

**UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**



**Sistema de reconocimiento de texto mecanografiado mediante redes neuronales para la gestión de boletas de pago en la Ugel Ferreñafe**

**TESIS PARA OPTAR EL TÍTULO DE  
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

**AUTOR**

**Jonathan Alonso Bonilla Vilchez**

**ASESOR**

**Guadalupe Teresa Lip Curo**

<https://orcid.org/0000-0002-0353-939X>

**Chiclayo, 2024**

**Sistema de reconocimiento de texto mecanografiado mediante redes neuronales para la gestión de boletas de pago en la Ugel Ferreñafe**

PRESENTADA POR

**Jonathan Alonso Bonilla Vilchez**

A la Facultad de Ingeniería de la  
Universidad Católica Santo Toribio de Mogrovejo  
para optar el título de

**INGENIERO DE SISTEMAS Y COMPUTACIÓN**

APROBADA POR

Maria Ysabel Aranguri Garcia  
PRESIDENTE

Ricardo David Iman Espinoza  
SECRETARIO

Guadalupe Teresa Lip Curo  
VOCAL

## **Dedicatoria**

A mi familia que fue mi mayor apoyo durante todo este proceso.  
A mi asesora por toda su guía para la elaboración de esta tesis.

## **Agradecimientos**

A mis padres, quienes siempre me dieron el aliento y la fuerza para lograr mis metas, y a mi asesora de tesis por guiarme durante el desarrollo de este proyecto.

## INFORME DE ORIGINALIDAD

22%

INDICE DE SIMILITUD

22%

FUENTES DE INTERNET

2%

PUBLICACIONES

5%

TRABAJOS DEL  
ESTUDIANTE

### FUENTES PRIMARIAS

1

[hdl.handle.net](http://hdl.handle.net)

Fuente de Internet

7%

2

[repositorio.espe.edu.ec](http://repositorio.espe.edu.ec)

Fuente de Internet

3%

3

[tesis.usat.edu.pe](http://tesis.usat.edu.pe)

Fuente de Internet

2%

4

[repositorio.ucv.edu.pe](http://repositorio.ucv.edu.pe)

Fuente de Internet

<1%

5

Submitted to Universidad Católica Santo  
Toribio de Mogrovejo

Trabajo del estudiante

<1%

6

[docplayer.es](http://docplayer.es)

Fuente de Internet

<1%

7

[idoc.pub](http://idoc.pub)

Fuente de Internet

<1%

8

[bibdigital.epn.edu.ec](http://bibdigital.epn.edu.ec)

Fuente de Internet

<1%

9

Submitted to Universidad Cesar Vallejo

Trabajo del estudiante

## Índice

<b>Resumen .....</b>	<b>6</b>
<b>Abstract .....</b>	<b>7</b>
<b>Introducción.....</b>	<b>8</b>
<b>Revisión de literatura.....</b>	<b>9</b>
<b>Materiales y métodos .....</b>	<b>18</b>
<b>Matriz de consistencia.....</b>	<b>24</b>
<b>Resultados y discusión .....</b>	<b>26</b>
<b>Conclusiones .....</b>	<b>33</b>
<b>Recomendaciones .....</b>	<b>33</b>
<b>Referencias.....</b>	<b>34</b>
<b>Anexos .....</b>	<b>37</b>

## Resumen

En este proyecto, se llevó a cabo un estudio con el objetivo de desarrollar un sistema de reconocimiento óptico de caracteres (OCR) diseñado para identificar y almacenar la información de las boletas de pago de docentes en la UGEL Ferreñafe. Esto se debió a la necesidad de agilizar la búsqueda de boletas en formato físico, un proceso que, en ocasiones, podía llevar semanas y requerir la contratación de personal adicional. Esta problemática impulsó la búsqueda de una solución eficaz y rentable.

Siguiendo las metodologías SCRUM y CRISP-DM, se optó por utilizar Redes Neuronales (RN) como la técnica principal. Esta elección se basó en investigaciones previas y tendencias identificadas en Google Trends. El objetivo fundamental era alcanzar un porcentaje de error bajo en la tasa de caracteres reconocidos, y se logró un hito significativo del 1.8%, a pesar de la degradación de la tinta en muchas boletas debido al paso del tiempo.

Para evaluar la usabilidad del sistema, se aplicó la escala SUS (System Usability Scale), y el sistema obtuvo una puntuación de 80, superando las expectativas iniciales. Esto resalta la alta usabilidad y satisfacción de los usuarios finales con la aplicación desarrollada.

Palabras clave: Modelo de reconocimiento, Redes neuronales.

## **Abstract**

In this project, a study was carried out with the objective of developing an optical character recognition (OCR) system designed to identify and store information from teacher pay slips at UGEL Ferreñafe. This was due to the need to expedite the search for physical ballots, a process that could sometimes take weeks and require the hiring of additional staff. This problem prompted the search for an effective and profitable solution.

Following the SCRUM and CRISP-DM methodologies, it was decided to use Neural Networks (RN) as the main technique. This choice was based on previous research and trends identified in Google Trends. The fundamental objective was to achieve a low error rate in the rate of recognized characters, and a significant milestone of 1.8% was achieved, despite the degradation of the ink on many ballots due to the passage of time.

To evaluate the usability of the system, the SUS scale (System Usability Scale) was applied, and the system obtained a score of 80, exceeding initial expectations. This highlights the high usability and satisfaction of end users with the developed application.

Keywords: Recognition model, Neural networks.

## Introducción

En la actualidad, las organizaciones dependen en gran medida de herramientas tecnológicas para llevar a cabo sus operaciones diarias. Sin embargo, hace algunos años, la gestión de información se realizaba manualmente, lo que llevó a la necesidad de mantener almacenes con documentación física o digitalizar documentos relevantes. Para abordar este desafío, se han desarrollado diversas técnicas de reconocimiento de caracteres, entre las cuales se destacan las redes neuronales artificiales (RN) [1]. La digitalización eficiente de documentos es fundamental, ya que a nivel mundial, el 15% de los documentos empresariales se almacenan incorrectamente, y en Europa, el 45% de las empresas afirman que sus sistemas de búsqueda no son óptimos [2]. Esta problemática no es ajena a Perú, donde muchas organizaciones, en particular en el sector público, mantienen documentos en formato físico.

La Unidad de Gestión Educativa Local (UGEL) de Ferreñafe, posee boletas de pago mecanografiadas de profesores que abarcan un periodo de 34 años. La búsqueda de estos documentos puede llevar entre 1 y 3 semanas, dependiendo de su antigüedad, lo que afecta la eficiencia de la institución y la satisfacción de los usuarios. Ante esta situación, surge la pregunta: ¿Cómo se puede mejorar la gestión de boletas de pago en la UGEL de Ferreñafe? La respuesta a esta interrogante se encuentra en el Reconocimiento Óptico de Caracteres (OCR), una técnica que utiliza redes neuronales para digitalizar documentos y facilitar la búsqueda de información.

Esta investigación se justifica científicamente porque servirá un precedente para futuros proyectos que trabajen haciendo uso de OCR para adquirir datos de documentos escaneados o imágenes. Se justifica de manera tecnológica por qué hace uso de tecnologías y herramientas actuales para el desarrollo del modelo. Por último, se justifica de manera empresarial porque será un apoyo directo al en la gestión documental con el manejo de data de las boletas de pago de una manera más eficaz.

En ese sentido se plantea como objetivo general: Implementar un sistema de reconocimiento de texto mecanografiado mediante la utilización de redes neuronales para la gestión de boletas de pago en la UGEL de Ferreñafe ; para lograrlo se plantearon los siguientes objetivos específicos: Comparar técnicas de reconocimiento de caracteres para seleccionar la que mejor reconozca el texto mecanografiado, alcanzar un bajo porcentaje en la tasa de error de caracteres y obtener una valoración mayor a 68 de la usabilidad de la aplicación por parte de los usuarios finales



## **Revisión de literatura**

### **Antecedentes**

Se han considerado para esta investigación los siguientes antecedentes:

#### **Antecedentes internacionales**

López [5] en su trabajo, se abordó la implementación de un sistema de digitalización y reconocimiento óptimo de caracteres (OCR). Esta iniciativa llevó a la optimización de los tiempos de atención al usuario, simplificando considerablemente la búsqueda de documentos y mejorando el control sobre las carpetas donde se almacenaba la información. Para la creación de la aplicación de escritorio en este proyecto, se empleó el lenguaje de programación C#, lo que permitió administrar y visualizar documentos gráficos de diversos formatos, como TIFF, JPG, BMP, PNG y PDF. Es importante destacar que el escáner utilizado para capturar las imágenes utilizó tecnología TWAIN, lo que aseguró la calidad y eficiencia en la adquisición de datos. Es relevante subrayar que todo el desarrollo del proyecto se llevó a cabo siguiendo la metodología SCRUM, lo que garantizó una gestión ágil y eficiente de cada etapa del proceso.

El Battioui [6] llevo a cabo una investigación centrada en la aplicación de CNNs (Redes Neuronales Convolucionales) para el reconocimiento de caracteres en imágenes en el contexto industrial, con el objetivo de optimizar algunas de las tareas preexistentes. En este proceso, se abordó la definición de una CNNs, considerando imágenes sometidas a condiciones del mundo real, incluyendo aspectos como la iluminación, nitidez y otros factores similares. Como resultado de este esfuerzo, se logró alcanzar una tasa de error extremadamente baja, del 0.47%.

Talat et al. [7] se enfocaron en la selección clave de características para la eficacia de un sistema de reconocimiento de caracteres. Se presenta una propuesta que combina aprendizaje automático con conjuntos aproximados y un algoritmo de optimización para identificar características óptimas en el reconocimiento de caracteres árabes manuscritos. Utilizaron el conjunto de datos CENPARMI, ampliamente aceptado en experimentos de aprendizaje automático con caracteres árabes manuscritos, para evaluar nuestra aproximación. Sus resultados destacan las ventajas evidentes en términos de precisión, eficiencia de memoria y tiempo de procesamiento en comparación con métodos sin las características propuestas. También supera las redes neuronales profundas como VGGnet, Resnet, Nasnet, Mobilenet, Inception y Xception en términos de tasa de reconocimiento y tiempo. Comparaciones con trabajos recientes que utilizan el mismo conjunto de datos refuerzan nuestra alta precisión y eficiencia temporal. Además, examinaron las ocasiones

en que se produjeron fallos en la clasificación, señalando que incluso hablantes nativos de árabe pueden encontrar ciertos casos confusos debido a la dependencia del contexto en la interpretación correcta de los caracteres.

Nasir et al. [8] en su investigación, se introdujo un extenso corpus en urdu, un idioma de gran prominencia en el sur de Asia, denominado MMU-OCR-21. Este corpus es excepcionalmente amplio y abarca una variedad de niveles y estilos de escritura. De hecho, es el corpus más extenso creado hasta la fecha para texto impreso en urdu y ha sido especialmente diseñado para funcionar de manera efectiva con enfoques de aprendizaje profundo. En su totalidad, este corpus incluye más de 602,472 imágenes, que engloban tanto líneas de texto como palabras escritas en tres fuentes de gran relevancia, acompañadas de sus correspondientes referencias precisas. Además, se llevaron a cabo una serie de experimentos utilizando diversas técnicas de aprendizaje profundo de vanguardia, con un enfoque específico en imágenes de líneas de texto y palabras. Los resultados del estudio indican que la mayoría de los modelos de aprendizaje profundo utilizados en los experimentos se adaptaron bien al corpus desarrollado y lograron puntuaciones notables en términos de CER (Tasa de Error de Caracteres) y WER (Tasa de Error de Palabras).

Cao et al. [9] este artículo presenta una propuesta basada en el aprendizaje profundo para la detección de texto. Este enfoque mejora el modelo You Only Look Once versión 3 (YOLOv3) al modificar el objeto de regresión, pasando de un solo carácter a un texto de ancho fijo. Además, se implementa una estrategia de ensamblaje que construye líneas de texto a partir de la matriz de relaciones, logrando una mejora significativa en la precisión del 9.8%. Los resultados experimentales, tanto en el conjunto de datos de detección de texto chino RCTW como en un entorno real de ejercicios, demuestran que este modelo puede aumentar la efectividad en la detección de texto. También se llevó a cabo una comparación con dos enfoques líderes en la detección de texto en ejercicios, donde se discutieron las capacidades y limitaciones de cada método. En adición, se proporciona una plataforma en esta propuesta que permite detectar líneas de texto en los ejercicios diarios de los estudiantes o en los exámenes, mejorando de manera significativa la experiencia del usuario.

### **Antecedentes nacionales**

Luna [10], en su proyecto, se llevó a cabo el desarrollo de una aplicación móvil equipada con tecnología OCR, permitiendo la captura y traducción de textos. Esto se logra mediante un sistema de búsqueda dinámico, respaldado por un diccionario de idiomas que muestra la traducción correspondiente de cada término. En esta investigación, el autor optó por utilizar la metodología de desarrollo RUP. Las conclusiones del estudio señalan que la aplicación

móvil desempeñó un papel fundamental en la conversión de diversos tipos de imágenes en texto. Las imágenes, capturadas por una cámara digital y escritas en un idioma desconocido para el turista, se resolvieron fácilmente gracias a la opción de búsqueda en el diccionario de datos de idiomas. Este enfoque, a su vez, mejoró el proceso de promoción turística en la provincia de San Martín.

Espejo [11] Se nos plantea un desafío relacionado con el tiempo y los posibles errores inherentes a la transcripción de datos desde formularios a un sistema informático, particularmente en lo que respecta a los números. Con el objetivo de agilizar el proceso de llenado de la base de datos y prevenir los errores humanos comunes en esta transcripción, hemos concebido este proyecto. La meta es lograr que los empleados simplemente coloquen los formularios en una plataforma para luego capturar una fotografía. A través de un algoritmo que emplea redes neuronales, se llevará a cabo la transferencia de los datos del formulario al sistema informático y el posterior llenado en la base de datos. Esta solución se implementará utilizando Matlab y estará vinculada con las bases de datos de Microsoft Access, donde se almacenarán los números reconocidos por el programa. Es importante destacar que esta tesis proporcionó una guía valiosa para comprender cómo se pueden reconocer los caracteres numéricos escritos a mano, abordando una problemática muy similar a la que se abordará en este proyecto.

Núñez [12] en su investigación, se ha desarrollado una plataforma que cumple con los estándares nacionales para la digitalización de documentos físicos en formatos electrónicos. Esto se logró utilizando escáneres de diversas marcas y modelos para la digitalización de los documentos. Los documentos digitalizados fueron debidamente indexados, lo que implica que se vinculan con datos específicos para facilitar su recuperación a través de un módulo de búsqueda. Además, esta plataforma incorpora capacidades de reconocimiento óptico de caracteres (OCR), lo que permite buscar contenido dentro de las imágenes escaneadas. El sistema también integra un gestor documental que facilita la administración y organización de los documentos electrónicos en carpetas y subcarpetas. El módulo de búsqueda simplifica la localización rápida de documentos e imágenes, ya sea utilizando campos indexados o palabras clave reconocidas por el OCR. Esto resulta en una drástica reducción del tiempo necesario para acceder a documentos en comparación con la búsqueda de documentos físicos. Adicionalmente, el sistema permite el acceso simultáneo de múltiples usuarios a la misma información, lo que agiliza el flujo de trabajo y fomenta la colaboración en la gestión de documentos.

Aquino [13] para la elaboración de esta investigación, de carácter descriptivo, el autor se optó por una metodología que combina Programación Extrema (Extreme Programming - XP) y UML. Dentro de este marco, se desarrolló un sistema de gestión documental que incorpora tecnología OCR. Los resultados obtenidos brindaron al usuario la capacidad de administrar información de manera altamente eficiente. Esto se logra al permitir al usuario organizar y dar seguimiento a sus documentos, realizar búsquedas personalizadas mediante índices, y visualizar tanto los documentos como el historial de cada usuario. De esta manera, se mejora la facilidad y la disponibilidad de acceso a todo tipo de información relevante para la empresa

### **Antecedentes locales**

Santisteban [14] El propósito fundamental de esta investigación fue establecer la relación entre la digitalización de documentos y el sistema de gestión documental. Para lograrlo, se implementó un enfoque cuantitativo con un diseño no experimental de tipo transversal. La muestra de estudio incluyó a 385 usuarios pertenecientes a la Municipalidad Distrital de Picsi, ubicada en la provincia de Chiclayo. La metodología empleada para la recopilación de datos consistió en la aplicación de encuestas mediante cuestionarios diseñados para ambas variables de estudio. Los resultados obtenidos revelaron una relación directa entre la digitalización de documentos y la eficacia del sistema de gestión documental en una municipalidad del distrito de Chiclayo. Esto se confirmó mediante un nivel de significancia inferior al 0.05 y un coeficiente de Spearman de ( $r= 0,571$ ). En conclusión, se demostró que a medida que aumenta el grado de digitalización de documentos, se incrementa la eficiencia del sistema de gestión documental.

Loconi y Leiva [15] implementó el Modelo de Gestión Documental utilizando la metodología DIRKS (Designing and Implementing Recordkeeping Systems) como una solución a la gestión ineficiente de la información. A través del diseño de este modelo, se logró identificar las funciones, actividades y tareas involucradas en el proceso de gestión documental. Además, se determinaron los grupos documentales pertinentes en cada caso, lo que proporcionó directrices claras para la implementación del Sistema de Gestión de Documentos. Como resultado de esta implementación, se concluye que ha habido una mejora significativa en la administración de expedientes en las oficinas de la entidad. Esta mejora, a su vez, tiene un impacto positivo en la gestión de documentos y contribuye al logro de los objetivos organizacionales.

Vega [16] este proyecto se llevó a cabo siguiendo una metodología cuantitativa y adoptando un enfoque de investigación no experimental. La población objeto de estudio estuvo compuesta en su totalidad por el personal empleado en la Municipalidad Provincial de Ayabaca, Chiclayo. Se empleó la encuesta como técnica principal para la recopilación de datos, con el propósito de obtener información relevante acerca de las variables relacionadas con el Sistema de Información y la Gestión Documental.

### **Bases teórico científicas**

#### ***Reconocimiento de texto***

Para la situación problemática que se tuvo fue necesario aplicar el reconocimiento de texto debido a que se contó con documentación física, este es un proceso que consiste en aplicar métodos que comparen la forma de los caracteres leídos por un computador para conseguir determinar qué caracteres alfanuméricos o qué tipo de signos de puntuación representan dichas formas [16]. Tiempo atrás que se utilizaran los escáneres y los programas de OCR (Reconocimiento óptico de caracteres), se utilizaba una técnica titulada MICR (Magnetic Ink Character) para el reconocimiento de los caracteres con tinta magnética, que aún es utilizada hoy en día para el tratamiento de los cheques y otros elementos bancaros. Estos lectores de MICR detectan los caracteres mediante las propiedades magnéticas que tiene la tinta con la que estos fueron impresos, los convierten a datos digitales y listos para ser procesados por un computador [17].

La mayor ventaja que conlleva el reconocer un texto mecanografiado luego de escanearlo es la notable reducción del tamaño que este representa, contando con libros enteros físicos pero guardados como archivos de imagen en un disco duro. Dándonos la posibilidad de guardar grandes cantidades de información y ahorrando los costos de la entidad [18]. El proceso del reconocimiento del texto tiene las siguientes etapas:

Escaneo del texto, donde se comienza por la pre-digitalización de la imagen y para darle los ajustes necesarios de contraste, colores o intensidad propias de los escáneres.

Reconocimiento del texto, donde en muchos programas se muestran una pantalla dividida, la cual nos permite escoger qué tipo de archivo necesitamos guardar independientemente del otro (un archivo de imagen o uno del texto obtenido).

La corrección de todo el texto reconocido y la aplicación de los formatos de estilo.

#### ***Reconocimiento óptico de caracteres***

Es el proceso que permite convertir las imágenes escaneadas de texto impreso o manuscritos en un formato que sea procesable por las computadoras (ASCII), tal y como ya se había mencionado se trabajó con documentación física y para poder extraer la

información de dichos documentos que en este caso fueron boletas de pago se aplicó este proceso. Con el reconocimiento óptico de caracteres se analizan los caracteres impresos y determinan su forma utilizando patrones de claros y oscuros. Comúnmente se utiliza para poder reconocer los caracteres que han sido escritos con un computador o máquina de escribir, aunque con sistemas muchos más avanzados se pueden identificar también caracteres que han sido escritos de manera manual [19].

En este procedimiento suele demandar el uso de un escáner óptico, con el que se usa para obtener la imagen del texto para continuar comparando los claros-oscuros de la imagen con patrones de caracteres mediante un software especializado; al identificarse los caracteres, estos se convierten en formato ASCII para poder ser procesado por la computadora [19].

Cabe resaltar que el mayor éxito alcanzado hasta el momento se ve directamente en el reconocimiento de escritura que se encuentra impresa, que resulta de fácil segmentación por sus caracteres. Posibilitando la aparición de sistemas de OCR comerciales que ofrecen muy buenas prestaciones, con tasas de identificación que rondan el 99%. Sin embargo, el tema del reconocimiento de escritura manuscrita prevalece aún como un problema con pocas soluciones [19].

### **Redes Neuronales**

Para el desarrollo del modelo fue necesario seleccionar una técnica que fuera capaz de adaptarse a la solución propuesta, por ese motivo se optó por hacer uso de las redes neuronales con el lenguaje de programación Python.

Con respecto a las redes neuronales, no se tiene una definición general de lo que es una red neuronal artificial, sino que existen diferentes definiciones dependiendo del texto o del artículo que se está tratando [20]. Dentro de las cuales se pueden encontrar las siguientes definiciones:

Una red neuronal es un tipo de modelo computacional y que está compuesto por unidades procesadoras adaptables con una gran interconexión entre las mismas.

Es un sistema de procesamiento de la información que posee características de funcionamiento muy comunes comparándolas con las redes neuronales biológicas.

Las redes neuronales son modelos simplificados de las redes de neuronas que conforman el cerebro humano. Y al igual que éste, simulan “aprender” partiendo de los datos que se le ingresan.

Las redes neuronales están conformadas por un conjunto de neuronas artificiales que se encuentran interconectadas y distribuidas en diversas capas. Y de esta forma las neuronas de una capa se encuentran conectadas con las neuronas de la siguiente capa; de manera

general, siempre existirá una primera capa de entrada, una o dos capas que están ocultas y una capa que será la de salida.

En las definiciones que se mencionan aparece el componente de la simulación del comportamiento biológico; sobre todo al tratar el perceptrón multicapa que no todas las redes van a emular una determinada estructura neuronal. Lo que tienen en común dichos

elementos con el cerebro humano sería la distribución de todas las operaciones a realizar en un conjunto de elementos básicos, donde por la analogía con los sistemas de biología, son llamados neuronas. Estos a su vez, están interconectados entre ellos mediante series de conexiones que, si seguimos la estructura de la analogía biológica, se llamarían pesos sinápticos. Estos pesos van a variar con respecto al tiempo mediante un proceso que se suele llamar como “aprendizaje”. Por lo tanto, se concluye que el aprendizaje de una red neural como el proceso que modifica las conexiones entre todas las neuronas y pesos sinápticos para poder realizar la tarea deseada [21].

### ***Características de las redes neuronales***

Las características más esenciales del funcionamiento del cerebro como un sistema son las siguientes [22]:

Adaptabilidad y autoorganización, donde se va a utilizar algoritmos de aprendizaje de autoorganización y también adaptativo, ofreciéndose posibilidades de un procesamiento adaptativo y robusto.

Facilidad de aprendizaje, por medio de diversos entrenamientos se van a justar los pesos de las conexiones que tiene la red, optimizando las tareas que se realizan en la misma red.

Procesado no lineal, aumentando la capacidad que tendrá la red para poder clasificar y aproximar su inmunidad ante cualquier ruido.

Facilidad de representación, brindando una forma muy eficiente de poder representar el conocimiento, sobre todo en áreas de clasificación o reconocimiento con problemas específicos.

Paralelismo teniendo varias neuronas que se encuentran interconectadas pueden trabajar todas al mismo tiempo, ofreciendo un alto rendimiento, puesto que, todas las neuronas de una sola capa pueden dar resultados de manera simultánea.

Tolerancia al error: Porque tienen una gran cantidad de unidades de neuronas y algún error que pueda producirse en alguna de ellas no lograría afectar de manera significativo el resultado final de la red.

Se concluye que las redes neuronales buscan conseguir todas estas características imitando el comportamiento del cerebro humano, pretendiendo desarrollar un algoritmo equivalente a los procesos de aprendizaje y reconocimiento humanos.

### ***Elementos de las redes neuronales***

Las redes neuronales tienen elementos en común que son los siguientes [23]:

**Capa de entrada:** Será el primer elemento de la red neuronal y el que tiene contacto con los datos originales, dentro de esta capa el número de neuronas puede variar según criterios del desarrollados.

**Pesos:** A cada neurona que se ubique dentro de la red neuronal se le asignará un peso, este define la influencia que tiene la salida de una neurona sobre toda la red, de esta forma se puede ponderar y variar la salida final y también realizar la retropropagación de errores.

**Salidas:** es el resultado de procesamiento realizado en cada neurona, ya sea de la capa de entrada o de cualquiera otra ubicación dentro de la red neuronal.

### ***Gestión documental***

La gestión documental es un concepto que parte desde la captura, almacenamiento y recuperación de documentos dentro de cualquier organización, actualmente las organizaciones ya realizan estas actividades, sea con documentos en formato físico o digital por ellos resulta de suma importancia que los encargados de la gestión documental estén orientados en el marco conceptual y normativo para de esta forma poder aplicar las técnicas, herramientas y metodologías más eficaces en la gestión documental moderna [24].

### ***Características de la gestión documental***

Dentro de la gestión documental existen algunas características claves, las más resaltantes son [24]:

**Revisión de entrada, salida y bloqueo:** esta característica permite la edición simultánea de un documento evitando que los cambios de una persona sobrescriban los de otra.

**Control de versiones:** Garantiza que se mantenga un historial sobre cómo estaba el documento en un determinado punto del tiempo.

**Registros de auditoría:** Permiten la reconstrucción de una determinada versión del documento para poder identificar a la persona que realizó los cambios y cuales fueron cada uno de dichos cambios.

### ***Sistemas de gestión documental***

El tipo de sistema trabajado fue un sistemas de gestión documental, estos son considerados herramientas que han sido diseñadas y construidas con la finalidad de digitalizar y administrar grandes cantidades de documentos y archivos [25].



Un sistema de gestión documental tiene los siguientes objetivos [26]:

Controlar y coordinar todas las actividades relacionadas con la creación, recepción, ubicación, acceso y preservación de los documentos.

Asegurar que el proceso de recuperación y custodia de los documentos se realice de forma eficaz.

Garantizar que la organización tenga un correcto funcionamiento y se alinee con la normatividad y legislación vigente en cuando a la gestión documental.

## Materiales y métodos

### Tipo de investigación

El tipo de investigación del presente proyecto a desarrollar es aplicada, ya que se espera probar el producto acreditable dentro de la organización, a fin de contribuir a la reducción de los tiempos de respuesta en la búsqueda de las boletas y de la migración de toda la data recolectada [27].

### Métodos de investigación

Los métodos de investigación empleados serán los siguientes:

TABLA I  
MÉTODOS DE INVESTIGACIÓN

Método	Descripción
Analfítico	Este método será utilizado para analizar la situación de la empresa en la actualidad, junto con las causas y las consecuencias de las deficiencias en las gestión documental existente.
Deductivo	Este método será útil para realizar la fase de diseño y arquitectura del sistema, así como los requerimientos funcionales y no funcionales necesarios para dar una solución óptima.

### Técnicas e instrumentos de recolección de datos

A continuación, en la siguiente tabla se muestra las técnicas e instrumentos que fueron útiles para la recolección de datos.

TABLA II  
TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Técnicas	Instrumentos	Elementos de la población	Propósito
Entrevista	Guía de entrevista	Personal del área de sistemas de información	Obtener la información necesaria sobre la aplicación que gestiona las boletas de pago de los años 2015- 2021.
Encuesta	Cuestionario	Personal del área de trámite documental	Conocer el grado de satisfacción del personal del área de trámite documental sobre la inclusión de un sistema que gestione las boletas de pago en su totalidad.

### Procedimientos

## Metodología de desarrollo

Con respecto a la metodología, se decido trabajar con una para la parte del desarrollo del aplicativo, en este caso SCRUM y otra para el desarrollo del modelo de reconocimiento con el uso de CRISP-DM.



Fig. 1. Ciclo Scrum [28]

A continuación se describirá un poco acerca de la metodología SCRUM con la cual se trabajará el aplicativo de escritorio. Además considerando el tamaño reducido del equipo de trabajo esta metodología nos brindará flexibilidad y comodidad durante la ejecución de las actividades [28] [29].

De entre los aspectos importantes dentro de Scrum se tiene a los roles:

**Dueño del producto:** Es el responsable de definir, mantener y priorizar las tareas dentro del producto backlog que se maneje en el proyecto, en otras palabras, es la representación del cliente y quien vela por que finalmente se obtenga lo que se pidió.

**Scrum master:** Es el encargado de manejar el trabajo de todo el equipo, es su responsabilidad que se cumplan las bases de Scrum y solucionar los inconvenientes que se puedan presentar.

**Equipo de desarrollo:** Son los encargados de ejecutar las tareas que se hayan definido e ir construyendo el producto.

Las fases que comprende esta metodología son las siguientes:

### **FASE #1: Inicio**

Durante esta fase se deberá analizar y estudiar las necesidades que se tienen con respecto al problema que se está buscando solucionar.

Dentro de esta fase se pueden identificar seis procesos:

*Creación de la visión del proyecto:*

Se debe definir claramente cuáles son las ambiciones iniciales y expectativas que se tienen para poder realizar una correcta planificación de las necesidades en cuanto a los recursos.

*Identificación de los roles:*

Para asegurar una buena organización del proyecto, además de definir las responsabilidades que permitan a cada integrante del proyecto reconocer sus funciones y en base a ello actuar según corresponda.

*Formar el equipo Scrum:*

En base a la visión que se tenga del proyecto se deberá escoger a las personas idóneas y que cumplan con los requisitos y características fundamentales que el proyecto implica.

*Recolectar las historias de usuario:*

Para comprender a fondo el problema y las necesidades que tiene nuestro usuario final.

*Crear el producto backlog y priorizarlo:*

A partir de las historias de usuario se debe definir y priorizar cada uno de los requerimientos a fin de poder distribuir el trabajo en cada uno de los sprints.

*Planificar el lanzamiento:*

Se deberá incluir una limitación para el alcance y fechas límites junto con las especificaciones de los entregables a realizar.

***FASE #2: Planificación y estimación***

A partir de la fase anterior se debe tener ya identificado un backlog general del proyecto a partir del análisis de las historias de usuarios ya que aquí se tomarán en consideración los siguientes procesos: crear, estimar y comprometer historias de usuario, identificar y estimar tareas y crear el sprint backlog

***FASE #3: Implementación***

Es importante reconocer que durante esta fase no debería realizarse ningún cambio innecesario ya que a este nivel ya se cuenta con una planificación de las actividades en base a la proyección del tiempo y recursos con los que se cuenta.

En esta fase se deben cumplir con los siguientes procesos: crear entregables, realizar el daily stand-up y refinar el producto backlog priorizado.

***FASE #4: Revisión y retrospectiva***

Cuando se haya cumplido con la maquetación, prototipado o implementación programada para la fase anterior se procederá a revisar el resultado buscando autocríticas y soluciones viables a los problemas que se hayan presentado, para ello se deben cumplir los siguientes procesos: demostrar y validar el sprint y retrospectiva del sprint.

#### ***FASE #5: Lanzamiento***

En esta última fase se debe considerar realizar una revisión general tanto de los sprint realizados como de la documentación que se haya generado a fin de constatar que todos lo realizado y logrado haya sido lo que se tenía previsto, además, se deberá considerar los siguientes procesos: enviar entregables y enviar retrospectiva del proyecto.

Con respecto a la metodología CRIPS-DM con la cual se trabajará el modelo de reconocimiento de caracteres, contempla un total de 6 fases como se observa en la Fig.3, las salidas que se esperan de cada fase, se muestran en la Fig.4.

A continuación se dará una pequeña descripción de lo que se va a contemplar en cada fase de la metodología:

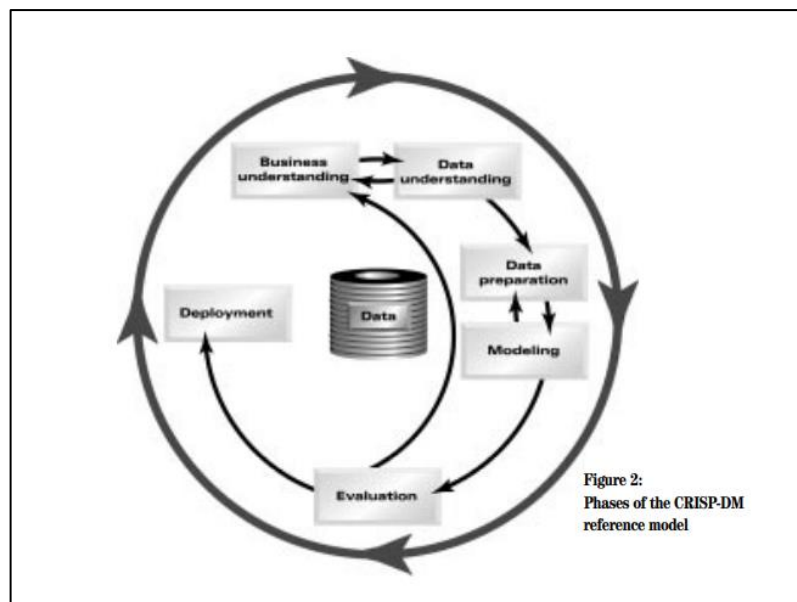


Fig. 2. Crisp-DM Fases [30]

#### ***FASE #1: Compresión del negocio***

Son las razones por las cuales el negocio necesita la solución a implementar.

#### ***FASE #2: Compresión de los datos***

Esta fase sirve para tener en claro el valor de los datos del negocio, para ello es necesario acceder a ellos para explorarlos y determinar la calidad que tienen.

### ***FASE #3: Preparación de los datos***

Se estima que esta fase suele llevar entre el 50-70% del tiempo y esfuerzo del proyecto, ya que se trabajan los datos antes explorados.

### ***FASE #4: Modelado***

Se ejecuta en múltiples iteraciones.

### ***FASE #5: Evaluación***

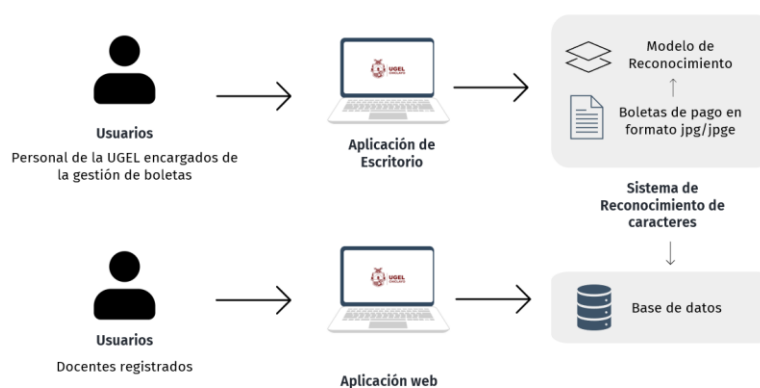
Se deben tomar en cuenta los criterios de rendimiento de minería de datos que ya definidos.

### ***FASE #6: Despliegue***

Consiste en implementar la solución desarrollada dentro de la organización.

Se adaptaron ambas metodologías para poderlas trabajar conjuntamente, lo cual se muestra en el apartado de resultados y discusión.

## **Arquitectura**



*Fig. 3. Arquitectura de la solución*

Usuarios internos: Representan al personal responsable de tramite documentario, los cuales interactuarán con el sistema mediante una interfaz que permitirá la búsqueda o el reconocimiento de los caracteres mecanografiados de las boletas de pago mediante un escáner. Será necesario contar con una computadora de escritorio o una laptop para poder ejecutar la aplicación.

Modelo de detección de caracteres: Representa la Red Neuronal de Back-Propagation cuya función será que a partir de una boleta de pago escaneada reconozca los caracteres necesarios para asignar dicha boleta a un docente.

Base de datos: Es el componente que se encarga de almacenar los datos que el usuario acaba de realizar, optándose por el gestor de base de datos Microsoft SQL Server, porque es el que tiene a su disposición la Ugel de Ferreñafe y con el cual tienen todos sus servicios enlazados.

Aplicación de escritorio: Es el componente que permitirá la interacción de los usuarios internos con el modelo de reconocimiento de caracteres y la posterior transferencia de la información hacia la base de datos para que los docentes puedan consultarla según sus necesidades.

### **Infraestructura tecnológica**

Considerando la arquitectura anteriormente descrita, la infraestructura tecnológica estará conformada por:

Usuario interno: Encargado de conectarse al sistema.

Escáner: Será necesario que el usuario tenga conectada un escáner para el caso de que necesite subir una boleta una boleta de pago en el sistema, y en el caso de que quiera buscar una boleta solo será necesario una conexión activa de internet.

Ordenadores: Dispositivo por el cual se conectará el usuario interno, en este caso PC/Laptop.

Router: Dispositivo por el cual se tiene acceso a internet.

Internet: Se necesitará tener acceso a internet para el acceso a la base de datos.

Servidor web: En este se almacenará la Aplicación web, la API web y la base datos, por ello será necesario 8GB de memoria RAM, 2TB de almacenamiento y 1TB de transferencia mensual para que funcione correctamente. El ambiente donde se ejecutará se tiene previsto que se desarrolle en el sistema operativo Linux ya que es un SO Open Source y reducirá significativamente los costos.

## Matriz de consistencia

TABLA III  
MATRIZ DE CONSISTENCIA

<u>FORMULACIÓN DEL PROBLEMA</u>		<u>MÉTODOLÓGÍA DE INVESTIGACIÓN</u>			
¿De qué manera se puede ayudar a los trabajadores del área de trámite documentario de la UGEL de Ferreñafe a realizar una gestión de boletas de pago más eficaz y eficiente?		<u>TIPO DE INVESTIGACIÓN</u> Investigación aplicada			
<u>OBJETIVO GENERAL</u>	<u>MÉTODO</u>	<u>DESCRIPCIÓN</u>			
Implementar un sistema de reconocimiento de texto mecanografiado mediante la utilización de redes neuronales para la gestión de boletas de pago en la UGEL de Ferreñafe.	Análítico	Este método será utilizado para analizar la situación de la empresa en la actualidad, junto con las causas y las consecuencias de las deficiencias en la gestión documental existente.			
	Deductivo	Este método será útil para realizar la fase de diseño y arquitectura del sistema, así como los requerimientos funcionales y no funcionales necesarios para dar una solución óptima.			
	<u>TÉCNICAS</u>	<u>INSTRUMENTOS</u>	<u>ELEMENTOS DE LA POBLACIÓN</u>	<u>PROPÓSITO</u>	
	Observación	Guía de observación	Área de trámite documentario	Conocer el proceso de solicitud de boletas de pago	
	Entrevista	Guía de entrevista	Personal del área de sistemas de información	Obtener la información necesaria sobre la aplicación que gestiona las boletas de pago de los años 2015-2021.	
	Encuesta	Cuestionario	Personal del área de trámite documentario	Conocer el grado de satisfacción del personal del área de trámite documentario sobre la inclusión de un sistema que gestione las boletas de pago en su totalidad.	
<u>OBJETIVOS ESPECÍFICOS</u>	<u>DESCRIPCIÓN DEL LOGRO DE LOS OBJETIVOS ESPECÍFICOS</u>			<u>INDICADORES</u>	



---

Comparar técnicas de reconocimiento de caracteres para seleccionar la que mejor reconozca el texto mecanografiado.

Realizar una comparación de las técnicas óptimas existentes para el reconocimiento de caracteres mecanografiados para seleccionar el más preciso.

Grado de precisión del algoritmo.

Alcanzar un bajo porcentaje en la tasa de error de caracteres.

Alcanzar un porcentaje bajo en la tasa de error de caracteres (CER).

$$CER = \frac{S + D + I}{N}$$

Obtener una valoración mayor a 68 de la usabilidad de la aplicación por parte de los usuarios finales.

Mediante la System Usability Scale se medirá la usabilidad de la aplicación de reconocimiento de caracteres en textos mecanografiados, la cual deberá alcanzar un valor mayor a 68 que indicaría que es aceptable.

Puntaje obtenido en la System Usability Scale

---

## **Consideraciones éticas**

Se mantuvo se discreción los datos referentes a las boletas de pago brindadas por los docentes, es decir, que no se divulgo ni se dio mal uso a la información, únicamente se utilizó para fines académicos, tal y como se había acordado previamente con la organización. Los aspectos que se han considerado son:

Aplicación de técnicas de recolección de datos: Se realizó una entrevista presencial a un integrante del personal de la UGEL relacionado con el área de documentación.

Seguridad de la Información: La empresa brindo para este proyecto las boletas físicas de diferentes años a fin de poder hacer uso de ellas y pasar la información al sistema desarrollado.

## **Resultados y discusión**

### **Sprint N°1**

En este sprint se realizaron varias actividades clave, incluyendo la arquitectura de la solución y la creación de prototipos de las interfaces de usuario. Además, se definió el conjunto de tecnologías a utilizar para diferentes aspectos del proyecto las cuales fueron:

#### ***Modelo de Predicción***

Se optó por utilizar Python para el desarrollo del modelo de reconocimiento de caracteres debido a su idoneidad para tareas de IA.

Se eligió Visual Studio Code como el editor para trabajar cómodamente con Python.

#### ***Aplicación de Escritorio***

C# se seleccionó como el lenguaje de programación para el desarrollo de la aplicación de escritorio.

Visual Studio .Net fue elegido como el entorno de desarrollo, dado que ofrece experiencia y facilidad para trabajar con C#.

#### ***Aplicación de Web***

React, una biblioteca de código abierto de JavaScript, se utilizó para facilitar el desarrollo de la aplicación web.

Tailwind CSS, un framework de CSS, se empleó para crear una apariencia moderna y atractiva para la aplicación web.

JavaScript se utilizó para trabajar con React, ya que es un lenguaje orientado a objetos y fácil de entender.

Visual Studio Code se eligió como el editor para desarrollar la aplicación web, con la adición de algunas extensiones necesarias.

### ***Base de Datos***

Se optó por utilizar *SQL Server* como el sistema de gestión de base de datos. Esta elección se hizo para asegurar una buena integración con las herramientas seleccionadas previamente.

Este sprint se centró en la planificación y definición de tecnologías clave para el proyecto, lo que sienta las bases necesarias para el desarrollo posterior de la solución.

### **Sprint N°2**

En este sprint, se realizaron actividades fundamentales para el proyecto, que incluyeron la identificación de problemas en la UGEL relacionados con la gestión de boletas de docentes, la evaluación de sistemas existentes y la elaboración de objetivos para la minería de datos.

Se desarrolló un plan de proyecto que integra metodologías CRISP-DM y SCRUM, y se obtuvieron datos iniciales escaneando boletas de pago físicas desde 1988 hasta principios de los 2000. Se describieron estos datos, se exploraron las variaciones en la estructura de las boletas a lo largo del tiempo y se destacó la importancia de preservar información en boletas degradadas. Este sprint sentó las bases para la implementación de la solución en fases posteriores del proyecto.

### **Sprint N°3**

Aquí se realizaron varias etapas esenciales en la preparación de los datos para el proyecto. Se seleccionaron los datos clave necesarios para encontrar las boletas en el sistema, que incluyen el apellido y nombre del docente, el código del docente, el mes de emisión y el año de emisión de la boleta. La limpieza de datos fue un desafío, ya que la información no estaba registrada sistemáticamente y algunas boletas ilegibles no se iban a poder reconocer. Además, no fue necesario construir nuevos datos ni integrar información adicional. Los valores de los datos no se transformaron, ya que existen solo en formato físico. Se mencionó la estructura del modelo entidad-relación que respalda el aplicativo de escritorio y móvil, lo que indicó la importancia de estos pasos en la preparación de los datos para el proyecto.

### **Sprint N°4**

Este sprint contempló la gestión de acceso al sistema y además la gestión de los roles de los usuarios.

Para poder habilitar el acceso a la aplicación de escritorio, se implementó la clase *Session* en la capa de negocios (*Business Layer*) con las siguientes funcionalidades:

***CurrentUser:*** Es una propiedad estática que representa al usuario actualmente conectado.

**login:** Permite el inicio de sesión en el sistema. Se busca un usuario en la base de datos que coincida con las credenciales proporcionadas (nombre de usuario y contraseña). Si se encuentra un usuario válido, activo y personal de la institución, se establece como *CurrentUser* y se devuelve *true*; de lo contrario, se devuelve *false*.

***HasPermission:*** Verifica si el usuario actual tiene un permiso específico. Primero, se comprueba si *CurrentUser* es null (es decir, si no hay un usuario conectado), y en ese caso se devuelve *false*. Luego, se verifica si el usuario actual es un superusuario, en cuyo caso se devuelve *true*. Si no es un *superusuario*, se examinan los permisos del rol del usuario actual y se compara el identificador único del permiso con el proporcionado. Si se encuentra una coincidencia, se devuelve *true*; de lo contrario, se devuelve *false*.

Para interactuar con la base de datos, se utiliza la biblioteca *Entity Framework Core*, y se garantiza la persistencia de cambios utilizando el método *SaveChanges()* del contexto. Además, se emplea la clase *Encrypt* del espacio de nombres *Common* para cifrar la contraseña antes de la comparación en el inicio de sesión.

Para la gestión de roles que se llevó a cabo en la aplicación de escritorio a través del código del *RolesController* en la capa de negocios (*Business Layer*). Este controlador desempeña varias funciones clave:

***createRole:*** Crea un nuevo rol en la base de datos, asignándole un nombre y una lista de permisos. Recopila los permisos correspondientes de la base de datos y los agrega al rol, luego guarda el rol en el contexto y actualiza la base de datos.

***getRoles:*** Recupera una lista de todos los roles de la base de datos utilizando una consulta LINQ, y proyecta los datos en instancias de *Role* para devolver la lista resultante.

***removeRole:*** Elimina un rol basado en su ID, buscándolo por este identificador y eliminándolo del contexto antes de guardar los cambios en la base de datos.

***updateRole:*** Actualiza un rol existente, modificando su nombre y lista de permisos. Esta función busca el rol por su ID, actualiza sus datos y persiste los cambios en la base de datos.

El código interactúa con la base de datos a través de la clase *DatabaseContext* y garantiza que los cambios se guarden adecuadamente utilizando el método *SaveChanges()*.

## Sprint N°5

En el siguiente sprint, se abordó la gestión de usuarios en el sistema mediante la implementación de la clase *UsersController* en la capa de negocios (*Business Layer*), con las siguientes funcionalidades:

***createUser:*** Crea un nuevo usuario en la base de datos, tomando múltiples parámetros, como nombre completo, nombre de usuario, contraseña, ID de rol, estado de activación, estado de personal, estado de *superusuario* y una lista de docentes asociados. Se crea una instancia de *User* y se asignan los valores proporcionados. Luego, se agregan los docentes correspondientes a la lista de docentes del usuario y se guardan los cambios en la base de datos.

***updateUser:*** Actualiza un usuario existente en la base de datos, tomando una serie de parámetros similares al crear usuario. Se busca al usuario por su ID, se actualizan los valores y se actualizan también los docentes asociados.

***removeUser:*** Elimina un usuario basado en su ID, eliminándolo del contexto y guardando los cambios en la base de datos.

***getUsers:*** Recupera una lista de todos los usuarios de la base de datos mediante una consulta LINQ.

***isUsedUsername:*** Verifica si un nombre de usuario ya está en uso por otro usuario en la base de datos.

***isOwnUsername:*** Verifica si un nombre de usuario ya está en uso por otro usuario, excluyendo al usuario con el ID proporcionado.

El código interactúa con la base de datos a través de la clase *DatabaseContext* y asegura que los cambios se guarden utilizando el método *SaveChanges()*. Además, utiliza la clase *Encrypt* del espacio de nombres *Common* para cifrar la contraseña antes de almacenarla en la base de datos.

## Sprint N°6

En este sprint, se implementó la capa de negocios (*Business Layer*) que incluye un controlador de Docentes (*TeachersController*) con las siguientes funciones clave:

***createTeacher:*** Crea un nuevo docente en la base de datos especificando el nombre completo y el código del docente, utilizando el contexto de *DatabaseContext*.

***getTeachers:*** Recupera una lista de todos los docentes en la base de datos, incluyendo sus datos de alternativas, utilizando el contexto de *DatabaseContext*.

***filterTeachers:*** Filtra la lista de docentes en base a un filtro proporcionado, buscando docentes cuyo nombre completo o código contengan la cadena de filtro. También incluye los datos de alternativas y utiliza el contexto de *DatabaseContext*.

***removeTeacher:*** Elimina un docente de la base de datos según su ID, buscándolo mediante el contexto de *DatabaseContext*.

***updateTeacher:*** Actualiza el nombre completo y el código de un docente existente en la base de datos, buscando el docente por su ID a través del contexto de *DatabaseContext* y actualizando los datos correspondientes.

Además, el código utiliza la biblioteca *Entity Framework Core* para interactuar con la base de datos y se asegura de que los cambios se guarden mediante el método *SaveChanges()*.

## **Sprint N°7**

Se evaluaron tres modelos utilizados para el desarrollo de *Optical Character Recognition* (OCR) con el objetivo de seleccionar el más adecuado para la solución. Estos tres modelos incluyen *ANN*, *CNN* y *SVM*. La elección se basó en una comparación que consideró las tendencias actuales y el conocimiento previo en el desarrollo de estos modelos. Las *ANN* (Redes Neuronales Artificiales) se seleccionaron debido a su mayor impacto y uso actual, así como su capacidad de adaptación sin requerir muchos recursos.

El diseño de pruebas se centró en la precisión del modelo, utilizando la función de pérdida *categorical\_crossentropy* para evaluar si el modelo identifica correctamente los caracteres de palabras y fechas. La función de pérdida se enfoca en penalizar las predicciones incorrectas, lo que es apropiado para modelos de clasificación multiclase que involucran etiquetas de salida codificadas.

Luego, se describió la construcción del modelo OCR en Python, utilizando bibliotecas como *Transformers*, *TensorFlow*, *Pillow (PIL)*, *SpaCy*, y otras. Se definió una clase *Voucher* para almacenar caracteres y sus propiedades, y se detalló la lógica de agrupación de caracteres en líneas y palabras.

El proceso de identificación de caracteres se llevó a cabo mediante la función *identify\_character*, que utiliza un modelo OCR para reconocer caracteres en una imagen. También se discutió la corrección ortográfica utilizando un modelo y se describió la función *check\_spell*.

Además, se presentó la función *detecte\_char*, que se encarga de detectar y reconocer caracteres en una imagen, realizar corrección ortográfica si es necesario y reconocer entidades

como nombres, meses, años y códigos. El resultado del procesamiento se almacena en una lista y se imprime en formato *JSON*.

Este sprint se centró en la selección y construcción de un modelo OCR, así como en la corrección ortográfica y la identificación de entidades en imágenes de boletas.

### **Sprint N°8**

Se desarrolló el componente de inicio de sesión de la aplicación web dirigida a docentes utilizando React. El componente Login se creó como una función que hace uso de ganchos de React, como *useState*, *useEffect*, *useRef* y *useContext*.. Este componente se encarga de gestionar el estado y los efectos secundarios relacionados con la autenticación del usuario.

El componente Login muestra un formulario de inicio de sesión que permite al usuario ingresar su nombre de usuario y contraseña. Cuando se hace clic en el botón de inicio de sesión, el componente envía una solicitud POST a un punto final de inicio de sesión. Si la autenticación es exitosa, el componente establece los datos del usuario en el contexto de la aplicación y redirige al usuario a la página de inicio. Además, el componente maneja mensajes de error y muestra el estado de carga durante el proceso de inicio de sesión.

### **Sprint N°9**

Se llevó a cabo una revisión de los resultados del modelo OCR, centrándose en la evaluación de la tasa de error de caracteres (CER). Se seleccionaron segmentos de imágenes con caracteres legibles para realizar una evaluación objetiva del modelo. El CER se calculó utilizando la fórmula  $CER = (S + D + I) / N$ , donde S es la cantidad de caracteres sustituidos, D es la cantidad de caracteres eliminados, I es la cantidad de caracteres insertados y N es la cantidad de caracteres originales en la frase.

Los resultados de la evaluación arrojaron un CER promedio de 0.018, lo que equivale a una tasa de error del 1.8%. Esto demuestra la efectividad del modelo OCR en el reconocimiento de caracteres.

Además de la evaluación del modelo, se llevaron a cabo pruebas de caja blanca y caja negra en otras funcionalidades del proyecto, como el inicio de sesión y las operaciones CRUD, y se confirmó que todas funcionan correctamente.

### **Sprint N°10**

Se planificaron aspectos relacionados con la implementación del sistema desarrollado en la entidad UGEL. Se destaca la necesidad de obtener la aprobación y el permiso de la UGEL para

llevar a cabo la implementación. Se acordará una fecha y hora que no interrumpa las labores cotidianas del personal, y se brindará capacitación al personal para asegurar la comprensión y resolver sus dudas.

Además, se contemplo la posibilidad de alquilar un dominio y hosting, lo que permitiría alojar el sistema en la nube en lugar de operarlo localmente. Esto proporcionaría la capacidad de trabajar con una mayor cantidad de datos de manera simultánea en el futuro.

En cuanto al seguimiento y mantenimiento, se propuso establecer una programación regular que incluya la realización de extracciones trimestrales o semestrales de los datos almacenados para crear copias de seguridad, mejorar o actualizar el modelo y las librerías utilizadas, y evaluar y revisar periódicamente el rendimiento del sistema.

En este proyecto, se siguió una metodología que combina elementos de la metodología SCRUM y la metodología CRISP-DM, aprovechando las fortalezas de ambas para optimizar el desarrollo el modelo de reconocimiento. Durante los Sprints se llevó a cabo, utilizamos el lenguaje de programación Python para desarrollar las redes neuronales, en línea con investigaciones previas [6] [7].

La investigación [7], que forma parte de las investigaciones revisadas, se centró en casos de fallos en la clasificación, destacando la complejidad de estos incluso para hablantes nativos de árabe. En este proyecto, se enfrentaron desafíos similares, especialmente en el procesamiento de imágenes de boletas gravemente deterioradas debido al paso del tiempo y su mala conservación. En comparación con las boletas en mejor estado de legibilidad, se obtuvo un índice de error del 1.8% en la identificación de caracteres.

A diferencia del enfoque de [9], que se centró en el desarrollo de su modelo mediante Deep Learning, este proyecto adoptó un enfoque basado en redes neuronales después de una cuidadosa evaluación de diversas técnicas, que demostró ser la más adecuada para la solución planteada. A pesar de las diferencias, estos antecedentes sirvieron como base y referencia para mejorar el reconocimiento de caracteres, especialmente teniendo en cuenta la variabilidad en la tipografía de las boletas y su estado de conservación.

En [10], se desarrolló una aplicación móvil con tecnología OCR para capturar y traducir textos a través de un sistema de búsqueda dinámico. El proyecto abordó una tarea similar al identificar los caracteres clave de las variables de interés para agilizar la búsqueda de boletas de pago de docentes, lo que implica que solo algunos de los datos de las boletas se almacenaron en la base de datos.

El estudio [11] enfatizó la importancia del OCR para acelerar la transferencia de datos a una base de datos y mejorar la gestión de documentos. Estos hallazgos respaldan la propuesta



planteada en esta investigación en relación con el uso del OCR. Además, observamos un patrón similar en [12] [13] [14] [15], donde se demostró cómo la gestión documental se mejoró significativamente tras la implementación del modelo OCR en sus sistemas. Es relevante destacar que en nuestra investigación logramos un nivel de aceptación del 80% por parte de los usuarios, lo cual no se ha observado en otras investigaciones.

### **Conclusiones**

Luego de un análisis comparativo, las redes neuronales se destacaron como la técnica de modelamiento preferida tanto en la teoría como en la práctica, demostrando ser la elección más acertada y respaldada por evidencia sólida debido a que se adaptan a datos no estructurados, como texto manuscrito en imágenes.

Se logró de un error del 1.8% en comparación con los modelos existentes constituye una prueba concluyente de que nuestro modelo es altamente preciso y cumple con creces las expectativas, lo que refuerza su validez y eficacia en la tarea de reconocimiento óptico de caracteres.

Se obtuvo un puntaje sobresaliente de 80 de 100 en la escala de Usabilidad del Sistema (System Usability Scale) aplicada al personal de la UGEL, lo cual superó las expectativas originales. Esto confirma de manera inequívoca la alta usabilidad y satisfacción del personal con la aplicación desarrollada, validando así su implementación exitosa.

### **Recomendaciones**

En este proyecto, el enfoque primordial se centró en el desarrollo del modelo OCR para la identificación de las variables principales de la boleta. Con miras a futuras investigaciones, se recomienda encarecidamente considerar la inclusión y almacenamiento adecuado de la totalidad de los datos presentados en las boletas. Esto permitirá un aprovechamiento integral de la información y su análisis para una gama más amplia de propósitos.

Adicionalmente, se sugiere ampliar la investigación para abordar la problemática de reconocer caracteres con poca nitidez o desgastados, lo que representaría un avance significativo en la capacidad del modelo OCR para lidiar con situaciones del mundo real.

Asimismo, para investigaciones posteriores, se recomienda la digitalización completa del documento, generándolo en un formato actual como un archivo PDF en lugar de limitarse a formatos de imagen. Esto facilitaría la extracción y manipulación de datos de manera más eficiente, lo que a su vez mejoraría la accesibilidad y la utilidad de la información contenida en las boletas.

## Referencias

- [1] L. Solares de Oliveira, R. Sabourin, F. Bortolozzi y C. Suen, «A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition.,» *International Journal of Pattern Recognition and Artificial Intelligence.*, p. 903.929, 2003.
- [2] TIC Portal, «La función de búsqueda en un gestor documental: ¿es indispensable para encontrar sus documentos rápido?,» European Knowledge Center for Information Technology, [En línea]. Available: <https://n9.cl/hdk6v>. [Último acceso: 10 Septiembre 2021].
- [3] R. T. Ramírez, Interviewee, *Entrevista para caracterización de la situación del área de trámite documentario de la UGEL de Ferreñafe*. [Entrevista]. 10 Octubre 2021.
- [4] C. J. Sánchez Fernández y V. Sandonís Consuegra, «Reconocimiento Óptico de Caracteres (OCR),» 2008. [En línea]. Available: <http://www.it.uc3m.es/jvillena/irc/practicass/08-09/09.pdf>.
- [5] M. M. López Ulloa, «“DESARROLLO DE UN SISTEMA DE DIGITALIZACIÓN Y RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR) PARA LOS DOCUMENTOS DE LA NOTARIA PRIMERA DE QUILANGA-LOJA”»,» GUAYAQUIL, 2019.
- [6] S. A. El Battioui, «Reconocimiento de caracteres en imágenes mediante el uso de CNNs,» Gijón, 2019.
- [7] A. TALAT SAHLOL, M. ABD ELAZIZ, M. AL-QANESS y S. KIM, «Handwritten Arabic Optical Character Recognition Approach Based on Hybrid Whale Optimization Algorithm With Neighborhood Rough Set,» *IEEE Access*, 2020.
- [8] T. NASIR , M. KAMRAN MALIK y K. SHAHZAD, «MMU-OCR-21: Towards End-to-End Urdu Text Recognition Using Deep Learning,» *IEEE Access*, 2021.
- [9] L. CAO, H. LI, R. XIE y J. ZHU, «A Text Detection Algorithm for Image of Student Exercises Based on CTPN and Enhanced YOLOv3,» *IEEE Access*, 2020.

- [10] P. Luna Grande, «“Implementación de una Aplicación Móvil con reconocimiento óptico de caracteres para la difusión turística de la provincia de San Martín, 2018”,» Lima, 2019.
- [11] C. M. Espejo Pérez, *Sistema de Reconocimiento de caracteres numéricos para actualización de base de datos*, Lima: Universidad Peruana de Ciencias Aplicadas, 2021.
- [12] J. J. Núñez Quiroz, «Adquisición e Implementación de un Sistema de Digitalización de Documentos y Gestión Documental’,» Lima, 2019.
- [13] G. Aquino Espinoza, «Desarrollo e Implementación del Sistema de Gestión de Documentos ScanViewer Para la Empresa Global Factoring S.A.,» Lima, 2019.
- [14] L. A. Santisteban Sanchez, «Digitalización de la documentación en el sistema de gestión,» Chiclayo, 2023.
- [15] M. R. G. Loconi Yahuana y A. R. Leiva Yzquierdo, «Propuesta Metodológica para la Gestión documental de los procedimientos de la Gerencia de Desarrollo Vial y Transporte de la Municipalidad Provincial de Chiclayo,» Chiclayo, 2020.
- [16] Amity, «Sistemas de reconocimiento óptico de caracteres (OCR), el futuro de la captura de datos,» 16 Diciembre 2019. [En línea]. Available: <https://n9.cl/ochtx>. [Último acceso: 25 Octubre 2021].
- [17] Investopedia team, «Magnetic Ink Character Recognition (MICR) Line,» 15 Abril 2021. [En línea]. Available: <https://n9.cl/6gu2x>. [Último acceso: 10 Noviembre 2021].
- [18] F. Debandi y M. Redi, «El reconocimiento de textos,» [En línea]. Available: <https://n9.cl/0nmmj9>. [Último acceso: 12 Noviembre 2021].
- [19] Signaturit, «OCR: ¿Qué es el reconocimiento óptico de caracteres?,» 17 Noviembre 2021. [En línea]. Available: <https://n9.cl/csrnk>. [Último acceso: 18 Noviembre 2021].
- [20] IBM, «El modelo de redes neuronales,» 2021. [En línea]. Available: <https://n9.cl/7q7tm>. [Último acceso: 12 Noviembre 2021].
- [21] Xeridia, «Redes neuronales artificiales: Qué son y cómo se entrenan,» 6 Septiembre 2019. [En línea]. Available: <https://n9.cl/a0ir8>. [Último acceso: 10 Noviembre 2021].

- [22] Three Points, «Redes neuronales: definición y características,» 21 Enero 2021. [En línea]. Available: <https://n9.cl/u1v8q>. [Último acceso: 5 Noviembre 2021].
- [23] Grupo de Tecnologías Avanzadas en Computación, «Elementos básicos de una red neuronal artificial,» 31 Agosto 2007. [En línea]. Available: <https://n9.cl/ecat0>. [Último acceso: 20 Octubre 2021].
- [24] Kyocera, «La gestión documental. Definición, conceptos clave e importancia,» 2021. [En línea]. Available: <https://n9.cl/85o7>. [Último acceso: 18 Noviembre 2021].
- [25] Lemon Tech, «¿Qué es gestión documental? Importancia, objetivos y ejemplos,» 23 Diciembre 2020. [En línea]. Available: <https://n9.cl/p069r>. [Último acceso: 16 Noviembre 2021].
- [26] Universidad Oberta de Catalunya, «Qué es un sistema de gestión documental (SGD),» 2021. [En línea]. Available: <https://n9.cl/hmt44>. [Último acceso: 16 Noviembre 2021].
- [27] OCDE, «Manual de Frascati 2015,» Fundación Española para la Ciencia y la Tecnología, 2018.
- [28] J. Sutherland y K. Schwaber, «La Guía de Scrum,» 2020.
- [29] Estrategia digital, «Metodología Scrum en proyectos digitales,» 5 Abril 2017. [En línea]. Available: <https://n9.cl/9fme>. [Último acceso: 16 Octubre 2021].
- [30] P. Chapman, J. Clinton, R. Kerbe, T. Khabaza , T. Reinartz, C. Shearer y R. Wirth , «CRISP-DM 1.0,» 2000.

**Anexos****ANEXO N° 01. CONSTANCIA DE APROBACIÓN DEL PRODUCTO  
ACREDITABLE DE LA ENTIDAD DONDE SE EJECUTÓ LA TESIS**

Ferreñafe, 30 de octubre del 2023

**CONSTANCIA DE APROBACIÓN DEL PRODUCTO ACREDITABLE**

En referencia a la solicitud realizada por el estudiante JONATHAN ALONSO BONILLA VILCHEZ de la UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO identificado con DNI N° 72795259 de la escuela de INGENIERA DE SISTEMAS Y COMPUTACIÓN en la que solicita la evaluación del cumplimiento de su producto acreditable presentando en la tesis denominada "SISTEMA DE RECONOCIMIENTO DE TEXTO MECANOGRAFIADO MEDIANTE REDES NEURONALES PARA LA GESTIÓN DE BOLETAS DE PAGO EN LA UGEL FERREÑAFE" se comunica que, se aprueba el cumplimiento y funcionamiento en su totalidad del producto acreditable presentado, demostrando su efectividad y eficiencia en el área de gestión documentaria. Reduciendo la carga de trabajo constante de los trabajadores del área, así como también demostrando que será de gran apoyo para los docentes al momento de solicitar sus boletas de pagos.

Se exhorta al estudiante a continuar desarrollando herramientas tecnológicas que permitan la optimización de procesos para una correcta gestión dentro de las organizaciones y/o entidades.

Atentamente,

Firma del Representante de la Entidad

Cargo: DIRECTORA

Nombre y Apellidos: GLORIA ELIZABETH JIMENEZ PEREZ

DNI: 16704478

## ANEXO N° 02. VALIDACIÓN DEL SISTEMA

Participante	Fecha
<u>Gloria Jimenez Perez</u>	<u>30/10/23</u>
	Extremadamente en desacuerdo      Extremadamente en de acuerdo
	1    2    3    4    5
El sistema de reconocimiento óptico de caracteres (OCR) desarrollado fue fácil de aprender	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
En general, el sistema era fácil de usar	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Me sentí confiado al utilizar el sistema para identificar y almacenar datos de boletas de pago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Necesité aprender muchas cosas antes de poder usar el sistema	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Las funciones del sistema estaban bien integradas	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Creo que la mayoría de las personas aprendería a usar el sistema de manera rápida	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Me sentí cómodo utilizando el sistema	<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
El sistema respondió de manera rápida a mis acciones	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
En general, estaba satisfecho con la facilidad de uso del sistema	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Recomendaría este sistema a otros para el procesamiento de datos de boletas de pago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>



Firma del Director de la Entidad

Cargo: DIRECTORA

Nombre y Apellidos: GLORIA ELIZABETH JIMENEZ PEREZ

DNI: 16704478

### ANEXO N° 03. CÓDIGO FUENTE DEL MODELO DE PREDICCIÓN

Con respecto al desarrollo del modelo de reconocimiento de imágenes, se hizo uso del lenguaje de programación Python, a continuación se explica el archivo Main.py en el cual se encuentra la parte principal del modelo OCR.

```
1     from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
2     from tensorflow import keras
3     from PIL import Image
4     import numpy as np
5     import spacy
6     import json
7     import cv2
8     import sys
9     import os
```

*Fig.1. Librerías*

Primeramente lo que se hizo fue importar las librerías necesarias en este caso:

**Transformers:** Se desarrolló como una respuesta a los desafíos del aprendizaje supervisado en el campo del Procesamiento del Lenguaje Natural. Introdujo una solución innovadora para la traducción de idiomas al permitir el entrenamiento en paralelo, lo que resultó en una mejora significativa en la velocidad y la reducción de costos. Además, se aprovechó el mecanismo de atención como un elemento potenciador en este enfoque [40].

**Tensorflow:** Te permite construir y entrenar redes neuronales para detectar patrones y razonamientos usados por los humanos [41].

**Pil:** Permite la edición de imágenes directamente desde Python. Soporta una variedad de formatos, incluidos los más utilizados como GIF, JPEG y PNG [42].

**Spacy:** Diseñada para facilitar tareas de procesamiento avanzado del lenguaje natural [43].

**Json:** JavaScript Object Notation es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript [44].

**Cv2:** Es una potente librería para trabajar con la carga y visualización de imágenes, la manipulación de imágenes y el filtrado de imágenes [45].

**Sys:** Este módulo provee acceso a algunas variables usadas o mantenidas por el intérprete y a funciones que interactúan fuertemente con el intérprete [46].

Os: Este módulo provee una manera versátil de usar funcionalidades dependientes del sistema operativo [46].

```

10
11 MONTHS = ['ENERO', 'FEBRERO', 'MARZO', 'ABRIL', 'MAYO', 'JUNIO', 'JULIO', 'AGOSTO', 'SEPTIEMBRE', 'OCTUBRE', 'NOVIEMBRE', 'DICIEMBRE']
12

```

*Fig. 2. Arreglo de meses*

Se creo un arreglo para cada uno de los meses que pueden estar registrados en las boletas.

```

13 ✓ class Character:
14 ✓     def __init__(self, x, y, w, h, char):
15         self.x = x
16         self.y = y
17         self.w = w
18         self.h = h
19         self.char = char
20         self.text = ""
21

```

*Fig.3 Class Character*

Se creo la clase Class Character y se definieron algunas variables.

```

22 ✓ class Voucher:
23     def __init__(self):
24         self.characters = []
25         self.text = ""
26
27     def append(self, char):
28         if(char.char != ""):
29             self.characters.append(char)
30

```

*Fig. 4: Class Voucher*



Se creo la clase Class Voucher y se definieron algunas variables, además de ello en la Evidencia XL se define a la función llamada append y esta función acepta un parámetro llamado char.

El código comprueba si el valor de la propiedad char del objeto char es diferente de una cadena vacía. Si el valor no es una cadena vacía, se agrega el objeto char a una lista llamada characters.

```

30
31 ✓ def sort_characters(self):
32     self.characters = sorted(self.characters, key=lambda c: c.y)
33
34     lines = []
35     if (len(self.characters)):
36         current_line = [self.characters[0]]
37         tolerance = 7
38         base_line = self.characters[0].y + self.characters[0].h
39         for char in self.characters[1:]:
40             if char.y > base_line:
41                 lines.append(current_line)
42                 current_line = [char]
43                 base_line = char.y + char.h
44             else:
45                 current_line.append(char)
46         lines.append(current_line)
47
48         lines = [sorted(line, key=lambda c: c.x) for line in lines]
49         tolerance = 7
50         for i, line in enumerate(lines):
51             words = []
52             current_word = [line[0]]
53             for c1, c2 in zip(line, line[1:]):
54                 if c2.x - (c1.x + c1.w) > tolerance:
55                     words.append(current_word)
56                     current_word = [c2]
57                 else:
58                     current_word.append(c2)
59             words.append(current_word)
60             lines[i] = words
61
62     self.characters = lines
63     self.generate_text()

```

Fig. 5. Class Voucher

En la Figura 5 se muestra la función llamada `sort_characters`; la función comienza ordenando la lista de caracteres almacenada en la propiedad `characters` en función del atributo "y" de cada objeto de carácter. Esto se logra utilizando la función `sorted` y una función lambda como clave de ordenamiento.

Luego de ello, crea una lista vacía llamada `lines` para almacenar las líneas de caracteres. Si la lista de caracteres no está vacía, se realiza un proceso de agrupación de caracteres en líneas. Se utiliza una variable llamada `current_line` para rastrear la línea actualmente en construcción. Se establece una variable llamada `tolerance` determinar si un carácter pertenece a la misma línea o a una línea nueva.

Se itera sobre los caracteres restantes de la lista `characters` a partir del segundo carácter. Si la coordenada "y" del carácter actual es mayor que la línea base (calculada sumando la coordenada "y" del primer carácter y su altura), se agrega la línea actual a la lista `lines` y se crea una nueva línea con el carácter actual como único elemento. De lo contrario, el carácter actual se agrega a la línea actual.

Después del bucle, se agrega la última línea actual a la lista `lines`.

A continuación, se ordeña cada línea en la lista `lines` en función del atributo "x" de cada carácter, utilizando la función `sorted` y una función lambda como clave de ordenamiento.

Luego, se itera sobre cada línea en `lines` y se realiza un proceso de agrupación de caracteres en palabras. Se utiliza una lista vacía llamada `words` para almacenar las palabras de cada línea. Se itera sobre cada par de caracteres adyacentes en una línea y se verifica si la diferencia en la coordenada "x" entre ellos es mayor que la `tolerance`. Si es así, se agrega la palabra actual a la lista `words` y se crea una nueva palabra con el carácter actual como único elemento. De lo contrario, el carácter actual se agrega a la palabra actual.

Después del bucle, se agrega la última palabra actual a la lista `words` y se actualiza la línea actual con la lista de palabras.

Finalmente, se asigna la lista `lines` a la propiedad `characters` de la instancia actual y se llama a un método llamado `generate_text` para generar el texto correspondiente a partir de las líneas de caracteres.

Para resumir, la función `sort_characters` ordena y agrupa los caracteres en líneas y palabras, basándose en sus coordenadas "x" e "y", respectivamente. Luego, actualiza la lista de caracteres con las líneas y palabras agrupadas y genera el texto correspondiente.

```

64
65 ✓ def generate_text(self):
66     for line in self.characters:
67         for i, word in enumerate(line):
68             if i != 0: self.text += " "
69             self.text += "".join([c.char for c in word])
70         self.text += "\n"
71
72
73 ✓ def identify_character(ocr_model, char_image):
74     img = char_image.convert('L')
75     img = img.resize((28, 28))
76     x = np.asarray(img, dtype=np.float32)
77     x = np.expand_dims(x, axis=0) / 255.
78
79     predictions = ocr_model.predict(x, verbose=False)
80     predicted_class = np.argmax(predictions[0])
81     character = ocr_labels[predicted_class]
82
83     return character
84

```

*Fig.6. Class Voucher*

En la Figura 6 se muestra:

La función `generate_text(self)` concatena los caracteres en líneas y palabras para formar un texto, para ello, recorre la lista de caracteres almacenada en la propiedad `characters` de la instancia actual. Para cada línea en `characters`, itera sobre cada palabra en la línea. Dentro del bucle interno, si no es el primer carácter de la palabra, se agrega un espacio en blanco al atributo de texto llamado `text`. Luego, se agrega el valor del atributo `char` de cada carácter en la palabra a `text`. Después de completar una palabra, se agrega un salto de línea a `text`. En resumen, la función genera un texto concatenando los caracteres en líneas y palabras basándose en la estructura de `characters`.

La función `identify_character(ocr_model, char_image)` utiliza un modelo OCR para identificar el carácter en una imagen proporcionada. Para ello, toma como entrada `ocr_model` y `char_image`. Primero, la imagen se convierte a escala de grises y se redimensiona a un tamaño de 28x28 píxeles. Luego, se convierte en una matriz numpy de tipo `float32` y se normaliza dividiendo por 255. Luego, se realiza una predicción utilizando el modelo OCR pasando la imagen normalizada. La función devuelve el

carácter identificado basado en la clase con la mayor probabilidad predicha por el modelo OCR. La variable `ocr_labels` se asume que contiene una lista de etiquetas correspondientes a las clases del modelo OCR.

```

85 ✓ def check_spell(model, text):
86     input_tokens = tokenizer.encode_plus(
87         text,
88         return_tensors="pt"
89     )
90
91     corrected_ids = model.generate(
92         input_ids=input_tokens["input_ids"],
93         attention_mask=input_tokens["attention_mask"],
94         max_length=700,
95         num_beams=4,
96         early_stopping=True
97     )
98
99     corrected_text = tokenizer.decode(corrected_ids[0], skip_special_tokens=True)
100
101     return corrected_text.replace("_", " ")
102
103 ✓ def detecte_char(img, model, contours, voucher, x_=0, y_=0):
104     for contour in contours:
105         (x, y, w, h) = cv2.boundingRect(contour)
106         x, y = x+x_, y+y_
107
108         if w > 1000 and h > 900:
109             new_img = img[y+10:y+h-10, x+10:x+w-10]
110             new_gray = cv2.cvtColor(new_img, cv2.COLOR_BGR2GRAY)
111             new_blur = cv2.GaussianBlur(new_gray, (5,5), 0)
112             new_thresh = cv2.adaptiveThreshold(new_blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, 2)
113             new_contours, _ = cv2.findContours(new_thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
114             voucher = detecte_char(img, model, new_contours, voucher, x+10, y+10)
115         elif w < 3 or h < 8 or w > 40 or h > 40:
116             continue
117         else:
118             char_img = Image.fromarray(cv2.cvtColor(img[y:y+h, x:x+w], cv2.COLOR_BGR2RGB))
119             character = identify_character(ocr_model, char_img)
120             voucher.append(Character(x,y,w,h,character))
121
122     return voucher

```

*Fig. 7. Class Voucher*

En la Figura 7 se muestra la función `check_spell(model, text)` toma un modelo y un texto como entrada. En primer lugar, se utiliza un tokenizer para codificar el texto y obtener los tokens de entrada necesarios para el modelo. Luego, se utiliza el modelo para generar correcciones de texto. Se especifica una longitud máxima permitida de 700, se utilizan 4 beams para la generación y se detiene tempranamente. El resultado de la generación se decodifica utilizando el tokenizer para obtener el texto corregido, y luego se reemplaza el token especial "\_" por un espacio en blanco. Finalmente, la función devuelve el texto corregido.

Además se tiene a la función `detecte_char(img, model, contours, voucher, x_=0, y_=0)` toma como entrada una imagen, un modelo, una lista de contornos, voucher y coordenadas opcionales. La función itera sobre los contornos proporcionados y realiza varias operaciones para detectar y reconocer los caracteres en la imagen.

```

124     if __name__ == '__main__':
125         root_path = sys.argv[1]
126         images_path = sys.argv[2]
127
128         # OCR
129         print("<PROGRESS>:Cargando modelo de OCR...;25", flush=True)
130         ocr_model = keras.models.load_model(f"{root_path}/models/ocr_model.h5")
131         with open(f"{root_path}/models/ocr_labels.json") as file:
132             ocr_labels = {int(k): v for k, v in json.load(file).items()}
133
134         # Spell checker
135         print("<PROGRESS>:Cargando modelo de corrección...;35", flush=True)
136         tokenizer = AutoTokenizer.from_pretrained("Helsinki-NLP/opus-mt-es-en")
137         checker_model = AutoModelForSeq2SeqLM.from_pretrained(f"{root_path}/models/spell_checker_model")
138
139         # NER
140         print("<PROGRESS>:Cargando modelo de reconocimiento de entidades...;45", flush=True)
141         ner_model = spacy.load(f"{root_path}/models/ner_model")
142
143         images_list = os.listdir(images_path)
144         texts = []
145
146         print("<PROGRESS>:Iniciando procesamiento de imágenes...;50", flush=True)
147         current_progress = 50;
148         progress_batch = 35.0/len(images_list);
149         img_count = 0;

```

*Fig.8.Class Voucher*

```

150     for image_name in images_list:
151         img_count += 1
152         current_progress += progress_batch
153         print(f"<PROGRESS>:Procesando imagen: {img_count} de {len(images_list)}...;{int(current_progress)}", flush=True)
154         img = cv2.imread(f'{images_path}/{image_name}')
155         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
156         blur = cv2.GaussianBlur(gray, (5,5), 0)
157         thresh = cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, 2)
158         contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
159
160         voucher = Voucher()
161         voucher = detecte_char(img, ocr_model, contours, voucher)
162         voucher.sort_characters()
163         corrected_text = ""
164         if len(voucher.text) <= 512:
165             corrected_text = check_spell(checker_model, voucher.text)
166         else:
167             for j in range(round(len(voucher.text) / 512)):
168                 corrected_text += check_spell(checker_model, voucher.text[j*512:512*(j+1)])
169
170         doc = ner_model(corrected_text)
171         data = {'NAME': None, 'MONTH': None, 'YEAR': None, 'CODE': None}
172         for ent in doc.ents:
173             if ent.label_ == 'NAME' and data['NAME'] == None: data['NAME'] = ent.text
174             if ent.label_ == 'CODE' and data['CODE'] == None: data['CODE'] = ent.text
175             if ent.label_ == 'YEAR' and data['YEAR'] == None: data['YEAR'] = int(ent.text)
176             if ent.label_ == 'MONTH' and data['MONTH'] == None:
177                 if ent.text in MONTHS: data['MONTH'] = MONTHS.index(ent.text) + 1
178             if ent.label_ == 'DATE':
179                 if data['MONTH'] == None: data['MONTH'] = int(ent.text.split("/")[1])
180                 if data['YEAR'] == None: data['YEAR'] = int(ent.text.split("/")[2])
181
182         texts.append({
183             'image_name': image_name,
184             'original_text': voucher.text,
185             'text': corrected_text,
186             'teacher_name': data['NAME'],
187             'month': data['MONTH'],
188             'year': data['YEAR'],
189             'code': data['CODE']
190         })
191
192     print(f"<RESPONSE>:{json.dumps(texts)}")

```

*Fig.9.Class Voucher*

En las Figura 8 y 9 se toman dos argumentos de línea de comandos: la ruta principal `root_path` y la ruta de las imágenes `images_path`, se carga el modelo de OCR desde el archivo `ocr_model.h5` ubicado en la carpeta de modelos dentro de la ruta principal. También carga las etiquetas de OCR desde el archivo `ocr_labels.json` y las almacena en un diccionario.

Luego de ello se carga el tokenizer necesario para el modelo de corrección ortográfica desde la biblioteca `Transformers`. El tokenizer se inicializa con el modelo `Helsinki-NLP/opus-mt-es-en`.

Después se carga el modelo de corrección ortográfica desde el archivo `spell_checker_model` ubicado en la carpeta de modelos dentro de la ruta principal y se carga el modelo de reconocimiento de entidades (NER) de `Spacy` desde la carpeta de modelos dentro de la ruta principal. Se obtiene una lista de nombres de archivos de imágenes en la carpeta `images_path`.

Inicia el procesamiento de las imágenes en un bucle. Para cada imagen, realiza las siguientes operaciones:

Lee la imagen y realiza una serie de operaciones de preprocesamiento, como convertir a escala de grises, aplicar desenfoque gaussiano y umbral adaptativo para obtener una imagen binaria.

Encuentra los contornos en la imagen binaria.

Crea un objeto voucher y llama a la función `detecte_char` para detectar y reconocer los caracteres en la imagen. La función `sort_characters` se llama después para ordenar los caracteres.

Luego, se verifica la longitud del texto en voucher. Si es menor o igual a 512 caracteres, se llama a la función `check_spell` para corregir el texto utilizando el modelo de corrección ortográfica. Si el texto es más largo, se divide en segmentos de 512 caracteres y se aplica la corrección a cada segmento.

A continuación, se utiliza el modelo de reconocimiento de entidades (NER) para identificar entidades relevantes en el texto corregido. Los campos de interés son "NAME" (nombre), "MONTH" (mes), "YEAR" (año) y "CODE" (código).

Los resultados del procesamiento se agregan a una lista llamada `texts`, que contiene información como el nombre de la imagen, el texto original, el texto corregido, el nombre del docente, el mes, el año y el código.

Finalmente, se imprime la lista `texts` en formato JSON como resultado del procesamiento de imágenes.

**ANEXO N° 04. PRUEBAS DE CAJA NEGRA Y CAJA BLANCA**

TABLA IV  
PRUEBA DE CAJA NEGRA N°1

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Login aplicación de web	Funcional	01
<b>Descripción</b>		
El usuario del sistema puede ingresar correctamente con sus credenciales		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN01	Login	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Ingreso al sistema	Credenciales del usuario	Ingreso correcto y sin errores al sistema
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema permitió el correcto ingreso al usuario con sus credenciales	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	



TABLA V  
PRUEBA DE CAJA NEGRA N°2

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Inicio aplicación de web	Funcional	02
<b>Descripción</b>		
El usuario puede ver la interfaz de home (mis boletas de pago) en la aplicación web y buscar sus boletas		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN02	Inicio	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada</b> <b>/ Acciones de entrada</b>	<b>Resultado esperado</b>
Interfaz principal de la aplicación web	Credenciales del usuario	Se muestra correctamente la interfaz de inicio de la aplicación y el usuario puede buscar sus boletas sin ningún inconvenientes
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema mostro correctamente la interfaz de inicio de la aplicación y el usuario puedo buscar sus boletas sin ningún inconvenientes	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA VI  
PRUEBA DE CAJA NEGRA N°3

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Login aplicación de escritorio	Funcional	03
<b>Descripción</b>		
El usuario del sistema puede ingresar correctamente con sus credenciales		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN03	Login	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Ingreso al sistema	Credenciales del usuario	Ingreso correcto y sin errores al sistema
<b>Requerimientos de ambiente de pruebas</b>		
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b>
		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema permitió el correcto ingreso al usuario con sus credenciales	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA VII  
PRUEBA DE CAJA NEGRA N°4

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Inicio aplicación de escritorio	Funcional	04
<b>Descripción</b>		
El usuario puede ver los gráficos de los procesos OCR realizados en ciertos periodos		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN04	Inicio	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Interfaz principal de la aplicación de escritorio	Credenciales del usuario	Se muestra correctamente la interfaz de inicio de la aplicación de escritorio y el usuario puede visualizar los gráficos correspondientes a los procesos OCR realizados
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema mostro correctamente la interfaz de inicio de la aplicación de escritorio y el usuario puede visualizar los gráficos correspondientes a los procesos OCR realizados sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	

TABLA VIII  
PRUEBA DE CAJA NEGRA N°5

PRUEBAS DE CAJA NEGRA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b> Usuario	<b>Tipo de requisito</b> Funcional	<b>Código del requisito</b> 05
<b>Descripción</b> El usuario administrador puede gestionar los datos de los usuarios, es decir realizar las operaciones de: crear, ver, editar, eliminar, listar mediante el aplicativo de escritorio desarrollado		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PCN05	<b>Caso de prueba</b> Gestión de usuarios	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Ver datos del usuario 2.Listar datos del usuario 3.Editar los datos del usuario 4.Registrar al usuario 5.Eliminar usuario 6.Buscar usuario	<b>Datos de entrada / Acciones de entrada</b> Probar las operaciones CRUD un usuario	<b>Resultado esperado</b> Correcto funcionamiento de todas y cada una de las funciones CRUD correspondientes a un usuario mediante la interfaz del aplicativo de escritorio
<b>Requerimientos de ambiente de pruebas</b> La aplicación debe de estar en ejecución y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> El sistema permitió la correcta realización de las operaciones CRUD del usuario mediante la aplicación de escritorio	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b> -	<b>Observaciones</b> -	

TABLA IX  
PRUEBA DE CAJA NEGRA N°6

PRUEBAS DE CAJA NEGRA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Rol de usuario	Funcional	06
<b>Descripción</b>		
El usuario administrador puede gestionar los datos de los roles de los usuarios, es decir realizar las operaciones de: crear, ver, editar, eliminar, listar mediante el aplicativo de escritorio desarrollado		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN06	Gestión de rol de usuarios	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
1.Ver datos del rol de usuario 2.Listar datos del rol de usuario 3.Editar los datos del rol de usuario 4.Registrar al rol de usuario 5.Eliminar rol de usuario 6.Buscar rol de usuario	Probar las operaciones CRUD un rol de usuario	Correcto funcionamiento de todas y cada una de las funciones CRUD correspondientes a un rol de usuario mediante la interfaz del aplicativo de escritorio
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema permitió la correcta realización de las operaciones CRUD del rol de usuario mediante la aplicación de escritorio	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA X  
PRUEBA DE CAJA NEGRA N°7

PRUEBAS DE CAJA NEGRA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b> Docente	<b>Tipo de requisito</b> Funcional	<b>Código del requisito</b> 07
<b>Descripción</b> El usuario administrador puede gestionar los datos de los docentes, es decir realizar las operaciones de: crear, ver, editar, eliminar, listar mediante el aplicativo de escritorio desarrollado		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PCN07	<b>Caso de prueba</b> Gestión de docentes	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Ver datos del docente 2.Listar datos del docente 3.Editar los datos del docente 4.Registrar docente 5.Eliminar docente 6.Buscar docente	<b>Datos de entrada / Acciones de entrada</b> Probar las operaciones CRUD de un docente	<b>Resultado esperado</b> Correcto funcionamiento de todas y cada una de las funciones CRUD correspondientes a un docente mediante la interfaz del aplicativo de escritorio
<b>Requerimientos de ambiente de pruebas</b> La aplicación debe de estar en ejecución y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> El sistema permitió la correcta realización de las operaciones CRUD del docente mediante la aplicación de escritorio	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b> -	<b>Observaciones</b> -	

TABLA XI  
PRUEBA DE CAJA NEGRA N°8

PRUEBAS DE CAJA NEGRA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b> Detalle de Boleta	<b>Tipo de requisito</b> Funcional	<b>Código del requisito</b> 08
<b>Descripción</b> El usuario administrador puede ver el detalle de la boleta almacenada		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PCN08	<b>Caso de prueba</b> Boleta	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Ver datos de la boleta 2.Editar los datos de la boleta	<b>Datos de entrada</b> <b>/ Acciones de entrada</b> -	<b>Resultado esperado</b> Visualización de los datos de la boleta y edición de los mismos en caso sea necesario
<b>Requerimientos de ambiente de pruebas</b> La aplicación debe de estar en ejecución y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> El sistema permitió la correcta visualización de los datos de la boleta, así como la realización de cambios y correcto guardado.	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b> -	<b>Observaciones</b> -	

TABLA XII  
PRUEBA DE CAJA NEGRA N°9

PRUEBAS DE CAJA NEGRA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Creación y ejecución del modelo OCR.	Funcional	09
<b>Descripción</b>		
El usuario puede crear un nuevo proceso OCR cargando la boleta y el sistema se encarga de ejecutar el modelo		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN09	Modelo OCR	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
1. Creación del modelo OCR. 2. Ejecución del modelo OCR.	Credenciales del usuario	El usuario puede realizar la creación de un nuevo proceso OCR y el sistema se encarga de ejecutar el modelo
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema le permitió al usuario crear un nuevo modelo OCR y realizó la ejecución del mismo sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	



TABLA XIII  
PRUEBA DE CAJA NEGRA N°10

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Listado de procesos OCR	Funcional	10
<b>Descripción</b>		
El usuario puede ver el listado de procesos OCR realizados		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCN10	Modelo OCR	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Listado de procesos OCR	Credenciales del usuario	El usuario puede ver el listado de los procesos OCR realizados
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
La aplicación debe de estar en ejecución y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
El sistema le permitió al usuario puede ver el listado de los procesos OCR realizados sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XIV  
PRUEBA DE CAJA BLANCA N°1

**PRUEBAS DE CAJA BLANCA**

<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código del login aplicación de web	Funcional	11
<b>Descripción</b>		
Funcionalidad del login		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB01	Login	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Ingreso al sistema	Credenciales del usuario	Ingreso correcto y sin errores al sistema
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Se permitió el correcto ingreso al usuario con sus credenciales al sistema	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XV  
PRUEBA DE CAJA BLANCA N°2

PRUEBAS DE CAJA BLANCA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código del inicio aplicación de web	Funcional	12
<b>Descripción</b>		
Extracción de los datos de las boletas del usuario y filtros de búsqueda		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB02	Inicio	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada</b>	<b>Resultado esperado</b>
1.Extracción de los datos de las boletas del usuario	/ Acciones de entrada	Se extraen los datos de las boletas del usuario y se pueden filtrar las búsquedas
2. Filtros de búsqueda para as boletas	Credenciales del usuario	
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Se extraen los datos de las boletas del usuario y se pueden filtrar las búsquedas	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XVI  
PRUEBA DE CAJA BLANCA N°3

<b>PRUEBAS DE CAJA BLANCA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código del login de la aplicación de escritorio	Funcional	13
<b>Descripción</b>		
El usuario del sistema puede ingresar correctamente con sus credenciales		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB03	Login	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada</b>	<b>Resultado esperado</b>
Ingreso al sistema	/ <b>Acciones de entrada</b> Credenciales del usuario	Ingreso correcto y sin errores al sistema
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Se permitió el correcto ingreso al usuario con sus credenciales al sistema	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XVII  
PRUEBA DE CAJA BLANCA N°4

PRUEBAS DE CAJA BLANCA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código de inicio aplicación de escritorio	Funcional	14
<b>Descripción</b>		
Extracción de los datos de los procesos OCR realizados en ciertos periodo		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB04	Inicio	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada</b>	<b>Resultado esperado</b>
Extracción de los datos de los procesos OCR realizados	/ <b>Acciones de entrada</b> Credenciales del usuario	Se obtienen los datos correspondientes a los procesos OCR realizados
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Se obtuvieron los datos correspondientes a los procesos OCR realizados sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	

TABLA XVIII  
PRUEBA DE CAJA BLANCA N°5

PRUEBAS DE CAJA BLANCA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código de los CRUD del usuario	Funcional	15
<b>Descripción</b>		
Operaciones CRUD del usuario		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PC05	CRUD de usuarios	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada</b>	<b>Resultado esperado</b>
1.Extrar datos del usuario	/ <b>Acciones de entrada</b> Probar las operaciones CRUD un usuario	Correcto funcionamiento de todas y cada una de las funciones CRUD
2.Listar datos del usuario		
3.Editar los datos del usuario		
4.Registrar al usuario		
5.Eliminar usuario		
6.Buscar usuario		
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Funcionamiento de todas las operaciones CRUD del usuario sin ningún inconveniente	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XIX  
PRUEBA DE CAJA BLANCA N°6

<b>PRUEBAS DE CAJA BLANCA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b> Código de los CRUD del rol de usuario	<b>Tipo de requisito</b> Funcional	<b>Código del requisito</b> 16
<b>Descripción</b> Operaciones CRUD del rol de usuario		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PC0B6	<b>Caso de prueba</b> Gestión de rol de usuarios	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Extraer datos del rol de usuario 2.Listar datos del rol de usuario 3.Editar los datos del rol de usuario 4.Registrar al rol de usuario 5.Eliminar rol de usuario 6.Buscar rol de usuario	<b>Datos de entrada</b> <b>/ Acciones de entrada</b> Probar las operaciones CRUD un rol de usuario	<b>Resultado esperado</b> Correcto funcionamiento de todas y cada una de las funciones CRUD del rol de usuario
<b>Requerimientos de ambiente de pruebas</b> Ejecución del código y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> Correcta realización de las operaciones CRUD del rol	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b> -	<b>Observaciones</b> -	

TABLA XX  
PRUEBA DE CAJA BLANCA N°7

<b>PRUEBAS DE CAJA BLANCA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b> Código de los CRUD del docente	<b>Tipo de requisito</b> Funcional	<b>Código del requisito</b> 17
<b>Descripción</b> Operaciones CRUD del docente		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PCB07	<b>Caso de prueba</b> Gestión de docentes	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Extraer datos del docente 2.Listar datos del docente 3.Editar los datos del docente 4.Registrar docente 5.Eliminar docente 6.Buscar docente	<b>Datos de entrada</b> <b>/ Acciones de entrada</b> Probar las operaciones CRUD de un docente	<b>Resultado esperado</b> Correcto funcionamiento de todas y cada una de las funciones CRUD correspondientes a un docente
<b>Requerimientos de ambiente de pruebas</b> Ejecución del código y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> Correcta realización de las operaciones CRUD del docente	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	



TABLA XXI  
PRUEBA DE CAJA BLANCA N°8

**PRUEBAS DE CAJA BLANCA**

<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código del detalle de boleta	Funcional	18
<b>Descripción</b>		
Operaciones de extraer datos y editar boleta de pago		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b> PCB08	<b>Caso de prueba</b> Boleta	<b>Fecha de prueba</b> 17/06/2023
<b>Funcionalidad / Característica a evaluar</b> 1.Extracción de datos de la boleta 2.Editar los datos de la boleta	<b>Datos de entrada / Acciones de entrada</b> -	<b>Resultado esperado</b> Extracción de los datos de la boleta y edición de los mismos en caso sea necesario
<b>Requerimientos de ambiente de pruebas</b> Ejecución del código y la base de datos conectada correctamente		<b>Condiciones / Restricciones</b> Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b> Correcta extracción de los datos de la boleta, así como la realización de cambios y correcto guardado.	<b>Estado actual</b> Conforme	<b>Observaciones</b> Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b> -	<b>Observaciones</b> -	

TABLA XXII  
PRUEBA DE CAJA BLANCA N°9

PRUEBAS DE CAJA BLANCA		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código de la creación y ejecución del modelo OCR	Funcional	19
<b>Descripción</b>		
Creación de un nuevo proceso OCR cargando la boleta y ejecución del modelo		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB09	Modelo OCR	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
1. Creación del modelo OCR. 2. Ejecución del modelo OCR.	Credenciales del usuario	Creación de un nuevo proceso OCR y ejecución del modelo
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Creación de un nuevo proceso OCR y ejecución del modelo sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	

TABLA XXIII  
PRUEBA DE CAJA NEGRA N°10

<b>PRUEBAS DE CAJA NEGRA</b>		
<b>REQUISITO</b>		
<b>Módulo / Área Funcional / Sub proceso</b>	<b>Tipo de requisito</b>	<b>Código del requisito</b>
Código del listar procesos OCR	Funcional	20
<b>Descripción</b>		
Listado de procesos OCR realizados		
<b>CASO DE PRUEBA</b>		
<b>Código de prueba</b>	<b>Caso de prueba</b>	<b>Fecha de prueba</b>
PCB10	Modelo OCR	17/06/2023
<b>Funcionalidad / Característica a evaluar</b>	<b>Datos de entrada / Acciones de entrada</b>	<b>Resultado esperado</b>
Extracción de procesos OCR realizados	Credenciales del usuario	Listado de los procesos OCR realizados
<b>Requerimientos de ambiente de pruebas</b>		<b>Condiciones / Restricciones</b>
Ejecución del código y la base de datos conectada correctamente		Ninguna
<b>SEGUIMIENTO</b>		
<b>Resultado obtenido</b>	<b>Estado actual</b>	<b>Observaciones</b>
Listado de los procesos OCR realizados sin ningún inconveniente.	Conforme	Ninguna
<b>CORRECCIONES</b>		
<b>Fecha de cambio de estado</b>	<b>Observaciones</b>	
-	-	