



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 Junio del 2019

Índice general

Índice de figuras	III
Índice de tablas	IV
I title	1
1. Introducción	2
1.1. Palabras Clave	2
1.2. Planteamiento del problema	2
1.3. Objetivos	6
1.4. Estado del arte	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
2.1.4. Extracción de características	10
2.1.5. Clasificación e interpretación	10
2.2. Métodos de reconocimiento de expresiones matemáticas	10
2.2.1. Análisis sintáctico dirigido	10
2.2.2. Redes Neuronales	10
2.2.2.1. Conjunto de entrenamiento	12
2.2.2.2. Aprendizaje Profundo	12
2.2.2.3. Redes Neuronales Convolucionales	13
2.2.2.4. Arquitectura Encoder-Decoder	13
2.2.2.5. Image Captioning	14
2.2.3. Análisis estructural	14
2.3. Aplicación web	14
2.3.1. Django	15
2.3.1.1. Modelo-Vista-Template	16
2.3.2. API REST	17

2.3.2.1. Autenticación	17
2.4. Aplicación Android	17
2.4.1. Arquitectura Clean	17
2.4.2. SOLID	18
3. Análisis del sistema	19
3.1. Requerimientos funcionales	19
3.1.1. Módulo de usuarios	19
3.1.2. Módulo de proyectos	19
3.1.3. Módulo de análisis	20
3.1.4. Módulo de traducción	20
3.2. Requerimientos No Funcionales	20
3.3. Requerimientos técnicos	20
3.3.1. Aplicación móvil	21
3.3.1.1. Requerimientos mínimos de software	21
3.3.1.2. Requerimientos mínimos de hardware	21
3.3.2. Aplicación web	21
3.3.2.1. Requerimientos mínimos de software	21
3.3.2.2. Requerimientos mínimos de hardware	21
3.4. Reglas de Negocio	21
3.4.1. RN-001 Campos obligatorios	22
3.4.2. RN-002 Datos correctos	22
3.4.3. RN-003 Unicidad de identificadores	22
3.4.4. RN-004 Calificación proyecto	22
3.4.5. RN-005 Fecha de modificación	23
3.4.6. RN-006 Usuario verificado	23
3.4.7. RN-007 Información necesaria para descargar un proyecto	23
3.5. Análisis de Riesgos	24
3.6. Descripción del software	26
3.6.1. Android	26
3.6.2. Base de datos	27
3.6.3. Framework de desarrollo web	28
Bibliografía	30

Índice de figuras

1.1. Arquitectura del sistema.	5
2.1. Imagen con bloques (Izquierda) y conjunto de segmentos de línea extraídos (Derecha).	9
2.2. 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.	11
2.3. Representación de un perceptrón con entradas x_1 , x_2 y x_3	11
2.4. Red neuronal de 3 capas	12
2.5.	13
2.6. Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.	15
2.7. Utilización de un árbol de recubrimiento mínimo para el reconocimiento de expresiones matemáticas.	15
2.8. Representación del patrón de diseño Modelo-Vista-Template.	16
2.9. Arquitectura Clean [?]	17

Índice de tablas

1.1. Resumen de productos similares	7
3.1. Análisis de riesgos.	25
3.2. Tabla comparativa de sistemas operativos de dispositivos móviles	27
3.3. Comparación de diferentes gestores de bases de datos.	28
3.4. Comparación de frameworks de desarrollo web.	29

Parte I

title

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 del usuario 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

La comunidad que utiliza LaTeX es muy grande en el ámbito académico y científico, sus principales usuarios son investigadores y alumnos de ciencias exactas tales como matemáticas, física, química e incluso ciencias de la computación[?]. Esto es debido a que la escritura de ecuaciones en LaTeX es mucho más eficiente que en algún otro editor de texto [6]. Sin embargo, debido a que la escritura de expresiones matemáticas requiere cierto conocimiento de cómo se tienen que estructurar esto puede llegar a ser algo difícil para personas que empiezan a utilizar LaTeX.

El enfoque del presente Trabajo Terminal es amenizar la interacción de nuevos usuarios y reducir el tiempo de la escritura de expresiones matemáticas mediante su reconocimiento en imágenes.

El sistema está dirigido a estudiantes y científicos que tengan que elaborar documentos con contenido matemático. Esta aplicación está enfocada para usuarios de LaTeX sin importar que tanta experiencia tengan en el lenguaje.

A diferencia del reconocimiento del texto plano el cual es un problema que se considera resuelto por los buenos resultados que se han obtenido, el reconocimiento de expresiones matemáticas sigue siendo un problema abierto debido a su naturaleza jerárquica y bidimensional y que en la actualidad las soluciones existentes no proporcionan un buen resultado [7 Agregar Referencia], por lo que mejorar los resultados del estado del arte no es el objetivo de este trabajo.

El tiempo estimado de desarrollo del Trabajo Terminal es de diez meses, empezando en el mes de agosto 2019 y concluyendo en el mes de mayo del 2020 (como se puede ver en la Sección 6).

PRODUCTOS O 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.

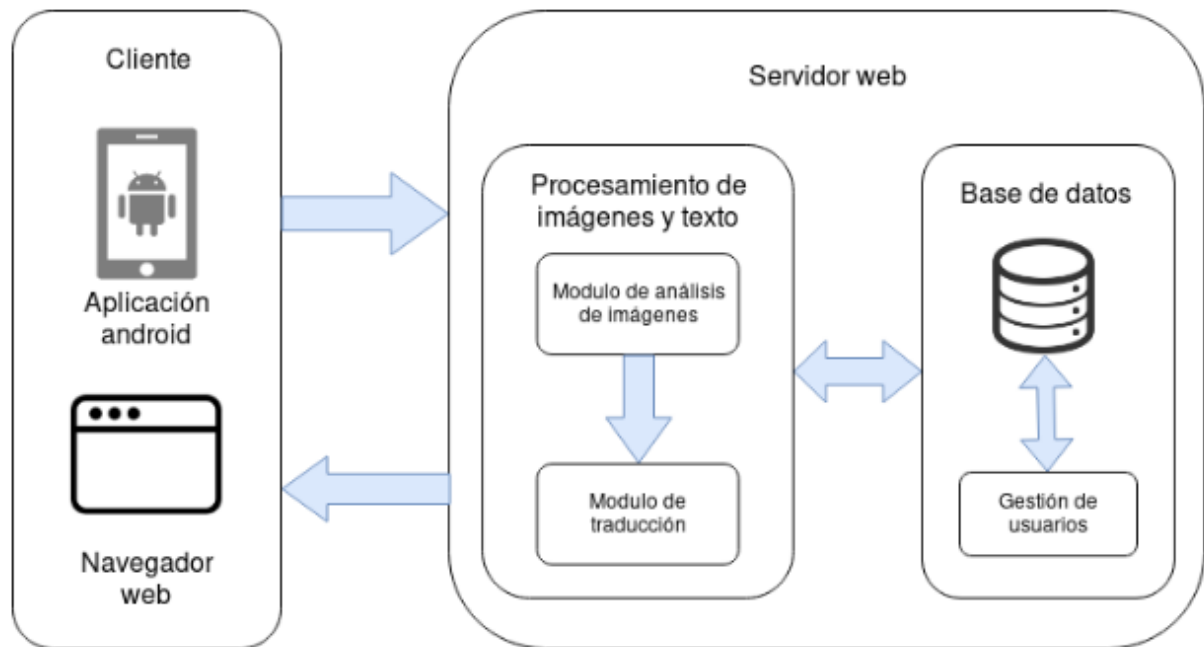


Figura 1.1: Arquitectura del sistema.

METODOLOGÍA Para el desarrollo del proyecto se eligió la metodología incremental debido a que permite dividir el proyecto en pequeñas iteraciones secuenciales [?]. En cada iteración, se tendrá la libertad de implementar los requerimientos más apropiados para un correcto flujo en el desarrollo de la aplicación. A pesar de que la metodología por incrementos permite flexibilidad para modificar los requerimientos funcionales, se tienen planeadas las siguientes actividades que se llevarán a cabo en distintas iteraciones:

- Se realizará obtención de requerimientos, el análisis y diseño (modelado) del módulo de análisis de imágenes, construcción del módulo y finalmente se harán pruebas sobre este.
- El módulo de traducción se realizará paralelamente al 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.
- 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 el módulo de servidor web ya que es lo que nos servirá para conectar el módulo de análisis de imágenes con la aplicación móvil y se probará la comunicación entre cada uno de estos elementos.

1.3. Objetivos

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. Objetivos específicos:

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

1.4. Estado del arte

Algunos sistemas similares que se han desarrollado son:

- Mathpfix
- MyScript Nebo
- SESHAT
- IDEAL Math Writer

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

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.

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. [?]

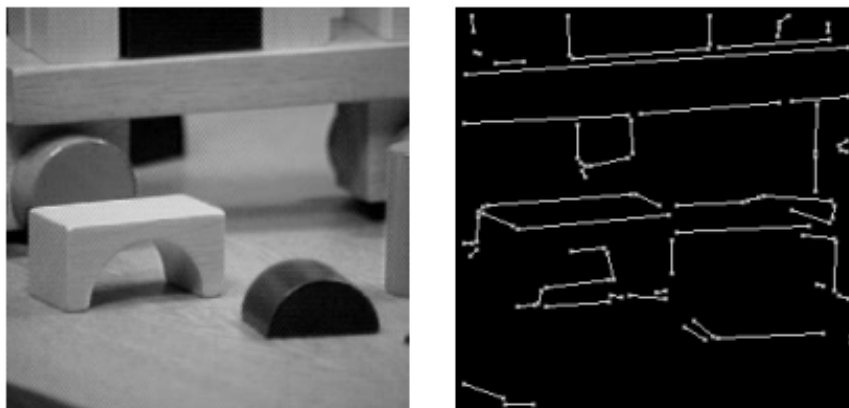


Figura 2.1: Imagen con bloques (Izquierda) y conjunto de segmentos de linea extraídos (Derecha).

2.1.4. Extracción de características

2.1.5. Clasificación e interpretación

2.2. Métodos de reconocimiento de expresiones matemáticas

2.2.1. 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:

$$G = (V, T, P, S) \quad (2.1)$$

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 [$?$].

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 [$?$]. 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 2.2, se puede ver un ejemplo del proceso de reconocimiento propuesto por este método.

2.2.2. Redes Neuronales

Existe un tipo de neurona artificial llamada perceptrón desarrolladas entre 1950 y 1960 por el científico Frank Rosenblatt, sin embargo este modelo ha ido evolucionando y hoy en día se utilizan otros modelos de neurona como el llamado sigmoid neuron, que para entender, primero se debe definir al perceptrón como modelo. Un perceptrón toma varias entradas binarias $x_1, x_2, x_3, \dots, x_n$ y produce una sola salida binaria:

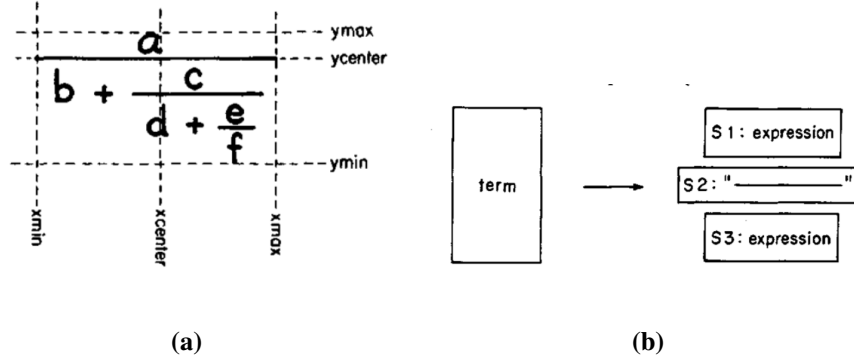


Figura 2.2: a) La primera etapa del método de reconocimiento, se anotan las coordenadas de de cada caracter. b) Se realiza un análisis sintáctico con las gramáticas libres de contexto definidas.

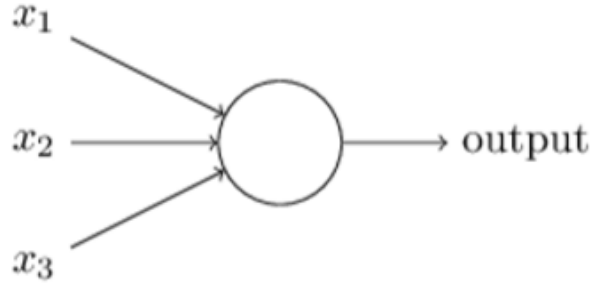


Figura 2.3: Representación de un percetrón con entradas x_1 , x_2 y x_3

La regla que Rosenblatt introdujo para calcular la salida involucra valores llamados pesos w_1, w_2, w_3, \dots que pueden tomar valores reales expresando la importancia de las respectivas entradas a la salida, la salida de la neurona (0 o 1) es determinado por la suma ponderada $\sum_j w_j x_j$ es menor o igual a un valor de umbral. Así como los pesos, el valor de umbral es un número real el cuál es un parámetro de la neurona:

$$output = \begin{cases} 0 & \text{si } \sum_j w_j x_j \leq threshold \\ 1 & \text{si } \sum_j w_j x_j \geq threshold \end{cases} \quad (2.2)$$

Una red neuronal es una red de perceptrones que agrega complejidad a las decisiones.

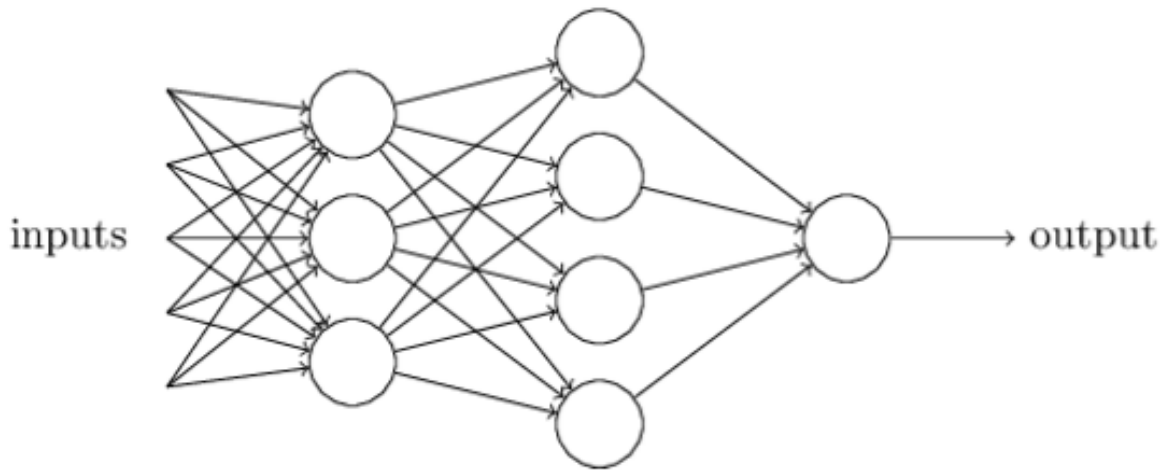


Figura 2.4: Red neuronal de 3 capas

2.2.2.1. Conjunto de entrenamiento

El conjunto de entrenamiento es un conjunto de datos que se utiliza a modo de ejemplo para que un modelo pueda aprender. En el aprendizaje supervisado el conjunto de entrenamiento se compone de muchas tuplas (X,Y) siendo X un vector de características del problema y Y un vector del resultado al que queremos llegar [?].

2.2.2.2. Aprendizaje Profundo

El aprendizaje profundo de forma clásica se refiere al tratamiento de redes neuronales con mas de dos capas, sin embargo, debido al cambio que esta sub-area ha sufrido, se debe definir como el tratamiento de redes neuronales con un largo número de parámetros y capas en una de las cuatro arquitecturas de redes fundamentales:

- Redes pre-entrenadas no supervisadas
- Redes Neuronales Convolucionales
- Redes neuronales recurrentes
- Redes neuronales recursivas.

Siendo las Redes Neuronales Convolucionales de las que se expondrá a detalle en

Una de las propiedades mas importantes que permite el Aprendizaje Profundo es la extracción automática de características que puede ser clave en dicho paso del análisis de imágenes.

2.2.2.3. Redes Neuronales Convolucionales

Las redes neuronales convolucionales han ganado popularidad en los últimos años debido a su efectividad prometedora en el aprendizaje profundo. Partiendo en el procesamiento de imágenes, las capas convolucionales han encontrado su camino en otros subcampos del aprendizaje profundo y muestran éxito en la mayor parte de dichos campos.

La diferencia fundamental entre *fully connected* (completamente conectadas) y *convolutional neural networks* (redes neuronales convolucionales) es el patrón de conexión entre las capas consecutivas. En el caso de las completamente conectadas como el nombre lo sugiere, cada unidad es conectada a todas las unidades de la previa capa.

En una capa convolucional de una red neuronal, por otra parte, cada unidad está conectada a (típicamente pequeña) un número de unidades cercanas en la capa previa. Más aún, todas las unidades están conectadas a la capa previa de la misma forma con los pesos exactamente iguales y estructura. Esto lleva a una operación conocida como convolución, dando a la arquitectura este nombre, a grandes rasgos esto significa aplicar una pequeña "ventana" de pesos (conocida como filtro) sobre una imagen como se muestra en la figura

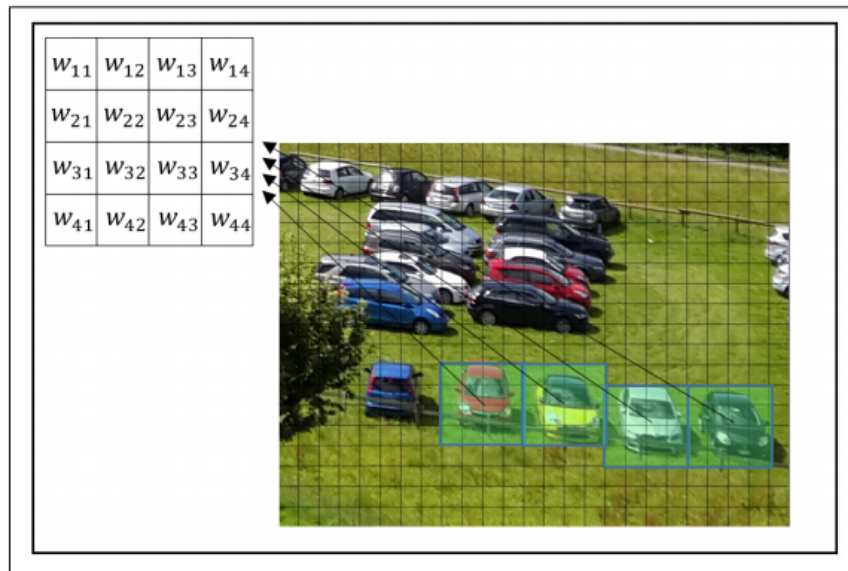


Figura 2.5: ...

2.2.2.4. Arquitectura Encoder-Decoder

En particular se discutirá la arquitectura de Long Short-Term Memory (LSTM). LSTM es una arquitectura de una red neuronal recurrente diseñada para resolver problemas secuenciales, normalmente conocidos como sequence-to-sequence. Los problemas secuenciales, tienen usualmente como reto, predecir el siguiente resultado encontrado. Un reto que surge naturalmente de

estos problemas, es la longitud variable de las entradas.

Una aproximación para resolver los problemas secuenciales es el Encoder-Decoder LSTM el cual ha probado ser efectivo en muchos casos. Esta arquitectura se compone de dos modelos [?]:

- Encoder: Es una red neuronal encargada de leer la entrada de longitud variable y convertirla en un vector de longitud fija.
- Decoder: Es una red neuronal encargada de tomar la salida del Encoder y como salida obtiene un valor predecido.

2.2.2.5. Image Captioning

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

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

La arquitectura que se esta utilizando actualmente para solucionar este problema se muestra en la Figura ?? [?][?][?].

2.2.3. Análisis estructural

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

La diferencia en este método, radica en su segunda etapa. No se realizará un análisis con gramáticas, en su lugar se intentará deducir la estructura jerárquica con algún otro método. El artículo [?] propone utilizar el algoritmo de Kruskal para obtener un árbol de recubrimiento mínimo por niveles, de este modo se puede saber la estructura de la expresión matemática recorriendo el grafo resultante. La Figura 2.7 muestra un ejemplo de como funciona este método.

2.3. Aplicación web

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

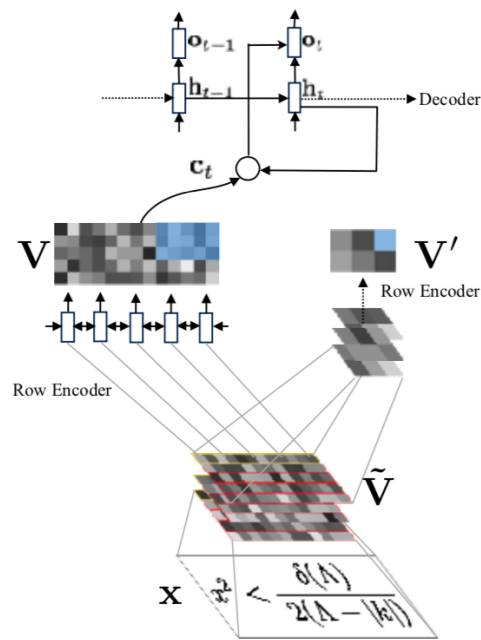


Figura 2.6: Arquitectura de image captioning para reconocer expresiones matemáticas en imágenes.

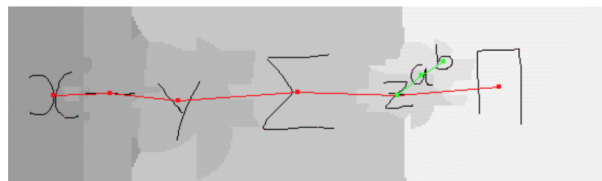


Figura 2.7: Utilización de un árbol de recubrimiento mínimo para el reconocimiento de expresiones matemáticas.

2.3.1. 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.

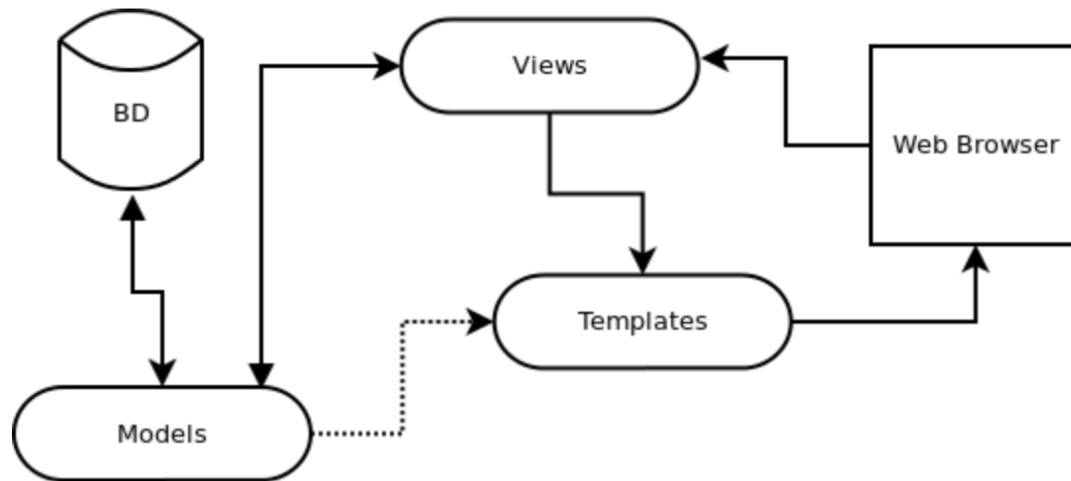


Figura 2.8: Representación del patrón de diseño Modelo-Vista-Template.

- **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.

2.3.1.1. Modelo-Vista-Template

El Modelo-Vista-Template (MTV) es el patrón de diseño que Django implementa. Este patrón es una modificación al conocido Modelo-Vista-Controlador (MVC). La diferencia radica en que Django se encarga de hacer la parte del controlador, por ende el desarrollador solamente tiene que preocuparse por implementar la lógica de negocio y de como mostrará los datos. En la Figura 2.8, se puede ver una representación del patrón MTV. En el patrón de diseño MTV [?]:

- **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.

2.3.2. API REST

2.3.2.1. Autenticación

2.4. Aplicación Android

2.4.1. 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 2.9: Arquitectura Clean [?]

La arquitectura Clean como se muestra en la figura 2.9 fue descrita por Robert C. Martin en 2012 y al utilizarla se puede llegar a tener las siguientes características en los sistemas [?]:

- 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.

2.4.2. 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. [?]

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. [?]

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.[?]

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.[?]

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.[?]

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. [?]

Capítulo 3

Análisis del sistema

3.1. Requerimientos funcionales

Los requerimientos funcionales describen los actividades y comportamientos que tendrá el sistema bajo ciertas condiciones.

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.

RF2. Mecanismo de autenticación de usuarios

RF3. Mecanismo de recuperación de contraseñas.

RF4. Mecanismo para la comunicación entre la aplicación android y la aplicación web.

3.1.2. Módulo de proyectos

RF5. Mecanismo para la gestión de proyectos de \LaTeX .

RF6. Permitir descargar el archivo de \LaTeX que se haya traducido.

RF7. Permitir calificar una traducción realizada.

RF8. Permitir el uso de la capara del dispositivo android para tomar fotografías.

RF9. Permitir la visualización del resultado de la traducción.

RF10. Permitir añadir al portapapeles alguna traducción seleccionada.

RF11. Mecanismo para el envío de imágenes tomadas por la aplicación android a la aplicación web para su uso.

3.1.3. Módulo de análisis

RF12. Mecanismo para el tratamiento de la imagen previo a su análisis.

RF13. Mecanismo para el reconocimiento de un conjunto definido de expresiones matemáticas en imágenes.

3.1.4. Módulo de traducción

RF14. Mecanismo para la traducción a \LaTeX de las expresiones matemáticas encontradas en el modulo de análisis.

3.2. Requerimientos No Funcionales

A continuación se enlistan los requerimientos no funcionales.

RNF1. Escalabilidad. El sistema deberá ser fácilmente escalable con la finalidad de soportar a una mayor cantidad de usuarios permitiendo dar respuestas menores a **x minutos** por petición de traducción.

RNF2. Disponibilidad. El sistema debería estar disponible en la mayor parte del tiempo a excepción de operaciones de soporte planeadas y previamente notificadas al usuario final.

RNF3. Estabilidad.

RNF4. Seguridad.

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.

3.3.1.2. Requerimientos mínimos de hardware

1. Memoria RAM: 2 GB.
2. Espacio de almacenamiento de 50 MB.

3.3.2. Aplicación web

3.3.2.1. Requerimientos mínimos de software

1. Cualquiera de los siguientes navegadores web:
 - Google Chrome 7
 - Edge
 - Internet Explorer 10
 - Firefox 4
 - Opera 12
 - Safari 5

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

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 [?]	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 [?]	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 [?].

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 [?].

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.

Bibliografía
