoNombreValid("1"));
71     }
72 }
```

En el caso de estas pruebas presentadas en el código anterior, se utilizaron los métodos `Assert.assertTrue` y `Assert.assertFalse` para verificar que las pruebas fueran correctas. En el caso del primero de estos métodos la prueba es exitosa si el valor que retorna el método a probar es falso, mientras que para el segundo método si el valor de retorno es falso la prueba es correcta.

La ejecución de las pruebas se muestra en la figura 6.2.

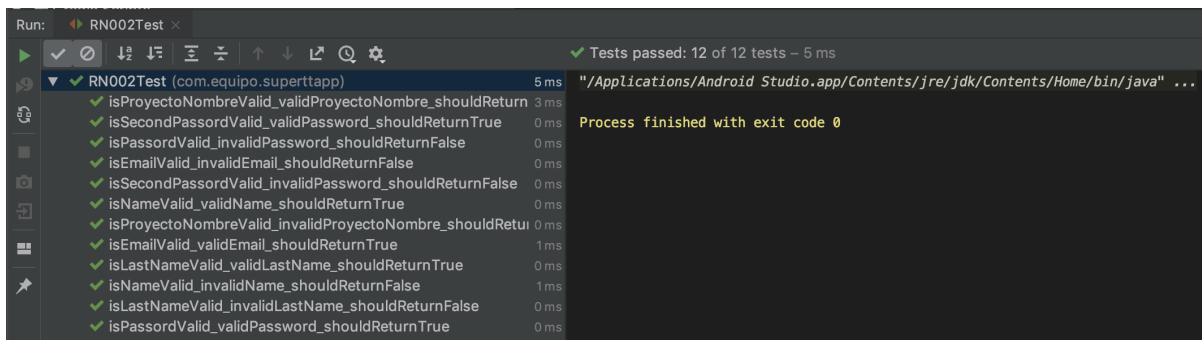


Figura 6.2: Resultados de las pruebas de la clase RNN002

6.1.2. Aplicación web

Para realizar pruebas unitarias a la aplicación web desarrollado con Django se utilizo la herramienta que permite analizar que partes del código de un programa se están ejecutando y con ello determinar que bloques de código se deben de someter a pruebas.

Al utilizar coverage.py sobre el proyecto de Django se obtuvieron los resultados que se muestran en la figura 6.3. El reporte que coverage.py arroja muestra la cantidad de código que se tiene que probar, el código que falta por probar, el excluido y el que se tienen cubierto con las pruebas.

Se logró cubrir el 100% del código y se excluyeron algunas partes debido a que forman parte de los archivos de configuración de Django y porque fueron pruebas complicadas de elaborar.

Coverage report: 100%				
Module	statements	missing	excluded	coverage ↑
general/util.py	136	0	1	100%
manage.py	7	0	2	100%
proyectos/__init__.py	0	0	0	100%
proyectos/admin.py	1	0	0	100%
proyectos/apps.py	3	0	0	100%
proyectos/models.py	17	0	0	100%
proyectos/serializers.py	14	0	0	100%
proyectos/tests.py	1	0	0	100%
proyectos/urls.py	9	0	0	100%
proyectos/views.py	130	0	189	100%
supertt/__init__.py	0	0	0	100%
supertt/settings.py	31	0	0	100%
supertt/urls.py	12	0	0	100%
users/__init__.py	0	0	0	100%
users/admin.py	11	0	0	100%
users/apps.py	3	0	0	100%
users/models.py	49	0	0	100%
users/tests.py	1	0	0	100%
users/urls.py	6	0	0	100%
users/views.py	263	0	26	100%
utils/__init__.py	0	0	0	100%
Total	694	0	218	100%

coverage.py v5.1, created at 2020-06-07 21:43

Figura 6.3: Reporte del código sometido a pruebas

Para realizar pruebas sobre estos bloques de código encontrados, se utilizó el módulo de pruebas unitarias con el que cuenta Django y se realizaron pruebas sobre los modelos, vistas y las

clases de utilitaria que se desarrollaron.

Como se puede apreciar en la figura 6.4, se realizaron 71 pruebas sobre modelos, vistas y utilería, al final siendo exitosas cada una de ellas.

```
(venv) Tonas-MacBook-Pro:supertt tonatihu$ coverage run --omit='*/venv/*,*/migrations/*' manage.py test -v 1
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....http://192.168.1.68:8000/media/proyectos/1/1/imagen
.....url: http://192.168.1.68:8000/usuarios/contrasena?token=b214ffdda0f04225c0a
.Ocurrio un error: 'correo'
.....Un error: UserHashes matching query does not exist.
.....
-----
Ran 71 tests in 18.837s
OK
Destroying test database for alias 'default'...
```

Figura 6.4: Resultado de las pruebas de Django

La forma en la que se elaboraron estas pruebas fue la siguiente.

6.1.2.1. Pruebas sobre los modelos

Un ejemplo de las pruebas sobre modelos es el siguiente código, en el cual se prueba el modelo del usuario y en cada uno de los métodos que comienzan con la palabra test determinan los diferentes casos de prueba a elaborar, que en este ejemplo son sobre el guardado de información de un usuario.

```
1 from django.test import TestCase
2 from users.models import User
3
4 class UserManagerTest(TestCase):
5     def test_create_user(self):
6         user = User.objects.create_user(email="hola@gmail.com", password="hola")
7         user.save()
8         self.assertEqual(user.email, "hola@gmail.com")
9
10    def test_create_superuser(self):
11        user = User.objects.create_superuser(email="hola@gmail.com", password="hola")
12        user.save()
13        self.assertEqual(user.email, "hola@gmail.com")
14
```

```

15     def test_create_user_not_email(self):
16         with self.assertRaises(ValueError):
17             user = User.objects.create_user(email=None, password="hola")
18
19     def test_create_superuser_is_staff_false(self):
20         with self.assertRaises(ValueError):
21             user = User.objects.create_superuser(email="hola", password="hola",
22                                                 is_staff=False)
23
24     def test_create_superuser_is_superuser_false(self):
25         with self.assertRaises(ValueError):
26             user = User.objects.create_superuser(email="hola", password="hola",
27                                                 is_superuser=False)

```

En las primeras dos pruebas se utiliza el método assertEquals el cual permite comparar el resultado de una prueba con el esperado para determinar si la prueba fue exitosa.

Para las siguientes tres pruebas como se espera que se arroje una excepción se utiliza la declaración with para determinar si la excepción esperada ocurrió y con ello asegurar que la prueba fue exitosa, si esto no ocurre la prueba no es correcta.

6.1.2.2. Pruebas sobre los vistas

Las vistas, al ser el componente encargado de presentar el contenido al usuario se tiene que simular la petición HTTP que se hará y se tiene que verificar si la respuesta que se genera es la adecuada. Esto se puede apreciar en el siguiente código, en el cual se prueba el correcto funcionamiento de la vista encargada de enviar el correo electrónico.

Para realizar esto, lo primero es crear un usuario de prueba en el método setUp de la clase, este método se ejecuta por cada test dentro de la clase y se elimina lo creado en este método al finalizar cada prueba.

Después, utilizando el método reverse se obtiene la URL asociado a la vista a probar, se realiza la petición correspondiente y se evalúa el resultado obtenido.

```

1 from django.test import TestCase
2 from django.urls import reverse
3 from users.models import User
4
5 class EnviarCorreoRecuperacionView(TestCase):
6     def setUp(self):
7         self.user = User.objects.create_user(email="hola@gmail.com", password="hola")

```

```

8     ="hola")
9
10    def test_enviar_correo_error(self):
11        url = reverse("enviar_correo_recuperacion_view")
12        response = self.client.post(url)
13        self.assertEqual(response.status_code, 302)
14
15    def test_enviar_correo(self):
16        url = reverse("enviar_correo_recuperacion_view")
17        response = self.client.post(url, {"correo": "hola@gmail.com"})
18        self.assertTrue(self.client.session["resp"] is not None)

```

En la primera prueba, se espera que se arroje un error debido a que al ser una petición de tipo post se espera recibir información la cual no se envía, por lo que el código de respuesta será 302 indicando que se hará una redirección a otra pagina.

En la segunda prueba, ya que si se envía información correcta lo que se espera es que la respuesta que se obtiene contenga la variable de sesión con el nombre: resp.

6.1.2.3. Pruebas sobre las clases de utilería

Un ejemplo del como se probaron las clases de utilería se muestra en el siguiente código, en el cual se prueba la clase Validator que se uso para asegurar que los datos ingresados en los formularios fueran correctos.

Para este caso se crea una instancia de la clase Validator en el método setUp de la clase junto a un usuario de prueba. En cada una de las pruebas que se tiene se verifica el funcionamiento de ca método de la clase Validator mediante el uso de los métodos assertFalse y assertTrue que se encargan de validar que el valor de retorno de cada método sometido a prueba sea el esperado.

```

1 from django.test import TestCase
2 from general.util import Validator
3 from users.models import User
4 from proyectos.models import Proyecto
5
6 class ValidatorTest(TestCase):
7     def setUp(self):
8         self.validator = Validator()
9         self.user = User.objects.create_user(email="hola@gmail.com", password
10         ="hola")

```

```
10     proyecto = Proyecto.objects.create(usuario=self.user, nombre=""
11         Proyecto 1", calificacion=0)
12
13     def test_are_the_same_true(self):
14         self.assertTrue(self.validator.are_the_same(1, 1))
15
16     def test_are_the_same_false(self):
17         self.assertFalse(self.validator.are_the_same(1, 2))
18
19     def test_field(self):
20         self.assertFalse(self.validator.field(None))
21
22     def test_user_exists_true(self):
23         self.assertTrue(self.validator.user_exists("hola@gmail.com"))
24
25     def test_user_exists_false(self):
26         self.assertFalse(self.validator.user_exists("error@gmail.com"))
27
28     def test_email(self):
29         self.assertFalse(self.validator.email(""))
30
31     def test_min_len_true(self):
32         self.assertTrue(self.validator.min_len("contrasegura"))
33
34     def test_min_len_false(self):
35         self.assertFalse(self.validator.min_len("error"))
36
37     def test_max_len_false(self):
38         self.assertFalse(self.validator.max_len("contrasegura"*20))
39
40     def test_max_len_true(self):
41         self.assertTrue(self.validator.max_len("error"))
42
43     def test_name_true(self):
44         self.assertTrue(self.validator.name("Carlos Tonatiuh"))
```

```
44
45     def test_name_false(self):
46         self.assertFalse(self.validator.name("Tona777"))
47
48     def test_proyect_exists_true(self):
49         self.assertTrue(self.validator.proyect_exists(self.user, "Proyecto 1"))
50
51     def test_proyect_exists_false(self):
52         self.assertFalse(self.validator.proyect_exists(self.user, "Proyecto 2"))
```

Bibliografía

- [1] R. C. Martin, “The Clean Architecture.” <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>, 2012. [Consultado: 2019-09-21]. VI, 29
- [2] K. Singhal, “CROHME Data Extractor.” <https://github.com/karan1149/crohme-data-extractor>, 2017. [Consultado: 2018-11-01]. VI, VI, VI, VI, VII, VII, 46, 47, 48, 49, 50, 51
- [3] M. S. de Lorenzo, “Clean Architecture Guide (with tested examples): Data Flow != Dependency Rule.” <https://proandroiddev.com/clean-architecture-data-flow-dependency-rule-615ffdd79e29>, 2018. [Consultado: 2019-09-21]. VII, 122
- [4] F. Cejas, “Architecting Android...The clean way?” <https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/>, 2014. [Consultado: 2019-09-21]. VII, VII, 124, 129
- [5] F. Cejas, “Architecting Android...Reloaded.” <https://fernandocejas.com/2018/05/07/architecting-android-reloaded/>, 2018. [Consultado: 2019-09-21]. VII, 135
- [6] “Mahpix.” <https://mathpix.com/about>. 3
- [7] “MyScript Nebo.” <https://www.nebo.app/es/>. 3
- [8] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, “An integrated grammar-based approach for mathematical expression recognition,” *Pattern Recognition*, vol. 51, pp. 135 – 147, 2016. 3
- [9] “IDEAL Math Writer.” <https://ideal-math-writer.soft112.com/>. 3
- [10] C. Larman, *Agile and Iterative Development: A Manager’s Guide*. 01 2004. 6
- [11] I. Sommerville, *Ingeniería del software*. 2005. 6
- [12] A. Seitz, S, “Computer vision.” <https://courses.cs.washington.edu/courses/cse576/>, 2000. [CSE576: Computer Vision]. 9, 12

- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>. 14, 16, 169
- [14] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015. 15
- [15] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2014. 22, 159
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. 22, 157
- [17] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. 2001. 23
- [18] R. H. Anderson, “Syntax directed recognition of hand-printed two-dimensional mathematics,” *Harvard University*. 23
- [19] E. Tapia and R. Rojas, “Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance,” *Freie Universität at Berlin, Institut für Informatik*. 24
- [20] G. Genthal and R. Sauvestre, “Image to latex,” *Stanford University*. 25
- [21] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, “Image-to-markup generation with coarse-to-fine attention,” *34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017*, 2017. 25
- [22] J. Zhang, J. Dua, S. Zhang, 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. 25, 46, 51, 52, 53, 53, 137, 159
- [23] S. Luján-Mora, “Programación en internet: clientes web,” 2001-10-08. 25
- [24] A. Holovaty and J. Kaplan-Moss, “El libro de Django 1.0.” <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>. 26
- [25] M. Meng, S. Steinhardt, and A. Schubert, “Application programming interface documentation: What do software developers want?,” *Journal of Technical Writing and Communication*, vol. 48, p. 295–330, 07 2018. 27
- [26] E. Sundvall, M. Nyström, D. Karlsson, M. Eneling, R. Chen, and H. Orman, “Applying representational state transfer (rest) architecture to archetype-based electronic health record systems,” *BMC medical informatics and decision making*, vol. 13, p. 57, 05 2013. 27
- [27] “Identificación y autentificación.” https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009740_.htm. 28

- [28] C. Salazar, “Los principios de SOLID de la programación orientada a objetos.” <https://www.codesolt.com/tutoriales/fundamentos/solid/>, 2018. [Consultado: 2019-09-21]. 30
- [29] A. Murthy, “Exploring S.O.L.I.D Principle in Android.” <https://proandroiddev.com/exploring-s-o-l-i-d-principle-in-android-a90947f57cf0>, 2018. [Consultado: 2019-09-21]. 30, 31
- [30] 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]. 42
- [31] “System Properties Comparison MySQL vs. Oracle vs. PostgreSQL.” <https://db-engines.com/en/system/MySQL%3BOracle%3BPostgreSQL>, 2019. 43
- [32] 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. 44
- [33] H. Mouchere, “CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions.” http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions, 2014. [Consultado: 2018-11-01]. 46
- [34] Z. Hadjadj, A. Meziane, Y. Cherfa, M. Cheriet, and I. Setitra, “Isauvola: Improved sauvola’s algorithm for document image binarization,” vol. 9730, pp. 737–745, 07 2016. 145
- [35] J. Zhang, J. Du, and L. Dai, “Multi-scale attention with dense encoder for handwritten mathematical expression recognition,” *CoRR*, vol. abs/1801.03530, 2018. 155
- [36] Y. Deng, A. Kanervisto, and A. M. Rush, “What you get is what you see: A visual markup decompiler,” *CoRR*, vol. abs/1609.04938, 2016. 143, 157, 159, 169
- [37] Z. Wang and J.-C. Liu, “Translating math formula images to latex sequences using deep neural networks with sequence-level training,” vol. abs/1908.11415, 2019. 157
- [38] K. Cho, B. van Merriënboer, Ç. Gülcühre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014. 167