



Instituto Politécnico Nacional

ESCUELA SUPERIOR DE CÓMPUTO

Trabajo Terminal

2019-A009

Ingeniería en Sistemas Computacionales

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

PRESENTAN:

Carlos Tonatihu Barrera Pérez

Juan Carlos Garcia Medina

Ian Mendoza Jaimes

DIRECTORES:

Dr. Jorge Cortés Galicia



Ciudad de México, 6 de noviembre de 2019

Índice general

Índice de figuras	V
Índice de tablas	VII
1. Introducción	1
1.1. Palabras Clave	1
1.2. Planteamiento del problema	2
1.3. Estado del arte	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	4
1.5. Resultados esperados	4
1.6. 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
2.1.3. Segmentación de la imagen	9
2.2. Deep Learning	10
2.2.1. Gráfos computacionales	10
2.2.2. Redes Neuronales Convolucionales	10
2.2.2.1. Pooling	12
2.3. Redes Neuronales Recurrentes	13
2.3.1. Arquitectura Encoder-Decoder	14
2.3.2. Long Short-Term Memory	14
2.3.3. Aprendizaje profundo como servicio	16
2.4. Métodos de reconocimiento de expresiones matemáticas	17
2.4.1. Análisis sintáctico dirigido	17
2.4.2. Análisis estructural	18
2.4.3. Image Captioning	18
2.5. Aplicación web	18
2.5.1. Django	19

2.5.1.1.	Modelo-Vista-Template	19
2.5.2.	API REST	20
2.5.2.1.	Application Programming Interface	20
2.5.2.2.	Representational State Transfer	21
2.5.2.3.	Autenticación	21
2.6.	Aplicación Android	22
2.6.1.	Arquitectura Clean	22
2.6.2.	Componentes	23
2.6.2.1.	Entidades	23
2.6.2.2.	Casos de uso	23
2.6.2.3.	Adaptadores	23
2.6.2.4.	Frameworks y controladores	23
2.6.3.	SOLID	23
3.	Análisis del sistema	25
3.1.	Requerimientos funcionales	25
3.1.1.	Módulo de usuarios	25
3.1.2.	Módulo de proyectos	26
3.1.3.	Módulo de análisis	27
3.1.4.	Módulo de traducción	27
3.2.	Requerimientos No Funcionales	27
3.3.	Requerimientos técnicos	28
3.3.1.	Aplicación móvil	28
3.3.1.1.	Requerimientos mínimos de software	28
3.3.1.2.	Requerimientos mínimos de hardware	28
3.3.2.	Aplicación web	28
3.3.2.1.	Requerimientos mínimos de software	28
3.3.2.2.	Requerimientos mínimos de hardware	29
3.4.	Reglas de Negocio	29
3.4.1.	RN-001 Campos obligatorios	29
3.4.2.	RN-002 Datos correctos	29
3.4.3.	RN-003 Unicidad de identificadores	29
3.4.4.	RN-004 Calificación proyecto	30
3.4.5.	RN-005 Fecha de modificación	30
3.4.6.	RN-006 Usuario verificado	31
3.4.7.	RN-007 Información necesaria para descargar un proyecto	31
3.5.	Análisis de Riesgos	31
3.6.	Descripción del software	33
3.6.1.	Android	33
3.6.2.	Base de datos	34
3.6.3.	Framework de desarrollo web	35
3.7.	Delimitación de expresiones matemáticas	37
3.7.1.	Nivel dimensional	42

4. Diseño del sistema	43
4.1. Arquitectura del sistema	43
4.1.1. Cliente web	43
4.1.2. Cliente android	43
4.1.3. Servidor web	43
4.1.4. Servidor de base de datos	44
4.1.5. Servidor cloud	44
4.2. Modelo de datos del sistema	46
4.2.1. Entidad: Usuario	46
4.2.1.1. Atributos	47
4.2.2. Entidad: Proyecto	47
4.2.2.1. Atributos	47
4.2.3. Entidad: Traducción	48
4.2.3.1. Atributos	48
4.2.4. Entidad: Tipo estudios	49
4.2.4.1. Atributos	49
4.2.5. Entidad: Usuario Token	49
4.2.5.1. Atributos	49
4.3. Aplicación de Android	51
4.3.1. Módulo usuarios	51
4.3.1.1. Diagrama de casos de uso	51
4.3.1.2. A-USR-CU1 Crear cuenta	51
4.3.1.3. A-USR-CU2 Iniciar sesión	54
4.3.1.4. A-USR-CU3 Recuperar contraseña	56
4.3.1.5. A-USR-CU4 Editar perfil	58
4.3.1.6. A-USR-CU5 Cerrar Sesión	60
4.3.2. Módulo proyectos	61
4.3.2.1. Diagrama de casos de uso	61
4.3.2.2. A-PR-CU1 Gestionar proyectos	61
4.3.2.3. A-PR-CU1.1 Visualizar proyecto	63
4.3.2.4. A-PR-CU1.1.2 Editar proyecto	64
4.3.2.5. A-PR-CU1.1.3 Eliminar proyecto	66
4.3.2.6. A-PR-CU1.1.4 Eliminar traducción	68
4.4. Aplicación Web	70
4.4.1. Módulo usuarios	70
4.4.1.1. Diagrama de casos de uso	70
4.4.1.2. W-USR-CU1 Crear cuenta	70
4.4.1.3. W-USR-CU1.1 Verificar cuenta	73
4.4.1.4. W-USR-CU2 Iniciar sesión	74
4.4.1.5. W-USR-CU3 Recuperar contraseña	76
4.4.1.6. W-USR-CU4 Editar perfil	78
4.4.1.7. W-USR-CU5 Cerrar sesión	80
4.4.2. Módulo proyectos	81

4.4.2.1.	Diagrama de casos de uso	81
4.4.2.2.	W-PR-CU1 Gestión de proyectos	81
4.4.2.3.	W-PR-CU1.1 Visualizar Proyecto	83
4.4.2.4.	W-PR-CU1.1.1 Editar proyecto	84
4.4.2.5.	W-PR-CU1.1.2 Descargar proyecto	86
4.4.2.6.	W-PR-CU1.1.3 Eliminar traducción	88
4.4.2.7.	W-PR-CU1.1.4 Copiar traducción	90
4.4.2.8.	W-PR-CU1.1.5 Eliminar proyecto	91
4.4.2.9.	W-PR-CU1.1.6.1 Calificar traducción	93
4.4.2.10.	W-PR-CU1.2 Crear proyecto	95
4.5.	Mensajes	97
4.6.	Interfaces	99
4.6.1.	Aplicación en Android	99
4.6.2.	Aplicación en Web	106
5.	Desarrollo del sistema	109
5.1.	Android	109
5.1.1.	Arquitectura de la aplicación	109
5.1.1.1.	Capa de datos	110
5.1.1.2.	Capa de dominio	117
5.1.1.3.	Capa de presentación	123
5.2.	Web	124
5.2.1.	Arquitectura de la aplicación	124
5.2.1.1.	Modelo	124
5.2.1.2.	Vista	124
5.2.1.3.	Template	124
5.3.	Conjunto de entrenamiento	125
5.3.1.	Descripción	125
Bibliografía		127

Índice de figuras

1.1. Arquitectura del sistema.	6
2.1. Imagen con bloques (Izquierda) y conjunto de segmentos de línea extraídos (Derecha).	9
2.2. 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.	11
2.3. Matriz de pesos conocida como kernel o filtro aplicada a una imagen.	12
2.4. 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 $s^{(t-1)}$	13
2.5. Una compuerta LSTM típica.	15
2.6. Ejemplo de sistema con módulo de Machine Learning	16
2.7. a) La primera etapa del método de reconocimiento, se anotan las coordenadas de de cada caracter. b) Se realiza un análisis sintáctico con las gramáticas libres de contexto definidas.	17
2.8. Utilización de un árbol de recubrimiento mínimo para el reconocimiento de expresiones matemáticas.	18
2.9. Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.	19
2.10. Representación del patrón de diseño Modelo-Vista-Template.	20
2.11. Consumidor y proveedor de servicios comunicandose mediante solicitudes y respuestas REST.	21
2.12. Arquitectura Clean [1]	22
3.1. Frecuencia de aparición de dígitos del sistema de numeración arábigo	37
3.2. Frecuencia de aparición de símbolos matemáticos	38
3.3. Frecuencia de aparición de letras mayúsculas	39
3.4. Frecuencia de aparición de letras minúsculas	40
3.5. Frecuencia de aparición de letras griegas	41
3.6. Frecuencia de aparición de símbolos especiales	42
4.1. Arquitectura del sistema.	44

4.2.	Modelo de datos del sistema.	46
4.3.	Diagrama de casos de uso del módulo Usuarios Android	51
4.4.	Diagrama de casos de uso del módulo Proyectos Android	61
4.5.	Diagrama de casos de uso del módulo Usuarios Web	70
4.6.	Diagrama de casos de uso del módulo Proyectos Web	81
4.7.	MIUA 1 Iniciar Sesión	99
4.8.	MIUA 2 Registro	100
4.9.	MIUA 3 Recuperar Contraseña	101
4.10.	MIUA 4 Menú de Opciones	102
4.11.	MIUA 5 Lista de Proyectos	103
4.12.	MIUA 6 Lista de Traducciones	104
4.13.	MIUA 7 Editar Perfil	105
4.14.	MIUW 1 Iniciar Sesión	106
4.15.	MIUW 2 Registro	106
4.16.	MIUW 3 Editar Perfil	107
4.17.	MIUW 4 Recuperar Contraseña	107
4.18.	MIUW 5 Lista de Proyectos	108
4.19.	MIUW 6 Lista de Traducciones	108
4.20.	MIUW 7 Menú de Opciones	108
5.1.	Tres capas que se tienen al utilizar la arquitectura Clean [2]	110
5.2.	Capa de datos [3]	112
5.3.	Capa de dominio [3]	117
5.4.	Capa de presentación [4]	123
5.5.	Imagen de ejemplo del conjunto de datos de CROHME	125
5.6.	Imagen de ejemplo del conjunto de datos de Im2latex-100k	126

Índice de tablas

1.1. Resumen de productos similares	4
3.1. Análisis de riesgos.	32
3.2. Tabla comparativa de sistemas operativos de dispositivos móviles	34
3.3. Comparación de diferentes gestores de bases de datos.	35
3.4. Comparación de frameworks de desarrollo web.	36
4.1. Tabla de los atributos de la entidad usuario	47
4.2. Tabla de los atributos de la entidad proyecto	48
4.3. Tabla de los atributos de la entidad traducción	49
4.4. Tabla de los atributos de la entidad tipo de estudios	49
4.5. Tabla de los atributos de la entidad usuario token	50

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. Palabras Clave

[Aplicación móvil, LaTeX, Image Captioning]

1.2. 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 LaTeX, 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 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.

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 incluyendo la clasificación de las mismas mediante usuarios y proyectos asociados a dichas traducciones.

Con el desarrollo de este prototipo de sistema se debe tener en cuenta que específicamente se ataca el problema de expresiones matemáticas (que a diferencia del reconocimiento de texto plano el cual es un problema que se considera resuelto por los buenos resultados que se han obtenido) este sigue siendo un problema abierto debido a su naturaleza jerárquica y bidimensional y que en la actualidad las soluciones existentes no brindan buenos resultados aún [5], por lo que mejorar los resultados del estado del arte no es el objetivo de este trabajo.

1.3. 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 LaTeX. También cuenta con un API de pago	Este producto cuenta con diferentes planes de pago según su uso o el tiempo que decidas pagarlo. 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 LaTeX. También cuenta con un API de pago	Este producto cuenta con diferentes planes de pago según su uso o el tiempo que decidas pagarlo. Tiene distinto trato para empresas. Un ejemplo de suscripción mensual es \$99 dólares el mes.

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

Tabla 1.1: Resumen de productos similares

1.4. Objetivos

1.4.1. 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.

1.4.2. 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.

1.5. 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 LaTeX 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 LaTeX 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.

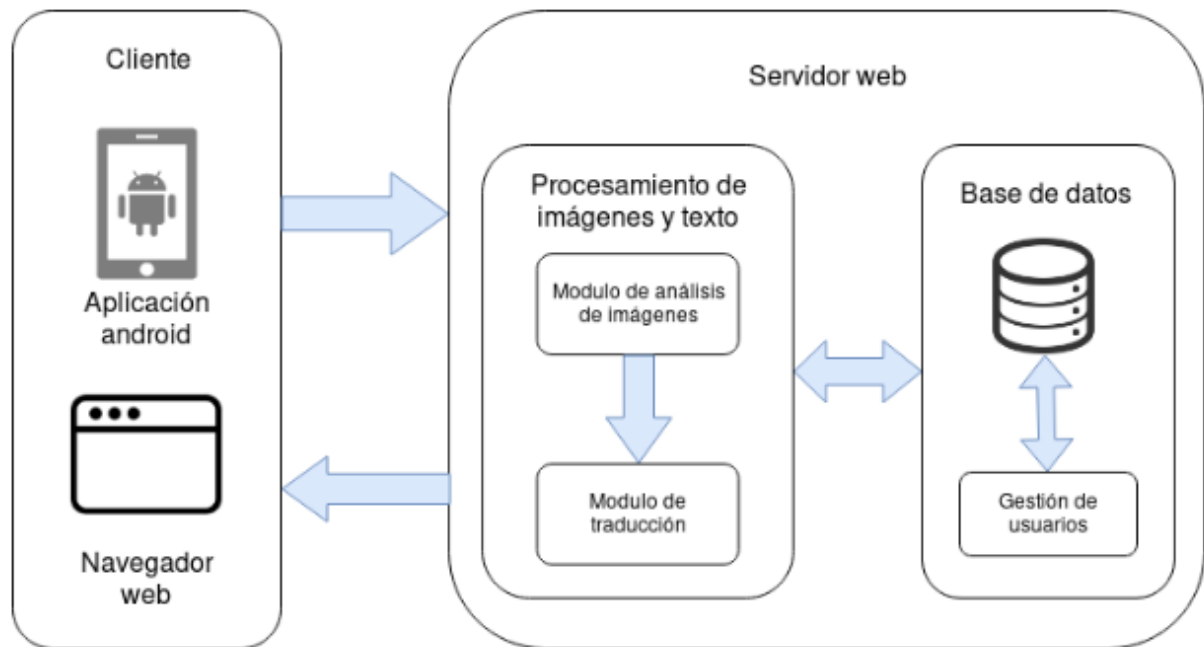


Figura 1.1: Arquitectura del sistema.

1.6. 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

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 distorciones 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.

Provee mejor entrada para otras técnicas avanzadas de procesamiento automatizadas de imágenes.

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 or polígono. Cuando las regiones de interés no cubren la imagen completa, aún se requiere el proceso de segmentación en regiones de y de fondo para ignorarse. [12]

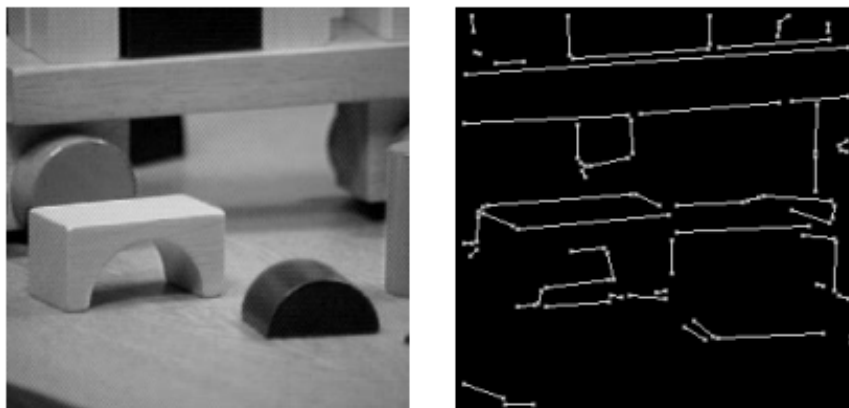


Figura 2.1: Imagen con bloques (Izquierda) y conjunto de segmentos de linea extraídos (Derecha).

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 logra 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. Gráfos computacionales

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

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

En la Figura 2.2 se puede ver un ejemplo de un perceptrón de una sola capa que es representado mediante un gráfo computacional. Gracias a esta definición, se puede extender fácilmente el gráfo mostrado en 2.2 para modelar a un perceptrón de n capas.

2.2.2. 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

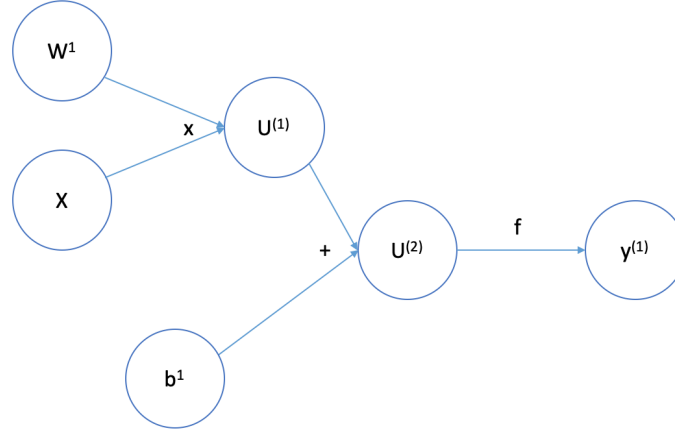


Figura 2.2: Ejemplo de un perceptrón de una sola capa representado mediante un gráfico computacional, siendo x la entrada, w^1 la matriz de pesos, b^1 la matriz de bias, $u^{(1)}$ y $u^{(2)}$ nodos intermedios en el gráfico y y^1 la salida de la red.

una multiplicación tensorial normal [13].

La operación de convolución que se utiliza en matemáticas e ingeniería esta definida de la siguiente manera:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)w(t - \tau)d\tau \quad (2.1)$$

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:

$$y(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)w(t - \tau) \quad (2.2)$$

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

$$y(i, j) = \sum_m \sum_n x(i, j)w(i - m, j - n) \quad (2.3)$$

En la Figura 2.3 se puede ver un ejemplo de cómo una imagen es procesada por la etapa de convolución en una CNN.

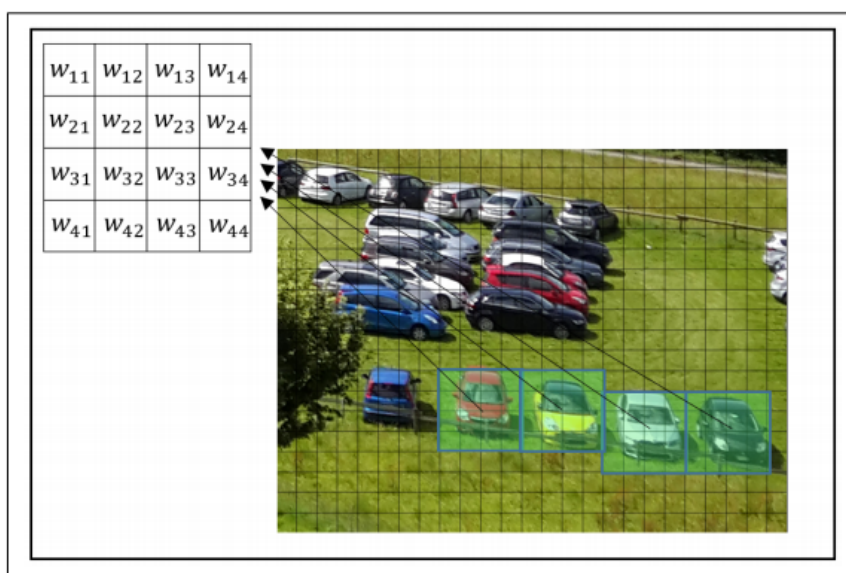


Figura 2.3: Matriz de pesos conocida como kernel o filtro aplicada a una imagen.

2.2.2.1. 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.

2.3. 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 2.4 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).

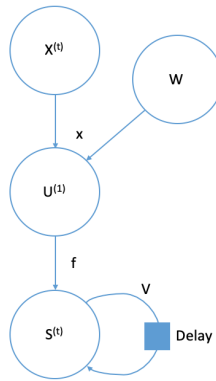


Figura 2.4: Ejemplo de una RNN que modela la ecuación $s(t) = f(s^{(t-1)}, \theta)$, donde θ representa los parámetros W y V , f la función no lineal aplicada a $Wx^{(t)}$ y V la matriz de pesos aplicada a $s^{(t-1)}$.

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 sí tienen una longitud fija. De este modo se puede decir que funcionan muy parecido a una CNN, ya que suman 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.

2.3.1. 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.

2.3.2. 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 esten condicionados con el contexto, de modo que la red pueda decidir cuando olvidar la información previa.

La LSTM es puede verse como una compuerta lógica que puede ser implementada en otra RNN para crear una arquitectura más robusta. En la Figura 2.5 se muestra un ejemplo de la compuerta LSTM. La LSTM implementa diferentes puertas las cuales son: **input gate**, **forget gate** y el **output gate**. Todas estas puertas ayudan a la LSTM a controlar en que momento es oportuno olvidar la información previamente encontrada en los anteriores estados.

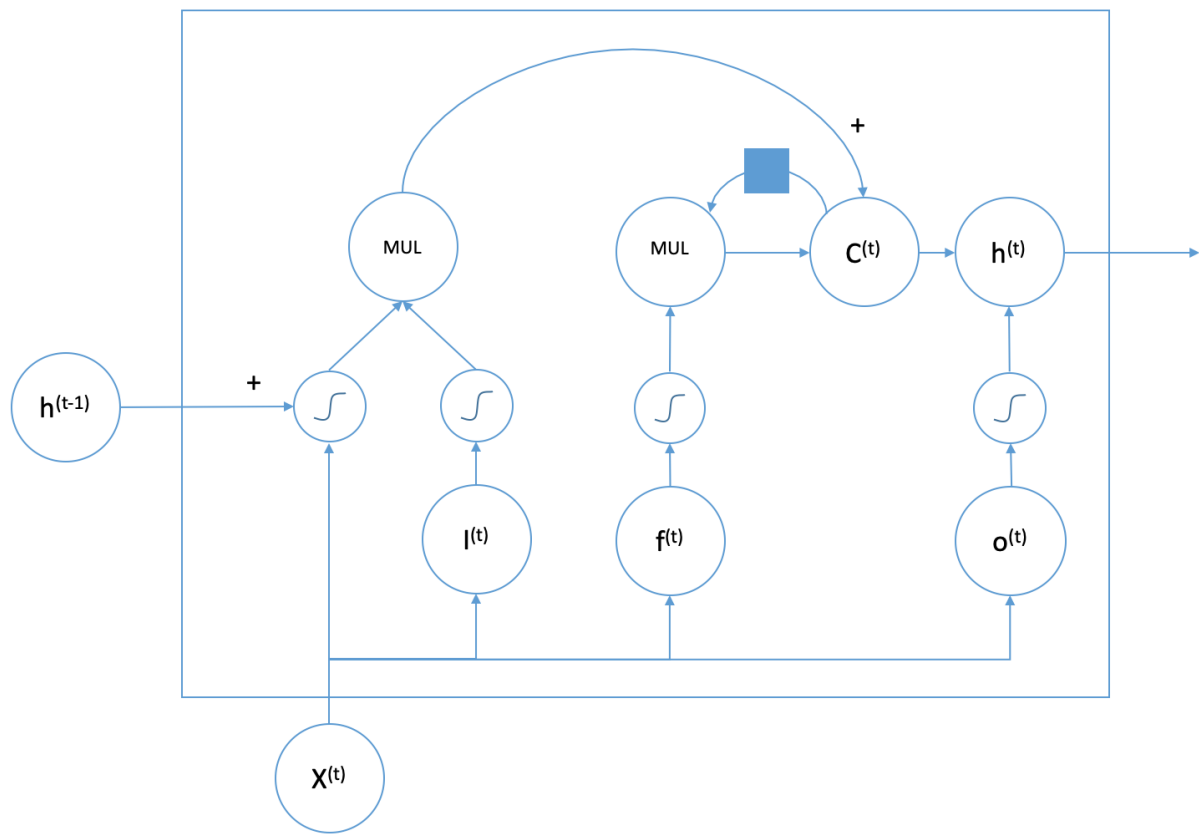


Figura 2.5: Una compuerta LSTM típica.

2.3.3. Aprendizaje profundo como servicio

El entrenamiento de redes neuronales profundas, conocidas como aprendizaje profundo es en la actualidad áltamente 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 2.6 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.

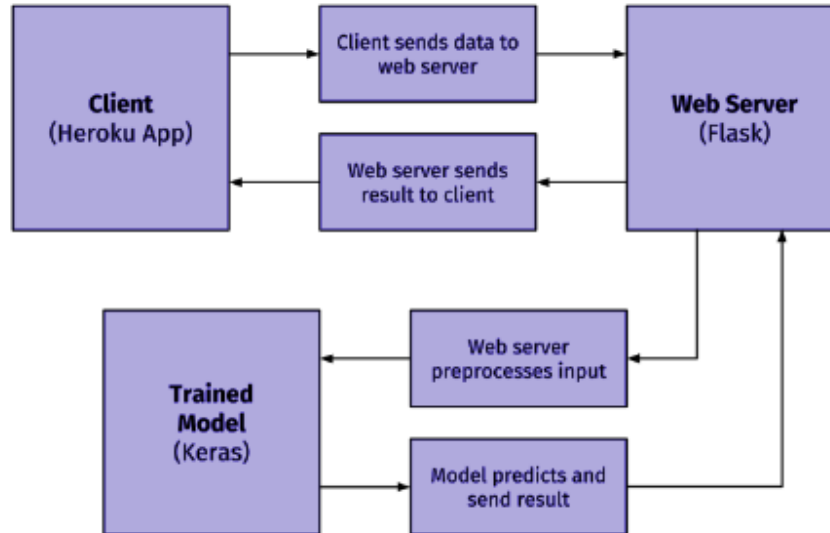


Figura 2.6: Ejemplo de sistema con módulo de Machine Learning

2.4. Métodos de reconocimiento de expresiones matemáticas

2.4.1. 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:

$$G = (V, T, P, S) \quad (2.4)$$

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 [14].

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 [15]. 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 2.7, se puede ver un ejemplo del proceso de reconocimiento propuesto por este método.

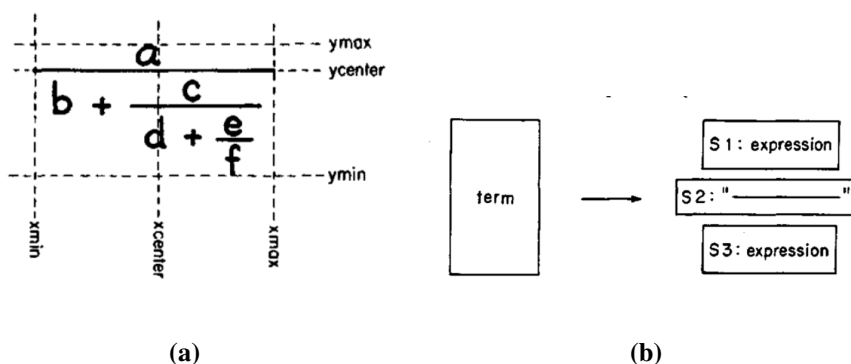


Figura 2.7: a) La primera etapa del método de reconocimiento, se anotan las coordenadas de cada carácter. b) Se realiza un análisis sintáctico con las gramáticas libres de contexto definidas.

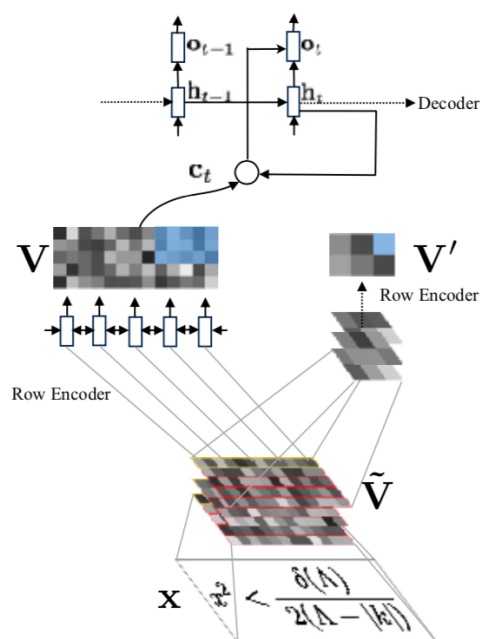


Figura 2.9: Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.

2.5.1. 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.

2.5.1.1. 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

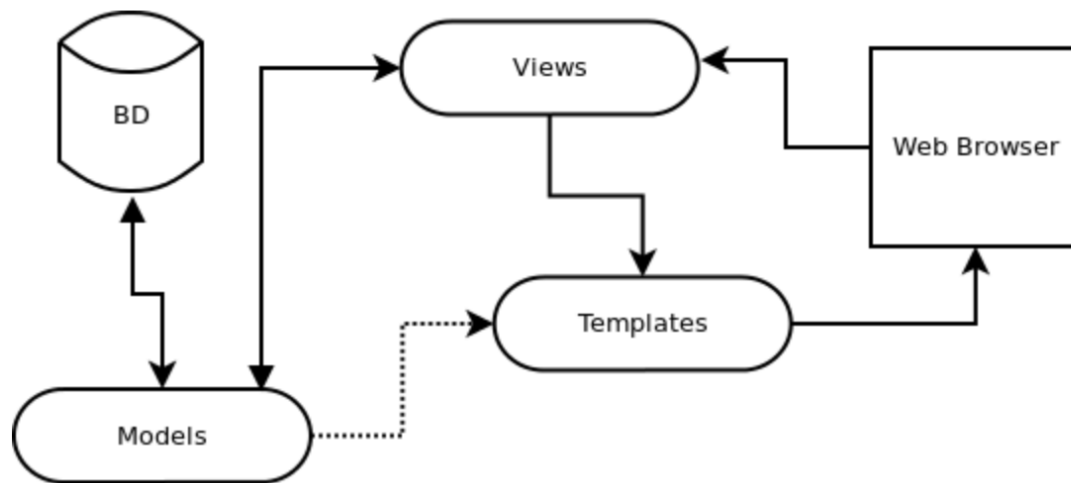


Figura 2.10: Representación del patrón de diseño Modelo-Vista-Template.

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 2.10, se puede ver una representación del patrón MTV. En el patrón de diseño MTV [20]:

- **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.

2.5.2. API REST

2.5.2.1. 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.

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 [21].

2.5.2.2. Representational 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 [22].

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

- La funcionalidad y estado son divididos 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 2.11 ilustra el uso de REST para servicios Web.

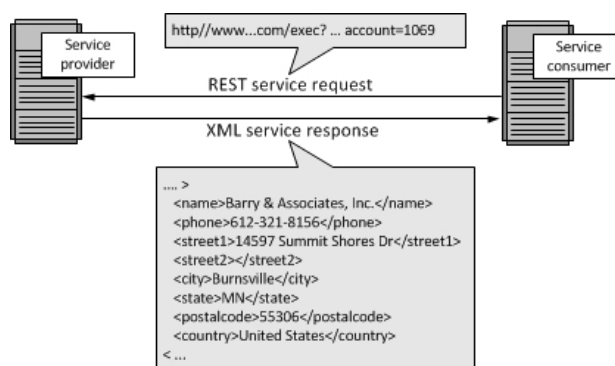


Figura 2.11: Consumidor y proveedor de servicios comunicandose mediante solicitudes y respuestas REST.

2.5.2.3. Autenticación

Para hablar de autenticación es necesario primero entender la diferencia entre identificación y autenticación, 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 autenticación es la capacidad de demostrar que un usuario o una aplicación es quien dicha persona o aplicación asegura ser [23].

Para el caso de una REST API es en muchas ocasiones necesario que se lleve a cabo la autenticación 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 autenticación para así asegurar que los datos sean visibles únicamente a aquellos que proporcionen las credenciales adecuadas y dispongan de los permisos necesarios.

2.6. Aplicación Android

2.6.1. 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.

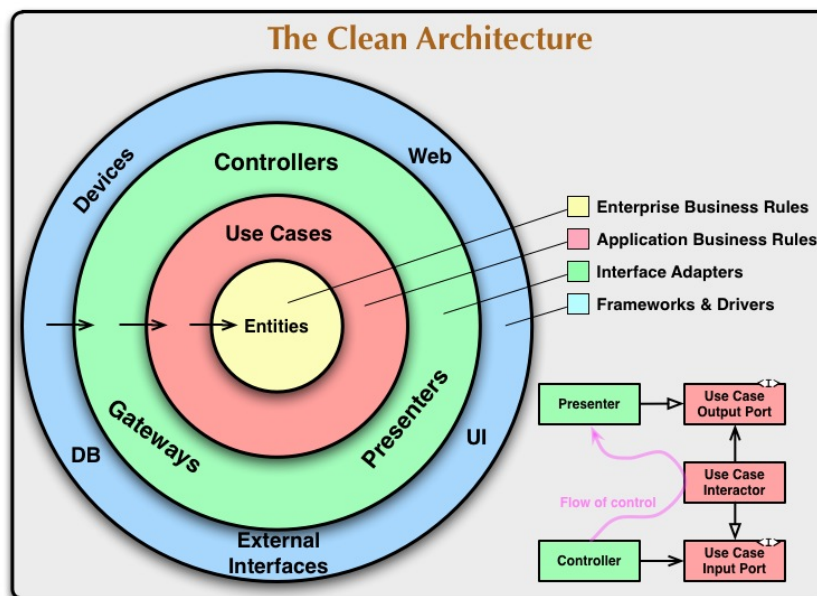


Figura 2.12: Arquitectura Clean [1]

La arquitectura Clean como se muestra en la figura 2.12 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.

2.6.2. Componentes

2.6.2.1. 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.

2.6.2.2. 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.

2.6.2.3. 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.

2.6.2.4. 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.

2.6.3. 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. [24]

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. [25]

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.[25]

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.[25]

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.[25]

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. [25]

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 debe de estar presentes en la aplicación web y de android y solo serán permitidas para usuarios con cuenta verificada.
- RF2. Mecanismo de autenticación de usuarios.** Proporcionar un mecanismo para el inicio y cierre de sesión de la cuenta del usuario cuya cuenta haya sido verificada en la aplicación web y android.
- RF3. Mecanismo de verificación de cuenta.** Proporcionar al usuario una forma para verificar que una cuenta creada es valida a través de una verificación basada en el envío de un correo electrónico para la confirmar dicha validez y hacer uso del resto de funcionalidad del sistema. Esta verificación solo se podra realizar a través de la aplicación web.
- RF4. Mecanismo de recuperación de contraseñas.** Proporcionar al usuario la posibilidad de recuperar la contraseña asociada a su cuenta a través de un envío de correo electrónico con el cual podrá acceder a una interfaz en la aplicación web para recuperar el acceso a su cuenta. La recuperación de contraseñas estará disponible en la aplicación android y web.

- RF5. Mecanismo para la comunicación entre la aplicación android y web.** El sistema debe de contar con una interfaz para la comunicación entre las dos aplicaciones que se tienen, la forma en que se realizara sera un API REST y el formato para el 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 \LaTeX .** Proporcionar al usuario un mecanismo para visualizar, editar, crear o borrar proyectos asociados a su cuenta. La gestión se podrá realizar en la aplicación web y android.
- RF6. Permitir descargar el archivo de \LaTeX** que se haya traducido. Proporcionar al usuario la funcionalidad de generar un archivo \LaTeX que contenga la traducción de expresiones matemáticas que se encuentren en un proyecto. La descarga de dicho archivo solo estará disponible en la aplicación web.
- RF7. Permitir calificar una traducción realizada.** Proporcionar al usuario la funcionalidad de calificar que tan buena fue la traducción de una expresión matemática en una imagen para brindar retroalimentación, dicha calificación podrán ser valores enteros entre 1 y 5. Para poder calificar una traducción se tendrá que hacer uso de la aplicación web.
- RF8. Permitir el uso de la cámara del dispositivo android para tomar fotografías.** Proporcionar un mecanismo en la aplicación web que permita acceder a la cámara del dispositivo, tomar una fotografía, visualizarla.
- RF9. Permitir la visualización del resultado de la traducción.** Proporcionar al usuario una interfaz en la cual pueda observar la traducción a \LaTeX que se realizó a partir de una imagen. Esto se podrá hacer en la aplicación web y android.
- RF10. Permitir añadir al portapapeles alguna traducción seleccionada.** Proporcionar al usuario la funcionalidad para copiar el código de una traducción a su portapapeles para su posterior uso. Esta funcionalidad está limitada a solo la aplicación web.
- RF11. Mecanismo para el envío de imágenes tomadas por la aplicación android a la aplicación web para su uso.** El sistema debe de tener un mecanismo que a través del uso del API REST permita a la aplicación android el enviar una imagen a la aplicación web para que esta sea tratada.

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

3.2. Requerimientos No Funcionales

A continuación se enlistan los requerimientos no funcionales.

- RNF1. Usabilidad.** La interfaz de usuario debe de ser intuitiva con el objetivo de hacer uso de las funcionalidades del sistema de una forma fácil para el usuario.
- RNF2. Seguridad.** La seguridad se solventa al realizar un cifrado de las contraseñas de los usuarios así como una verificación de la cuenta del usuario para poder hacer uso del sistema. Por otro lado, en la comunicación que se realiza entre la aplicación móvil y el API REST se utiliza un token para autenticar las peticiones que se realizan, dicho token es único para cada usuario.
- RNF3. Escalabilidad.** El sistema deberá ser fácilmente escalable y con ello poseer la cualidad de que si se agregan nuevas funcionalidades, estas sean fáciles de acoplar con lo ya desarrollado. Además, de que debe de ser posible aumentar sus capacidades para brindar servicio a más usuarios.
- RNF4. Disponibilidad.** El sistema debería estar disponible en la mayor parte del tiempo, para lograr esto el uso de un proveedor de hosting es necesario. Entre las posibles opciones están Amazon Web Services, Azure o Google Cloud

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 Megapíxeles.

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 Unicidad de identificadores

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

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

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

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

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

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 [26]	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 [26]	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 seria mayor debido a la curva de aprendizaje de Swift y Objective-C a la cual nos enfrentaríamos.

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

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 [27].

Características	MySQL	Oracle	PostgreSQL
Modelo de base de datos primario	Relacional	Relacional	Relacional
Modelo de base de datos secundario	Documento	Documento, gráfico, 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 [28].

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

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

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

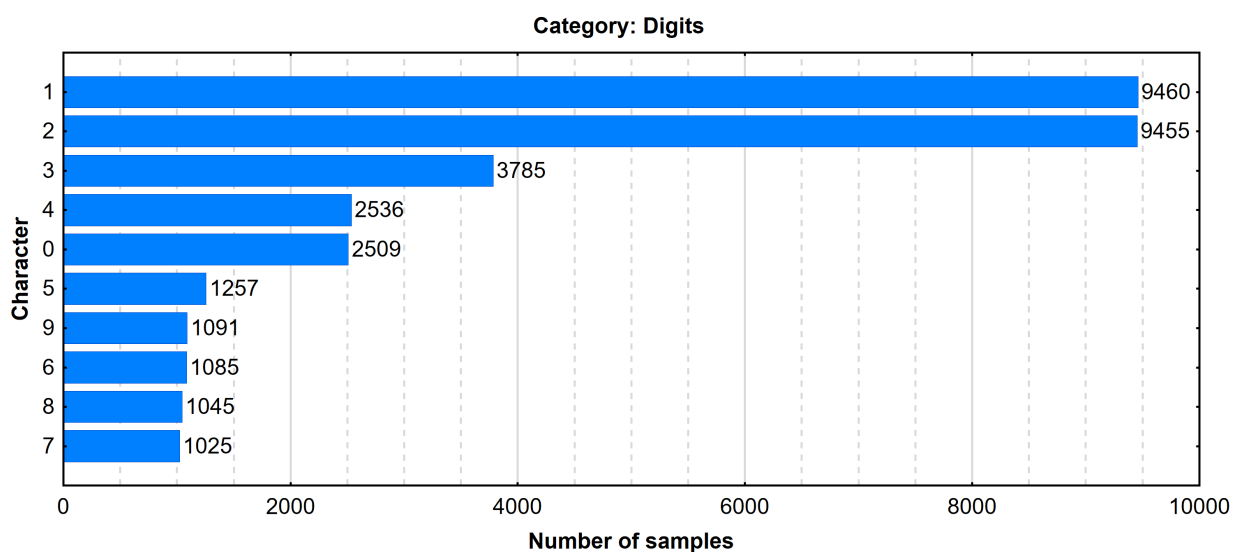


Figura 3.1: Frecuencia de aparición de dígitos del sistema de numeración arábigo

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

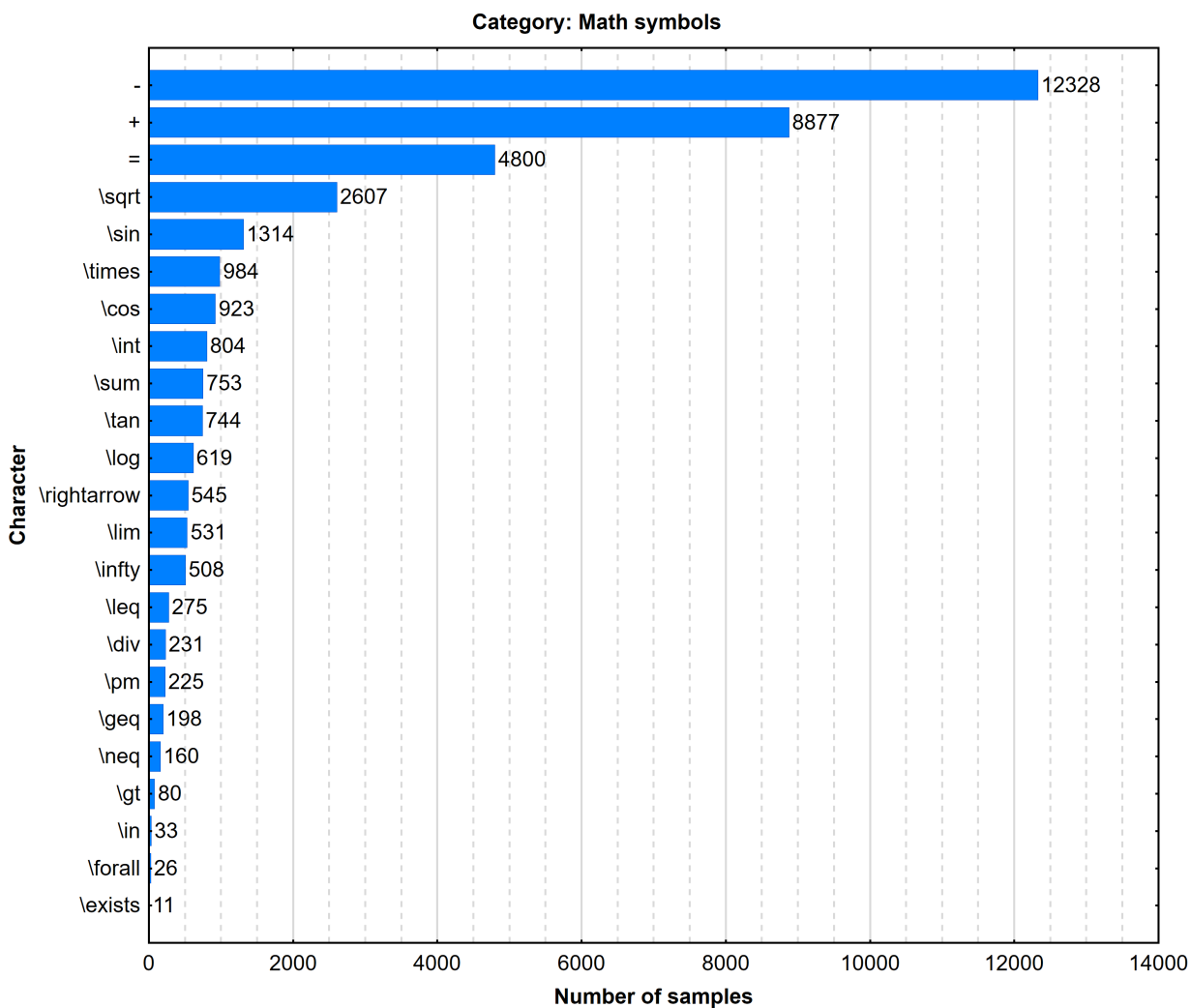


Figura 3.2: Frecuencia de aparición de símbolos matemáticos

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

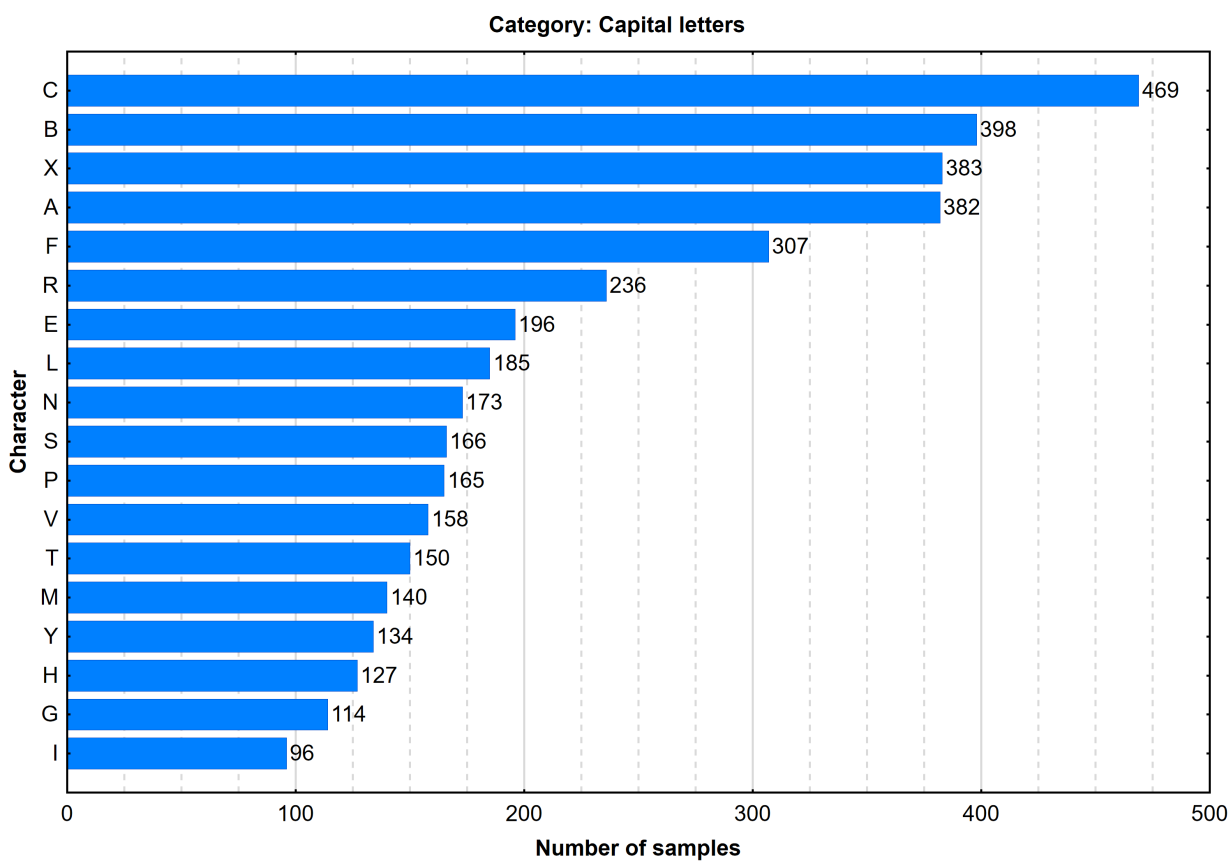


Figura 3.3: Frecuencia de aparición de letras mayúsculas

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

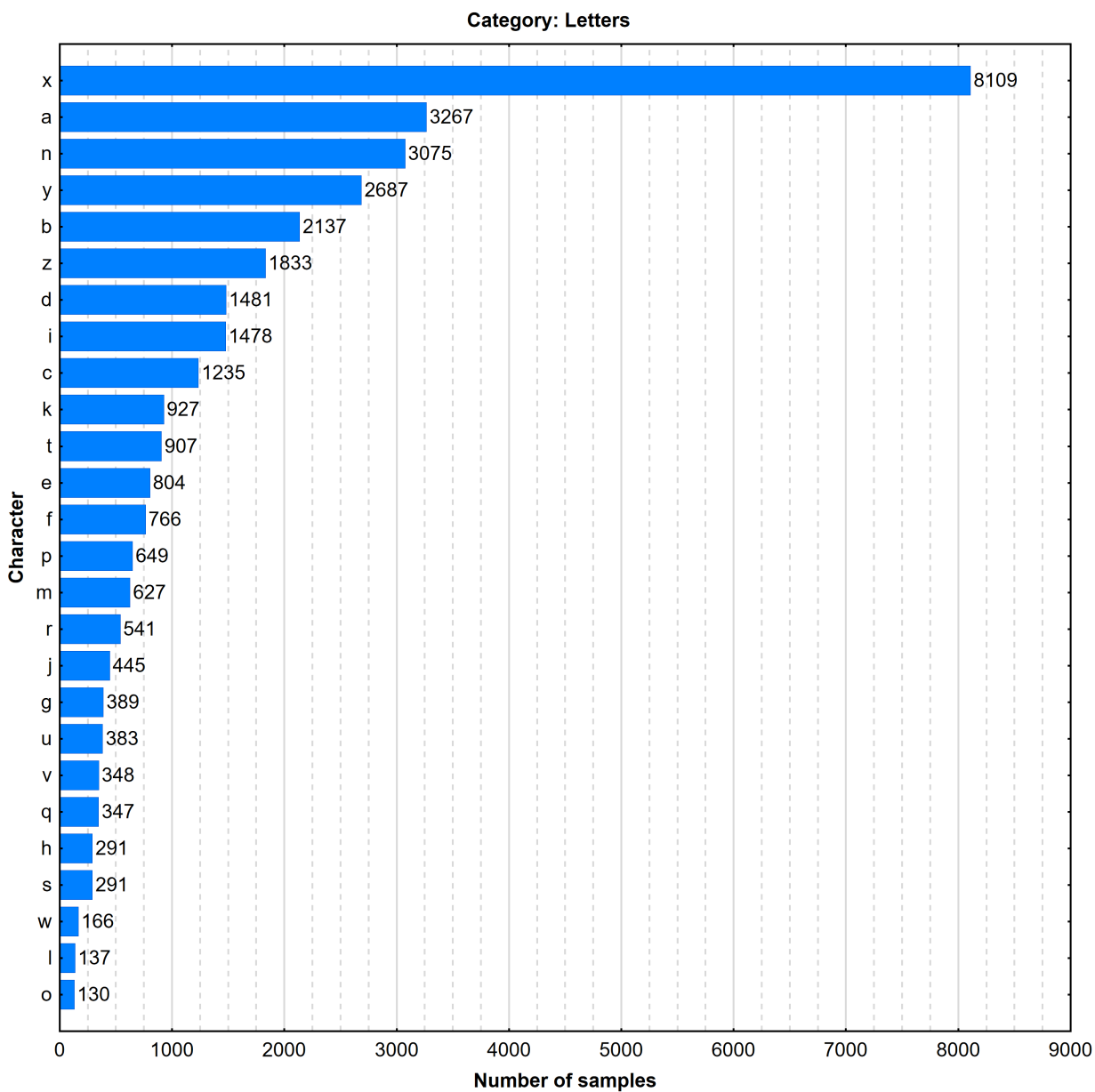


Figura 3.4: Frecuencia de aparición de letras minúsculas

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

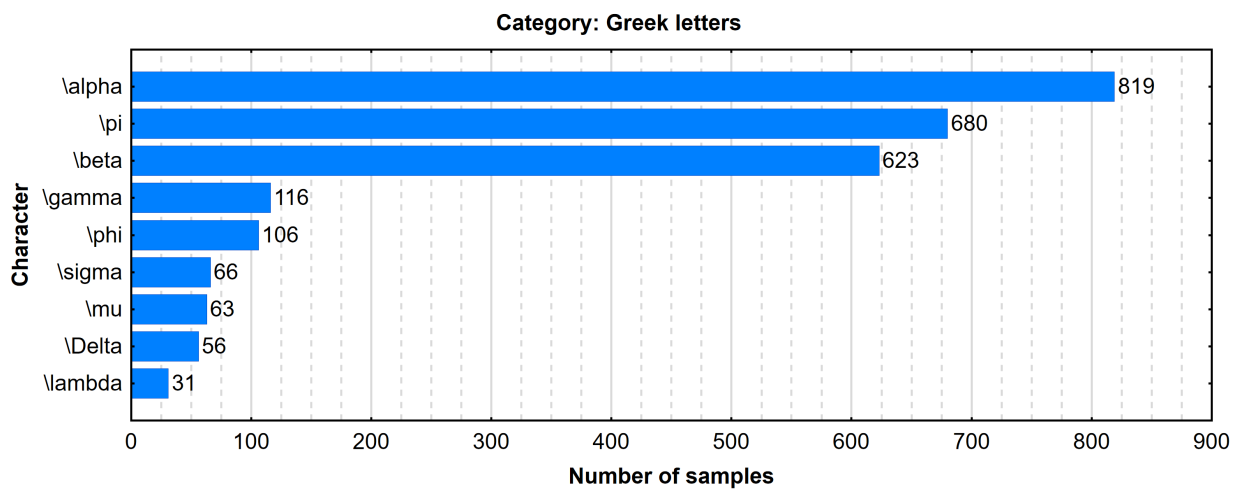


Figura 3.5: Frecuencia de aparición de letras griegas

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.

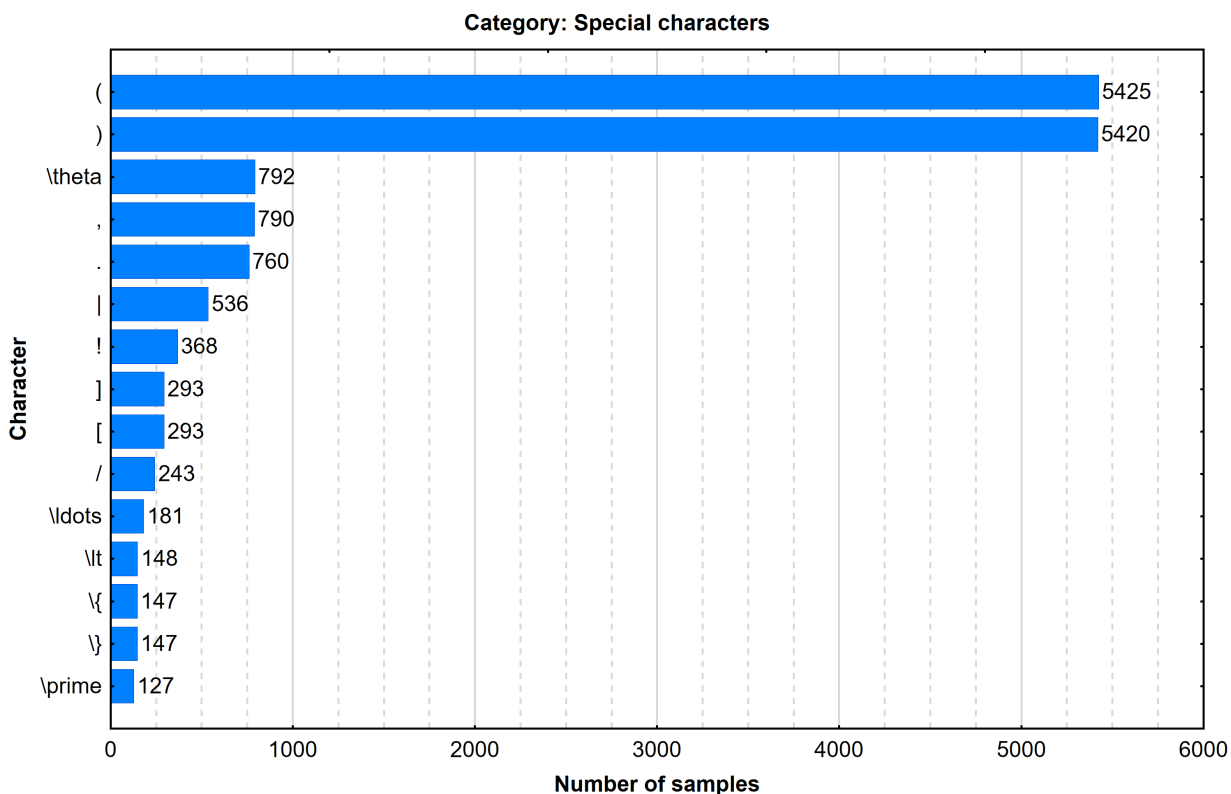


Figura 3.6: Frecuencia de aparición de símbolos especiales

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 con el mismo conjunto de entrenamiento.

Capítulo 4

Diseño del sistema

En este capítulo se detallará en el cómo se desarrollará el sistema, incluyendo una descripción de la arquitectura del sistema, la explicación del modelo de datos con el cual se trabajará. Se detallan los casos de uso que indican el cómo interactúa el usuario con cada módulo 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 cómo está 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 plantillas 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 sí, es decir, por un lado tenemos el presentador de vistas de Django para controlar al cliente

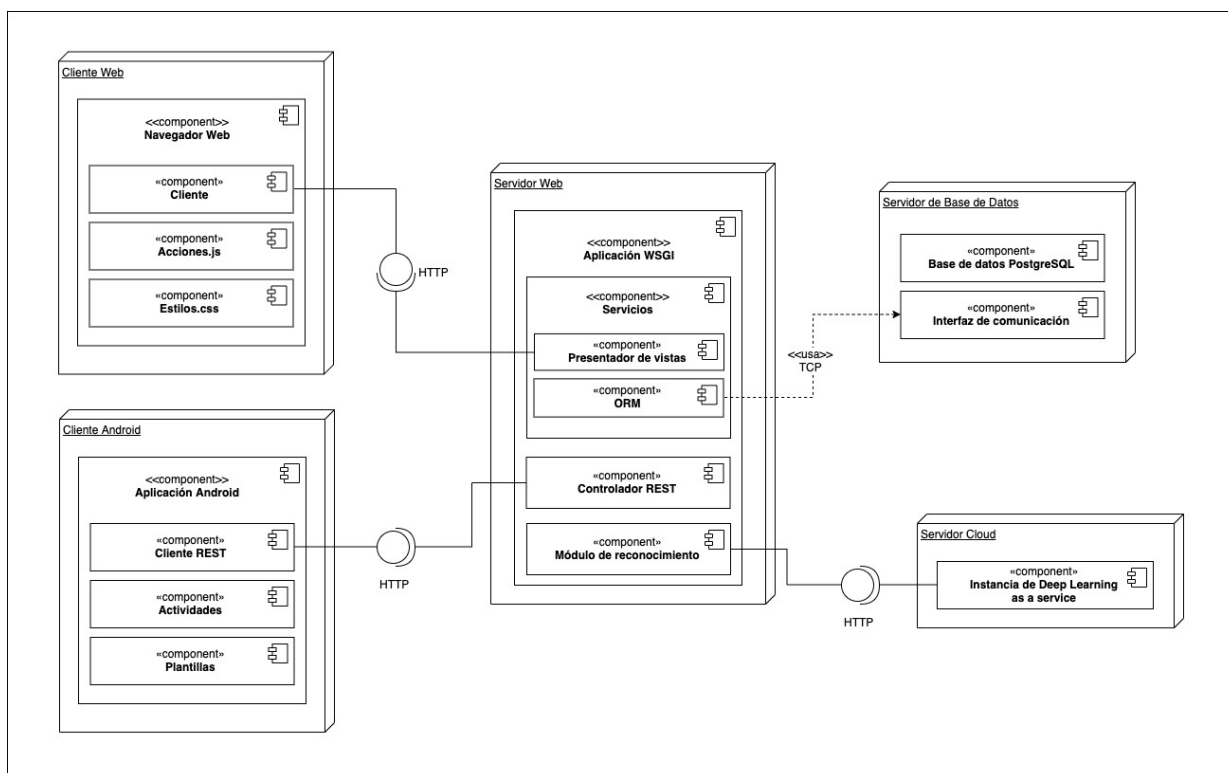


Figura 4.1: Arquitectura del sistema.

web, para la interacción con el servidor web se utiliza directamente el ORM de Django para evitar lidiar con queries 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 postgresSQL y con ello permitir que el sistema este mejor modularizado y tener una independencia en capa complemente de la arquitectura.

4.1.5. Servidor cloud

Servidor cloud es en dónde se realizará la tarea de realizar la traducción a \LaTeX 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.

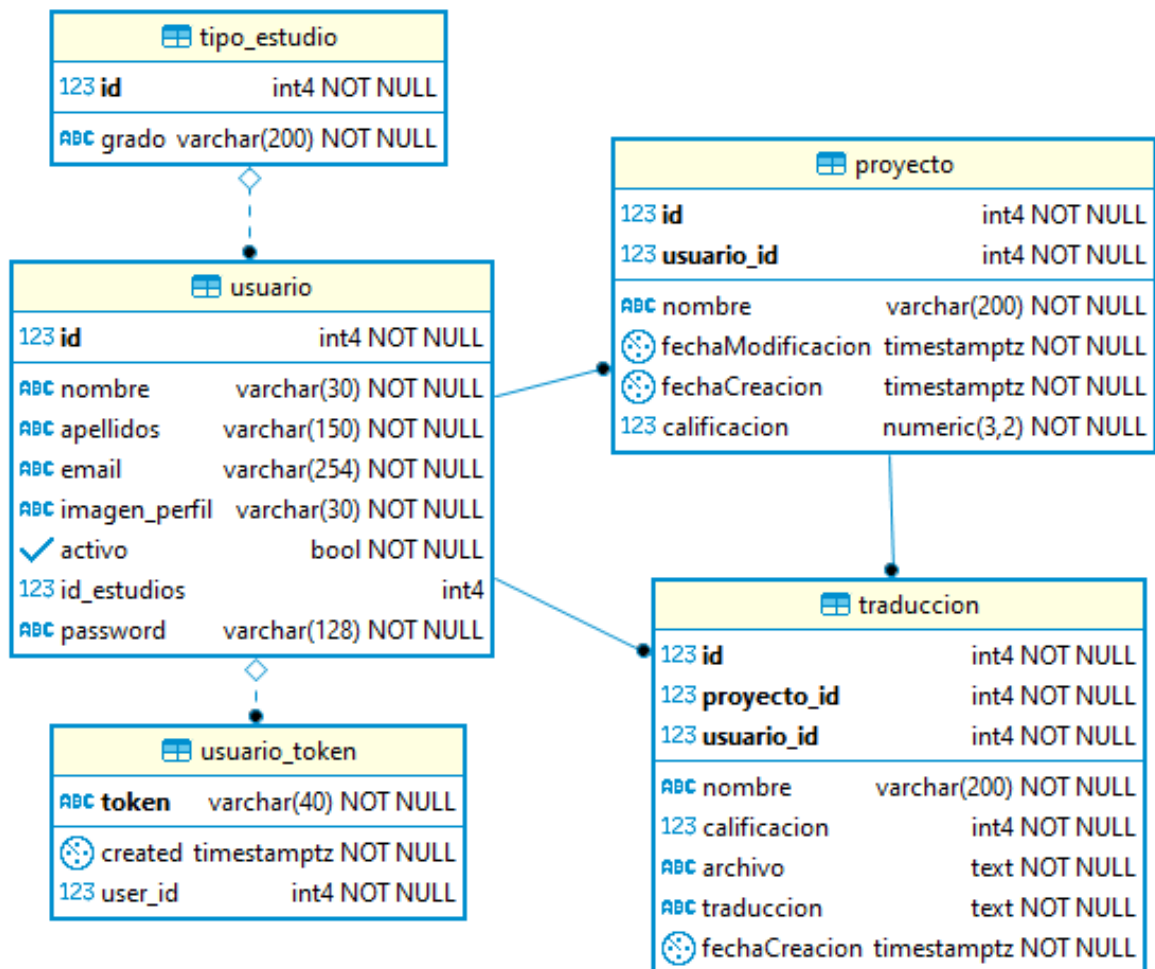


Figura 4.2: Modelo de datos del sistema.

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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 se 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.

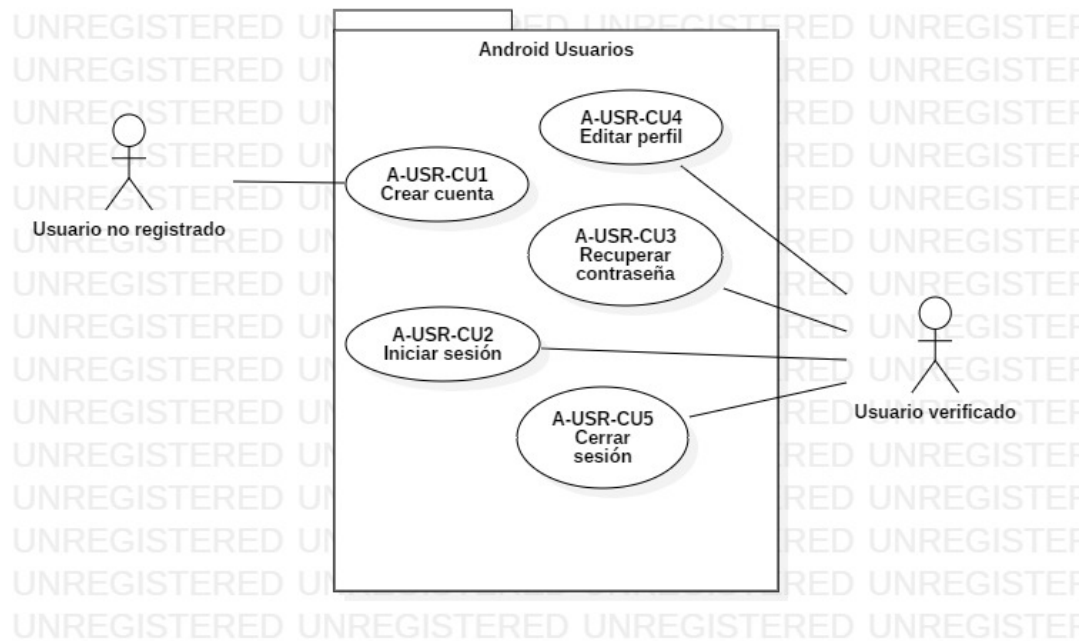


Figura 4.3: Diagrama de casos de uso del módulo Usuarios Android

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 Envío 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.

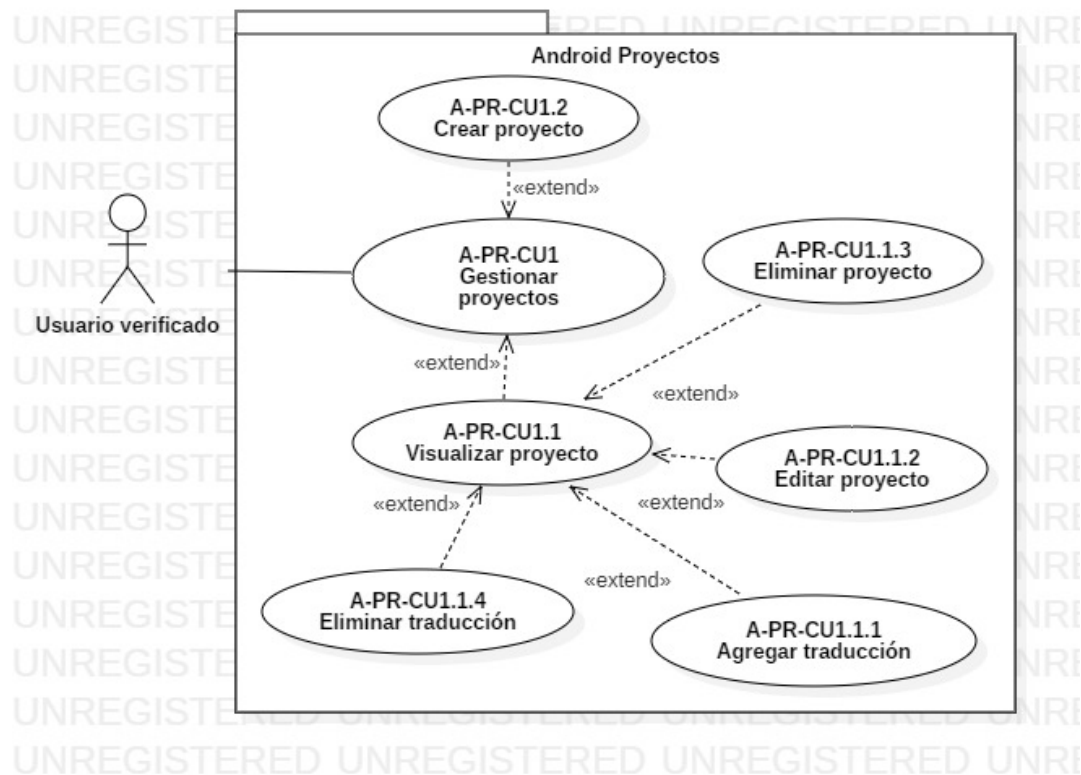


Figura 4.4: Diagrama de casos de uso del módulo Proyectos Android

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 puede 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 especifico
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 icono *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 el 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 especifico
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 especifico.
Prioridad	Baja.
Reglas de Negocio	Ninguna.

Trayectoria Principal

1. [Actor] Presiona el icono *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 icono *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.

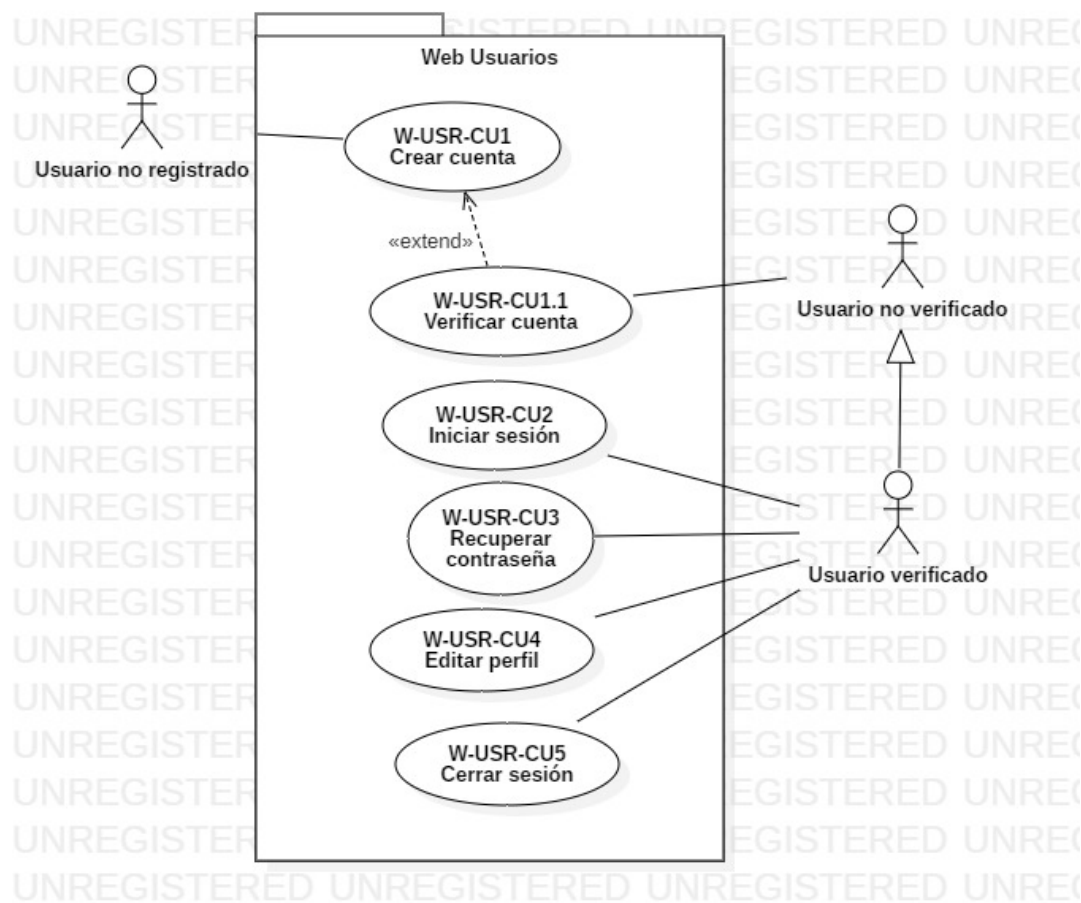


Figura 4.5: Diagrama de casos de uso del módulo Usuarios Web

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 Envío 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.

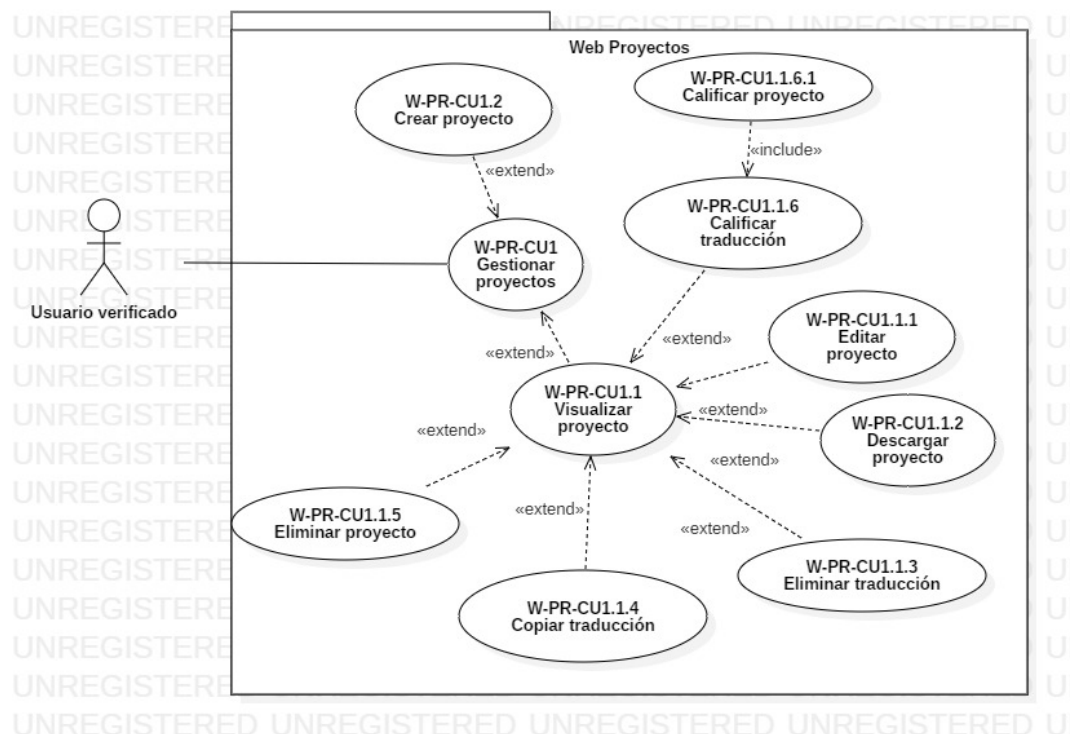


Figura 4.6: Diagrama de casos de uso del módulo Proyectos Web

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 puede 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] Continúa 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] Continúa 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

Marvel 13:00

← Registrate

Correo

Contraseña

Confirmar contraseña

Nombre

Crear Cuenta

Figura 4.8: MIUA 2 Registro



Figura 4.9: MIUA 3 Recuperar Contraseña

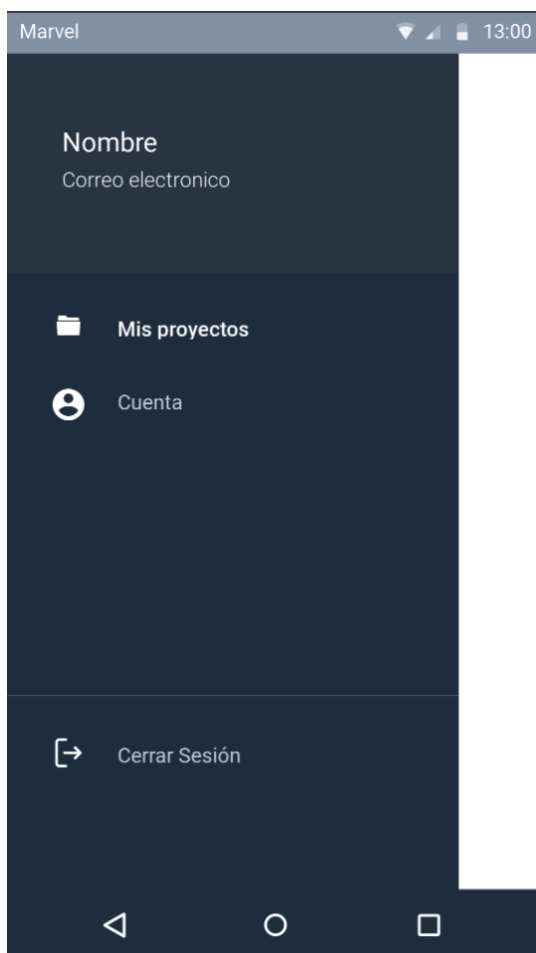


Figura 4.10: MIUA 4 Menú de Opciones



Figura 4.11: MIUA 5 Lista de Proyectos



Figura 4.12: MIUA 6 Lista de Traducciones

Marvel 13:00

Cuenta

Correo electronico
correo@gmail.com

Nombre
Nombre y apellido

Contraseña

Confirmar Contraseña

Grado de estudio
Seleccione su grado de estudio ▼

Guardar Cambios

Figura 4.13: MIUA 7 Editar Perfil

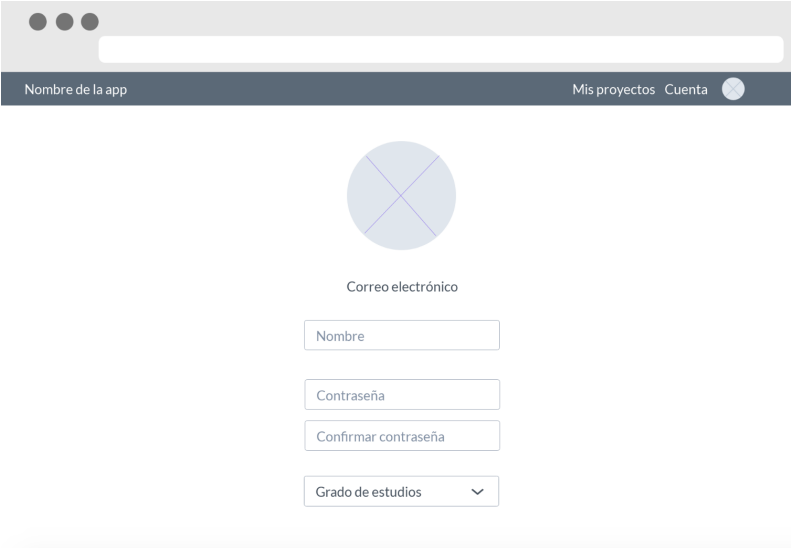
4.6.2. Aplicación en Web

The image shows a web browser window with a header bar containing three dots and a text input field labeled "Nombre de la app". Below the header, the page title is "Inicio de Sesión". The form consists of two text input fields labeled "Correo" and "Contraseña". Below the "Contraseña" field is a link that says "Olvidaste tu contraseña, recuperala aquí.". There are two buttons: "Iniciar sesión" (dark blue) and "Regístrate" (dark blue), separated by a horizontal line with a small circle in the middle.

Figura 4.14: MIUW 1 Iniciar Sesión

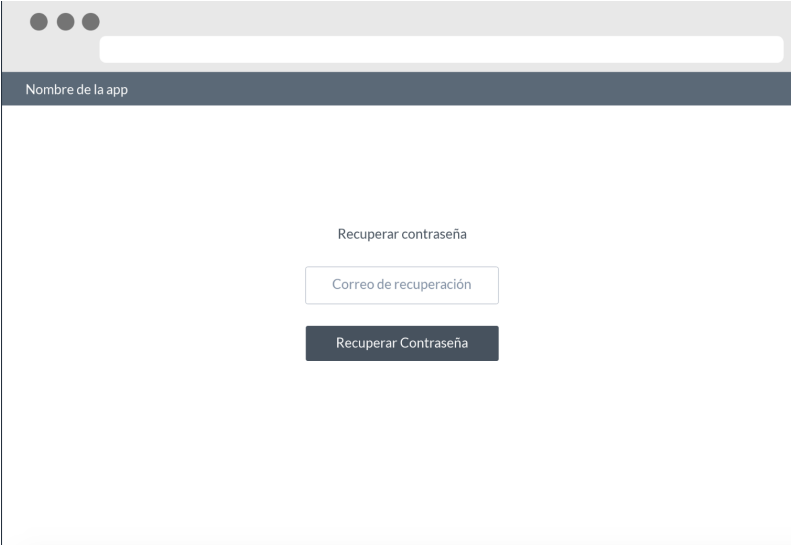
The image shows a web browser window with a header bar containing three dots and a text input field labeled "Nombre de la app". Below the header, the page title is "Registro". The form consists of five text input fields labeled "Nombre", "Correo", "Contraseña", "Confirmar contraseña", and "Registrarse". Below the "Registrarse" field is a horizontal line with a small circle in the middle, followed by a button labeled "Iniciar sesión" (dark blue).

Figura 4.15: MIUW 2 Registro



The image shows a web form for editing a profile. At the top, there is a header bar with three dots on the left, a search bar, and the text "Nombre de la app" on the left and "Mis proyectos Cuenta" on the right. Below the header, there is a circular profile picture placeholder with a blue 'X' inside. Underneath the profile picture is the label "Correo electrónico". Below this label are four input fields: "Nombre", "Contraseña", "Confirmar contraseña", and "Grado de estudios" with a dropdown arrow. The form is set against a light gray background.

Figura 4.16: MIUW 3 Editar Perfil



The image shows a web form for recovering a password. At the top, there is a header bar with three dots on the left, a search bar, and the text "Nombre de la app" on the left. Below the header, there is a section titled "Recuperar contraseña". Underneath this title is a single input field labeled "Correo de recuperación". Below the input field is a dark gray button labeled "Recuperar Contraseña". The form is set against a light gray background.

Figura 4.17: MIUW 4 Recuperar Contraseña

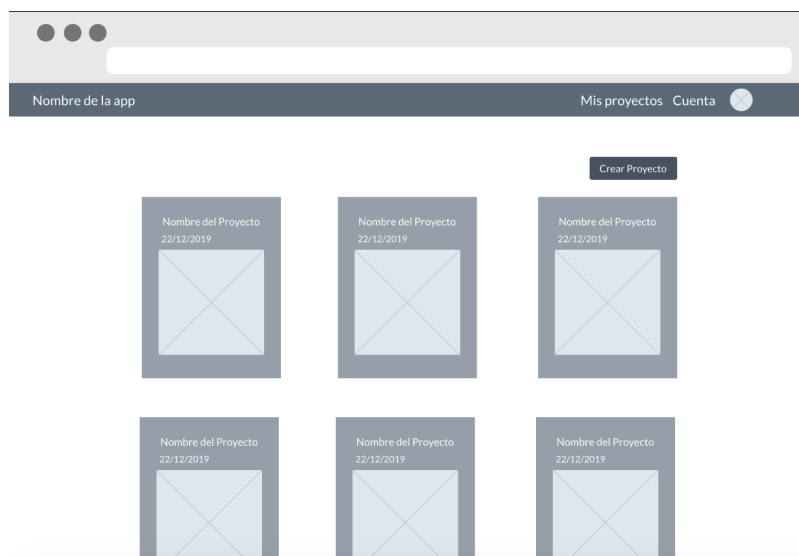


Figura 4.18: MIUW 5 Lista de Proyectos



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.

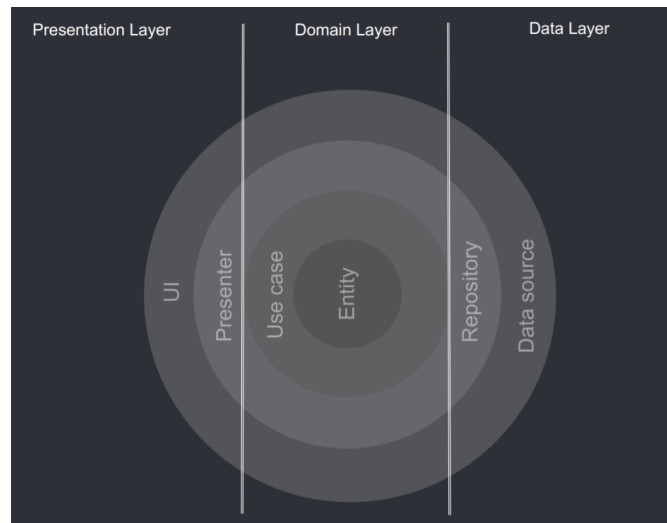


Figura 5.1: Tres capas que se tienen al utilizar la arquitectura Clean [2]

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 desarrollado 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
5     private static final String TAG = ProjectRepositoryImpl.class.
6         getCanonicalName();
7
8     @Override
9     public MutableLiveData<BusinessResult<ProyectoModel>>
10    findAllProyectosByUser(Integer id, String key) {

```

```

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

```

```

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

```

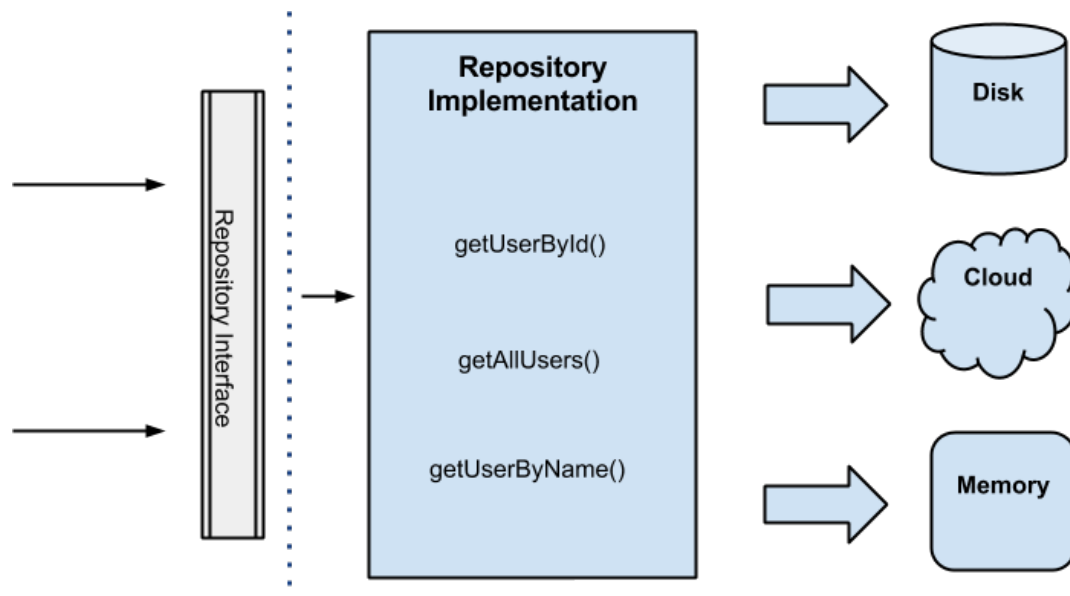


Figura 5.2: Capa de datos [3]

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 realizara la petición con alguno de los métodos que tiene HTTP, se tienen los parámetros que se envían y cada método nos regresa una llamada asincrónica 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,
        @Header("Authorization") String token);

```



```

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

```

```

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

```

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 utilizara para poder observar las peticiones que se realizan y brindar una retroalimentación a la hora de hacer pruebas y por ultimo el parser que se utilizara para trabajar y pasar de clases a datos que el API REST entienda y pueda utilizar, en este caso se utilizo 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(
    BASE_URL)
4         .addConverterFactory(GsonConverterFactory.create());
5
6     private static Retrofit retrofit = builder.build();
7     private static OkHttpClient.Builder httpClient = new OkHttpClient.Builder
    ();
8     private static HttpLoggingInterceptor loggingInterceptor = new
    HttpLoggingInterceptor();
9
10    public static <S> S createService(Class<S> serviceClass) {
11        if (!httpClient.interceptors().contains(loggingInterceptor)) {
12            loggingInterceptor.level(HttpLoggingInterceptor.Level.BODY);
13            httpClient.addInterceptor(loggingInterceptor);
14            builder.client(httpClient.build());
15            retrofit = builder.build();
16        }
17        return retrofit.create(serviceClass);
18    }
19 }

```

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     }
```

67 }

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 interactores 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.

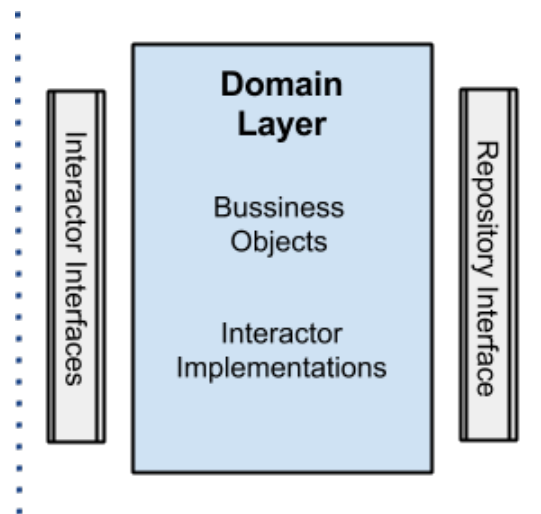


Figura 5.3: Capa de dominio [3]

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.
    getCanonicalName();
3     private UserRepository repository;
4
5     public UserInteractorImpl() {
6         repository = new UserRepositoryImpl();
7     }
8
9     @Override
```

```

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

```

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.

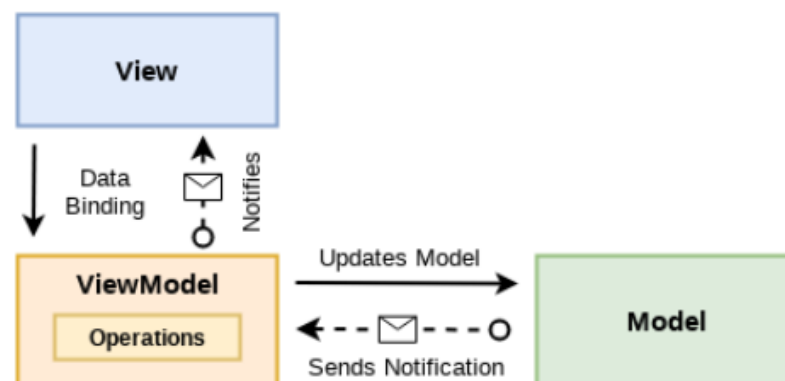


Figura 5.4: Capa de presentación [4]

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. [4]

- **Modelo** Se encarga de representar la información que será 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 interactores 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. Conjunto de entrenamiento

5.3.1. Descripción

Uno de los principales retos al trabajar con redes neuronales es la obtención de grandes conjuntos de entrenamiento que permitan mejores resultados, para el problema de reconocimiento de expresiones matemáticas existe una competencia llamada **Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME)** que pone disponible de manera libre un conjunto de entrenamiento con mas de diez mil expresiones matemáticas escritas a mano provenientes de muchos usuarios voluntarios de distintos países mezclando los conjuntos de entrenamiento de 3 competencias CROHME. Se le pidió a los voluntarios copiar expresiones impresas de un corpus de expresiones. El corpus ha sido diseñado para cubrir la diversidad compuesta por las distintas tareas elegidas de un corpora matemático y de expresiones embebidas en páginas de wikipedia. Distintos dispositivos han sido utilizados (plumas digitales de distintas tecnologías, dispositivos de entrada como pizarrones blancos, tablets con pantalla sensible) de tal modo que distintas escalas y resoluciones son usadas



A handwritten mathematical expression in black ink on a white background. The expression is $p = \sum_{i \leq d} a_i 2^i$. The variable p is on the left, followed by an equals sign. The summation symbol \sum is in the middle, with $i \leq d$ as the subscript. To the right of the summation is $a_i 2^i$.

Figura 5.5: Imagen de ejemplo del conjunto de datos de CROHME

A pesar de parecer un conjunto de datos prometedor para el desarrollo del presente trabajo terminal se deben considerar factores como el ruido que puede tener una imagen a partir de una fotografía tomada con un smartphone, por lo que se considera también otro conjunto de entrenamiento llamado **im2latex-100k** [17]. El cual a diferencia de CROHME no se enfoca en expresiones matemáticas escritas a mano sino extraídas de documentos generados por computadora como pueden ser libros o artículos impresos. [29].

$$Q = (b + 1/b)\rho, \quad \rho = \frac{1}{2} \sum_{\alpha > 0} \alpha,$$

Figura 5.6: Imagen de ejemplo del conjunto de datos de Im2latex-100k

Para la segunda parte de este trabajo terminal se utilizarán algunas métricas de evaluación donde el principal parámetro será la precisión. Se propondrán varias arquitecturas de redes neuronales de prueba y se entrenarán con ambos conjuntos para así comparar la precisión que ofrecería un modelo entrenado por cada conjunto, una vez hecho esto se elegirá el modelo y el conjunto que presente mayor precisión.

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]. v, 22
- [2] 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]. vi, 110
- [3] F. Cejas, “Architecting Android...The clean way?.” <https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/>, 2014. [Consultado: 2019-09-21]. vi, vi, 112, 117
- [4] F. Cejas, “Architecting Android...Reloaded.” <https://fernandocejas.com/2018/05/07/architecting-android-reloaded/>, 2018. [Consultado: 2019-09-21]. vi, 123
- [5] J. Zhanga, J. Dua, S. Zhanga, D. Liub, Y. Hub, J. Hub, S. Weib, and L. Daia, “Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition,” *Pattern Recognition*, p. 196–206, 2017. 2, 18
- [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]. 8, 9

- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>. 10, 11
- [14] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. 2001. 17
- [15] R. H. Anderson, “Syntax directed recognition of hand-printed two-dimensional mathematics,” *Harvard University*. 17
- [16] E. Tapia and R. Rojas, “Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance,” *Freie Universit at Berlin, Institut fur Informatik*. 18
- [17] G. Genthial and R. Sauvestre, “Image to latex,” *Stanford University*. 18, 126
- [18] 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. 18
- [19] S. Luján-Mora, “Programación en internet: clientes web,” 2001-10-08. 18
- [20] 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>. 20
- [21] 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. 20
- [22] 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. 21
- [23] “Identificación y autenticación.” https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_7.5.0/com.ibm.mq.sec.doc/q009740_.htm. 21
- [24] C. Salazar, “Los principios de SOLID de la programación orientada a objetos.” <https://www.codesolt.com/tutoriales/fundamentos/solid/>, 2018. [Consultado: 2019-09-21]. 23
- [25] 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]. 24
- [26] 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]. 33

- [27] “System Properties Comparison MySQL vs. Oracle vs. PostgreSQL.” <https://db-engines.com/en/system/MySQL%3BOracle%3BPostgreSQL>, 2019. 34
- [28] 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. 35
- [29] A. Kanervisto, “im2latex-100k , arxiv:1609.04938,” June 2016. 126