



ESCUELA DE INFORMÁTICA Y MULTIMEDIA

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
INFORMÁTICA Y MULTIMEDIA**

AUTOR: David Fernando Salinas Herrera.

Director: Ing. Luis Roberto Jácome Galarza, *Mg.Sc.*

**IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA EL
RECONOCIMIENTO DE VEHÍCULOS Y AUTOMATIZACIÓN DEL LIBRO DIARIO DE
REGISTRO PARA LA EMPRESA LAVADORA LISTOCAR.**

LÍNEA DE INVESTIGACIÓN: Tecnología de la Información y Procesamiento de Imágenes.

FECHA: 24/11/2021.

CERTIFICACIÓN DE AUTORÍA.

Yo, David Fernando Salinas Herrera, declaro bajo juramento, que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado académico o título profesional y que se ha consultado la bibliografía necesaria para su elaboración. Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.



David Fernando Salinas Herrera

Yo, Ing. Luis Roberto Jácome Galarza, Mg. Sc, certifico que conozco al autor del presente trabajo, siendo el responsable exclusivo tanto de su originalidad y autenticidad, como de su contenido.



Firmado electrónicamente por:
**LUIS ROBERTO
JACOME GALARZA**

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DIRECTOR DE TESIS

DEDICATORIA

Este trabajo de investigación está dedicado a mis padres y hermanos, Luis, Ana, Guillermo y Fernando, por su apoyo y tiempo que generosamente brindado durante mi formación educativa, sus aportes, consejos y cariño han permitido el avance de este proyecto hasta su formalización.

David Salinas.

AGRADECIMIENTO

Extiendo mis agradecimientos a la Universidad Internacional del Ecuador, por la aportación de conocimientos y oportunidades para mi formación profesional y adquisición de experiencia.

Agradezco a la Magíster Lorena Conde, Directora de la Carrera de Informática y Multimedia por su guía y la impartición de sus conocimientos que facilitaron la elaboración de esta investigación.

Agradezco al Magíster Roberto Jácome, Director de Tesis, con su asesoramiento y tutela en la investigación de este trabajo y en el cumplimiento de los objetivos y estándares establecidos al inicio de la investigación de este proyecto.

David Fernando Salinas Herrera.

ÍNDICE.

CERTIFICACIÓN DE AUTORÍA.....	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
INDICE	v
ÍNDICE DE FIGURAS	ix
RESUMEN	xii
INTRODUCCIÓN	xiv
PROBLEMATIZACIÓN	xvi
OBJETIVOS	xviii
METODOLOGÍA	xix
CAPÍTULO 1. Introducción	1
CAPÍTULO 2. Marco Teórico	3
2.1. Lenguaje De Programación Python	3
2.1.1. ¿Qué es Python?	3
2.2. ¿Qué Es Jupyter?	4
2.3. Jupyter Notebook	5
2.4. Deepnote	6
2.5. Reconocimiento Óptico De Caracteres	7
2.5.1. ¿Qué Es El Reconocimiento De Caracteres?	7
2.6. Inteligencia Artificial	15
2.7. Tipos De Inteligencia Artificial	17

2.7.1. Inteligencia Artificial Débil	17
2.7.2. Inteligencia Artificial Fuerte	18
2.7.3. Inteligencia Artificial Para La Automatización	18
2.8. Aprendizaje Artificial	19
2.8.1. Aprendizaje Artificial Supervisado	19
2.8.2. Aprendizaje Artificial No Supervisado	20
2.8.3. Aprendizaje Semi-Supervisado	22
2.9. Beneficios del Reconocimiento Óptico de Caracteres Avanzado	22
2.10. Procesamiento Inteligente De Documentos	25
2.11. Aprendizaje Profundo	26
2.12. ¿Qué es Opencv?	30
2.13. ¿Qué Es Easyocr?	33
2.14. Pytorch	36
2.15. Red Neuronal Convolutiva	37
2.15.1. Redes Neuronales	37
2.15.2. Redes Neuronales Convolucionales	39
2.16. Unidad Lineal Rectificada	47
2.16.1. Combinación De Capas	50
2.16.2. Capa Completamente Conectada	51
2.17. Red Neuronal Recurrente	51
2.18. Modelo De Reconocimiento De Texto Craft	54
2.19. Red Residual	55
2.20. Red Neuronal Convolutiva De Uso Común Vgg	57

2.21. Redes Neuronales De Memoria A Corto Plazo	59
2.21.1. Puerta De Olvido	63
2.21.2. Puerta de Ingreso	63
2.21.3. Nueva Información	64
2.21.4. Puerta de salida	64
2.22. Clasificación Temporal Conexionista	65
2.23. Aplicativo Web	66
2.23.1. ¿Qué es un Aplicativo Web?	66
2.24. ¿Qué es Anvil?	67
CAPÍTULO 3. Desarrollo Del Sistema	70
3.1. Antecedentes del Registro	70
3.2. Entrevistas con los Usuarios	71
3.2.1. Historias de Usuario	71
3.3. Experimentación.....	73
3.3.1. Experimentación 1	73
3.3.2. Experimentación 2	75
3.3.3. Experimentación 3	78
3.3.4. Experimentación 4	80
3.3.5. Experimentación 5	81
3.4. Utilización de la metodología Mobile-D	86
3.4.1. Análisis	87
3.4.1.1. Obtención de Requerimientos	87
3.4.1.2. Clasificación de los Requerimientos	87

3.4.1.3. Personalización del Servicio	88
3.4.2. Diseño	89
3.4.2.1. Definir el escenario	89
3.4.2.2. Estructura del Software	89
3.4.2.2.1. Esquema de navegabilidad	90
3.4.3. Desarrollo	91
3.4.3.1. Codificación	91
3.4.3.1.1. Aplicativo Web desarrollado en Anvil	91
3.4.3.2. Pruebas Unitarias	96
3.4.4. Pruebas de Funcionamiento	99
3.4.4.1. Emulación y Simulación	99
3.4.4.1.1. Simulación de la Interfaz	99
3.4.4.1.2. Instalación	101
CONCLUSIONES Y RECOMENDACIONES	104
CONCLUSIONES	104
RECOMENDACIONES	105
BIBLIOGRAFÍA	106

Índice de Figuras.

<i>Figura 1. Fases del ciclo de la metodología ágil Mobile-D</i>	xxi
<i>Figura 2. Representación del proceso de Aprendizaje Artificial Supervisado</i>	20
<i>Figura 3. Representación del proceso de Aprendizaje Artificial No Supervisado</i>	21
<i>Figura 4. Representación de una Red Neuronal Artificial</i>	38
<i>Figura 5. Ejemplificación del proceso de convolución</i>	40
<i>Figura 6. Representación de la capa convolucional de una red neuronal convolucional 1</i>	41
<i>Figura 7. Representación de la capa convolucional de una red neuronal convolucional 2</i>	42
<i>Figura 8. Representación de la capa convolucional de una red neuronal convolucional 3</i>	43
<i>Figura 9. Representación del modelo de Reconocimiento de objetos por capas</i>	46
<i>Figura 10. Gráfico de línea de activación lineal rectificadora para entradas negativas y positivas</i>	48
<i>Figura 11. Interpretación gráfica de la aplicación método de aprendizaje residual y la utilización de la unidad de rectificación lineal para el aprendizaje artificial</i>	56
<i>Figura 12. Interpretación de las secciones de las Redes Neuronales de Memoria a Corto Plazo</i>	61
<i>Figura 13. Arquitectura de las Redes Neuronales de Memoria a Corto Plazo</i>	62
<i>Figura 14. Ficha de registro de la Historia de Usuario</i>	71
<i>Figura 15. Ficha de registro de la Historia de Usuario</i>	72
<i>Figura 16. Ficha de registro de la Historia de Usuario</i>	72
<i>Figura 17. Experimentación con el código resultando en errores y conflictos con los paquetes y módulos de TensorFlow 1.x</i>	74

<i>Figura 18. Experimentación con el código resultando en errores en la utilización de escala de grises</i>	<i>76</i>
<i>Figura 19. Experimentación con el código resultando en problemas al realizar la detección de bordes de la placa y aplicación de máscaras en la imagen debido a errores en la utilización de escala de grises</i>	<i>77</i>
<i>Figura 20. Experimentación con el código resultando en problemas al tratar de acceder al ejecutable de tesseract instalado en la computadora donde se realiza la experimentación</i>	<i>78</i>
<i>Figura 21. Experimentación con el código en Visual Code, requiere la clonación del repositorio YOLOv5 y su entrenamiento para el caso específico a ser aplicado</i>	<i>80</i>
<i>Figura 22. Experimentación con el código utilizando el servicio Deepnote en lugar de Jupyter Notebook</i>	<i>81</i>
<i>Figura 23. Instalación del paquete Anvil Uplink e importación</i>	<i>82</i>
<i>Figura 24. Experimentación con la función de Anvil, Anvil.media</i>	<i>83</i>
<i>Figura 25. Experimentación con la función de aprendizaje de la lista de resultados obtenidos por el reconocimiento de texto</i>	<i>85</i>
<i>Figura 26. Experimentación con la conexión entre el aplicativo web y la llamada de las funciones en el código</i>	<i>86</i>
<i>Figura 27. Abstracción gráfica del diseño del sistema simplificado</i>	<i>90</i>
<i>Figura 28. Esquema gráfico de la navegabilidad de la interfaz del registro vehicular</i>	<i>90</i>
<i>Figura 29. Diseño y construcción de la interfaz en Anvil Works</i>	<i>91</i>
<i>Figura 30. Interfaz de Propiedades de los widgets</i>	<i>92</i>
<i>Figura 31. Eventos en la interfaz de Propiedades de los widgets</i>	<i>92</i>
<i>Figura 32. Editor de código en Anvil para los widgets y otras funciones</i>	<i>93</i>

<i>Figura 33. Diseño del formulario para los datos de los vehículos</i>	<i>94</i>
<i>Figura 34. Código de la asignación de variables para los campos del registro y datos de la tabla en la base de datos</i>	<i>95</i>
<i>Figura 35. Tabla en la base de datos utilizada para almacenar los datos de los vehículos registrados.....</i>	<i>96</i>
<i>Figura 36. Captura de pantalla y descripción de la interfaz 1</i>	<i>97</i>
<i>Figura 37. Captura de pantalla y descripción de la interfaz 2</i>	<i>97</i>
<i>Figura 38. Captura de pantalla y descripción de la interfaz 3</i>	<i>98</i>
<i>Figura 39. Captura de pantalla y descripción de la interfaz 4</i>	<i>98</i>
<i>Figura 40. Captura de la interfaz de desarrollo del aplicativo web, el botón Run utilizado para la simulación.</i>	<i>99</i>
<i>Figura 41. Captura de la interfaz de desarrollo del aplicativo web, presentado en pantalla dividida para la simulación.</i>	<i>100</i>
<i>Figura 42. Captura de la interfaz de desarrollo del aplicativo web, maximizada para la simulación.</i>	<i>100</i>
<i>Figura 43. Captura de la interfaz de desarrollo del aplicativo web, presentado en una nueva pestaña para la simulación.</i>	<i>101</i>
<i>Figura 44. Captura de pantalla del navegador y descripción de los pasos 1</i>	<i>102</i>
<i>Figura 45. Captura de pantalla del navegador y descripción de los pasos 2</i>	<i>102</i>
<i>Figura 46. Captura de pantalla del navegador y descripción de los pasos 3</i>	<i>103</i>
<i>Figura 47. Captura de pantalla del navegador y descripción de los pasos 4</i>	<i>103</i>

RESUMEN

El presente proyecto de investigación tiene como objetivo agilizar los procesos documentación y registro vehicular de las actividades de servicio en el establecimiento de lavado de vehículos LISTOCAR, utilizando aplicativos web, y facilitar mediante el procesamiento de imágenes a través de reconocimiento óptico de caracteres e interpretación de texto de fotos tomadas de placas de los vehículos con teléfono móvil y cargados en el formulario del aplicativo web, accesibles desde un ordenador o un teléfono móvil permitiendo que la administración del establecimiento se logre de manera más eficiente y la comunicación del personal sea más eficaz.

PALABRAS CLAVE: Reconocimiento, Placa vehicular, Aplicativo Web.

ABSTRACT

The objective of this research project is to streamline the documentation and vehicular registration processes of the service activities in the LISTOCAR vehicle wash establishment, using web applications, and to facilitate image processing through optical character recognition and interpretation of text of photos taken from license plates of vehicles with a mobile phone and uploaded to the web application form, accessible from a computer or a mobile phone, allowing the administration of the establishment to be achieved more efficiently and the communication of the staff to be more effective.

KEY WORDS: Recognition, Vehicle license plate, Web application.

INTRODUCCIÓN

En la ciudad de Loja, existen múltiples negocios que se dedican al mantenimiento, reparación, cuidado y personalización de vehículos, estas actividades por lo general registran los vehículos que han sido recibidos en negocio. En su mayoría, los negocios de lavado vehicular de la ciudad de Loja requieren registrar los vehículos que han ingresado requiriendo sus servicios para mantener un control sobre el tiempo requerido y usado en los vehículos para su limpieza y/o mantenimiento, valor a cobrar, en caso de reclamos o pérdidas, etc.

El método al momento de mantener este registro es a través de cuadernos físicos, cuadernillos de tickets, y en el mejor de los casos hojas de Excel. Este método consume tiempo y es propenso a errores, malas interpretaciones, falta de uniformidad en la redacción, etc. Además, que luego estos registros en físico pueden sufrir daños por el cansancio de los materiales, mal almacenamiento, factores ambientales como calor o humedad, constituir un riesgo de incendio en el caso del papel almacenado, y perderse.

Esto hace que un aplicativo web accesible a través de ordenadores y teléfonos móviles facilita el obtener los listados de los vehículos y descripciones de los servicios requeridos, disponible tanto en oficina como permitiéndonos la revisión de los registros en cualquier lugar a través del teléfono móvil, haciendo el uso de herramientas de reconocimiento óptico de caracteres además de realizar copias y poder subir duplicados de los registros en la nube y permitiéndoles almacenarlos electrónicamente automatizando y mejorando los procesos existentes, evitando la aglomeración de papeles y documentos que puedan ser dañados o destruidos .

El propósito de este proyecto de investigación se cimienta en la implementación de reconocimiento óptico de caracteres en aplicativos webs de fácil acceso, diseñados a través de herramientas que permitan su creación y manejo de manera fácil y rápida para ser implementados prontamente y tener la elasticidad para ser utilizados en áreas relacionadas con el sector objetivo de este proyecto.

Para proporcionar un registro más ágil y práctico, se desarrollará un aplicativo web que permita llenar un formulario con los datos del vehículo recibido, con las características del mismo junto a un procesamiento de imágenes de la foto de la placa del vehículo, este formulario registra el vehículo en una lista que luego puede ser guardada.

PROBLEMATIZACIÓN

Para la supervisión y administración del establecimiento Lavadora LISTOCAR, requiere la comunicación entre los individuos que integran el personal del establecimiento y obtener la información del registro de los vehículos de los clientes para cumplir con sus funciones, debido a que hasta ese momento el registro es un cuaderno físico, las actividades eran interrumpidas debido a la necesidad de consultar los registros vehiculares.

Causas y Efectos del problema

Causas del Problema:

- El registro de los vehículos se realiza en un cuaderno físico manejado por el cajero, quien recibe al cliente en el establecimiento y registra los servicios solicitados y productos de venta en el almacén.
- La gerencia o dueño del establecimiento, debido a que en ocasiones debe realizar trámites o diligencias fuera del establecimiento, tiene que comunicarse con la persona encargada de la caja para recibir la información del cuaderno de registro.
- Errores al momento de escribir las entradas al registro físico o daños al mismo comprometen el registro actual y los registros anteriores.

Efectos derivados de las Causas.

- Los lavadores deben parar sus actividades para revisar o consultar el registro vehicular, los lavadores debían ir a oficina a consultar el servicio solicitado por el cliente si no se les había

comunicado o lo había olvidado, o si el cliente cambiaba de parecer y solicitaba otro servicio más completo o servicios adicionales, o cancelaba el servicio.

- El registro físico dificulta el mantener al gerente o dueño al tanto del flujo de trabajo, requiriendo la presencia constante del gerente o dueño, o comunicarse con la persona a cargo de la caja.
- Los registros físicos pueden dañarse, perderse, y si su almacenamiento no es realizado correctamente pueden destruirse.

OBJETIVOS

Objetivo General

Implementar un sistema de visión artificial para el reconocimiento vehicular y automatización del registro de servicios en locales de lavado y mantenimiento.

Objetivos Específicos

- Investigar el estado del arte en visión artificial y reconocimiento de objetos y el reconocimiento óptico de caracteres.
- Identificar los requerimientos funcionales y no funcionales de la aplicación.
- Diseñar e implementar un aplicativo web multiplataforma para el registro de vehículos y servicios ofrecidos en el local de lavado vehicular.
- Implementar un módulo de visión artificial para el reconocimiento de placas vehiculares.
- Realizar las pruebas de integración del módulo de detección de placas y del módulo de registro de vehículos.

METODOLOGÍA DE DESARROLLO

La selección de la metodología a utilizar para el proceso del desarrollo de software debe ajustarse a las necesidades del cliente, el espacio de tiempo disponible, carga de trabajo del proyecto y el tamaño del personal participando en el desarrollo, a continuación, se hablará de las metodologías ágiles y el método elegido a utilizar para este proyecto.

Metodologías Ágiles.

Las metodologías Ágiles son un enfoque iterativo utilizado para la creación de productos de software basado en el lanzamiento rápido de un Producto Mínimo Viable (“Minimum Viable Product” o MVP), para su posterior personalización y la adición incremental de funcionalidades y características basadas en el comportamiento y la retroalimentación del usuario. Un equipo de desarrollo de software ágil está mucho más integrado, cuenta con una colaboración constante y, muchas de las veces, intercambios regulares entre especialistas de diferentes disciplinas. Primero el Producto Mínimo Viable, luego las iteraciones son planificadas, se diseñan, son construidas, se realizan pruebas y se hace el lanzamiento en un proceso cíclico rápidamente.

Las nuevas iteraciones del Producto Mínimo Viable con sus nuevas características, funcionalidades, cambios y correcciones realizados, y mejoras de cada iteración están basados en una comunicación constante y cooperación en cada etapa del proceso de desarrollo entre los especialistas. (*Agile model*. (s/f). [Www.javatpoint.com](http://www.javatpoint.com). Recuperado el 6 de diciembre del 2021)

Metodología Mobile-D

La Metodología Ágil Mobile-D estaba basada en las prácticas de desarrollo Extreme Programming y las metodologías Crystal del Instituto Técnico de Investigación de Finlandia, utilizado con resultados positivos observados gracias a su uso, como son: aumento en el progreso de visibilidad, identificación temprana de problemas técnicos y soluciones, responsabilidad compartida, compartir información de manera eficiente, baja densidad de errores en productos lanzados, y un ritmo de desarrollo constante.

Por otra parte, existen algunas dificultades que podrían encontrarse en el uso de este método de desarrollo: Se requieren habilidades de desarrollo avanzadas para plataformas específicas, pueden existir problemas en la variabilidad y portabilidad debido a la diversidad de características en diferentes dispositivos móviles, dificultad para el lanzamiento rápido de ciclos, usar métodos ágiles no convencionales puede causar problemas, entre otros. ((S/f-a). Researchgate.net. Recuperado el 12 de agosto de 2022)

La metodología Ágil Mobile-D se conforma de 5 partes destacables:

- Exploración: enfocado para establecer el proyecto inicial y preparar las bases para proyectos a futuro.
- Inicialización: en esta fase se describen y preparan todos los componentes de la aplicación, así como también se trata de predecir posibles problemas críticos que puedan presentarse en el desarrollo del proyecto.
- Producción: Incluye la implementación de las funcionalidades requeridas usando ciclos de desarrollo iterativos e incrementales.

- Estabilización: finalización de la implementación; incluyendo de ser necesarios; subsistemas.
- Pruebas del sistemas y reparaciones: se realizan revisiones del desarrollo de la aplicación de acuerdo a los requerimientos del cliente y si contiene las funcionalidades requeridas.

Figura 1

Fases del ciclo de la metodología ágil Mobile-D



Nota. El gráfico representa las fases de la metodología de desarrollo ágil de desarrollo Mobile-D junto con las tareas que se realizan en cada fase de esta metodología. Recreación de *Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone (S/f-b)*. *Researchgate.net*. Recuperado el 13 de agosto de 2022.

CAPÍTULO I

INTRODUCCIÓN

En el establecimiento objetivo de este proyecto bajo el nombre comercial LISTOCAR brinda sus servicios de lavado y limpieza de vehículos, venta de accesorios y artículos de limpieza y mantenimiento. Ubicado en la ciudad de Loja en la Avenida Universitaria, entre la calle Lourdes y Catacocha. En este establecimiento presta sus servicios aproximadamente de a 25 a 45 vehículos diarios según las condiciones del clima y época del año, debido a esto se requiere que los vehículos que ingresan sean registrados para mantener el orden de atención, especificación de los servicios solicitados, artículos vendidos, valor a pagar, anotaciones y otros datos vitales para el cumplimiento de los servicios que han sido pedidos por el cliente.

El uso de registros en físico causa inconvenientes en la administración del negocio, y la comunicación entre el personal del establecimiento se ve ralentizada por la necesidad de consultar este registro en la oficina donde se administra el establecimiento, área diferente a la utilizada para la limpieza de los vehículos.

Para la realización de este proyecto se ha utilizado el módulo de reconocimiento óptico de caracteres de Python EasyOCR para el procesamiento y extracción de texto de imágenes, el cual será usado para el procesamiento de las imágenes tomadas de las placas de los vehículos.

Se pretende digitalizar el registro de vehículos en el establecimiento de lavado de vehículos LISTOCAR mediante un aplicativo web que además permita el procesamiento de imágenes tomadas con teléfonos móviles.

Un aplicativo web utilizado para el registro de vehículos accesible a través del ordenador de la oficina, así como través de teléfonos móviles permitiendo al personal acceso al registro sin interrumpir sus actividades para consultar en oficina, y de la misma manera la administración del local se puede realizar de manera remota gracias a que está disponible en línea.

CAPÍTULO II

MARCO TEÓRICO

2.1. Lenguaje de Programación Python

2.1.1. ¿Qué es Python?

Python es un popular lenguaje de programación de propósito general que se puede usar para una variedad de aplicaciones. Incluye estructuras de datos de alto nivel, escritura dinámica, enlace dinámico y muchas otras características que lo hacen tan útil para desarrollar aplicaciones complejas como para secuencias de comandos o "código adhesivo" que une componentes.

También se puede ampliar para realizar llamadas al sistema para casi todos los sistemas operativos y ejecutar código escrito en C o C++. Por su capacidad para ejecutarse en casi cualquier arquitectura de sistema, Python es un lenguaje universal que se puede encontrar en una variedad de aplicaciones diferentes. Python permite varios estilos de programación ya sea programación orientada a objetos, imperativa y/o funcional. Otros tipos de paradigmas de programación están soportados mediante extensiones.

En Python incluso las clases son objetos gracias a esto las clases al ser objetos son instancias de una meta clase Python además soporta la herencia múltiple y el polimorfismo. Este lenguaje de programación contiene miles de módulos de terceros disponibles en Python Package Index (PyPI). PyPI ofrece estándares populares para varias disciplinas, como Django para

desarrollo web y varios paquetes para ciencia de datos (o data science, en inglés) populares como Matplotlib, Pandas y NumPy.

Python 2.0 se lanzó el 16 de octubre de 2000 con muchas características nuevas e importantes, incluido un recolector de elementos no utilizados con reconocimiento de ciclos (además del recuento de referencias) para la administración de memoria y compatibilidad con Unicode. Sin embargo, el cambio más significativo ha afectado el proceso de desarrollo en sí mismo, con un cambio hacia un proceso más transparente y apoyado por la comunidad.

Python 3.0, una versión principal incompatible con versiones anteriores, se lanzó el 3 de diciembre de 2008 después de un largo período de prueba. Muchas de sus funciones principales también se han adaptado a Python 2.6 y 2.7, compatibles con versiones anteriores, pero ahora no compatibles. (by: I. (s/f). *What is python?* Opensource.com. Recuperado el 12 de agosto de 2022)

2.2. ¿Qué es Jupyter?

El proyecto Jupyter es un proyecto impulsado por la comunidad con el objetivo de “desarrollar software de código abierto, estándares abiertos y servicios para computación interactiva en docenas de lenguajes de programación”. Se separó de IPython en 2014 por Fernando Pérez y Brian Granger. (Wikipedia contributors. (2022, agosto 9). *Project Jupyter*. Wikipedia, The Free Encyclopedia.)

El nombre del proyecto Jupyter hace referencia a los tres lenguajes de programación principales admitidos por Jupyter, a saber, Julia, Python y R, y también rinde homenaje a los cuadernos de notas de Galileo que narran el descubrimiento de las lunas de Júpiter. Project Jupyter ha desarrollado y respaldado los productos informáticos interactivos Jupyter Notebook, JupyterHub y JupyterLab.

2.3. Jupyter Notebook

Jupyter Notebook (anteriormente IPython Notebooks) es un entorno informático interactivo basado en la web para crear documentos de cuaderno. Un documento de Jupyter Notebook es un REPL (bucle Lectura-Evaluación-Impresión, REPL por las siglas en inglés de “Read-Eval-Print-Loop”); un entorno de programación interactivo basado en un proceso cíclico de lectura de código ingresado por el usuario, evaluación y devolución de un resultado al usuario, basado en navegador que contiene una lista ordenada de celdas de entrada/salida, que puede contener código, texto (usando Markdown), matemáticas, gráficos y medios enriquecidos. Debajo de la interfaz, un cuaderno es un documento JSON que sigue un esquema versionado y normalmente termina con la extensión. ipynb. (*1. What is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 documentation.* (s/f). Readthedocs.Io. Recuperado el 16 de febrero de 2022)

Jupyter Notebook puede conectarse a muchos kernels para permitir la programación en diferentes lenguajes. Un kernel de Jupyter es un programa responsable de manejar diferentes tipos de solicitudes (ejecución de código, finalización de código, inspección) y proporcionar una

respuesta. Además, a diferencia de muchas otras interfaces similares a portátiles, los núcleos en Jupyter no saben que están adjuntos a un documento específico y pueden conectarse a muchos clientes simultáneamente. Por lo general, los núcleos permiten que solo se ejecute un idioma, con algunas excepciones. De forma predeterminada, Jupyter Notebook se envía con el kernel de IPython. A partir de la versión 2.3 lanzada en octubre de 2014, hay 49 núcleos compatibles con Jupyter para muchos lenguajes de programación, incluidos Python, R, Julia y Haskell.

2.4. Deepnote

Deepnote es un cuaderno gratuito de ciencia de datos en línea que se centra principalmente en la colaboración (en tiempo real, similar a Google Docs) y en la abstracción de todos los conceptos que interfieren en el trabajo: entorno y configuración de la infraestructura. Deepnote tiene disponibles distintos conjuntos de características dependiendo del plan elegido, Deepnote es posible usar en el plan gratuito de manera indefinida, dependiendo del caso de uso puede requerirse características de un plan de pago de los disponibles de la página web de Deepnote. Además, Deepnote permite guardar archivos como imágenes, documentos, videos, etc. que pueden ser accedidos por el archivo de Jupyter Notebook (.ipynb) siendo usado en Deepnote, disponiendo de 5GB de almacenamiento.

Al ejecutar un cuaderno de Jupyter Notebook, generalmente se necesita utilizar un entorno virtual creado en el ordenador en el que se le está ejecutando, con las librerías necesarias, para evitar instalar las librerías requeridas para ejecutar el cuaderno de Jupyter a través de la consola

del ordenador, mientras se utilizan el CPU, RAM y de ser necesario o de estar disponible, GPU, pudiendo ser eliminadas al borrar la carpeta del entorno virtual creada para el cuaderno.

Deepnote tiene la ventaja de no necesitar un entorno virtual creado en el ordenador, este servicio dispone de RAM y CPU asignado según el plan utilizado, de manera gratuita se asigna 1.2GB de RAM y además ofrece entornos virtuales de Python con varias versiones o se puede añadir un entorno virtual personalizado. Deepnote puede programar las ejecuciones del cuaderno a horas y días personalizados. (Candido, T. (2020, mayo 26). *Reviewing deepnote — the new IDE for data scientists*. Towards Data Science.)

2.5. Reconocimiento Óptico de Caracteres

2.5.1. ¿Qué es el Reconocimiento de Caracteres?

El Reconocimiento Óptico de Caracteres (OCR por sus siglas en inglés); también referido como reconocimiento de texto; extrae y utiliza los datos escaneados de documentos, imágenes de cámara y archivos en .pdf de imágenes. El software de OCR aísla las letras de la imagen, las ordena en palabras y las palabras en oraciones, así permitiendo el acceso y edición del contenido original. También elimina la necesidad del ingreso manual de datos.

En 1974, Ray Kurzweil fundó Kurzweil Computer Products, Inc., cuyo producto OCR (reconocimiento óptico de caracteres Omni-Font) podía reconocer texto impreso en prácticamente cualquier fuente. Decidió que la mejor aplicación de esta tecnología sería un dispositivo de aprendizaje automático para ciegos, por lo que creó una máquina de lectura que pudiera leer texto

en voz alta en un formato de texto a voz. (SuperAnnotate. (2021, octubre 29). *What is Optical Character Recognition (OCR): Overview and use cases*. SuperAnnotate Blog. <https://blog.superannotate.com/ocr-overview-and-use-cases/>)

En 1980, Kurzweil vendió su empresa a Xerox, que estaba interesada en comercializar aún más la conversión de texto de papel a computadora. La tecnología OCR se hizo popular a principios de la década de 1990 cuando se digitalizaron periódicos históricos. Antes de que la tecnología OCR estuviera disponible, la única forma de formatear documentos digitalmente era volver a escribir manualmente el texto. Esto no solo requería mucho tiempo, sino que también implicaba inexactitudes y errores tipográficos inevitables.

Desde entonces, las soluciones actuales son capaces de ofrecer una precisión de OCR casi perfecta. Se utilizan métodos avanzados para automatizar flujos de trabajo de procesamiento de documentos complejos y los servicios de OCR están ampliamente disponibles para el público.

Los sistemas OCR utilizan una combinación de hardware y software para convertir documentos físicos impresos en texto legible por máquina. El hardware, como un escáner óptico o una placa de circuito especial, copia o lee texto; entonces el software normalmente maneja el procesamiento extendido. El software OCR puede usar inteligencia artificial para implementar métodos más avanzados de reconocimiento inteligente de caracteres (ICR), como la identificación de idiomas o estilos de escritura.

El proceso OCR se usa más comúnmente para convertir documentos legales o históricos en papel en documentos PDF, lo que permite a los usuarios editar, formatear y buscar los documentos como si hubieran sido creados con un procesador de textos. (*What is optical character recognition (OCR)?* (2022, enero 5). Ibm.com. <https://www.ibm.com/cloud/blog/optical-character-recognition>)

El reconocimiento óptico de caracteres (OCR) utiliza un escáner para procesar la forma física de un documento. Una vez que se copian todas las páginas, el software OCR convierte el documento en una versión en dos colores o en blanco y negro. La imagen escaneada o el mapa de bits se analiza en busca de áreas claras y oscuras y las áreas oscuras se identifican como caracteres reconocibles, mientras que las áreas claras se identifican como fondo. Luego, las áreas oscuras se procesan para encontrar letras o números alfabéticos. Esta fase generalmente se enfoca en un carácter, palabra o bloque de texto a la vez. Luego, los caracteres se identifican utilizando uno de dos algoritmos: reconocimiento de patrones o reconocimiento de características.

La coincidencia de patrones se utiliza cuando se envían muestras de texto en diferentes fuentes y formatos al programa OCR para comparar y reconocer caracteres en el documento escaneado o archivo de imagen.

El reconocimiento de características se produce cuando el OCR aplica reglas relacionadas con las características de una letra o número específico para reconocer caracteres en el documento escaneado. Las características incluyen el número de líneas en ángulo, líneas cruzadas o curvas en un carácter. Cuando se identifica un carácter, se convierte en un código ASCII (Código estándar

estadounidense para el intercambio de información), que los sistemas informáticos utilizan para realizar más manipulaciones. Por ejemplo, la letra "A" mayúscula se almacena como dos líneas diagonales que se encuentran con una línea horizontal en el medio.

Un programa de OCR también analiza la estructura de la imagen de un documento. Divide la página en elementos como bloques de texto, tablas o imágenes. Las líneas se dividen en palabras y luego en caracteres. Una vez que se han elegido los personajes, el programa los compara con una serie de imágenes de muestra. Después de procesar todas las coincidencias probables, el programa le presentará el texto reconocido.

El caso de uso más conocido para el reconocimiento óptico de caracteres OCR es la conversión de documentos impresos en papel en documentos de texto legibles por máquina. Una vez que un documento en papel escaneado ha pasado por el procesamiento de OCR, el texto del documento se puede editar con un procesador de textos como Microsoft Word o Google Docs.

OCR se usa a menudo como una tecnología oculta que impulsa muchos sistemas y servicios conocidos en nuestra vida diaria. Los casos de uso importantes para la tecnología OCR incluyen la automatización del ingreso de datos, la asistencia a personas ciegas y con problemas de visión, y la indexación de documentos para motores de búsqueda, como pasaportes, matrículas, facturas, extractos bancarios, tarjetas de visita y reconocimiento automático de matrículas. (*What is OCR?* - *The Ultimate Guide to OCR 2022*. (2022, Abril 26). Klippa. <https://www.klippa.com/en/blog/information/what-is-ocr/>)

OCR permite la optimización del modelado de big data al convertir documentos en papel e imágenes escaneadas en archivos PDF legibles por máquina y con capacidad de búsqueda. El procesamiento y la recuperación de información valiosa no se pueden automatizar sin aplicar primero OCR a los documentos que aún no tienen capas de texto.

Con el reconocimiento de texto OCR, los documentos escaneados se pueden integrar en un sistema de big data que ahora puede leer datos de clientes de extractos bancarios, contratos y otros documentos impresos importantes. En lugar de hacer que los empleados revisen innumerables documentos de imágenes, y luego ingrese manualmente la entrada en un flujo de trabajo de procesamiento de big data automatizado, las empresas pueden usar OCR para automatizar la fase de entrada de la extracción de datos. El software OCR puede identificar el texto en la imagen, extraer texto en imágenes, guardar el archivo de texto y admitir jpg, jpeg, png, bmp, tiff, pdf y otros formatos.

El proceso de Reconocimiento Óptico de Caracteres en términos generales sigue las siguientes fases:

Preprocesamiento. - El componente de hardware, sea una cámara o escáner, digitaliza la imagen, en esta digitalización la tecnología de OCR define las áreas que son de interés que contengan texto, considerando las áreas vacías como nulas, este proceso es referido como convertir imágenes en fondos (áreas en blanco) y caracteres (áreas en oscuro). En el paso de preprocesamiento, el motor de OCR comprueba automáticamente si hay errores y corrige los problemas.

Las técnicas para mejorar las imágenes o los documentos escaneados comúnmente utilizadas son:

- Enderezar. - El proceso de enderezar y corregir el ángulo de una fotografía o documento escaneado.
- Binarización. - El proceso de convertir una imagen o un documento escaneado a blanco y negro. La binarización permite una forma más precisa de separar el texto del fondo.
- Zonificación. - También conocido como análisis de diseño, se utiliza para identificar columnas, filas, bloques, etiquetas, párrafos, tablas y otros elementos.
- Normalización. - El proceso de reducción del ruido ajustando el valor de intensidad de los píxeles a los valores promedio de los píxeles circundantes.

Segmentación. - La segmentación es el proceso de reconocer el texto una línea a la vez. La segmentación incluye los siguientes pasos:

- Reconocimiento de palabras y líneas de texto. - Se refiere al reconocimiento de las líneas de texto y sus palabras asociadas.
- Reconocimiento de guiones. - El proceso de identificación de guiones en función de documentos, páginas, líneas de texto, párrafos, palabras y caracteres.

Reconocimiento de Caracteres. - En este paso, una imagen o documento se divide en partes, secciones o zonas. Una vez realizada la separación, se reconocen los caracteres que contiene. Una vez que los fondos y los caracteres han sido separados, es posible iniciar el proceso de identificación del contenido específico de los caracteres.

Existen 2 formas de resolver este paso:

- Coincidencia de matrices. - El proceso de comparar cada carácter con una biblioteca de matrices de caracteres. El modelo OCR completa una comparación píxel por píxel para hacer coincidir una imagen de un carácter con su carácter correspondiente.
- Reconocimiento de características. - El proceso de reconocer patrones de texto y características de los caracteres de imágenes. Por ejemplo, el tamaño, la altura, la forma, las líneas y la textura de un carácter se comparan con los de la biblioteca existente.

Post Procesamiento posterior del Resultado. - Este paso tiene que ver con las técnicas y algoritmos que mejoran la precisión de la extracción de datos para obtener un resultado óptimo, los datos se registran para luego corregirlos de ser necesario. Los datos extraídos se comparan con una biblioteca de vocabulario o caracteres para realizar comprobaciones gramaticales y consideraciones contextuales para completar la fase de postprocesamiento.

Este proceso tiene sus limitantes, ya que originalmente OCR fue creado para que los ciegos convirtieran los caracteres impresos en voz. Más tarde, la tecnología se utilizó para leer y reconocer texto negro sobre un fondo blanco. El OCR tradicional no fue pensado como una solución para la

extracción dinámica de datos. Estas limitantes se pueden presentar de diferentes maneras en los resultados.

Los problemas que comúnmente son identificados son:

- Dependencia de la calidad de entrada. - La calidad en el reconocimiento de caracteres y la precisión de la extracción de texto es dependiente directamente de la calidad de entrada de la imagen que se ha enviado al motor. Tomando el caso de un carácter cuya altura sea inferior a 20 píxeles, por ejemplo, la precisión cae drásticamente.
- Depende de las Plantillas y Reglas. - El OCR tradicional requiere plantillas y reglas para ejecutarse. Se deben establecer reglas estrictas programando el motor para recopilar datos de los campos y filas correctas. Debido a esto cuando se presentan con documentos no estructurados no puede hacer frente a la variedad de documentos.
- Costoso. - A medida que se necesitan desarrollar más reglas y algoritmos para aumentar la precisión, el OCR tradicional puede volverse muy costoso. Además, la creación de estas reglas y algoritmos no siempre garantiza un resultado de alta calidad, ya que también depende de la calidad de entrada de la imagen.
- Poca adaptabilidad a una gran variedad de documentos. - Con el OCR tradicional, el resultado generalmente es muy preciso cuando los documentos son sencillos y tienen pocas variaciones presentes. Sin embargo, muchas empresas necesitan procesar varios documentos en sus tareas de trabajo. Cuanto mayor es el volumen y la variedad de documentos que se requieren

ser procesados, más exigentes los recursos necesarios para el proceso se vuelve. Dado que el motor de OCR tradicional se entrena con plantillas, no puede mantenerse actualizado con una gran variedad de documentos nuevos o con formatos modificados o nuevos.

- Falta de automatización. - Debido a su dependencia de plantillas y reglas, el OCR tradicional carece de muchas opciones de automatización. Por ejemplo, si desea extraer datos estructurados de las facturas, cada campo de datos específico requerirá una nueva regla. Las facturas vienen en diferentes estilos y formatos, lo que llevará a numerosas reglas. Agregar más reglas requiere de más datos y recursos para entrenar el motor OCR. Con el enfoque tradicional puede convertirse en un serio cuello de botella ya que es necesario establecer cada vez más reglas.

La solución para estos problemas fue la integración de Inteligencia Artificial (AI) y Aprendizaje Artificial (Machine Learning). Esto permite a las empresas y usuarios lograr la automatización. Esta tecnología revolucionaria también se conoce como procesamiento inteligente de documentos (IDP o Intelligent Document Processing). (*What is optical character recognition (OCR)?* (2022, enero 5). Ibm.com. <https://www.ibm.com/cloud/blog/optical-character-recognition>)

2.6. Inteligencia Artificial

¿De acuerdo a John McCarthy en su paper ‘Que es la Inteligencia Artificial?’ de 2004, define a la Inteligencia Artificial como: “Es la ciencia y la ingeniería de fabricar máquinas inteligentes, especialmente programas informáticos inteligentes. Está relacionado con la tarea

similar de usar computadoras para comprender la inteligencia humana, pero la IA no tiene que limitarse a métodos que sean biológicamente observables”.

Sin embargo, varias décadas antes de esta definición, la conversación sobre la inteligencia artificial comenzó con Alan Turing; considerado como el "padre de la informática"; en su ensayo ‘Maquinaria de Computación e Inteligencia’ de 1950, planteaba la pregunta: "¿Pueden pensar las máquinas?" Se ofrece una prueba conocida como ‘La Prueba de Turing’, donde un interrogador humano trataría de distinguir entre una respuesta de una computadora y una respuesta de texto humana. Aunque esta prueba ha recibido mucho estudio desde su lanzamiento, sigue siendo una parte importante de la historia de la Inteligencia Artificial.

Uno de los principales libros de texto sobre Inteligencia Artificial es titulado “Inteligencia artificial: un Enfoque Moderno” escrito por Stuart Russell y Peter Norvig, en el que analizan cuatro posibles objetivos o definiciones de la Inteligencia Artificial, que se distinguen los sistemas informáticos en dos categorías, según el enfoque utilizado:

- Por el enfoque humano: Sistemas que piensan como humanos y Sistemas que se comportan como humanos.
- Por el enfoque ideal: Sistemas que piensan racionalmente y Sistemas que actúan racionalmente.

La definición dada por Alan Turing entraría en la categoría de sistemas que actúan como humanos. En su conceptualización más sencilla, la Inteligencia Artificial es un campo de la informática combinado con conjuntos de datos sólidos para la resolución de problemas.

Los sistemas expertos, una de las primeras aplicaciones exitosas de la Inteligencia Artificial, tenían como objetivo copiar el proceso que un ser humano utiliza para la toma de decisiones. En aquel momento, requería mucho tiempo extraer y codificar los conocimientos de las personas. (IBM Cloud Education. (2020a, Junio 3). *What is Artificial Intelligence (AI)?* Ibm.com. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>)

2.7. Tipos de Inteligencia Artificial

2.7.1. Inteligencia Artificial Débil

También conocida como Inteligencia Artificial Estrecha (ANI o Artificial Narrow Intelligence), es una Inteligencia Artificial entrenada para realizar tareas específicas. En lugar de ‘Débil’ (‘Weak’ en inglés), ‘Estrecha’ podría ser una descripción más precisa para este tipo de Inteligencia Artificial, ya que débil sería una subestimación de las capacidades de este tipo de Inteligencia Artificial; Habilita algunas de las aplicaciones más poderosas y populares entre los usuarios en el mundo, como por ejemplo Watson de IBM, Siri de la compañía Apple, Alexa de compañía Amazon, y en modelos de Inteligencia Artificial para vehículos autónomos.

La Inteligencia Artificial Débil integra la mayoría de la Inteligencia Artificial que nos rodea en el mundo hoy.

2.7.2. Inteligencia Artificial Fuerte

La Inteligencia Artificial Fuerte consiste en Inteligencia General Artificial (AGI) y Súper Inteligencia Artificial (ASI). La Inteligencia General Artificial (AGI o ‘Artificial General Intelligence’) es una forma teórica de Inteligencia Artificial en la que una máquina tendría una inteligencia igual a la de los humanos; tendría una conciencia con la capacidad de resolver problemas, ser capaz de aprender y realizar planificación para el futuro.

La Súper Inteligencia Artificial (ASI), también conocida como superinteligencia, superaría la inteligencia y la capacidad del cerebro del ser humano. Si bien la Inteligencia Artificial Fuerte todavía es completamente teórica y no hay ejemplos prácticos en la actualidad, los investigadores de Inteligencia Artificial están explorando su desarrollo.

2.7.3. Inteligencia Artificial para la automatización

Con la Inteligencia Artificial integrada en el software de Reconocimiento Óptico de Caracteres puede adaptarse y aprender constantemente para reconocer datos con una mejor precisión. Puede crear y aprovechar una comprensión más profunda de la semántica y ampliar la gama de idiomas disponibles, formatos de documentos, diseños en los que son presentados y tipos de documentos admitidos para su procesamiento.

La Inteligencia Artificial permite que el software o sistema de Reconocimiento Óptico de Caracteres analice los datos disponibles, encuentre e identifique correlaciones y pueda crear bases de conocimiento con la información.

La base de conocimientos creada por Inteligencia Artificial puede adaptarse con el tiempo para ayudarle a mejorar la precisión de la extracción de los datos. Lo mejor de la IA es que replica las habilidades humanas para escanear y comprender información clave con alta velocidad y precisión.

2.8. Aprendizaje Artificial

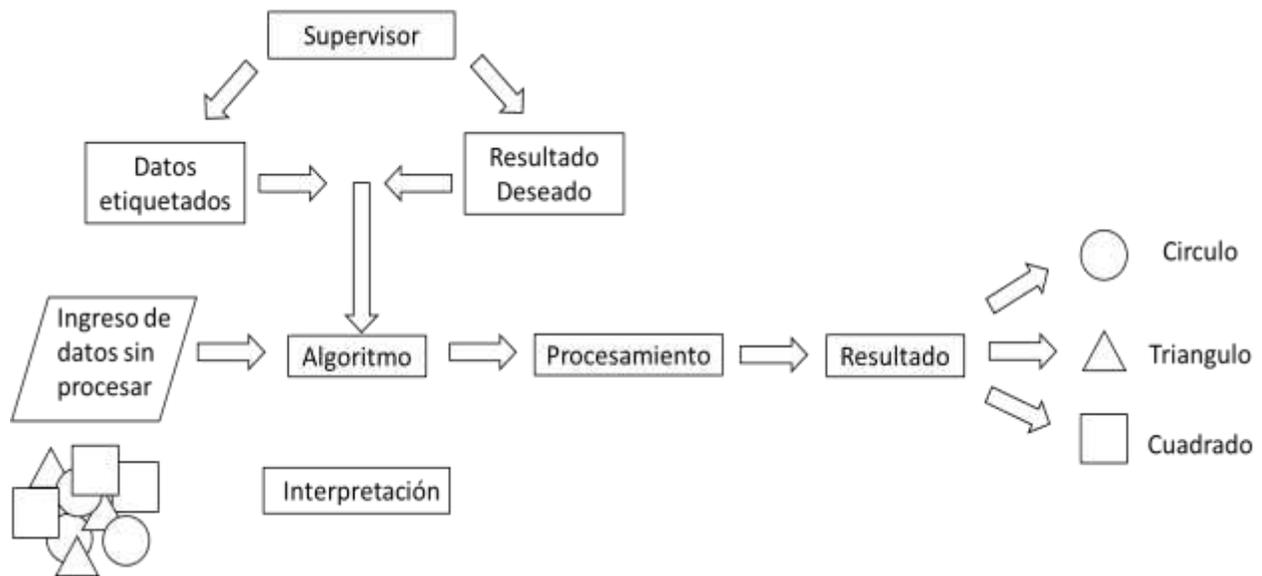
El software OCR integrado en el aprendizaje automático (Machine Learning) se puede entrenar para reconocer patrones y el significado del contenido a través de un conjunto de reglas establecidas. Esto se puede hacer a través del aprendizaje supervisado, el aprendizaje no supervisado o la combinación de ambos métodos de entrenamiento en inteligencia artificial.

2.8.1. Aprendizaje Artificial Supervisado

El Aprendizaje Supervisado en Aprendizaje Artificial ('Machine Learning' en inglés) se refiere al uso de conjuntos de datos etiquetados para entrenar algoritmos que clasifican datos y predicen resultados con alta precisión. Para lograr esto, el modelo tiene que ser alimentado con una gran cantidad de datos de entrada.

Figura 2

Representación del proceso de Aprendizaje Artificial Supervisado



Nota. Este diagrama representa la clasificación a través de inteligencia artificial de figuras geométricas en 3 categorías: círculo, triángulo y cuadrado. En este caso un usuario supervisa el aprendizaje del modelo al proveer datos etiquetados y los resultados esperados para ser comprobados con los obtenidos con el propósito de aumentar la precisión de la clasificación. Recreado de Do, T. (2022, marzo 21). *Supervised and unsupervised machine learning - explained through real world examples*. Omdena. <https://omdena.com/blog/supervised-and-unsupervised-machine-learning/>.

2.8.2. Aprendizaje Artificial No Supervisado

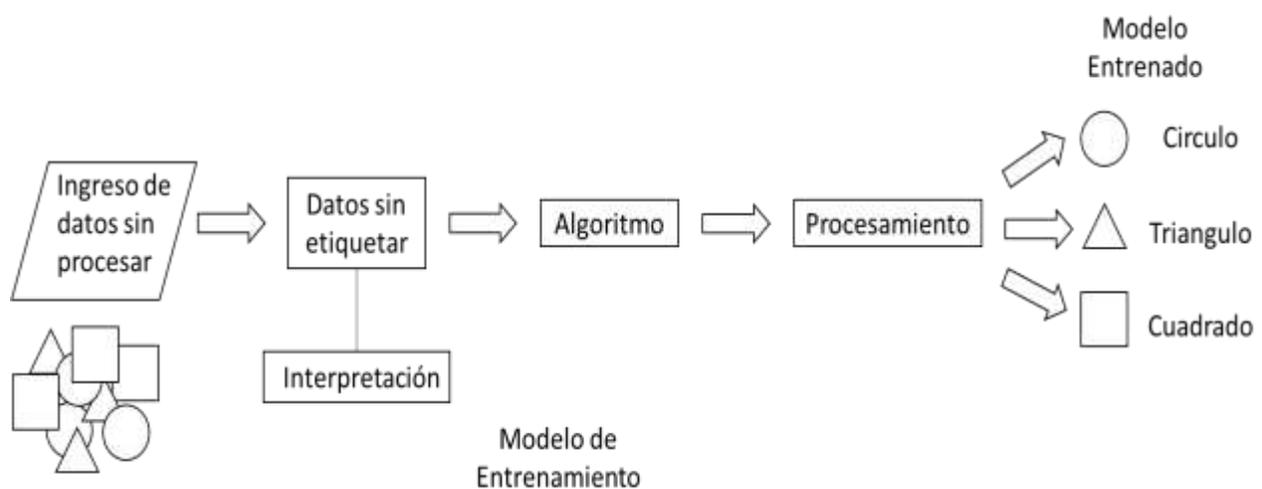
El aprendizaje no supervisado concretamente es similar al aprendizaje supervisado. La diferencia entre ambos es que en el aprendizaje no supervisado utiliza datos no etiquetados en lugar de datos previamente etiquetados. Este enfoque tiene mayor utilidad cuando existe dificultad

para identificar las características comunes dentro de un conjunto de datos, dándole mayor libertad al modelo.

Incluso si las etiquetas para los datos no están definidas, los puntos de datos reales permanecen, y de esta manera el modelo puede reconocer patrones al observar los datos de entrada. En resumen, el aprendizaje no supervisado puede replicar las habilidades humanas del proceso de aprendizaje y adaptación.

Figura 3

Representación del proceso de Aprendizaje Artificial No Supervisado



Nota. Este diagrama representa la clasificación a través de inteligencia artificial de figuras geométricas en 3 categorías: círculo, triángulo y cuadrado. A diferencia del caso anterior, no requiere la supervisión del aprendizaje de la red neuronal por un supervisor, y se deja que el modelo encuentre por sí mismo ejemplos y detecte por sí mismo datos no encontrados previamente. Recreado de Do, T. (2022, marzo 21). *Supervised and unsupervised machine learning - explained through real world examples*. Omdena. <https://omdena.com/blog/supervised-and-unsupervised-machine-learning/>.

2.8.3. Aprendizaje Semi-Supervisado

En este método, los datos de entrada en el aprendizaje semi supervisado están etiquetados y no etiquetados. En la mayoría de los casos este método se usa para resolver problemas de extracción de datos cuando se trata de grandes volúmenes de datos.

Debido a que el aprendizaje semi supervisado utiliza características de ambos métodos mencionados anteriormente, esto le ayuda a superar los desafíos que pueden presentarse en ambos enfoques: clasificación, tiempo, coste y big data.

Es preferible utilizar este método para casos en los que una pequeña cantidad de datos utilizados para el entrenamiento puede producir resultados notablemente más precisos.

(Do, T. (2022, marzo 21). *Supervised and unsupervised machine learning - explained through real world examples*. Omdena | Building AI Solutions for Real-World Problems; Omdena. <https://omdena.com/blog/supervised-and-unsupervised-machine-learning/>)

2.9. Beneficios del Reconocimiento Óptico de Caracteres Avanzado

Gracias al uso de Inteligencia Artificial y Aprendizaje Artificial se resuelven varios problemas encontrados en el Reconocimiento Óptico de Caracteres tradicional que se encontraban y permite nuevas funcionalidades.

Estas son algunas de las ventajas que conllevan la integración de estas tecnologías:

- Digitalización rápida de documentos. - Con el software de Reconocimiento Óptico de Caracteres, se puede dejar de usar papel y extraer datos en unos pocos segundos de documentos en formatos digitales como PDF, JSON, CSV, XML, etc.
- Menor tiempo necesario para la implementación. - Las soluciones de Reconocimiento Óptico de Caracteres más avanzadas no se basan únicamente en reglas y plantillas. Por lo tanto, lleva menos tiempo entrenar el motor e implementar la tecnología.
- Escalabilidad. - El Reconocimiento Óptico de Caracteres en la nube ofrece una escalabilidad que su predecesor tradicional carecía. El Reconocimiento Óptico De Caracteres tradicional puede adaptarse a través de un proceso basado en plantillas, pero puede volverse prohibitivo para el uso de las empresas.
- Mayor precisión. - Mientras que el Reconocimiento Óptico de Caracteres tradicional tiene una precisión de extracción de datos relativa del 60 % al 85 % aproximadamente, muchas soluciones más avanzadas que integran las tecnologías de inteligencia artificial y el aprendizaje automático, pueden alcanzar hasta el 99 % de precisión. Si bien la extracción manual de datos proporciona una precisión del 90 % al 95 %, es mucho más lenta e ineficiente para muchas organizaciones.

- Reducción de errores de entrada manual. - Los errores son comunes cuando las personas trabajan en tareas con tendencia a volverse tediosas y repetitivas, como al ingresar datos manualmente. El Reconocimiento Óptico de Caracteres puede automatizar estas tareas reduciendo los errores humanos de entrada manual de datos, la tasa de error se puede reducir aún más con Inteligencia Artificial y Aprendizaje Automático.
- Tiempo de respuesta reducido. - Los flujos de trabajo de procesamiento de documentos tradicionales muchas de las veces tienen tareas lentas y engorrosas que crean cuellos de botella que terminan siendo costosos. La revisión y extracción manual de datos puede llevar un aproximado de entre 10 a 20 minutos por documento digitalizado, mientras que el Reconocimiento Óptico de Caracteres tradicional puede hacerlo en menos de la mitad del tiempo. Soluciones de Reconocimiento Óptico de Caracteres que integran Inteligencia Artificial y Aprendizaje Artificial pueden ahorrar el 98% del tiempo.
- Reducción de costos: debido a que el Reconocimiento Óptico de Caracteres que integran Inteligencia Artificial y Aprendizaje Artificial permite tiempos de respuesta más rápidos, automatiza tareas tediosas y minimiza los errores al ingresar datos y los gastos se reducen, esto permite la reducción de costes convirtiéndose en una de las principales ventajas para las empresas.
- Detección de falsificaciones. - Un Reconocimiento Óptico de Caracteres más avanzado puede detectar fraudes a través del análisis de imágenes y EXIF, Pudiendo evitar la pérdida de capital por fraude interno y externo.

- Experiencia del cliente mejorada. - Existen muchos casos comerciales en los que la integración de Inteligencia Artificial en el Reconocimiento Óptico de Caracteres ayuda a mejorar la experiencia del cliente. Uno de estos casos a tomar en cuenta es cuando los bancos al incorporar nuevos clientes lo hagan de manera más fluida y ágil a través de la integración de la tecnología móvil.

2.10. Procesamiento Inteligente de Documentos

El IDP puede ofrecer mejores resultados que los obtenidos a través de las capacidades humanas cuando se consideran la eficiencia y el tiempo. Hace que los datos sean significativos, categoriza, organiza y convierte automáticamente los datos para el usuario en una ventana de tiempo menor a la que se esperaría con varias personas enfocadas en la misma tarea. Uno de los avances más importantes es que no se limita a plantillas o reglas como su antecedente tradicional permitiendo que el software OCR impulsado por IA sea más escalable y asequible para las empresas y particulares.

Del reconocimiento de caracteres se han derivado en diferentes ámbitos de trabajo, en el caso del reconocimiento de placas vehiculares, llamado en inglés License Plate Recognition o por sus siglas en inglés LPR, se han aplicado en diferentes áreas y ha progresado significativamente, las herramientas y métodos disponibles para realizar el reconocimiento de placas vehiculares son muy diversas, utilizables para diferentes necesidades y preferencias, en su aplicación en placas vehiculares algunos de los casos son el control vehicular de tráfico en ciudades, monitoreo de vehículos que ingresan o salen de estacionamientos privados o estatales, entradas a parqueaderos

de centros comerciales o negocios, peajes nacionales o internacionales, sistemas de fotorradars de controles de velocidad, etc.

Los propósitos para el uso de estas son igualmente diversos, como controles de acceso vehicular, haciendo uso de listados de vehículos autorizados o no autorizados para su ingreso en zonas o parqueaderos, control de cumplimiento en la asignación de estacionamientos de vehículos, control vehicular de infracciones de tránsito, sean de exceso de velocidad, cambios de carril, giros o conducción en contravía, detección de placas de vehículos con infracciones, de vehículos robados o cuya matrícula esté vencida. (*What is intelligent document processing & how does it work?* (s/f). ABBYY. Recuperado el 15 de agosto de 2022, de <https://www.abbyy.com/blog/intelligent-document-processing/>)

2.11. Aprendizaje Profundo

El aprendizaje profundo es un subconjunto del aprendizaje automático que maneja grandes cantidades de datos utilizando algoritmos inspirados en la estructura y función del cerebro humano, razón por la cual los modelos de aprendizaje profundo a menudo se denominan redes neuronales profundas. Es parte de una familia más amplia de métodos de aprendizaje automático basados en representaciones de datos de aprendizaje, a diferencia de los algoritmos tradicionales específicos de tareas.

En el Aprendizaje Profundo, un modelo informático aprende a realizar tareas de clasificación directamente a partir de imágenes, textos o sonidos, ejecutando una tarea

repetidamente y realizando ajustes para mejorar el resultado obtenido. Los modelos de aprendizaje profundo pueden superar el rendimiento obtenido a nivel humano. Los modelos se entrenan utilizando una gran cantidad de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas.

La parte más importante de una red neuronal de aprendizaje profundo es una capa de nodos computacionales llamados "neuronas". Cada neurona se conecta a todas las neuronas en la capa inferior. A través del Aprendizaje Profundo, la red neuronal utiliza al menos dos capas ocultas. Agregar las capas ocultas permite a los investigadores realizar cálculos más profundos.

El Algoritmo funciona gracias a que cada conexión tiene su peso o importancia. Pero con la ayuda de redes neuronales profundas, podemos encontrar automáticamente las características más importantes para la clasificación. Esto se hace mediante la función de activación, que evalúa cómo debe ir la señal para cada neurona, al igual que en el caso de un cerebro humano.

El paradigma informático de Aprendizaje Profundo se considera un estándar para la comunidad de aprendizaje automático en la actualidad. Además, se ha convertido gradualmente en el enfoque computacional más utilizado en el campo de Aprendizaje Artificial, logrando excelentes resultados en múltiples tareas cognitivas complejas que igualan o incluso superan las proporcionadas por el desempeño humano. Uno de los beneficios de Aprendizaje Profundo es la capacidad de aprender cantidades masivas de datos.

Algunas de las razones por las que el aprendizaje profundo se ha convertido en el estándar de la industria son:

- Manejo de datos no estructurados. - los modelos capacitados en datos estructurados pueden aprender fácilmente de los datos no estructurados, lo que reduce el tiempo y los recursos en la estandarización de conjuntos de datos.
- Manejo de Big Data. - Con la llegada de las unidades de procesamiento de gráficos (GPU), los modelos de aprendizaje profundo pueden procesar grandes cantidades de datos a la velocidad del rayo.
- Alta precisión. - Los modelos de aprendizaje profundo brindan los resultados más precisos en visión artificial, procesamiento de lenguaje natural (NLP) y procesamiento de audio.
- Reconocimiento de patrones. - La mayoría de los modelos requieren la intervención de un ingeniero de aprendizaje automático, pero los modelos de aprendizaje profundo pueden reconocer todo tipo de patrones automáticamente.

El área de Aprendizaje Profundo ha crecido rápidamente en los últimos años y se ha utilizado ampliamente para abordar una amplia gama de aplicaciones tradicionales con éxito. Más importante aún, Aprendizaje Profundo ha superado las conocidas técnicas de Aprendizaje Artificial en muchas áreas, como por ejemplo en ciberseguridad, procesamiento de lenguaje natural, bioinformática, robótica y control, procesamiento de información en el campo de la

medicina, entre muchos otros. Recientemente, el Aprendizaje Artificial se ha generalizado en la investigación y se ha integrado en una variedad de aplicaciones que incluyen minería de texto, detección de spam, recomendación de video, clasificación de imágenes y recuperación de conceptos multimedia. (Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>)

Entre los diversos algoritmos de Aprendizaje Artificial, el Aprendizaje Profundo es muy utilizado en estas aplicaciones. Otro nombre para Aprendizaje Profundo es Aprendizaje de Representación. Debido al crecimiento impredecible en la capacidad de obtener datos como también a los sorprendentes avances en tecnologías de hardware como Cómputo de Alto Rendimiento (High Performance Computing o HPC) han surgido una corriente de continuos nuevos estudios en los campos del aprendizaje profundo y distribuido.

El Aprendizaje Profundo se deriva de la red neuronal tradicional, pero supera significativamente a sus predecesores. Además, el Aprendizaje Profundo usa simultáneamente transformaciones y tecnologías gráficas para construir modelos de aprendizaje de varias capas. Las técnicas del Aprendizaje Profundo recientemente desarrolladas han logrado un rendimiento excelente en una gran variedad de aplicaciones que incluyen; por nombrar algunas; el procesamiento de lenguaje natural (Natural Language Processing o NLP), el procesamiento de audio y de voz, computación visual, etc.

Por lo general, la eficacia de un algoritmo de Aprendizaje Artificial depende en gran medida de la integridad de la representación de los datos de entrada utilizados. Se ha demostrado que una buena representación de datos proporciona un rendimiento mejorado en comparación con una mala representación de datos. Por esto, la Ingeniería de Características ha sido una importante tendencia de investigación en Aprendizaje Artificial durante muchos años, influyendo en numerosos estudios de investigación.

Este enfoque tiene como objetivo construir características a partir de datos sin procesar, además de que es extremadamente específico del tema y muchas de las veces requiere un esfuerzo humano significativo. Por ejemplo, se han introducido y comparado diversos tipos de funciones en el contexto de la visión artificial, un caso de esto sería una vez que se introduce y aprueba una característica novedosa, se convierte en una nueva dirección de investigación que se persigue durante varias décadas.

2.12. ¿Qué es OpenCV?

OpenCV es una biblioteca de software de aprendizaje automático y visión artificial de código abierto. OpenCV es una biblioteca de visión artificial de código abierto que se desarrolló para proporcionar una infraestructura común para aplicaciones de visión artificial y para acelerar el uso de la visión artificial en productos comerciales.

OpenCV está bajo la licencia BSD, una licencia de software libre permisiva que tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio

público. La licencia BSD al contrario que la GPL (Licencia Pública General); que es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto; permite el uso del código fuente en software no libre. La versión original ya se ha revisado y sus variantes son denominadas licencias BSD modificadas. Como producto con licencia BSD, OpenCV facilita el uso y la modificación del código. OpenCV tiene más de 2500 algoritmos optimizados, incluido un conjunto completo de algoritmos de aprendizaje automático y visión artificial, clásicos y de última generación.

Otros usos aplicables de esta tecnología pueden ser unir imágenes para producir imágenes de alta resolución, Tomas de una escena completa, búsqueda de imágenes similares a través de una base de datos de imágenes, eliminación de ojos rojos de tomas de fotos con flash, seguimiento de los movimientos oculares, detección de escenarios y configuración de marcadores para superponer con realidad aumentada, etc. Tiene interfaces C++, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS.

Estos algoritmos se pueden usar para:

- Detección y reconocimiento de rostros. - Un sistema de reconocimiento facial es una tecnología que puede hacer coincidir un rostro humano de una imagen digital o un cuadro de video con una base de datos de rostros. Se utiliza para autenticar a los usuarios a través de los servicios de verificación de identidad y funciona localizando y midiendo las características faciales en función de una imagen específica.

- La identificación de objetos. - La detección de objetos es un campo de la inteligencia artificial, que permite a los sistemas informáticos detectar objetos en imágenes o vídeos. Es usado para detectar diversas clases de objetos, sean humanos, de animales, vehículos o edificaciones en imágenes digitales como fotos o en los fotogramas de un vídeo.
- Clasificación de acciones humanas realizadas en videos. - Es una subdisciplina de la visión artificial, funciona con el objetivo de interpretar los gestos humanos mediante algoritmos matemáticos. Los gestos pueden provenir de cualquier movimiento o estado del cuerpo, pero generalmente provienen de la cara o la mano.
- Rastrear movimientos de la cámara. - El Rastreo de movimientos de la cámara o Egomotion se refiere al movimiento 3D de una cámara dentro de un entorno. En el contexto de la visión artificial, se ocupa de estimar el movimiento de una cámara en relación con una escena fija. La estimación del movimiento propio es importante en las aplicaciones de navegación de robots autónomos.
- Rastrear los objetos en movimiento. - El seguimiento de movimiento, también conocido como seguimiento de video, es el proceso de ubicar un objeto o varios objetos en movimiento a lo largo del tiempo con una cámara.
- Extracción de los modelos 3D de objetos. - Es un método fotogramétrico de imágenes a distancia para estimar estructuras tridimensionales a partir de secuencias de imágenes

bidimensionales que se pueden combinar con señales de movimiento locales. Se estudia en las áreas de visión artificial y percepción visual.

Segmentación y detección de imágenes:

- En imágenes digitales y visión artificial. - La segmentación de imágenes es el proceso de dividir una imagen digital en varios segmentos también conocidos como objetos de imagen que son conjuntos de píxeles. El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en algo que sea más significativo y más fácil de analizar.
- La realidad aumentada. - Es básicamente una experiencia interactiva de un entorno del mundo real en el que los objetos del mundo real se mejoran mediante información perceptiva generada por computadora, a menudo a través de múltiples habilidades sensoriales, ya sea visual, auditiva, háptica, somatosensorial y olfativa.

(OpenCV - overview. (2019, septiembre 23). GeeksforGeeks. <https://www.geeksforgeeks.org/opencv-overview/>)

2.13. ¿Qué es EasyOCR?

EasyOCR es un proyecto de Reconocimiento Óptico de Caracteres producido por Jaided AI, una empresa destinada al desarrollo de proyectos en Inteligencia Artificial fundada en 2020. Su primer proyecto fue EasyOCR, una librería de Reconocimiento Óptico de Caracteres open source o código abierto. Tras el éxito de su proyecto de código abierto, se propusieron llevar la Inteligencia Artificial a un público más amplio proporcionando una plataforma de Inteligencia

Artificial sin código para todos. Su servicio de Reconocimiento Óptico de Caracteres sin código se lanzó en mayo de 2021 para clientes que desean aplicar la tecnología de Reconocimiento Óptico de Caracteres a sus casos de uso específicos.

EasyOCR es una biblioteca de Reconocimiento Óptico de Caracteres basada en Python que extrae el texto de la imagen. Es un proyecto de código abierto con licencia Apache 2.0. En su versión Comunitaria tiene características como soportar cerca de 80 lenguajes, incluidos chino, japonés, coreano y tailandés.

Permite el procesamiento de 30 operaciones de reconocimiento de caracteres diarios en su página web, interfaz web, recibir imágenes en formatos .png, .jpg, .tiff, exportación a archivos json, herramientas avanzadas como “Humano en el Bucle: herramienta de anotaciones”, soporte técnico a través de GitHub en el apartado de Problemas.

Otra de las ventajas es que los foros de GitHub de EasyOCR contienen información de problemas y soluciones encontrados por los usuarios, ejemplos y casos de uso, útiles para encontrar e inferir soluciones al momento de encontrarse con inconvenientes relacionados a la integración de la librería EasyOCR. (Gulati, A. P. (2021, Junio 22). *Text detection from images using EasyOCR: Hands-on guide*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/>)

EasyOCR se implementa utilizando Python y la biblioteca PyTorch. Si se tiene una GPU habilitada para CUDA, la biblioteca de aprendizaje profundo PyTorch subyacente puede aumentar su velocidad de reconocimiento de texto y OCR.

EasyOCR utiliza el modelo de reconocimiento de texto CRAFT como detector, y el modelo de Reconocimiento utilizado por EasyOCR es una Red Neuronal Recurrente Convolutiva.

La Red Neuronal Recurrente Convolutiva incluye la Red Neural Convolutiva (“Convolutional Neural Network” o CNN), seguida de la Red Neuronal Recurrente (“Recurrent Neural Network” o RNN). La red propuesta es similar a la Red Neuronal Recurrente Convolutiva, pero produce resultados mejores y óptimos, particularmente en términos de procesamiento de señales de audio.

La Red Neuronal Recurrente Convolutiva utilizada por EasyOCR utiliza 3 componentes principales: para la extracción de características utilizando Resnet o Redes Neuronales Residuales y la Red Neuronal Convolutiva de uso común VGG (Visual Geometry Group), el etiquetado de secuencia se utiliza la red neuronal de Memoria a Corto Plazo (“Long Short-Time Memory” o LSTM), y para la decodificación se utiliza Clasificación Temporal Conexionista (“Connectionist Temporal Classification” o CTC). (GitHub. 2022. *EasyOCR/README.md at master · JaidedAI/EasyOCR*. <https://github.com/JaidedAI/EasyOCR/blob/master/README.md>)

2.14. PyTorch

PyTorch es una biblioteca de código abierto para programas de Python que facilita la creación de proyectos de aprendizaje profundo. PyTorch enfatiza la flexibilidad y permite que los modelos de aprendizaje profundo se representen en Python idiomático.

PyTorch se basa en una biblioteca de Python y Torch que admite el cálculo de tensores en unidades de procesamiento gráfico. Actualmente es la biblioteca más popular para la comunidad de investigación de aprendizaje profundo e inteligencia artificial.

Fernando Pérez anunció un proyecto derivado de IPython llamado Project Jupyter en 2014. IPython continúa existiendo como un shell de Python y como un kernel para Jupyter, mientras que el cuaderno y otras partes independientes del idioma de IPython han migrado bajo el nombre de Jupyter.

Jupyter es independiente del lenguaje y admite entornos de ejecución (también conocidos como kernels) en varias docenas de lenguajes, incluidos Julia, R, Haskell, Ruby y, por supuesto, Python (a través del kernel de IPython).

En 2015, GitHub y el proyecto Jupyter anunciaron la representación nativa del formato de archivo del cuaderno Jupyter (archivos. ipynb) en la plataforma GitHub. (Jwalapuram, N. (2021, abril 12). *PyTorch library*. Analytics Vidhya.)

2.15. Red Neuronal Convolutacional

2.15.1. Redes Neuronales

Las Redes Neuronales, también conocidas como Redes Neuronales Artificiales (“Artificial Neural Network” o ANN) o Redes Neuronales Simuladas (“Simulated Neural Network” o SNN), son un subconjunto del Aprendizaje Automático y son parte integral de los algoritmos de Aprendizaje Profundo. Su nombre y estructura están inspirados en el funcionamiento del cerebro humano, imitando la forma en que las neuronas biológicas se comunican entre sí.

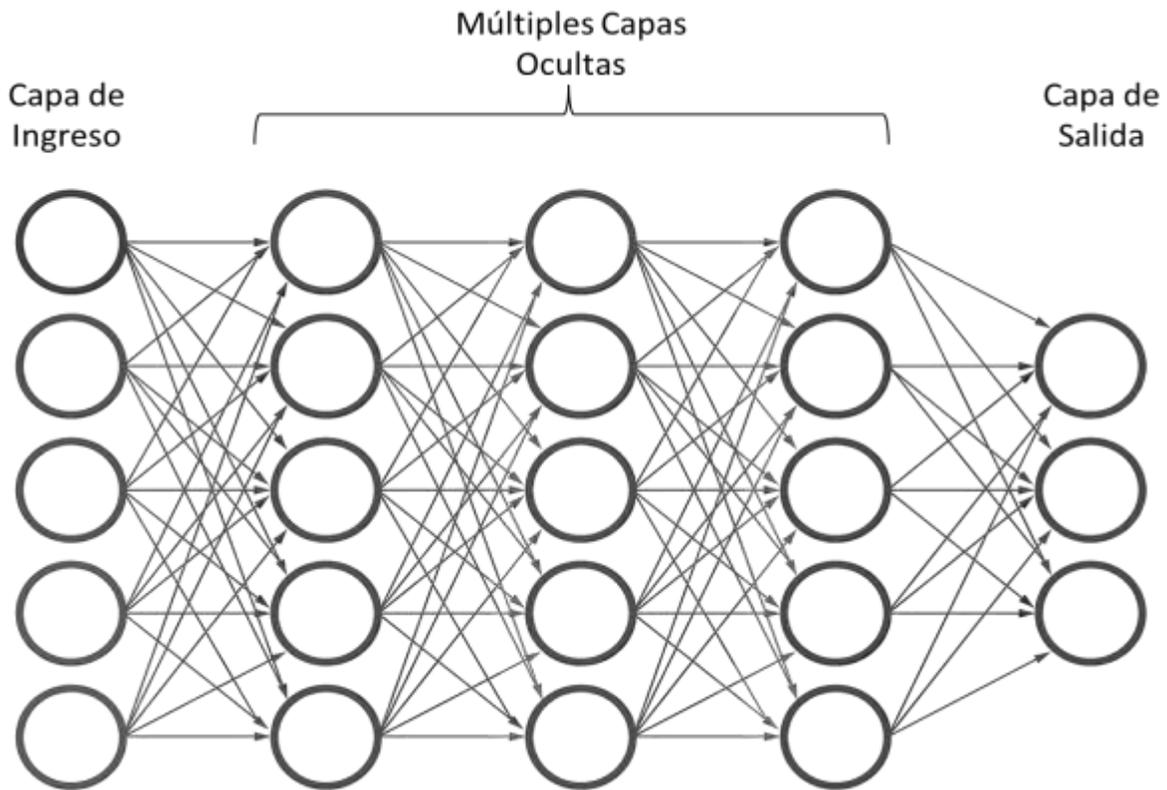
Las Redes Neuronales Artificiales se componen de capas de nodos, que contienen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo o neurona artificial está conectado a otro y tiene asociado un peso y un umbral. Cuando la salida de un nodo individual está por encima del umbral especificado, ese nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, no se pasan datos a la siguiente capa de la red.

Las redes neuronales se basan en datos de entrenamiento para aprender y mejorar su precisión con el tiempo. Sin embargo, una vez ajustados para la precisión, estos algoritmos de aprendizaje son poderosas herramientas que nos permiten clasificar y agrupar datos a alta velocidad en informática e inteligencia artificial.

Las tareas de reconocimiento de voz o reconocimiento de imágenes pueden llevar minutos en lugar de horas en comparación con la identificación manual por parte de expertos humanos. Una de las redes neuronales más conocidas es el algoritmo de búsqueda de Google.

Figura 4

Representación de una Red Neuronal Artificial



A diferencia de la computación tradicional, que procesan datos secuencialmente, las redes neuronales pueden aprender y realizar múltiples tareas. En otras palabras, mientras que las computadoras tradicionales simplemente siguen las instrucciones de su programación, las redes neuronales evolucionan constantemente a través de algoritmos avanzados. Tomado de *What is a neural network? Definition, working, types, and applications in 2022*. por Ashtari, H. 18 de mayo del 2022, Spiceworks.

(<https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-a-neural-network/>)

Se puede decir que las computadoras neuronales se “programan a sí mismas” para derivar soluciones a problemas nunca antes vistos. Además, las computadoras convencionales trabajan

con funciones lógicas basadas en un conjunto específico de cálculos y reglas. Por el contrario, las computadoras neuronales pueden procesar funciones lógicas y entradas sin procesar, como imágenes, videos y voz. Si bien las computadoras tradicionales funcionan de manera inmediata, las redes neuronales deben "entrenarse" con el tiempo para aumentar su precisión y eficiencia. El ajuste para la afinación de estas máquinas de aprendizaje para la precisión es muy productivo, proporcionando a los usuarios una eficiente herramienta informática para aplicaciones informáticas de inteligencia artificial.

2.15.2. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales se diferencian de otras redes neuronales debido a su rendimiento superior en las entradas de señales de imagen, voz o audio.

Se tienen tres tipos principales de capas, a saber:

Capa de convolución. - La capa convolucional es el bloque de construcción central de una Red Neuronal Convolucional, donde se lleva a cabo la mayor parte del cálculo. Requiere algunos componentes, estos serían, datos de entrada, un filtro y un mapa de características. Supongamos que el dato de entrada es una imagen en color compuesta por una matriz de píxeles en tres dimensiones (3D). Esto significa que la entrada tiene tres dimensiones (alto, ancho y profundidad) que corresponden a RGB en una imagen.

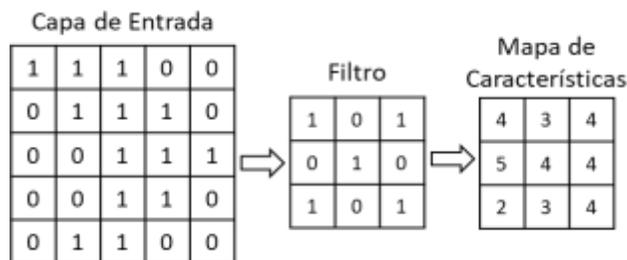
También tenemos un detector de características, también conocido como núcleo o filtro, que se mueve a través de los campos receptivos de la imagen y verifica si la característica está

presente. Este proceso se llama plegado. El detector de características es una matriz bidimensional (2-D) de pesos que representa parte de la imagen. Si bien pueden variar en tamaño, el tamaño del filtro típicamente es una matriz de 3x3; esto también determinará el tamaño del campo receptivo. Luego, el filtro se aplica a un área de la imagen y se calcula un producto escalar entre los píxeles de entrada y el filtro. Este producto escalar luego se introduce en una matriz de salida.

Después de eso, el filtro cambia un paso y repite el proceso hasta que el kernel haya barrido toda la imagen. El resultado final de la serie de productos escalares de la entrada y el filtro se conoce como mapa de características, mapa de activación o característica retro propagación. (Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>)

Figura 5

Ejemplificación del proceso de convolución.



Nota. En esta ejemplificación se muestran los componentes del proceso de convolución, las cuales son un ejemplo de datos ingresados en la capa de entrada de la red neuronal, el proceso de convolución al utilizar un filtro determinado y por último un mapa de características donde los

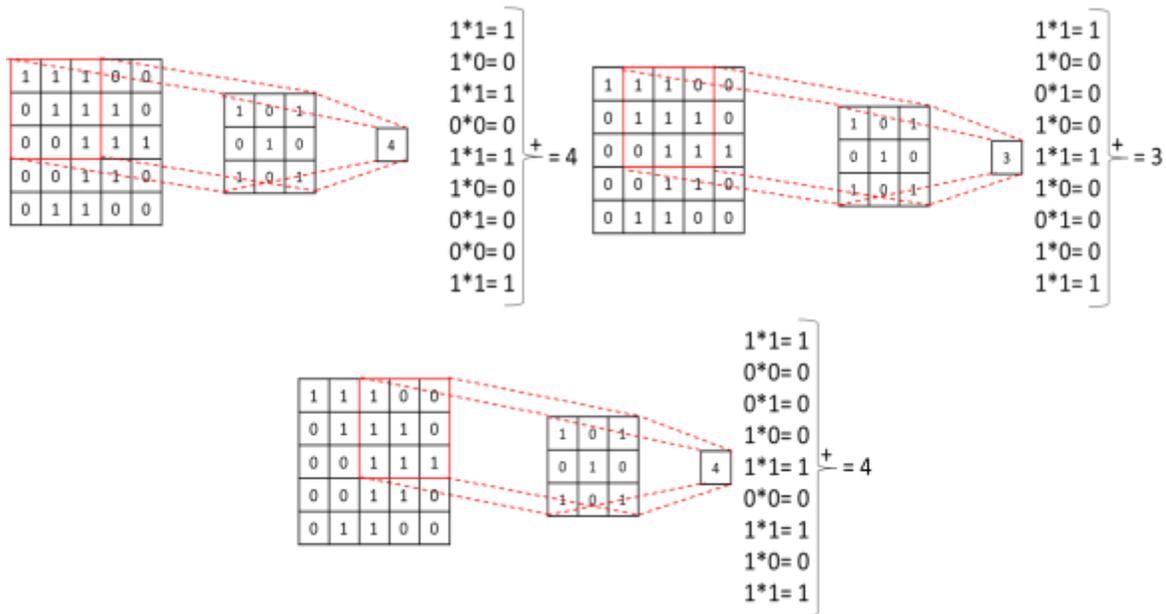
resultados del filtro son registrados para la siguiente capa. Recreado de *What are Convolutional Neural Networks?* IBM Cloud Education. 20 de octubre del 2020, Ibm.com.

(<https://www.ibm.com/cloud/learn/convolutional-neural-networks>)

Su proceso sería tal que:

Figura 6

Representación de la capa convolucional de una red neuronal convolucional 1

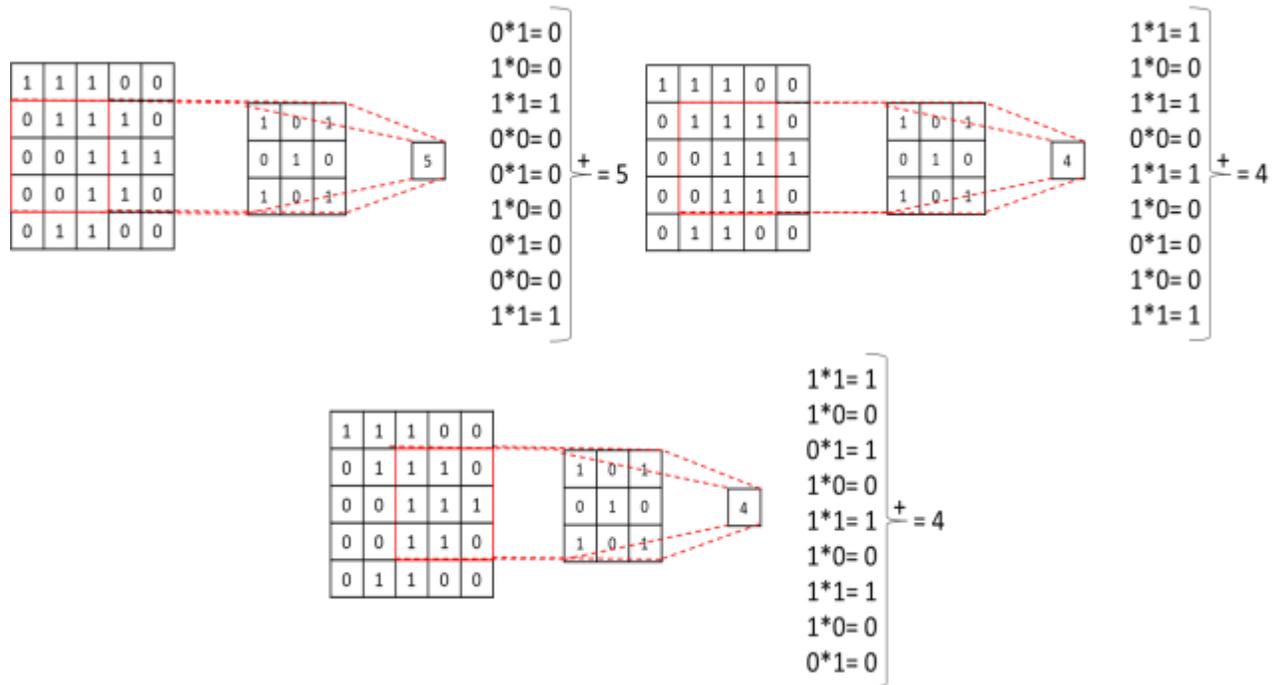


Nota. En este ejemplo detalla el uso del filtro en un dato de una capa de entrada y el registro de los resultados obtenidos por el filtro en su primera barrida por las celdas del dato ingresado. Se aprecia que el filtro de 3x3 realiza su barrido desde la izquierda superior a la derecha en la primera fila de datos según las dimensiones del filtro, obteniendo los tres primeros resultados del filtro. recreado de *Introduction to convolutional neural networks architecture*. Recreado de *Introduction to convolutional neural networks architecture*. 20 de junio del 2022, ProjectPro.

(<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>)

Figura 7

Representación de la capa convolucional de una red neuronal convolucional 2

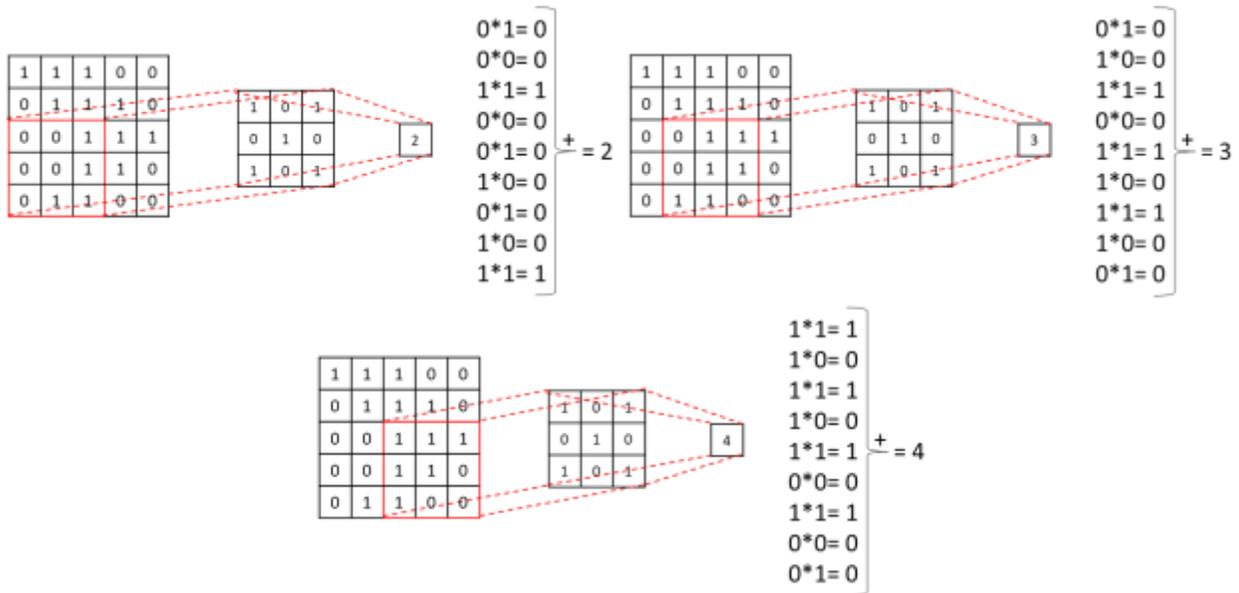


Nota. Nota. En este ejemplo detalla el uso del filtro en un dato de una capa de entrada y el registro de los resultados obtenidos por el filtro en su primera barrida por las celdas del dato ingresado. Se aprecia que el filtro de 3x3 realiza su barrido desde la izquierda superior a la derecha bajo la primera fila de datos según las dimensiones del filtro, obteniendo los tres segundos resultados del filtro. Recreado de *Introduction to convolutional neural networks architecture*. 20 de junio del 2022, ProjectPro.

<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>

Figura 8

Representación de la capa convolucional de una red neuronal convolucional 3



Nota. Nota. En este ejemplo detalla el uso del filtro en un dato de una capa de entrada y el registro de los resultados obtenidos por el filtro en su primera barrida por las celdas del dato ingresado. Se aprecia que el filtro de 3x3 realiza su barrido desde la izquierda superior a la derecha bajo la segunda fila de datos según las dimensiones del filtro, obteniendo los tres últimos resultados del filtro. Recreado de *Introduction to convolutional neural networks architecture*. 20 de junio del 2022, ProjectPro.

(<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>)

No es necesario que todos los valores de salida en el mapa de características necesiten estar conectados a cada valor de píxel en la imagen de entrada. Solo necesita estar conectado al campo receptivo donde se aplica el filtro. Gracias a que la matriz de salida no tiene que asignarse directamente a cada valor de entrada, las capas convolucionales y de agrupación se denominan

comúnmente Capas Parcialmente Conectadas. Sin embargo, esta propiedad también puede denominarse conectividad local. Hay que tener en cuenta que los pesos en el detector de características permanecen fijos a medida que se mueve por la imagen a analizar, lo que también se conoce como uso compartido de parámetros.

Algunos parámetros, como los valores de peso, se ajustan durante el entrenamiento a través del proceso de retro propagación y descenso de gradiente. Sin embargo, hay tres hiper parámetros que afectan el tamaño del volumen de la salida que se deben configurar antes de que comience el entrenamiento de la red neuronal:

El número de filtros. - Afecta la profundidad de la salida. Un caso a tomar, por ejemplo, sería el de tres filtros diferentes que darían como resultado tres mapas de características diferentes, dando una profundidad de tres. Paso. - Es la distancia o el número de píxeles que el núcleo se mueve a través de la matriz de entrada. Mientras los valores de paso de dos o más son raros, un paso más grande da como resultado menos salida.

El relleno cero. - Se usa normalmente cuando los filtros no coinciden con la imagen de entrada. Esto establece todos los elementos fuera de la matriz de entrada en cero, produciendo una salida mayor o igual.

Hay tres tipos de relleno:

- Relleno válido: esto también se conoce como sin relleno. En este caso, si las dimensiones no coinciden, se eliminará el último pliegue.

- Mismo relleno: este relleno garantiza que la capa de salida tenga el mismo tamaño que la capa de entrada.
- Relleno completo: este tipo de relleno aumenta el tamaño de la salida al agregar ceros al margen de la entrada.

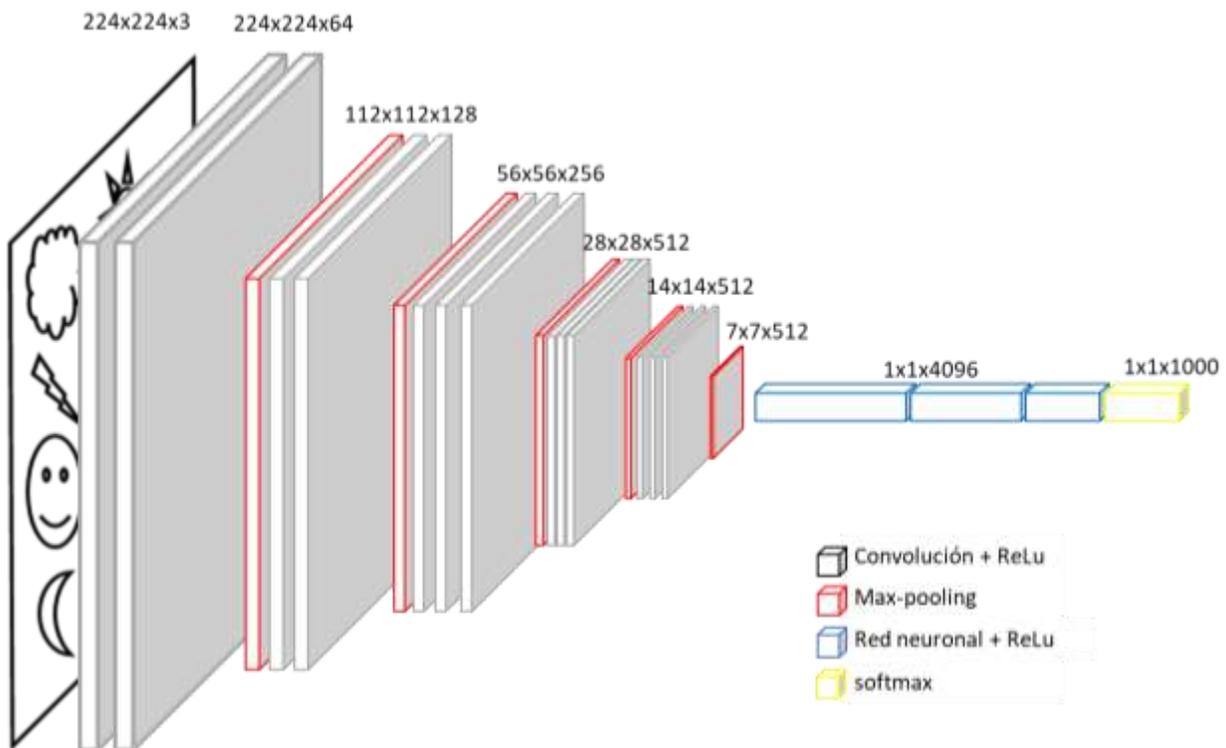
Después de cada operación de convolución, la Red Neuronal Convolutiva aplica una transformación Unidad Lineal Rectificada o ReLU al mapa de características, introduciendo la no linealidad en el modelo. La capa convolutiva original puede ser seguida por otra capa convolutiva. Cuando esto sucede, la estructura de la Red Neuronal Convolutiva puede volverse jerárquica ya que las capas posteriores pueden ver los píxeles dentro de los campos receptivos de las capas anteriores. La capa de convolución convierte la imagen en valores numéricos, lo que permite que la red neuronal interprete patrones relevantes y los extraiga.

Una Red Neuronal Convolutiva que recibe una imagen por ejemplo, y busque un valor en específico en la matriz de píxeles de la imagen, toma este dato digital de entrada y mediante el entrenamiento de múltiples capas convolutivas conformadas por filtros que realizan diferentes operaciones, extrayendo características más complejas progresivamente para lograr el reconocimiento de la imagen, cada bloque de filtros se encargará de extraer diferentes características, la salida de estos filtros se lleva a capas más profundas que permiten reducir la cantidad de datos, información, y extraer lo más representativos de los filtros que se utilizaron mediante máx-pooling y cada vez el tamaño de las imágenes resultantes va de-escalando y la cantidad de filtros usados se va incrementando.

A mayor profundidad en la red, se extraen más y más características, siendo cada vez más complejas hasta que resulta posible combinarlas en formas que permiten determinar el objeto que se quiere clasificar. En la parte final de la Red Neuronal Convolutiva se encuentra una Red Neuronal convencional de un número pequeño de capas, que permite tomar las características extraídas por las capas convolucionales, representadas como un vector de datos y se realiza la finalización de la clasificación de la imagen usando una función sigmoidea también conocida como “softmax”. (IBM Cloud Education. (2020c, octubre 20). *What are Convolutional Neural Networks?* Ibm.com. <https://www.ibm.com/cloud/learn/convolutional-neural-networks>)

Figura 9

Representación del modelo de Reconocimiento de objetos por capas



Nota. En esta representación gráfica, la imagen ilustrada a la izquierda tiene las dimensiones de 224 por 224 píxeles con 3 canales de color, procede a ser procesado por capas convolucionales integrando la unidad lineal rectificadora en color gris, la capa en color rojo es el proceso de Max-

pooling, en color azul se encuentra el procesamiento de la red neuronal integrando la unidad lineal rectificadora, en la última sección se utiliza la función Softmax en representado en color amarillo. Recreado de *VGG Very Deep Convolutional Networks (VGGNet) - What you need to know*. por Boesch, G., 6 de octubre del 2021, Viso.Ai. (<https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>)

2.16. Unidad Lineal Rectificada

La función de activación lineal rectificadora o ReLU para abreviar, es una función lineal por partes que generará la entrada directamente si es positiva; o de lo contrario, generará cero. Se ha convertido en la función de activación predeterminada para muchos tipos de redes neuronales porque un modelo que la usa es más fácil de entrenar y muchas de las veces funciona mejor por utilizarla. En una red neuronal, la función de activación es responsable de convertir la entrada ponderada sumada del nodo en la activación o salida del nodo para esa entrada.

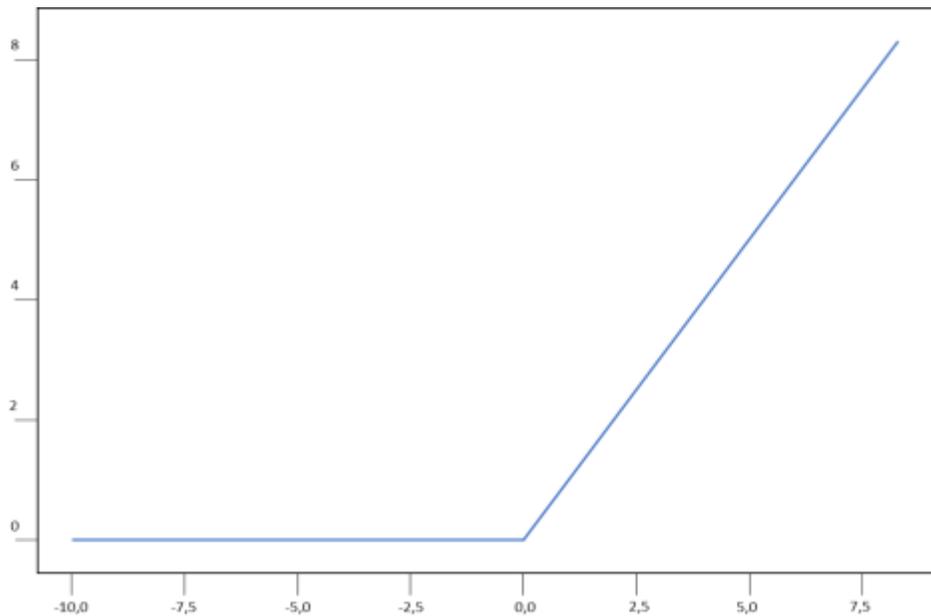
Un nodo o unidad que implementa esta función de activación se denomina unidad de activación lineal rectificadora, o ReLU para abreviar. Las redes que utilizan la función de rectificación de capa oculta a menudo se denominan redes rectificadas. La función de activación lineal rectificadora es un cálculo simple que devuelve el valor proporcionado como entrada directamente o el valor 0.0 si la entrada es 0.0 o menos.

La función es lineal para valores mayores que cero, y por esto se tiene muchas de las propiedades deseables de una función de activación lineal cuando se entrena una red neuronal

mediante retropropagación. Sin embargo, ya que los valores negativos siempre se reportan como cero, es una función no lineal.

Figura 10

Gráfico de línea de activación lineal rectificada para entradas negativas y positivas.



Nota. Todos los valores negativos y las entradas cero se establecen en 0,0 mientras que las salidas positivas se devuelven sin cambios, lo que da como resultado una pendiente que aumenta linealmente porque crea una serie de valores positivos que aumenta linealmente. Recreado de *A gentle introduction to the rectified linear unit (ReLU)*. Machine Learning Mastery. por Brownlee J. 8 de enero del 2019.

(<https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>)

La derivada de la función lineal rectificada también es sencilla de calcular. Cuando los pesos de un nodo se actualizan como parte de la propagación de errores la derivación de la función de activación es necesaria. La derivada de la función es la pendiente. La pendiente para valores

negativos es 0,0 y la pendiente para valores positivos es 1,0. El ReLU se puede utilizar con la mayoría de los tipos de redes neuronales, recomendando como estándar para perceptrones multicapa (MLP) y redes neuronales convolucionales (CNN).

Hay muchas ventajas al usar la función de activación lineal rectificadora, aquí podemos nombrar algunas de estas ventajas:

- Simplicidad computacional. - La función rectificadora es fácil de implementar y requiere una función $\max()$. Esto lo diferencia de las funciones de activación de Tanh y Sigmoide, que requieren el uso de un cálculo exponencial.
- Comportamiento lineal. - La función rectificadora se ve y se comporta principalmente como una función de activación lineal. Una red neuronal generalmente es más fácil de optimizar cuando su comportamiento es lineal o casi lineal. Las redes entrenadas con esta función de activación evitan el problema de los gradientes de fuga casi por completo, ya que los gradientes siguen siendo proporcionales a las activaciones de los nodos.
- Ahorro representativo. - Un beneficio importante de la función de rectificación es el que puede generar un verdadero valor cero. A diferencia de las funciones de activación Tanh y Sigmoidea, que aprenden a aproximarse a un valor muy cercano a cero, pero no un verdadero valor cero, pudiendo emitir valores cero verdaderos cuando las entradas son negativas, lo que permite la activación de capas ocultas en las redes neuronales para que contengan uno o más valores cero verdaderos. Esto se llama representación dispersa, una propiedad deseable en el aprendizaje

representacional porque puede acelerar el proceso de aprendizaje y ayudar a la simplificación del modelo.

- Entrenar redes profundas. - El redescubrimiento y adopción de la función de activación lineal rectificadora hizo posible entrenar con éxito redes multicapa profundas con una función de activación no lineal mediante retropropagación y aprovechar las mejoras en el hardware, al mismo tiempo que redes como las máquinas de Boltzmann, capacitación por turnos y la capacitación previa sin etiquetas podrían dejarse de lado.

2.16.1. Combinación de Capas

Conocida también como Reducción de Resolución, realiza una reducción de la dimensionalidad, lo que reduce la cantidad de parámetros en la entrada. Similar a la capa de convolución, la operación de agrupación barre un filtro sobre toda la entrada, pero en este caso la diferencia es que este filtro no tiene pesos. En su lugar, el kernel aplica una función de agregación a los valores en el campo de recepción y llena la matriz de salida. Hay dos tipos principales de agrupación:

Agrupación máxima. - a medida que el filtro se mueve a través de la entrada, selecciona el píxel con el valor máximo, que se envía a la matriz de salida. Dicho esto, este enfoque tiende a usarse con más frecuencia en comparación con la agrupación promedio.

Agrupación promedio. - a medida que el filtro se mueve a través de la entrada, calcula el valor promedio dentro del campo receptivo para enviarlo a la matriz de salida.

Mientras que se puede llegar a perder mucha información en la capa de agrupación, también tiene una serie de beneficios para la Red Neuronal Convolucional. Ayudan a reducir la complejidad, mejorar la eficiencia y limitar el riesgo de sobre adaptación.

2.16.2. Capa completamente conectada

El nombre de la capa completamente conectada se describe a sí mismo acertadamente. Antes se había mencionado que los valores de píxeles de la imagen de entrada en capas parcialmente conectadas no están conectados de forma directa a la capa de salida, pero en la capa totalmente conectada cada nodo de la capa de salida está directamente conectado a un nodo de la capa anterior. Esta capa realiza la tarea de clasificación en base a las características extraídas por las capas anteriores y sus distintos filtros. Mientras que las capas de convolución y agrupación tienden a usar funciones ReLu, las Capas Completamente Conectadas generalmente usan una función de activación softmax para clasificar adecuadamente las entradas y producir una probabilidad de 0 a 1.

2.17. Red Neuronal Recurrente

Una Red Neuronal Recurrente (“Recurrent Neural Networks” o RNN) es un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series temporales. Estos algoritmos de aprendizaje profundo se utilizan a menudo para problemas ordinarios o temporales, por ejemplo,

en el caso de traducción de voz, procesamiento de lenguaje natural (“Natural Language Processing” o NLP), reconocimiento de voz y anotación de imágenes.

Están integrados con aplicaciones de uso popular como Siri de Apple, Voice Search y Google Translate. Al igual que las Redes Neuronales Convolucionales y Redes Neuronales Pre Alimentadas, las Redes Neuronales Recurrentes utilizan datos de entrenamiento para aprender, sobresaliendo en su "memoria", ya que toman información de entradas anteriores para afectar la entrada y salida actual.

Mientras que las Redes Neuronales Profundas tradicionales se asume que las entradas y salidas son independientes entre sí, la salida de las Redes Neuronales Recurrentes depende de los elementos anteriores dentro de la secuencia. Si bien los eventos futuros también serían útiles para determinar el resultado de una secuencia en particular, las Redes Neuronales Recurrentes Unidireccionales no pueden tener en cuenta estos eventos en sus predicciones.

Otra característica que distingue a las Redes Neuronales Recurrentes de las Redes Neuronales Pre Alimentadas es que comparten parámetros en todas las capas de la red. Mientras que las Redes Neuronales Pre Alimentadas tienen diferentes pesos en cada nodo, las Redes Neuronales Recurrentes comparten el mismo parámetro de peso dentro de cada capa de la red. Estos pesos facilitan el aprendizaje por refuerzo al ajustarse a través de los procesos de retro propagación y descenso de gradiente.

Las Redes Neuronales Repetitivas utilizan el algoritmo de la Retro propagación a través del tiempo (“Backpropagation Through Time” o BPTT) para determinar los gradientes, que es ligeramente diferente de la propagación hacia atrás tradicional, ya que es específico de los datos de secuencia. Los principios de la Retro propagación a través del tiempo son los mismos que los de la Retro propagación tradicional, donde el modelo se entrena a sí mismo calculando errores desde su capa de salida hasta su capa de entrada. Estos cálculos nos permiten ajustar y ajustar los parámetros del modelo en consecuencia. La Retro propagación a través del tiempo difiere del enfoque tradicional en que suma los errores en cada paso de tiempo, mientras que las Redes Neuronales Pre Alimentadas no tienen que sumar los errores ya que no comparten parámetros en cada capa.

A través de este proceso, las Redes Neuronales Recurrentes tienden a encontrar dos problemas conocidos como gradientes explosivos y gradientes de fuga. Estos problemas se definen por la magnitud del gradiente que representa la pendiente de la función de pérdida a lo largo de la curva de error. Si el gradiente es demasiado pequeño, continuará disminuyendo, actualizando los parámetros de ponderación hasta que se vuelvan insignificantes, es decir, a 0. Al suceder esto, el algoritmo deja de aprender. Los gradientes explosivos se producen cuando el gradiente es demasiado grande, lo que crea un modelo inestable. En estos casos los pesos del modelo se vuelven demasiado grandes y terminan presentándose como NaN que es una propiedad global de un valor que representa Not-A-Number (No es Un Número). Una solución a estos problemas es reducir la cantidad de capas ocultas dentro de la red neuronal, eliminando de esta manera parte de la complejidad del modelo de Redes Neuronales.

2.18. Modelo de Reconocimiento de Texto CRAFT

Muchos modelos de aprendizaje profundo se han utilizado en visión por computadora para la tarea de reconocimiento de texto, sin embargo, estos modelos no funcionan bien cuando el texto de la imagen está torcido o el texto está curvado. El modelo CRAFT funciona bien en texto curvo, largo y deformado además del texto normal. Este modelo ha demostrado que supera a los modelos de última generación en varios conjuntos de datos de referencia, como TotalText, CTW-1500, etc.

Las técnicas tradicionales de detección de cuadros delimitadores a nivel de carácter no son adecuadas para abordar estas áreas. Además, la generación de datos reales para el procesamiento de texto a nivel de caracteres es una tarea tediosa y costosa. El modelo de reconocimiento de texto CRAFT utiliza un modelo de red neuronal convolucional para calcular puntuaciones de la región y la afinidad. La puntuación de región se usa para ubicar regiones de caracteres, mientras que la puntuación de la afinidad se usa para agrupar los caracteres en las regiones del texto.

CRAFT utiliza una red neuronal totalmente convolucional basada en el modelo VGG16. La inferencia se realiza proporcionando cuadros delimitadores a nivel de palabra. El modelo de reconocimiento de texto CRAFT funciona bien a diferentes escalas que podrían presentarse, desde textos grandes a pequeños, y ha demostrado su eficacia incluso en conjuntos de datos invisibles. Sin embargo, el rendimiento de reconocimiento de texto del modelo no es óptimo para texto en lenguas árabe y bengalí, que contiene caracteres continuos.

El tiempo que tardan los métodos de reconocimiento de texto CTPN, EAST y MSER es menor en comparación con el motor de reconocimiento de texto CRAFT. Sin embargo, CRAFT es más preciso y los cuadros delimitadores son más precisos cuando el texto es largo, curvo, es girado o está deformado. (Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. En *arXiv [cs.CV]*. <http://arxiv.org/abs/1904.01941>)

2.19. Red Residual

También conocido como ResNet o ‘Residual Network’ en inglés, es uno de los más famosos modelos de aprendizaje profundo presentados en el trabajo denominado "Aprendizaje residual profundo para el reconocimiento de imágenes" en 2015 por Shaoqing Ren, Kaiming He, Jian Sun y Xiangyu Zhang.

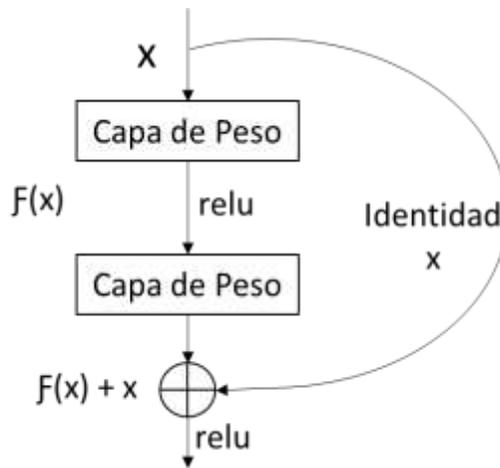
La Red Residual fue la respuesta para los problemas que se presentaban al entrenar Redes Neuronales Convolucionales, a lo largo de los años ha habido una serie de avances en el Aprendizaje Profundo y la Visión Artificial. En particular, con la llegada de redes neuronales convolucionales muy profundas, estos modelos han ayudado a lograr resultados de vanguardia en problemas como el reconocimiento de imágenes y la clasificación de imágenes.

Tras estos avances las arquitecturas de aprendizaje profundo se han ido profundizando al agregar más capas para resolver tareas cada vez de mayor complejidad, lo que también ayudó a la mejora del rendimiento de las tareas de clasificación y de reconocimiento, haciéndolas más robustas. Pero a medida que se agregan más capas a la red neuronal, esta se vuelve mucho más

difícil de entrenar apropiadamente y la precisión del modelo comienza a saturarse y a la vez luego degradarse. En el artículo "Aprendizaje residual profundo para el reconocimiento de imágenes" podemos ver el siguiente gráfico.

Figura 11

Interpretación gráfica de la aplicación método de aprendizaje residual y la utilización de la unidad de rectificación lineal para el aprendizaje artificial.



Nota. Representación de método de aprendizaje residual utilizando la unidad de rectificación lineal recreada de *programador clic*. (s/f). Programmerclick.com. Recuperado el 15 de agosto de 2022, de (<https://programmerclick.com/article/21771415339/>)

En la figura podemos notar que hay una conexión directa que salta algunas capas del modelo esta conexión se denomina "conexión de salto". La salida no es la misma debido a esta conexión de salto. Sin la conexión de salto, la entrada "X" se multiplica por los pesos de la capa, seguido de la suma de un término de sesgo. Luego viene la función de activación $f()$ y obtenemos la salida como $H(x)$ con la fórmula $H(x)=f(wx + b)$ o también $H(x)=f(x)$ Con la introducción de

una nueva técnica de conexión de salto, la salida ahora cambia a $H(x)$. Es así que nos encontramos con la fórmula $H(x)=f(x)+x$.

Sin embargo, la dimensión de la entrada puede diferir de la de la salida, lo que puede ser el caso de una capa convolucional o capas de agrupación. Por lo tanto, este problema se puede tratar con estos dos enfoques:

- Zero se rellena con la “conexión de salto” para aumentar sus dimensiones.
- Las capas de convolución 1×1 se agregan a la entrada para ajustar las dimensiones.

En tal caso, la salida es $H(x)=f(x)+w1.x$; Aquí se agrega un parámetro adicional $w1$ (peso 1), mientras que no se agrega ningún parámetro adicional cuando se usa el primer enfoque.

Esta técnica para omitir conexiones en Redes Residuales o ResNet resuelve el problema del gradiente de fuga en las Redes Neuronales Convolucionales (“Convolutional Neural Network” o CNN) profundas al permitir una ruta de atajo alternativa para el flujo de gradiente. Además, la conexión de omisión ayuda, si una capa afecta el rendimiento de la arquitectura, la regularización la omitirá. (*Residual networks (ResNet) - deep learning*. (2020, Junio 3). GeeksforGeeks.

<https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>)

2.20. Red Neuronal Convolucional de uso común VGG

El Grupo Geométrico Visual o VGG (Visual Geometry Group), también conocido como VGGNet, es una arquitectura clásica de Redes Neuronales Convolucionales. VGG fue diseñado

para aumentar la profundidad de tales Redes Neuronales Convolucionales para aumentar el rendimiento del modelo. Es una arquitectura estándar de Red Neuronal Convolutiva profunda multicapa.

La "Profundidad" hace referencia al número de capas, con VGG-16 o VGG-19 que constan respectivamente de 16 y 19 capas convolucionales. VGGNet fue diseñado como una red neuronal profunda que supera la línea de base en muchas tareas y conjuntos de datos más allá de ImageNet, además de ser una de las arquitecturas de reconocimiento de imágenes más populares.

El modelo VGG o VGGNet, que admite 16 capas, también se conoce como VGG16, es un modelo de red neuronal convolutiva propuesto por los investigadores de la Universidad de Oxford A. Zisserman y K. Simonyan.

Estos investigadores publicaron su modelo en el artículo de investigación titulado "Redes convolucionales muy profundas para el reconocimiento de imágenes a gran escala" que logra casi un 92,7 % de precisión en las 5 pruebas principales en ImageNet, siendo ImageNet un conjunto de datos que consta de más de 14 millones de imágenes pertenecientes a casi 1000 clases.

Además de que fue uno de los modelos más populares presentados al 'Reto de Reconocimiento Visual a Gran Escala ImageNet' ("ImageNet Large Scale Visual Recognition Challenge" ILSVRC-2014). Reemplaza los filtros grandes del tamaño del kernel uno por uno con múltiples filtros del tamaño del kernel 3×3 . El modelo VGG16 se entrenó durante varias semanas con la Unidad de Procesamiento Gráfico Nvidia Titan Black. VGGNet-16 admite 16 capas y es

capaz de clasificar imágenes en 1000 categorías de objetos, incluidos categorías de teclado, animales, lápiz, mouse, etc. Además, el modelo tiene un tamaño de entrada de imagen de 224x224.

Por otra parte, el concepto del modelo VGG19; también conocido como VGGNet-19; es el mismo que el VGG16 excepto que admite 19 capas. El "16" y el "19" representan el número de capas de peso en el modelo, conocidas como capas de convolución. Esto significa que VGG19 tiene tres capas convolucionales más que VGG16. A medida que aumenta el número de capas en Redes Neuronales Convolucionales, también aumenta la capacidad del modelo para adaptarse a funciones más complejas.

Por lo tanto, más capas prometen un mejor rendimiento. Evítese la confusión con una Red Neuronal Artificial, donde el aumento en el número de capas en el modelo no necesariamente resultaría en un mejor rendimiento. (Boesch, G. (2021, octubre 6). *VGG Very Deep Convolutional Networks (VGGNet) - What you need to know*. Viso.Ai. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>)

2.21. Redes Neuronales de Memoria a Corto Plazo

Son un tipo de red neuronal recurrente capaz de aprender la dependencia del orden en problemas de predicción de secuencias, este es un comportamiento requerido en áreas problemáticas complejas como la traducción automática o el reconocimiento de voz por nombrar algunos ejemplos. Los dos problemas técnicos superados gracias a las Redes Neuronales de

Memoria a Corto Plazo son los gradientes de fuga y los gradientes de explosión, ambos relacionados con la forma en que se entrena la red.

Es una Red Neuronal Recurrente avanzada, que permite la persistencia de la información. Es capaz de hacer frente al problema del gradiente de fuga al que se enfrenta la Red Neuronal Recurrente. Las Redes Neuronales Recurrentes recuerdan la información anterior y la usan para procesar la entrada actual. La desventaja de las Redes Neuronales Recurrentes es que no pueden recordar las dependencias a largo plazo debido al gradiente de fuga.

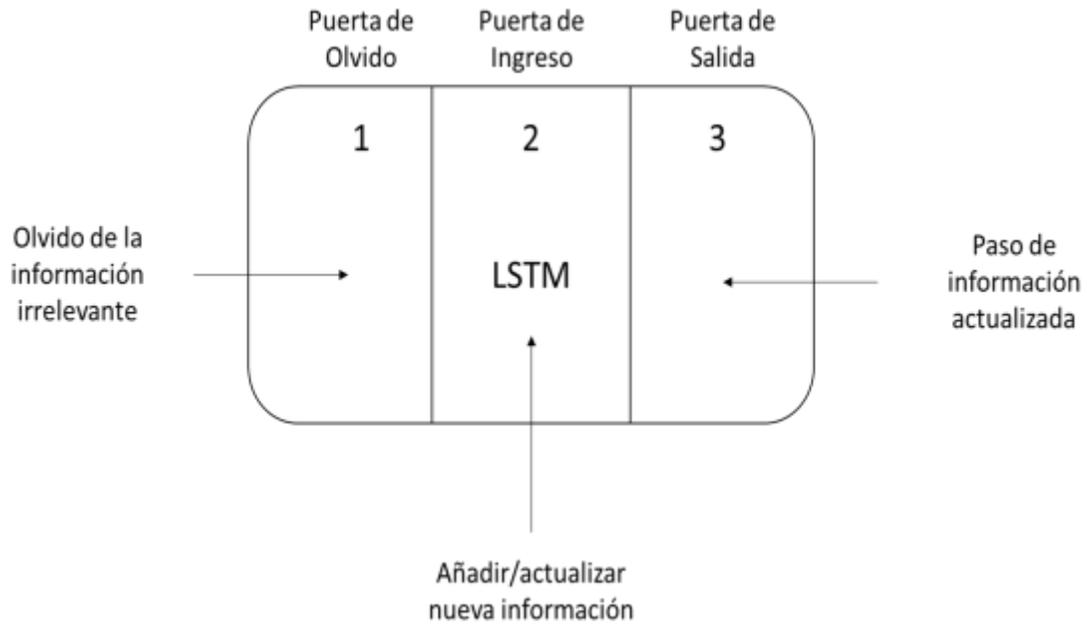
Las Redes Neuronales de Memoria a Corto Plazo están diseñadas explícitamente para evitar problemas de dependencia a largo plazo. Las Redes Neuronales de Memoria a Corto Plazo funcionan de manera muy similar a las Redes Neuronales Recurrentes. Consta de tres partes, cada parte cumple una función individual. La primera parte selecciona si la información proveniente de la marca de tiempo anterior debe recordarse o es irrelevante y puede olvidarse.

En la segunda parte, la celda intenta aprender nueva información de la entrada en esa celda. Finalmente, en la tercera parte, la celda pasa la información actualizada de la marca de tiempo actual a la siguiente marca de tiempo.

Estas partes se denominan puertas. La primera parte se llama Puerta de Olvido o Forget Gate, la segunda parte es la Puerta de Ingreso o Input Gate y la última es la Puerta de Salida u Output Gate.

Figura 12

Interpretación de las secciones de las Redes Neuronales de Memoria a Corto Plazo



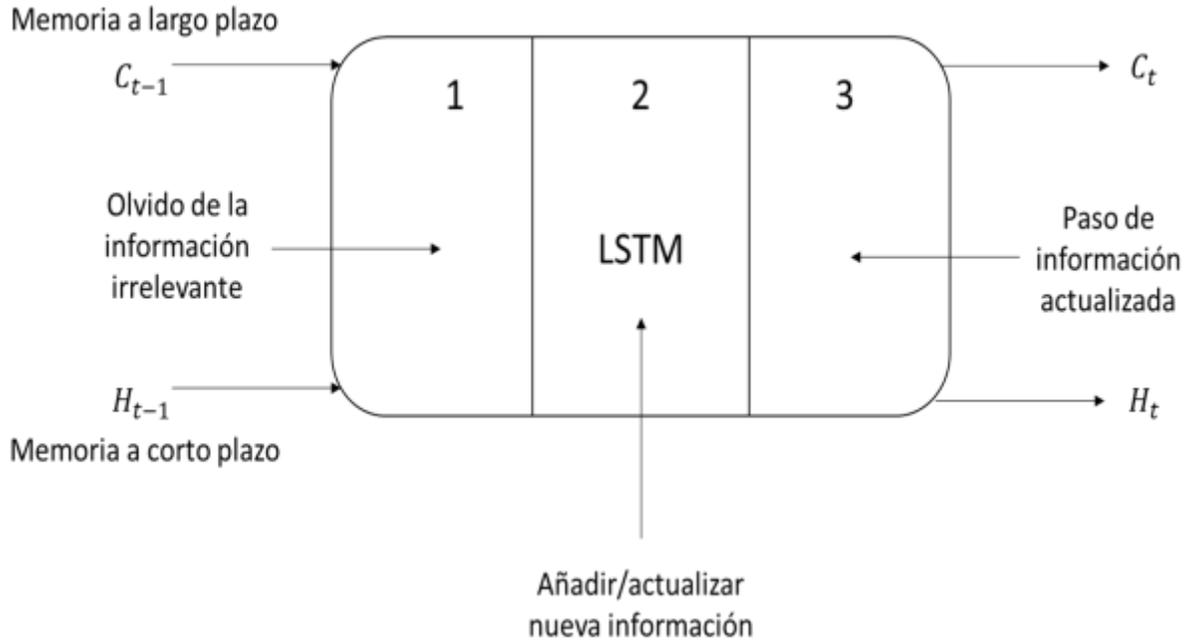
Nota. Gráfico recreado de *LSTM*. Saxena, S. (2021, marzo 16). Analytics Vidhya. (<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>)

Las Redes Neuronales de Memoria a Corto Plazo también tiene un estado oculto o “Hidden state”, donde $H(t-1)$ representa el estado oculto de la marca de tiempo anterior y H_t es el estado oculto de la marca de tiempo actual. Además, las Redes Neuronales de Memoria a Corto Plazo también tiene un estado de celda representado por $C(t-1)$ para la marca de tiempo anterior y $C(t)$ para la marca de tiempo actual.

Aquí el estado oculto se llama memoria a corto plazo y el estado celular se llama memoria a largo plazo. Es de destacar; además; que la celda lleva la información consigo junto con los registros de tiempo.

Figura 13

Arquitectura de las Redes Neuronales de Memoria a Corto Plazo



Nota. Gráfico recreado de *LSTM*. Saxena, S. (2021, marzo 16). Analytics Vidhya. (<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>)

Para detallar el funcionamiento de las Redes Neuronales de Memoria a Corto Plazo, a continuación, se presentan los roles que cumplen las puertas en la arquitectura de las Redes Neuronales de Memoria a Corto Plazo. (Brownlee, J. (2017, mayo 23). *A gentle introduction to long Short-Term Memory networks by the experts*. Machine Learning Mastery. <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>)

2.21.1. Puerta de Olvido

En una celda de Redes Neuronales de Memoria a Corto Plazo, primero debemos decidir si debemos mantener u olvidar la información de la marca de tiempo anterior. Aquí se presenta la ecuación para la Puerta de Olvido $\sigma_t = \sigma(U_{of} * X_t + W_{of}(t-1) * H_{t-1})$ en la que:

- X_t : Ingresa el registro de tiempo actual.
- U_{of} : Peso asociado con el ingreso.
- H_{t-1} : El estado oculto del registro de tiempo anterior.
- W_{of} : Es la matriz de pesos asociados con el estado oculto.

Se le aplica una función sigmoide, resultando en un valor entre 0 y 1. Esta función F_t se multiplica con la celda de estado del anterior registro de tiempo. Si es 0 en la ecuación $\sigma_t(t-1) * H_{t-1} = 0$... si $\sigma_t = 0$ (olvidar todo), entonces la red olvidará todo, si el valor es 1 en el resultado de la ecuación $\sigma_t(t-1) * H_{t-1} = H_{t-1}$... si $\sigma_t = 1$ (no olvidar) no olvidará nada.

2.21.2. Puerta de Ingreso

Esta puerta es usada para cuantificar la importancia de la nueva información obtenida por el ingreso, cuya ecuación es $\sigma_t = \sigma(U_{oi} * X_t + W_{oi}(t-1) * H_{t-1})$ en el cual:

- X_t : Ingreso con el registro actual de tiempo (t).
- U_{oi} : matriz de peso de ingreso.
- H_{t-1} : un estado oculto con el registro de tiempo anterior.
- W_{oi} : matriz de peso como ingreso asociado con estado oculto.

Se aplica una función sigmoide nuevamente. Como resultado el valor de (i) en el registro de tiempo (t) será entre 0 y 1.

2.21.3. Nueva Información

Tenemos la siguiente función para la información nueva $N_t = \tanh(w_{ij} * C_{t-1} + w_{ix} * x_t)$, la nueva información obtenida que necesita ser transferida a la celda del estado es una función de un estado oculto en el registro de tiempo anterior t-1 y la entrada x en la marca de tiempo t. La función de activación aquí es tanh, o tangente hiperbólica. Debido a la función tanh, el valor de la nueva información varía de -1 a 1. Si el valor de N_t es negativo, la información se resta del estado de la celda, y si el valor es positivo, la información se agrega a la celda de estado en la marca de tiempo actual. Sin embargo, el N_t no se agrega directamente al estado celular.

En la fórmula $C_t = C_{t-1} * \tanh(N_t) + C_{t-1} * C_{t-1}$ (actualización de la celda de estado), donde C_{t-1} es la celda de estado con la marca de tiempo actual y otros son valores que se calcularon previamente.

2.21.4. Puerta de Salida

Para la Puerta de Salida, tenemos la siguiente ecuación $O_t = \sigma(w_{oj} * C_t + w_{ox} * x_t)$, en este su valor se encuentra entre 0 y 1 debido a esta función sigmoide. Se calcula el estado oculto actual usando O_t y tanh del estado de la celda como en la ecuación $O_t * C_t = O_t * \tanh(C_t)$, el estado oculto resulta ser una función de la memoria a largo plazo (C_t)

y el rendimiento actual. Si necesita la salida de la marca de tiempo actual, simplemente aplique la activación de SoftMax en el estado oculto H_t tal que $Output = \text{SoftMax}(H_t)$, aquí, el token con la puntuación máxima en la salida es la predicción. Este es el diagrama más intuitivo de las Redes Neuronales de Memoria a Corto Plazo.

2.22. Clasificación Temporal Conexionista

La Clasificación Temporal Conexionista (“Connectionist Temporal Classification” o CTC) es una función de costo utilizada para entrenar Redes Neuronales Recurrentes para etiquetar datos no segmentados en el aprendizaje supervisado de la secuencia de entrada, mayormente utilizado en el reconocimiento y decodificación de caracteres deformados o manuscrita. La Clasificación Temporal Conexionista utiliza la matriz resultante de las Redes Neuronales Convolucionales para el entrenamiento de la red neuronal y/o para la decodificación de la salida de la red neuronal.

Está formulado de tal forma que solo se requiere el texto que aparece en la imagen en el caso de Reconocimiento de Caracteres, ignorando tanto el ancho como la posición de los caracteres en una imagen.

No tiene que ser procesada posteriormente, pudiendo obtener directamente el resultado de la red usando técnicas de decodificación. (Siddhant’s Scratch Book. (2019, octubre 19). Explanation of connectionist Temporal Classification. *Siddhant’s Scratch Book*. https://sid2697.github.io/Blog_Sid/algorithm/2019/10/19/CTC-Loss.html)

3.23. Aplicativo Web

3.23.1. ¿Qué es un Aplicativo Web?

Una aplicación web es un sitio web que contiene páginas con contenido parcial o totalmente indeterminado. El contenido final de una página no se determina hasta que el visitante solicita una página del servidor web. Debido a que el contenido final de la página varía de una solicitud a otra según las acciones del visitante, este tipo de página se denomina página dinámica.

Una aplicación web es una colección de páginas web estáticas y dinámicas. Una página web estática no cambia cuando un visitante del sitio la solicita: el servidor web envía la página al navegador web solicitante sin modificarla. Por el contrario, el servidor modifica una página web dinámica antes de enviarla al navegador solicitante. La naturaleza cambiante de la página es la razón por la que se llama dinámica. Las siguientes secciones describen con más detalle cómo funcionan las aplicaciones web.

Las aplicaciones web tienen muchos usos para los visitantes y desarrolladores de sitios web, permiten que los visitantes encuentren información rápida y fácilmente en un sitio web rico en contenido, este tipo de aplicación web brinda a los visitantes la capacidad de buscar, organizar y navegar por el contenido a su propia discreción. Recopilan, almacenan y analizan datos proporcionados por los visitantes del sitio web. En el pasado, los datos ingresados en formularios HTML se enviaban como mensajes de correo electrónico a colaboradores o aplicaciones CGI para su procesamiento. Una aplicación web puede almacenar datos de formularios directamente en una base de datos y también extraer los datos y generar informes basados en la web para su análisis.

Algunos ejemplos son páginas de banca en línea, de pago, encuestas y formularios de comentarios de los usuarios.

Las aplicaciones web son capaces de actualizarse con contenido en constante cambio, una aplicación web libera al diseñador web de actualizar constantemente el HTML del sitio web. Los proveedores de contenido, como los editores de noticias, proporcionan contenido a la aplicación web y la aplicación web actualiza automáticamente el sitio. (*No title.* (s/f). Stackpath.com. Recuperado el 15 de agosto de 2022, de <https://www.stackpath.com/edge-academy/what-is-a-web-application/>)

3.24. ¿Que es Anvil?

Anvil es una plataforma para crear y alojar aplicaciones web completas escritas completamente en Python. Anvil Runtime es el motor de código abierto que impulsa estas aplicaciones. Este repositorio contiene las bibliotecas de tiempo de ejecución de Anvil, así como un servidor de aplicaciones independiente que usa el tiempo de ejecución para servir una aplicación de Anvil desde el sistema de archivos local.

Anvil proporciona un editor gratuito en línea para construir aplicaciones Anvil llamado Anvil Works, con una interfaz de arrastrar y soltar, con elementos prediseñados como listas colapsables, áreas de imágenes, textos, listados, botones, etc.; Sin requerir HTML o CSS. La interfaz puede ser personalizada para las necesidades específicas de su caso de uso, además de la opción de añadir elementos personalizados del usuario.

Soporta varios tipos de librerías populares para Python, junto con guías de uso, algunas de estas son: Matplotlib, Plotly, Seaborn, Bokeh, Altair, Pygal. De estos, Plotly es el que tiene mayor soporte directo debido a que es una librería Javascript frontal. Dispone de soporte para bases de datos externas a través de librerías populares de Python como pymysql y psycopg2. Por otra parte, Anvil tiene bases de datos integradas para el desarrollo de aplicaciones, llamadas Data Tables, son sistemas de bases de datos basados en PostgreSQL que son accesibles directamente en Python.

Gracias a que la aplicación es almacenada en la nube del servicio Anvil, esta no requiere ser desplegada desde la computadora del usuario u otro equipo, sino que puede ser desplegada al seleccionar la opción Publicar en el editor proporcionado por Anvil. Al hacer clic en la opción Publicar, se dará un link que puede ser editado en parte a discreción del usuario, este link permite ingresar a la aplicación web con los cambios realizados hasta ese momento, ya que Anvil guarda los cambios realizados a la aplicación a través de un historial automático de versiones, esto además permite seleccionar la versión que se desea que sea publicada y utilizada mientras se trabaja en los cambios para una nueva versión de la misma aplicación.

Existen varias maneras de publicar la aplicación creada con Anvil, como pueden ser:

- Diseñar y Publicar en la nube de Anvil. - Este es el método con mayores facilidades para despliegue.
- Diseñar y Publicar a través de una nube privada. - Anvil tiene una versión para Empresas que permite desplegar el ecosistema de datos en sus propios servidores.

- Diseñar la aplicación con Anvil y hospedarla en un servidor propio. - Este método permite utilizar las herramientas de Anvil con la libertad de hospedar la aplicación en el servicio o método que sea de preferencia del usuario.
- Diseñar con un editor de texto. - Puede utilizarse código realizado en otro editor de texto y utilizar las herramientas de Anvil como la base de datos o de publicación sin utilizar la interfaz de Anvil.

Anvil anunció una extensión de su desarrollo de aplicaciones web, lo que permite a los desarrolladores de Python diseñar, crear e implementar aplicaciones web. Al hacer que el motor de tiempo de ejecución sea de código abierto, los desarrolladores que dominan el lenguaje Python ahora pueden optar por implementar sus aplicaciones en sus propias computadoras, o en dispositivos de IoT o Internet de las Cosas integrados o especializados, así como en las nubes de sus empleadores o el proporcionado por Anvil.

El nuevo servidor de aplicaciones de código abierto permite exportar una aplicación y ejecutarla en cualquier lugar sin depender de la nube de Anvil o de una conexión a Internet. (Ritz, R. (2020, agosto 11). *An Introduction to Anvil — Full-Stack Web Apps with Nothing but Python*. Towards Data Science. <https://towardsdatascience.com/an-introduction-to-anvil-full-stack-web-apps-with-nothing-by-python-cbab06392d13>)

CAPÍTULO III

DESARROLLO

3. DESARROLLO DEL SISTEMA

3.1. Antecedentes del Registro

En el establecimiento de lavado de vehículos LISTOCAR se realizan servicios de limpieza, venta de accesorios y mantenimiento vehicular. Debido al número de vehículos que ingresan diariamente a este establecimiento se requiere mantener un registro en el cual se describen las características del vehículo, el servicio o servicios solicitados, ítems vendidos del almacén a los clientes, valor a pagar y detalles necesarios para la identificación y cobranza de los servicios prestados. Este registro se realizaba hasta el año 2021 a mano escrita en cuadernos de contabilidad, este anticuado método de registro era propenso a errores, recibir daños, alteraciones, e incluso dificultar la búsqueda de entradas específicas en el registro. Debido a esto el uso de un registro electrónico es necesario, que cumpliera con los requerimientos de los administradores y empleados, y resuelva los problemas presentados previamente derivados del registro físico.

En el registro de los vehículos que ingresan en el establecimiento se encuentran los campos de tipo de vehículo o modelo, marca del vehículo, placa vehicular, color del vehículo, y el tipo de servicio solicitado, un espacio de anotaciones sobre ítems vendidos o detalles importantes sobre el vehículo o del servicio, si fue cancelado con tarjeta de crédito o si el cliente dispone de crédito en

el establecimiento, y una área donde se indique el valor a cancelar, y la hora en la que el vehículo fue recibido para estimar el tiempo en el que el vehículo estará listo para ser retirado por el cliente.

3.2. Entrevistas con los Usuarios

Para entender mejor las necesidades de los usuarios a los que está destinado el sistema de registro vehicular, se realizaron entrevistas con el personal del establecimiento, describiendo qué necesidades tienen, expectativas y requerimientos que el aplicativo web pueda satisfacer o complementar en sus actividades.

3.2.1. Historias de Usuario

Historia de Usuario #1

Figura 14

Ficha de registro de la Historia de Usuario

Historia de Usuario	
Numero: 1	Nombre: Acceso al Listado de Vehículos desde el Teléfono Móvil.
Usuario: Gerente.	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Puntos Reales: 1.3
Descripción: El registro debe ser accedido en cualquier momento tanto en ordenadores como en teléfonos móviles, visualizar el valor a cobrar, anotaciones y hora de ingreso del vehículo.	
Observaciones: El uso de aplicativos webs permite que el registro sea accesible en cualquier momento mientras se cuente con la conectividad a internet	

Nota. Registro de Historias de Usuarios realizada luego de la entrevista del 12 de febrero del 2022 al Ing. Luis Fernando Salinas Calva, Gerente de lavadora LISTOCAR.

Historia de Usuario #2

Figura 15

Ficha de registro de la Historia de Usuario

Historia de Usuario	
Numero: 2	Nombre: Ingreso del precio a pagar y anotaciones.
Usuario: Cajero.	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1.1
Riesgo en Desarrollo: Baja	Puntos Reales: 0.9
Descripción: El registro debe mostrar el valor estimado a pagar, y un marcador que indique si el valor fue cancelado.	
Observaciones: La interfaz para el registro debe contar con un medio para indicar si el valor de los servicios prestados fue cancelado o están pendientes de cancelar.	

Nota. Registro de Historias de Usuarios de la entrevista del día 12 de febrero del 2022 a Ana Soledad Herrera Vega, Contadora y Cajera de lavadora LISTOCAR.

Historia de Usuario #3

Figura 16

Ficha de registro de la Historia de Usuario

Historia de Usuario	
Numero: 3	Nombre: visualización de los tiempo de ingreso de los vehiculos.
Usuario: Lavador.	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 0.9
Riesgo en Desarrollo: Baja	Puntos Reales: 0.8
Descripción: El registro debe contener un área donde se pueda visualizar la hora de ingreso del vehículo del cliente, para estimar el tiempo que se dispone para la entrega del vehículo con los servicios solicitados concluidos.	
Observaciones: El tiempo que dispone cada lavador para realizar los servicios solicitados está determinado según la cantidad de vehículos en espera y el servicio o servicios solicitados, conocer la hora de ingreso permite priorizar según el tiempo que se dispone y la complejidad del servicio solicitado.	

Nota. Ficha de registro de Historias de Usuarios realizada luego de la entrevista del día 12 de febrero del 2022 al lavador de LISTOCAR.

3.3. Experimentación

Para el desarrollo de esta aplicación se investigó métodos, herramientas y procesos por los cuales el reconocimiento de texto en imágenes es aplicado para proyectos de similar aplicación, se experimentó con los documentos, tutoriales, recursos y proyectos utilizados por otros estudiantes, profesionales y organizaciones que han publicado y distribuido para su estudio, uso y aplicación. Estas experimentaciones fueron educativas en el uso de herramientas, librerías, módulos, entornos virtuales, etc. utilizados para el prototipado y desarrollo de proyectos y aplicaciones que derivaron en la solución desarrollada para el problema presentado en el establecimiento LISTOCAR.

3.3.1. Experimentación 1

El primer proyecto en el cual se experimentó para el estudio de soluciones para la detección de texto en imágenes y su procesamiento es el proyecto titulado “Curso de Detección de Objetos con TensorFlow” (“Tensorflow Object Detection” o TFODCourse), en el cual destaca el uso de TensorFlow, modelos previamente entrenados, y librerías, el uso de EasyOCR, Pytorch y datos de Kaggle para el reconocimiento óptico de caracteres. Este proyecto se encuentra publicado para la investigación del mismo y su uso para estudio y aplicación en GitHub y explicado a través de tutoriales publicados en la plataforma YouTube, en el canal Nicholas Renotte en su video Curso Completo de Reconocimiento Automático del Número de Placa utilizando TensorFlow y EasyOCR en 2 Horas | Python (“Automatic Number Plate Recognition using Tensorflow and EasyOCR Full Course in 2 Hours | Python”), en el que describe el uso y funcionamiento del proyecto.

Se procedió con la descarga del proyecto desde el repositorio en GitHub para su experimentación, al realizar los pasos descritos se encontraron los siguientes problemas.

Figura 17

Experimentación con el código resultando en errores y conflictos con los paquetes y módulos de TensorFlow 1.x

```
In [10]: !pip install numpy

Collecting numpy
  Downloading numpy-1.23.1-cp310-cp310-win_amd64.whl (14.6 MB)
  ----- 14.6/14.6 MB 6.5 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.23.1

In [11]: !pip uninstall tensorflow
!pip install tensorflow==1.13.2

WARNING: Skipping tensorflow as it is not installed.
ERROR: Could not find a version that satisfies the requirement tensorflow==1.13.2 (from versions: 2.8.0rc1, 2.8.0, 2.8.1, 2.8.2, 2.9.0rc0, 2.9.0rc1, 2.9.0rc2, 2.9.0, 2.9.1, 2.10.0rc0)
ERROR: No matching distribution found for tensorflow==1.13.2

In [12]: import tensorflow.compat.v1 as tf

-----
ModuleNotFoundError                                Traceback (most recent call last)
Input In [12], in <cell line: 1>()
----> 1 import tensorflow.compat.v1 as tf

ModuleNotFoundError: No module named 'tensorflow'

In [13]: import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

-----
ModuleNotFoundError                                Traceback (most recent call last)
Input In [13], in <cell line: 1>()
----> 1 import tensorflow.compat.v1 as tf
      2 tf.disable_v2_behavior()

ModuleNotFoundError: No module named 'tensorflow'

In [14]: import tensorflow as tf
print(tf.version)

-----
ModuleNotFoundError                                Traceback (most recent call last)
```

Entre los problemas encontrados, se encuentra el uso de TensorFlow 1.x, una librería de código abierto que ha sido reemplazada por el uso de TensorFlow 2, su nueva versión que

implementa cambios y optimizaciones, esto derivó en problemas y conflictos con otras librerías utilizadas en este proyecto, incluso en incompatibilidad con librerías que ya no son soportadas por TensorFlow en su nueva versión. Además de este problema, este proyecto requiere el entrenamiento de detección de caracteres con datos obtenidos de Kaggle.

3.3.2. Experimentación 2

Se experimentó con el código del proyecto “Reconocimiento Automático de Números de Placa” (“Automatic Number Plate Recognition” o ANPR) publicado en un repositorio en GitHub para su uso, investigación y aplicación.

Además, está explicado a través de tutoriales publicados en la plataforma YouTube, en el canal Nicholas Renotte en su video “Python ANPR con OpenCV y EasyOCR en 25 Minutos | Tutoría de Reconocimiento Automático de Número de Placa” (“Python ANPR with OpenCV and EasyOCR in 25 Minutes | Automatic Number Plate Recognition Tutorial”), el cual utiliza el módulo de Python EasyOCR para el Reconocimiento Óptico de Caracteres, y la biblioteca de aprendizaje automático de código abierto PyTorch.

Al realizar las pruebas de ejecución, se encontraron varios inconvenientes, podemos denotar la necesidad de la instalación de PyTorch en el dispositivo donde el entorno virtual es ejecutado, el uso de los filtros para la imagen causaba que fotos sin la iluminación adecuada quedaron oscurecidas por completo, causando problemas con la ejecución de los siguientes filtros y procesamientos.

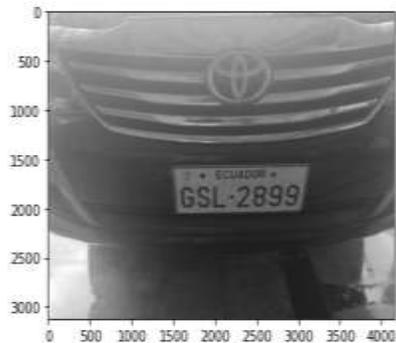
Figura 18

Experimentación con el código resultando en errores en la utilización de escala de grises.

1. Read in Image, Grayscale and Blur

```
In [16]: img = cv2.imread('18.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

Out[16]: <matplotlib.image.AxesImage at 0x2bedd7c4e20>



2. Apply filter and find edges for localization

```
In [34]: bfilter = cv2.bilateralFilter(gray, 5, 50, 50) #Woise reduction
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[34]: <matplotlib.image.AxesImage at 0x2be8261d270>



Debido a los problemas que puede causar el filtro de la imagen en gris se deriva en problemas en la detección de bordes y aplicación de máscaras, dificultando y hasta imposibilitando la identificación de la posición de la placa en la imagen, y por lo tanto afectando la capacidad de la detección de caracteres.

Figura 19

Experimentación con el código resultando en problemas al realizar la detección de bordes de la placa y aplicación de máscaras en la imagen debido a errores en la utilización de escala de grises.

```
In [37]: (x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]

-----
ValueError                                Traceback (most recent call last)
Input In [37], in <cell line: 2>():
      1 (x,y) = np.where(mask==255)
----> 2 (x1, y1) = (np.min(x), np.min(y))
      3 (x2, y2) = (np.max(x), np.max(y))
      4 cropped_image = gray[x1:x2+1, y1:y2+1]

File <_array_function__ internals>:180, in min(*args, **kwargs)

File ~\Documents\codigos de prueba para el proyecto\ANPR rapido\anprsys\lib\site-packages\numpy\core\fromnumeric.py:2918, in _
min(a, axis, out, keepdims, initial, where)
    2902 @array_function_dispatch(_amin_dispatcher)
    2903 def amin(a, axis=None, out=None, keepdims=np._NoValue, initial=np._NoValue,
    2904         where=np._NoValue):
    2905     """
    2906     Return the minimum of an array or minimum along an axis.
    2907     (...)
    2916     """
    2917     """
-> 2918     return _wrapreduction(a, np.minimum, 'min', axis, None, out,
    2919                          keepdims=keepdims, initial=initial, where=where)

File ~\Documents\codigos de prueba para el proyecto\ANPR rapido\anprsys\lib\site-packages\numpy\core\fromnumeric.py:86, in _
wrapreduction(obj, ufunc, method, axis, dtype, out, **kwargs)
      83     else:
      84         return reduction(axis=axis, out=out, **passkwargs)
----> 86 return ufunc.reduce(obj, axis, dtype, out, **passkwargs)

ValueError: zero-size array to reduction operation minimum which has no identity

In [ ]: plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))

In [38]: keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]

In [19]: location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break

In [20]: location

In [21]: mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

-----
error                                Traceback (most recent call last)
Input In [21], in <cell line: 2>():
      1 mask = np.zeros(gray.shape, np.uint8)
----> 2 new_image = cv2.drawContours(mask, [location], 0,255, -1)
      3 new_image = cv2.bitwise_and(img, img, mask=mask)

error: OpenCV(4.5.4) D:\a\opencv-python\opencv-python\opencv\modules\imgproc\src\drawing.cpp:2598: error: (-215:Assertion failed)
reader.ptr != NULL in function 'cvDrawContours'
```

Debido a estos problemas no se consideró prudente la utilización de este procedimiento para el desarrollo del código del proyecto.

3.3.3. Experimentación 3

En el caso del proyecto “Reconocimiento de Placa de Licencia” (“License Plate Recognition”), se utilizan las librerías OpenCV y Tesseract, si bien el reconocimiento óptico de caracteres cv2 funciona relativamente bien y haya podido detectar correctamente un considerable número de imágenes que fueron procesadas, este proyecto requiere la instalación de Tesseract en el dispositivo en el que será ejecutado para su funcionamiento, además, Tesseract no se ejecutará correctamente si el PATH; la variable del sistema que utiliza el sistema operativo para la búsqueda de ejecutables necesarios desde una línea de comandos o la ventana del terminal de Windows, no se encuentra definida.

Figura 20

Experimentación con el código resultando en problemas al tratar de acceder al ejecutable de tesseract instalado en la computadora donde se realiza la experimentación.

```
In [10]:
pytesseract.pytesseract.tesseract_cmd=r'C:\Program Files\Tesseract-OCR\tesseract.exe' #exe file for using ocr

text=pytesseract.image_to_string(Cropped_loc,lang='eng') #converts image characters to string
print("Number : ",text)

cv2.imshow("Original Image",img)
cv2.waitKey(0)
cv2.destroyAllWindows()

-----
FileNotFoundError                                Traceback (most recent call last)
File ~\Documents\codigos de prueba para el proyecto\Number Plate Recognition-main\nprmsys\lib\site-packages\pytesseract\pytesseract.py:254, in run_tesseract(input_filename, output_filename_base, extension, lang, config, nice, timeout)
    253 try:
--> 254     proc = subprocess.Popen(cmd_args, **subprocess_args())
    255 except OSError as e:

File C:\Python\lib\subprocess.py:966, in Popen.__init__(self, args, bufsize, executable, stdin, stdout, stderr, preexec_fn, close_fds, shell, cwd, env, universal_newlines, startupinfo, creationflags, restore_signals, start_new_session, pass_fds, user, group, extra_groups, encoding, errors, text, umask, pipesize)
    963     self.stderr = io.TextIOWrapper(self.stderr,
    964                                   encoding=encoding, errors=errors)
--> 966     self._execute_child(args, executable, preexec_fn, close_fds,
    967                        pass_fds, cwd, env,
    968                        startupinfo, creationflags, shell,
    969                        p2cread, p2cwrite,
    970                        c2pread, c2pwrite,
    971                        errread, errwrite,
    972                        restore_signals,
    973                        gid, gids, uid, umask,
    974                        start_new_session)
    975 except:
    976     # Cleanup if the child failed starting.
```

```

1445 # handles that only the child should have open. You need
(...)
1448 # pipe will not close when the child process exits and the
1449 # ReadFile will hang.

FileNotFoundError: [WinError 2] El sistema no puede encontrar el archivo especificado

During handling of the above exception, another exception occurred:

TesseractNotFoundError                                Traceback (most recent call last)
Input In [10], in <cell line: 3>()
      1 pytesseract.pytesseract.tesseract_cmd=r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe' #exe file for using ocr
----> 3 text=pytesseract.image_to_string(Cropped_loc,lang='eng') #converts image characters to string
      4 print("Number :",text)
      6 cv2.imshow("Original Image",img)

File ~\Documents\codigos de prueba para el proyecto\Number_Plate_Recognition-main\nprmsys\lib\site-packages\pytesseract\pytesse
ract.py:416, in image_to_string(image, lang, config, nice, output_type, timeout)
    411 """
    412 Returns the result of a Tesseract OCR run on the provided image to string
    413 """
    414 args = [image, 'txt', lang, config, nice, timeout]
--> 416 return {
    417     Output.BYTES: lambda: run_and_get_output(*(args + [True])),
    418     Output.DICT: lambda: {'text': run_and_get_output(*args)},
    419     Output.STRING: lambda: run_and_get_output(*args),
    420 }[output_type]()

File ~\Documents\codigos de prueba para el proyecto\Number_Plate_Recognition-main\nprmsys\lib\site-packages\pytesseract\pytesse
ract.py:419, in image_to_string.<locals>.<lambda>()
    411 """
    412 Returns the result of a Tesseract OCR run on the provided image to string
    413 """
    414 args = [image, 'txt', lang, config, nice, timeout]
    416 return {
    417     Output.BYTES: lambda: run_and_get_output(*(args + [True])),
    418     Output.DICT: lambda: {'text': run_and_get_output(*args)},
--> 419     Output.STRING: lambda: run_and_get_output(*args),
    420 }[output_type]()

File ~\Documents\codigos de prueba para el proyecto\Number_Plate_Recognition-main\nprmsys\lib\site-packages\pytesseract\pytesse
ract.py:286, in run_and_get_output(image, extension, lang, config, nice, timeout, return_bytes)
    275 with save(image) as (temp_name, input_filename):
    276     kwargs = {
    277         'input_filename': input_filename,
    278         'output_filename_base': temp_name,
    (...)
    283         'timeout': timeout,
    284     }
--> 286     run_tesseract(**kwargs)
    287     filename = kwargs['output_filename_base'] + extsep + extension
    288     with open(filename, 'rb') as output_file:

File ~\Documents\codigos de prueba para el proyecto\Number_Plate_Recognition-main\nprmsys\lib\site-packages\pytesseract\pytesse
ract.py:258, in run_tesseract(input_filename, output_filename_base, extension, lang, config, nice, timeout)
    256     if e.errno != ENOENT:
    257         raise e
--> 258     raise TesseractNotFoundError()
    260 with timeout_manager(proc, timeout) as error_string:
    261     if proc.returncode:

TesseractNotFoundError: C:\\Program Files\\Tesseract-OCR\\tesseract.exe is not installed or it's not in your PATH. See README f
ile for more information.

```

Estos problemas dificultan la portabilidad, ya que se necesita que se desarrolle una aplicación web que permita su acceso desde varios dispositivos.

3.3.4. Experimentación 4

En el proyecto “Como Entrenar YOLOv5 para detectar números de placas y extracción de texto usando OCR” (How to Train YOLOv5 to detect vehicle number plate and extraction of text using OCR) se realizaron pruebas siguiendo las instrucciones proveídas en el cuaderno de Jupyter del proyecto. Este proyecto hace uso del repositorio YOLOv5 además de indicar que se requiere el uso de la Unidad de Procesamiento de Gráficos o GPU para el procesamiento.

Figura 21

Experimentación con el código en Visual Code, requiere la clonación del repositorio YOLOv5 y su entrenamiento para el caso específico a ser aplicado.

```
Install Dependencies
Remember to choose GPU in Runtime if not already selected. Runtime --> Change Runtime Type --> Hardware accelerator --> GPU

# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
!git reset --hard 886f3c8d839579afac6896ac1742967ae395b6

Cloning into 'yolov5'...
remote: Enumerating objects: 10628, done.
remote: total 10628 (delta 0), reused 0 (delta 0), pack-reused 10628
Receiving objects: 100% (10628/10628), 10.85 MiB | 29.46 MiB/s, done.
Resolving deltas: 100% (7330/7330), done.
/content/yolov5
HEAD is now at 886f3c8 after autoanchor reorder (#3421)

# install dependencies as necessary
!pip install -qr requirements.txt # install dependencies (ignore errors)
!import torch

from IPython.display import Image, clear_output # to display images
from utils.google_utils import url_to_file, download # to download models/datasets

# clear output()
print("Setup complete. Using torch %s %s" % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))

Setup complete. Using torch 1.10.0+cu111 - cudaDeviceProperties(name='Tesla P100-PCIE-16GB', major=0, minor=0, total memory: 16280MB, multi-processor count=56)
```

Este proyecto requiere el entrenamiento de la red neuronal para que pueda reconocer correctamente las placas vehiculares y supervisar su entrenamiento, este proceso llega a consumir tiempo y se deben identificar errores al momento de evaluar el rendimiento de la red neuronal YOLOv5.

3.3.5. Experimentación 5

Se utilizó el material facilitado en el blog OMES en su artículo “Reconocimiento Óptico de Caracteres (OCR) con EASYOCR | Python – OpenCV” el cual explica la estructura y programación del código utilizado para el reconocimiento de caracteres en imágenes, utilizando el módulo de Python EasyOCR para el reconocimiento de caracteres, y el uso de la librería de código abierto OpenCV de visión artificial.

Al iniciar la escritura del código para realizar las pruebas se instaló la librería de visión artificial OpenCV, la librería de cálculo numérico y análisis de datos Numpy, la librería de procesamiento y edición de imágenes Pillow, el paquete de visión de computadora y reconocimiento óptico de caracteres EasyOCR, y la librería de lectura y escritura de datos Imageio.

Estas librerías son importadas al proyecto para su uso en las funciones del proyecto.

Figura 22

Experimentación con el código utilizando el servicio Deepnote en lugar de Jupyter Notebook

```
!pip install opencv-python
!pip install numpy
!pip install Pillow
!pip install easyocr
!pip install matplotlib
!pip install imageio

import imageio
import easyocr
import cv2
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
```

Una vez instaladas las librerías utilizadas para la ejecución de las funciones para el análisis y procesamiento de imágenes, se procede a instalar el paquete Anvil Uplink, el cual permite que el servicio Anvil pueda acceder a funciones específicas en el código.

Una vez instalado, se importa el paquete Anvil Uplink y se procede a conectar con el servidor de Anvil a través de una clave de acceso única proporcionada por Anvil.

Figura 23

Instalación del paquete Anvil Uplink e importación.

```
[2] !pip install anvil-uplink
✓
Delete this cell and move package "anvil-uplink" to requirements.txt

Collecting anvil-uplink
  Downloading anvil_uplink-0.4.0-py2.py3-none-any.whl (88 kB)
  _____ 88.5/88.5 KB 6.8 MB/s eta 0:00:00
Requirement already satisfied: six in /shared-libs/python3.7/py-core/lib/python3.7/site-packages (from anvil-uplink) (1.16.0)
Requirement already satisfied: future in /shared-libs/python3.7/py/lib/python3.7/site-packages (from anvil-uplink) (0.18.2)
Collecting argparse
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Collecting ws4py
  Downloading ws4py-0.5.1.tar.gz (51 kB)
  _____ 51.4/51.4 KB 9.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: ws4py
  Building wheel for ws4py (setup.py) ... done
  Created wheel for ws4py: filename=ws4py-0.5.1-py3-none-any.whl size=45215 sha256=de080016330f2944108327e3b9259822d3a9285d4925595e
  Stored in directory: /root/.cache/pip/wheels/29/ea/7d/3410aa0aa0e4402ead9a7a97ab2214804887e0f5c2b76f0c96
Successfully built ws4py
Installing collected packages: ws4py, argparse, anvil-uplink
Successfully installed anvil-uplink-0.4.0 argparse-1.4.0 ws4py-0.5.1
WARNING: You are using pip version 22.0.4; however, version 22.2.2 is available.
You should consider upgrading via the '/root/.venv/bin/python -m pip install --upgrade pip' command.

[3] import anvil.server

anvil.server.connect("")

Connecting to wss://anvil.works/uplink
Anvil websocket open
Connected to "listocar" as SERVER
```


En la siguiente línea de código contenida en la función “`análisis2`”, se indica en “`with anvil.media.TempFile(file) as image`” que el archivo enviado por el aplicativo web de Anvil se almacene en la variable “`img`”. A continuación, la línea “`reader= easyocr.Reader(["es"], gpu=False)`” es para cargar un modelo en la memoria. Lleva algo de tiempo, pero necesita ejecutarse sólo una vez.

Se definen las variables a utilizar, en este caso “`reader`” llama a EasyOCR y especifica el idioma en el que se realizará el Reconocimiento Óptico de Caracteres, “`es`” indicando el idioma español, seguido de “`gpu=False`” indicando que se utilizará la Unidad Central de Procesamiento o CPU en lugar de la Unidad de Procesamiento de Gráficos o GPU. “`easyocr.Reader`” indica que se utilizará esta función para detectar y leer el texto en la imagen en el idioma especificado.

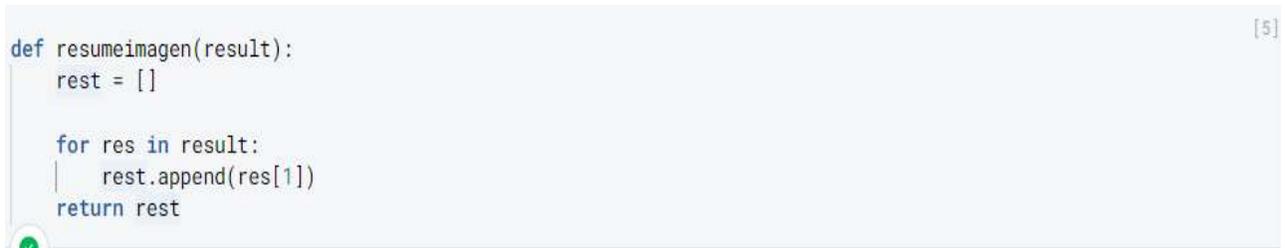
Luego de esta línea tenemos la siguiente variable declarada “`image`” siendo definida como “`image = cv2.imread(img)`” indicando que utilizando la librería de procesamiento de imágenes OpenCV, utilizando el módulo “`cv2.imread()`” que carga una imagen de un archivo especificado, en este caso de la variable “`img`” almacena el archivo enviado por el aplicativo web.

En la siguiente línea “`result = reader.readtext(image, paragraph=False)`” la variable “`result`” llama al método de EasyOCR “`reader.readtext`” indicando que se debe leer los caracteres de texto en la imagen almacenada en la variable “`image`” procesada por la librería OpenCV, se indica con “`paragraph=False`” que las líneas de texto encontradas en la imagen deben leerse por separado y no como un párrafo.

Figura 25

Experimentación con la función de aprendizaje de la lista de resultados obtenidos por el reconocimiento de texto.

```
def resumeimagen(result):  
    rest = []  
  
    for res in result:  
        rest.append(res[1])  
    return rest
```

A screenshot of a code editor showing a Python function named 'resumeimagen'. The function takes a parameter 'result' and returns a list 'rest'. It initializes 'rest' as an empty list, then iterates over 'result' and appends the second element of each item to 'rest'. The code is highlighted in a light blue box with a green checkmark in the bottom left corner and a '[5]' in the top right corner.

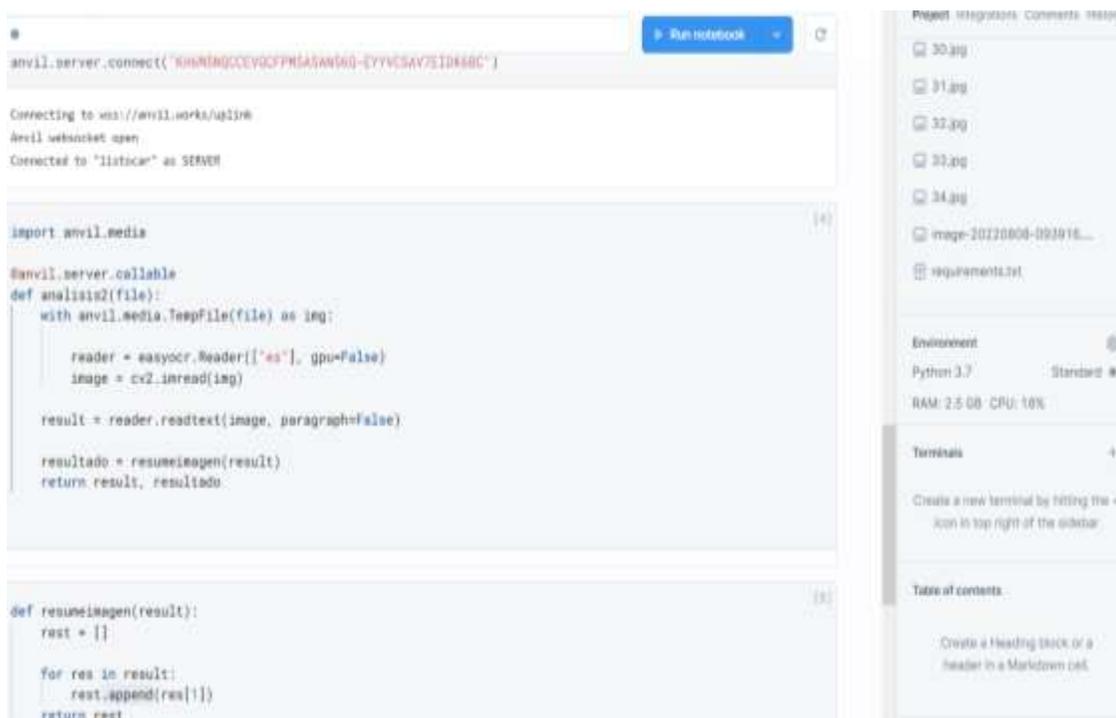
La variable “resultado” llama a la función “resumeimagen(result)” que recibe los datos almacenados en la variable “result”. En la función “resumeimagen(result)” se declara la variable “rest = []” donde los corchetes de índice ([]) se utilizan para definir "literales de lista", lo que le permite declarar una lista y su contenido en el programa, luego se indica que para la variable “res” se almacenará el resultado obtenido en la variable “result”, en este caso, “rest.append” utiliza el método “append” de Python.

Es un método que tiene toda lista de Python el cual permite agregar un elemento al final de la lista. Esto es debido a que al procesar el texto de la imagen enviada por el aplicativo web, se le instruyó a EasyOCR que cada línea de texto en la imagen sea procesada por separado en lugar de procesarse como un párrafo, por esto los resultados se almacenan en la variable “res”, esta función resulta en el valor devuelto en la variable “rest”. La función “analisis2” devuelve las variables “result” y “resultado”.

Debido a la simplificación de los procesos y la facilidad de uso del código, se decidió utilizar esta propuesta para el código del proyecto, al ser el uso de una Unidad de Procesamiento de Gráficos optativa, pudiendo utilizarse la Unidad Central de Procesamiento en su lugar, permitiendo la utilización del servicio Deepnote con el entorno virtual proveído por este servicio.

Figura 26

Experimentación con la conexión entre el aplicativo web y la llamada de las funciones en el código



3.4. Utilización de la metodología Mobile-D.

Una vez que se han investigado los distintos procedimientos utilizados para el reconocimiento óptico de caracteres y procesamiento de imágenes dirigidos para el desarrollo de la aplicación y las herramientas digitales elegidas para producir el aplicativo web, se siguen los pasos para la organización de las tareas a realizar utilizando la metodología Mobile-D.

3.4.1. Análisis.

3.4.1.1. Obtención de Requerimientos.

Utilizando las Historias de Usuarios obtenidas gracias a las entrevistas realizadas el personal y al gerente del establecimiento, se pudo clarificar los requerimientos funcionales, y con la investigación de las herramientas digitales

3.4.1.2. Clasificación de los Requerimientos.

Los requerimientos iniciales identificados fueron los siguientes:

Requisitos funcionales:

- El registro debe permitir seleccionar la fecha y hora del ingreso del vehículo.
- Las placas de los vehículos pueden ser registradas a través de fotografías.
- Alternativamente, las placas de los vehículos se pueden ingresar manualmente.
- El registro debe permitir la selección de la descripción del vehículo (tipo, marca, color).
- De haber sido ingresada la placa antes, debe poder mostrar los detalles del vehículo.
- Debe permitir asignar los servicios requeridos por el cliente al vehículo.
- Debe poderse listar el valor a pagar del cliente y un marcador para indicar si el monto a pagar del vehículo fue cancelado.
- El registro debe permitir crear anotaciones u observaciones con respecto al vehículo a registrar.

- Debe existir un método de búsqueda que permita visualizar los vehículos registrados en un espacio de tiempo determinado por el usuario.
- Debe poder filtrar los resultados utilizando la placa del vehículo opcionalmente.
- Se debe poder emitir un documento con extensión .pdf con las búsquedas realizadas como un reporte.

Requisitos no funcionales:

- El registro debe ser accesible desde el navegador de la computadora de oficina y el del teléfono celular del personal del establecimiento.
- Debe poder instalarse un aplicativo que permita acceder directamente al registro de vehículos.
- Según la función del personal en el establecimiento, se permitirá o restringirá el acceso a funciones o información del registro con el uso de roles manejados por el administrador del sistema.
- Los usuarios podrán registrarse en el aplicativo, sin embargo, el acceso al registro será habilitado solo por el administrador.
- El registro será accesible mediante conexión de internet, permitiendo al gerente mantener supervisado el registro de los vehículos de manera remota de ser necesario.

3.4.1.3. Personalización del Servicio

El aplicativo web debe ser fácil de entender, con iconos y textos claros, botones definidos y con los colores acorde al logotipo del establecimiento.

3.4.2. Diseño.

3.4.2.1. Definir el escenario.

El aplicativo funcionara con acceso a internet ya que las herramientas elegidas prestan sus servicios de hosting y almacenamiento de datos en la nube, permitiendo que el aplicativo sea portable y accesible siempre y cuando se cuente con acceso a internet, sea de puntos de acceso por cable LAN, Wi-Fi, o de datos móviles utilizando una tarjeta SIM de un proveedor de telefonía celular.

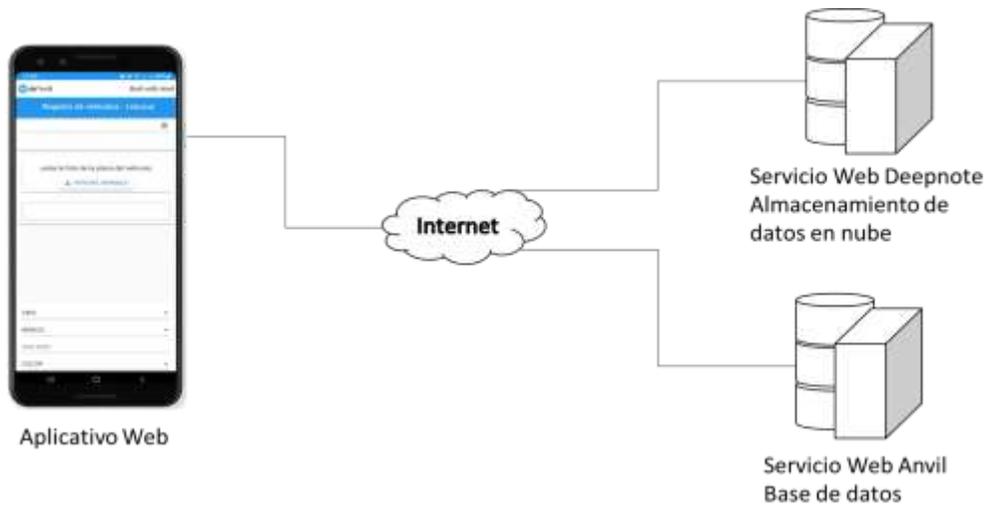
De esta manera, el aplicativo será accesible desde las computadoras utilizadas en la oficina del establecimiento, el teléfono móvil de la oficina, los dispositivos móviles del personal, y los dispositivos personales del gerente.

3.4.2.2. Estructura del Software

El sistema propuesto se encuentra formado por 3 componentes principales, un aplicativo web que se conecta al servicio web a través del internet, un servicio de nube que contiene un archivo IPYNB, el cual se ejecuta y procesa las imágenes subidas al aplicativo y que devuelve el texto procesado como respuesta, y un dispositivo sea móvil o de ordenador que permite acceder al aplicativo, el cual puede ser accedido a través del navegador o instalando el aplicativo en el ordenador.

Figura 27

Abstracción gráfica del diseño del sistema simplificado

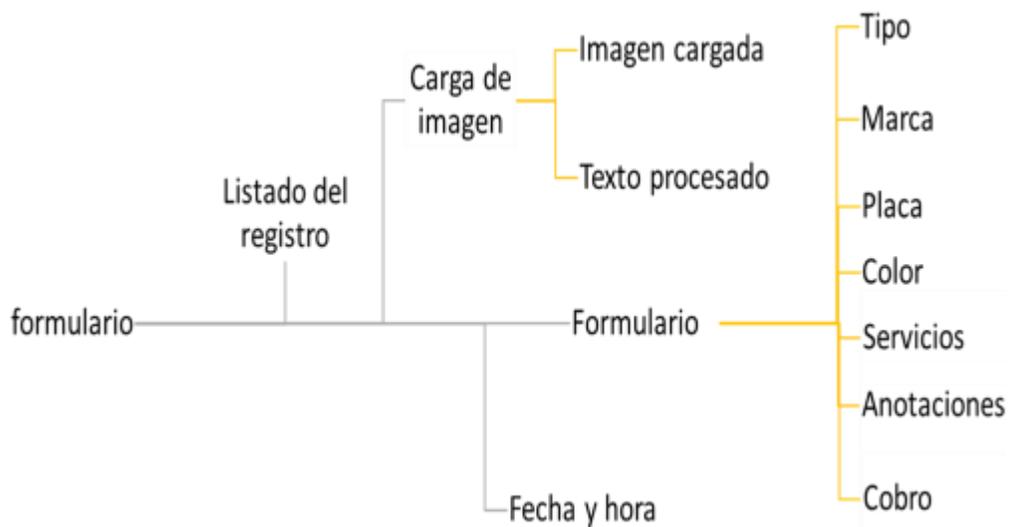


3.4.2.2.1. Esquema de navegabilidad

Se presenta un esquema de navegabilidad para identificar la interfaz que debe existir en el proyecto, utilizando la retroalimentación proveída por los usuarios objetivo.

Figura 28

Esquema gráfico de la navegabilidad de la interfaz del registro vehicular



3.4.3. Desarrollo.

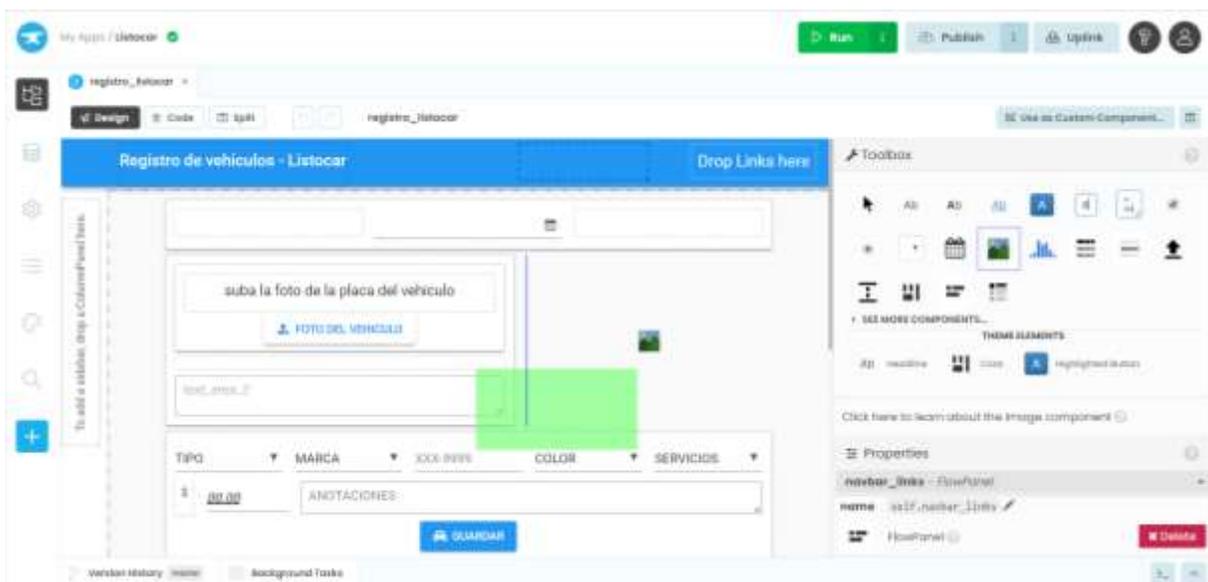
3.4.3.1. Codificación.

3.4.3.1.1. Aplicativo Web desarrollado en Anvil.

El modelo de construcción de interfaces a través de Anvil Works utilizando el método “drag-and-drop” o “arrastrar y soltar” para el diseño de formularios para el aplicativo web.

Figura 29

Diseño y construcción de la interfaz en Anvil Works



Anvil presenta a su derecha los “widgets” o “artilugios” a utilizar para formar la interfaz del aplicativo web en el espacio a la izquierda, su tamaño y ubicación son fáciles de modificar en el espacio de trabajo, se pueden añadir “widgets” personalizados de ser necesario. bajo la ventana de “widgets” se puede visualizar las propiedades del “widget” seleccionado, funciones y características, las cuales pueden ser editadas o modificadas a la preferencia del usuario.

Figura 30

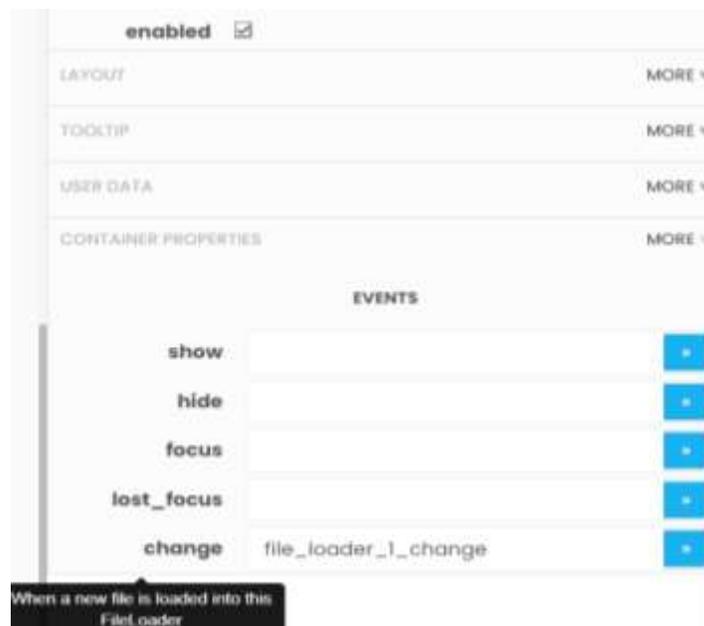
Interfaz de Propiedades de los widgets



Para esta interfaz se ubica en la parte superior un botón que desplegará un calendario y reloj para indicar la fecha y hora de llegada del vehículo. En el caso del botón de carga con el texto “foto del vehículo”, en sus propiedades se indica que, al realizarse un cambio de estado, siendo este la carga de una imagen, se ejecuten las funciones en su código.

Figura 31

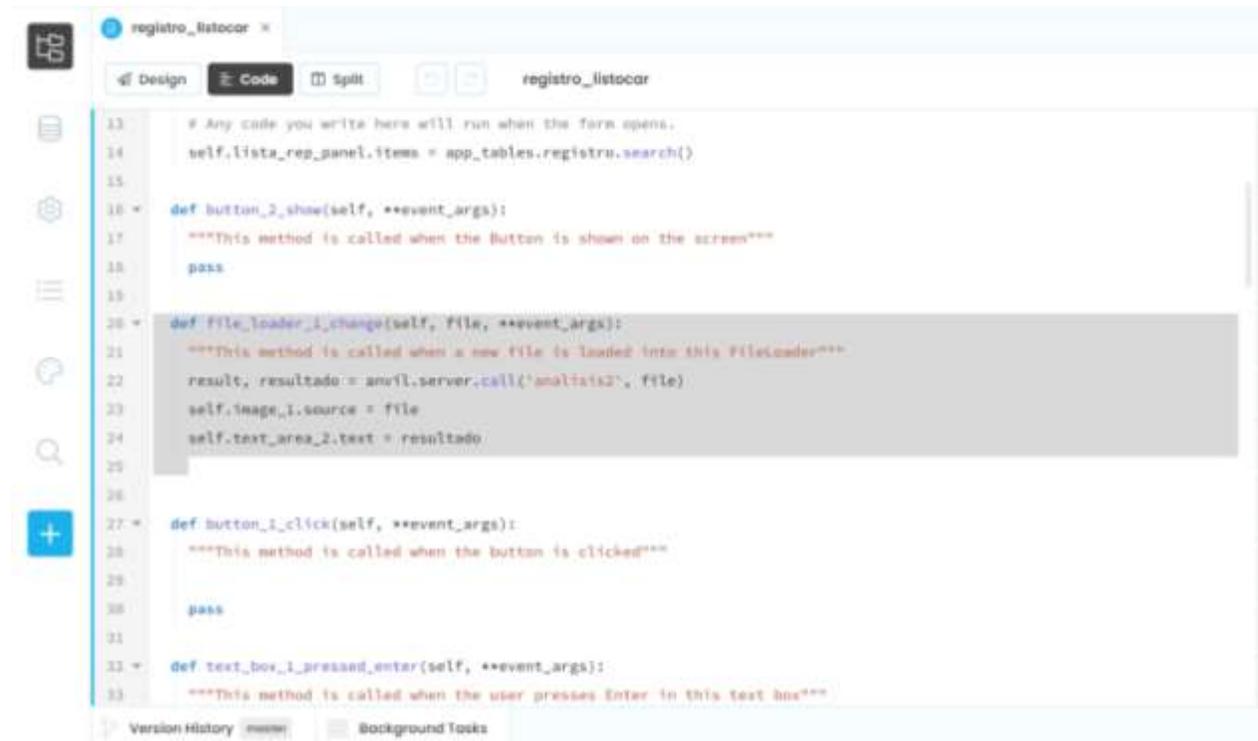
Eventos en la interfaz de Propiedades de los widgets



Junto a el botón de “Diseño” o “Design” se encuentra el botón de “Código” o “Code”, donde podemos acceder a un editor de texto con funciones de los “widgets” seleccionados para la interfaz, para codificar sus funciones según sea necesario.

Figura 32

Editor de código en Anvil para los widgets y otras funciones



```
13 # Any code you write here will run when the form opens.
14 self.lista_rep_panel.items = app_tables.registro.search()
15
16 def button_2_show(self, **event_args):
17     """This method is called when the Button is shown on the screen"""
18     pass
19
20 def file_loader_1_change(self, file, **event_args):
21     """This method is called when a new file is loaded into this Fileloader"""
22     result, resultado:= anvil.server.call('analisis2', file)
23     self.image_1.source = file
24     self.text_area_2.text = resultado.
25
26
27 def button_1_click(self, **event_args):
28     """This method is called when the button is clicked"""
29
30     pass
31
32 def text_box_1_pressed_enter(self, **event_args):
33     """This method is called when the user presses Enter in this text box"""
```

En el caso de el botón de carga para las imágenes, su función es definida como “file_loader__change”, el cual cuando la imagen a analizar es cargada, se llama la variable “resultado”, anidadas en el código dentro del archivo “.ipynb” cargado en el servicio Deepnote, utilizando el servicio Anvil-Uplink permite el llamar los datos en la variable “resultado”, los cuales son mostrados a través de los widgets “self.image_1.source” el cual mostrará la imagen cargada en la variable “file”, y el resultado de la función “analisis2” en la variable “resultado” el cual se mostrará en la caja de texto “self.text_area_2.text”.

Figura 33

Diseño del formulario para los datos de los vehículos

The image shows a web form for entering vehicle data. At the top, there are five dropdown menus: TIPO, MARCA, XXX-9999 (placa), COLOR, and SERVICIOS. Below these is a text input field for 'ANOTACIONES' and a numeric input field for '\$ 00.00'. A blue 'GUARDAR' button with a car icon is centered below the inputs. Below the main form is a repeating panel with five rows. Each row contains five dropdown menus labeled 'tipo_box', 'marca_box', 'placa_box', 'color_box', and 'serv_box'. Below each row is a sub-form with a numeric input for '\$ cobro_1', a checkbox for 'PAGADO', a text input for 'detalle_area', and a date input for 'fecha_label' with a red calendar icon.

Para el Registro del listado de vehículos, se utilizó el “widget” “repeating panel”, el cual se añadieron 4 listas colapsables para la selección del tipo de vehículo, marca del vehículo, color del vehículo y servicios solicitados, una caja de texto para la placa, una caja de texto para ingresar el valor estimado a pagar, y una caja de texto en el que se ingresaran anotaciones que se requieran, y un botón para guardar los datos ingresados en la tabla de la base de datos facilitados por el servicio de Anvil, y el panel repetitivo mostrará los vehículos ingresados en la base de datos a través del ingreso de datos al formulario presentado.

Figura 34

Código de la asignación de variables para los campos del registro y datos de la tabla en la base de datos

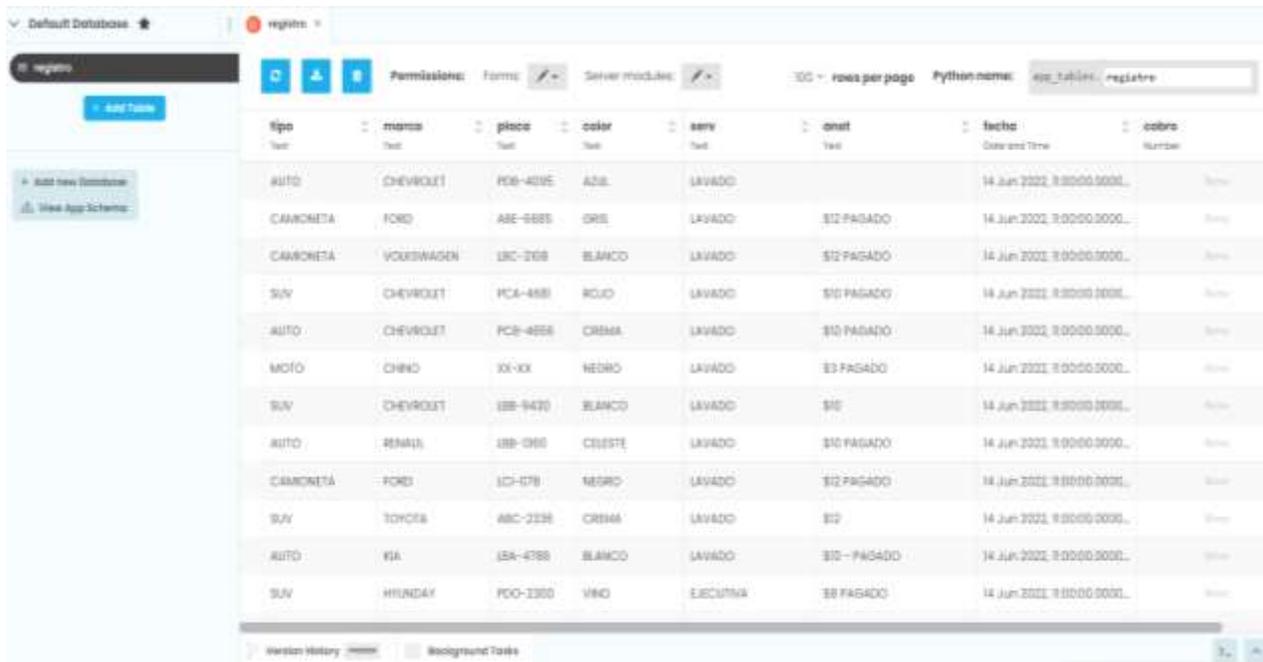
```
def guardar_vehic_click(self, **event_args):
    """This method is called when the button is clicked"""
    tipo_in = self.list_tipo.selected_value
    marca_in = self.list_marca.selected_value
    placa_in = self.placa_ingr.text
    color_in = self.list_color.selected_value
    serv_in = self.list_serv.selected_value
    anota_in = self.detalle_ingr.text
    fecha_in = self.date_picker_1.date
    cobro_in = self.cobro_box.text

    app_tables.registro.add_row(serv = serv_in,
                                placa = placa_in,
                                anot = anota_in,
                                tipo = tipo_in,
                                color = color_in,
                                marca = marca_in,
                                fecha = fecha_in,
                                cobro = cobro_in
                                )
    self.lista_rep_panel.items = app_tables.registro.search()
```

Anvil permite la creación de tablas en la base de datos que provee Anvil para el desarrollo de las aplicaciones, estas tablas almacenan los datos de los vehículos registrados, el valor pagado y su fecha de ingreso en el registro.

Figura 35

Tabla en la base de datos utilizada para almacenar los datos de los vehículos registrados



tipo	marca	placa	color	serv	onet	fecha	cobro
Text	Text	Text	Text	Text	Text	Date and Time	Number
AUTO	CHEVROLET	PCB-409E	AZUL	LAVADO		14 Jun 2022 3:00:00.000L	0.00
CAMIONETA	FORD	ARE-588E	GRIS	LAVADO	\$12 PAGADO	14 Jun 2022 3:00:00.000L	0.00
CAMIONETA	VOLKSWAGEN	IRC-208	BLANCO	LAVADO	\$10 PAGADO	14 Jun 2022 3:00:00.000L	0.00
SUV	CHEVROLET	PCA-488	ROJO	LAVADO	\$10 PAGADO	14 Jun 2022 3:00:00.000L	0.00
AUTO	CHEVROLET	PCB-405E	ORINA	LAVADO	\$10 PAGADO	14 Jun 2022 3:00:00.000L	0.00
MOTO	CHINO	XX-XX	NEGRO	LAVADO	\$3 PAGADO	14 Jun 2022 3:00:00.000L	0.00
SUV	CHEVROLET	IBB-942D	BLANCO	LAVADO	\$10	14 Jun 2022 3:00:00.000L	0.00
AUTO	RENAULT	IBB-086	CELESTE	LAVADO	\$10 PAGADO	14 Jun 2022 3:00:00.000L	0.00
CAMIONETA	FORD	ICJ-078	NEGRO	LAVADO	\$10 PAGADO	14 Jun 2022 3:00:00.000L	0.00
SUV	TOYOTA	ABC-233E	ORINA	LAVADO	\$10	14 Jun 2022 3:00:00.000L	0.00
AUTO	KIA	ISA-478E	BLANCO	LAVADO	\$10 - PAGADO	14 Jun 2022 3:00:00.000L	0.00
SUV	HYUNDAI	PCO-230D	VINO	EJECUTIVA	\$8 PAGADO	14 Jun 2022 3:00:00.000L	0.00

3.4.3.2. Pruebas Unitarias.

Anvil permite publicar versiones de la aplicación en cualquier estado de su desarrollo según la preferencia del usuario.

A continuación, se presentan capturas de pantalla del diseño de la interfaz de la aplicación a través de un dispositivo móvil para el usuario con su descripción.

Figura 36

Captura de pantalla y descripción de la interfaz 1

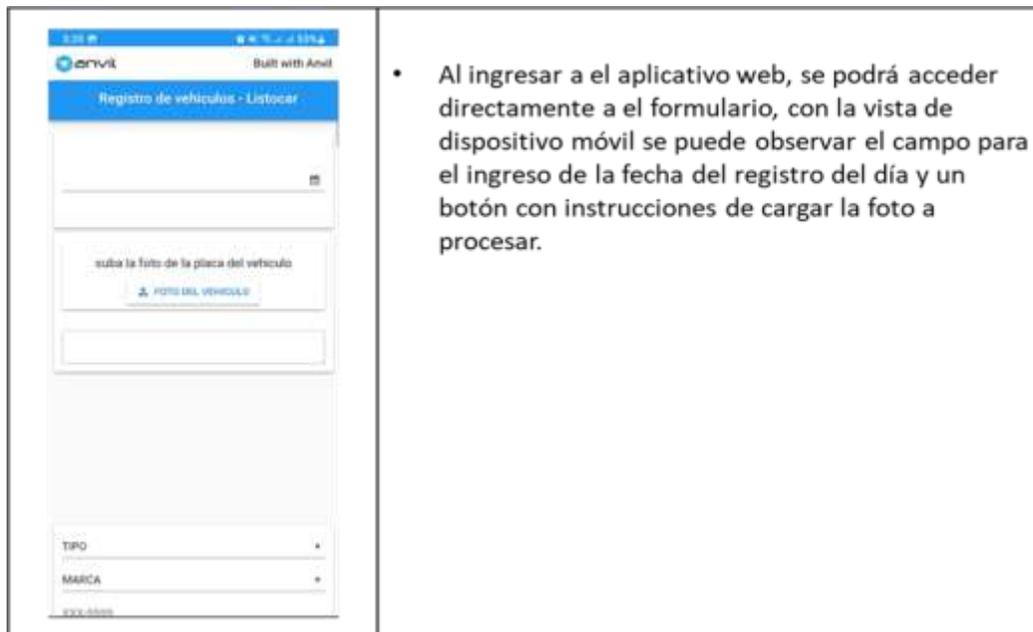


Figura 37

Captura de pantalla y descripción de la interfaz 2

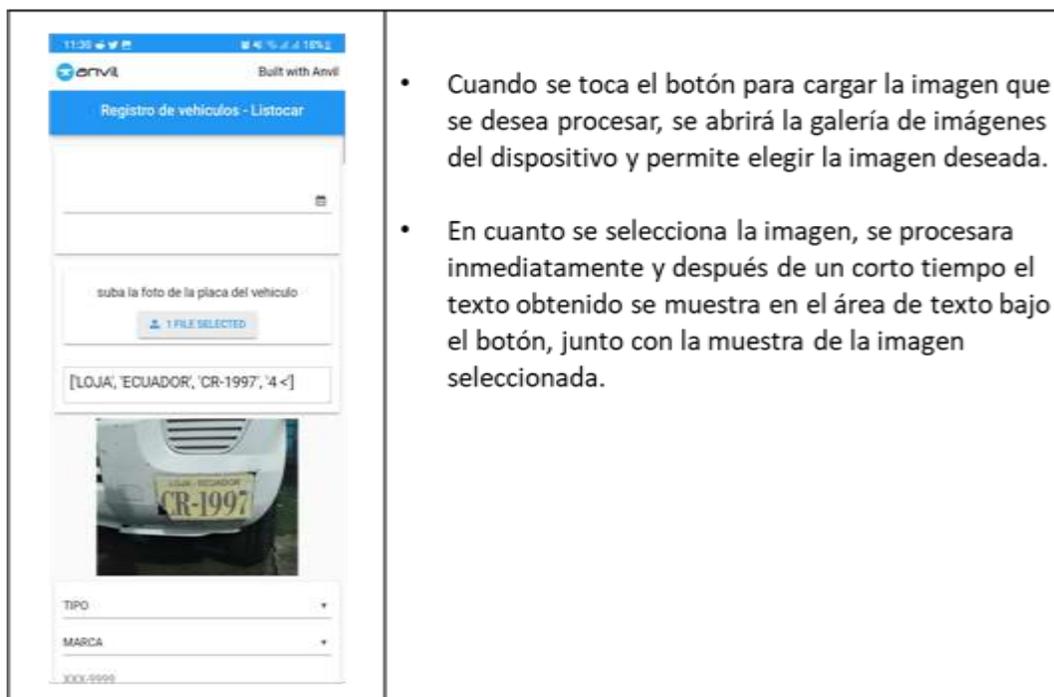


Figura 38

Captura de pantalla y descripción de la interfaz 3

	<ul style="list-style-type: none">• Bajo el área donde se da la opción de procesamiento de imágenes, se encuentra el formulario para ingresar la descripción del vehículo recibido en el establecimiento.• Se despliegan listas colapsables de cada característica, el tipo de vehículo, la marca del vehículo, un campo de ingreso manual de la placa, color del vehículo, el servicio o servicios solicitados, el valor a pagar y un área de texto para anotaciones importantes.
---	---

Figura 39

Captura de pantalla y descripción de la interfaz 4

	<ul style="list-style-type: none">• Bajo el área donde se da la opción de procesamiento de imágenes, se encuentra el formulario para ingresar la descripción del vehículo recibido en el establecimiento.• Se despliegan listas colapsables de cada característica, el tipo de vehículo, la marca del vehículo, un campo de ingreso manual de la placa, color del vehículo, el servicio o servicios solicitados, el valor a pagar y un área de texto para anotaciones importantes.
---	---

3.4.4. Pruebas de Funcionamiento.

3.4.4.1. Emulación y Simulación.

3.4.4.1.1. Simulación de la Interfaz.

Anvil permite la emulación del aplicativo web durante cualquier etapa del desarrollo de la interfaz y/o de las funcionalidades del aplicativo, desplegada a preferencia del usuario de entre las 3 opciones: Vista en pantalla dividida dentro de la interfaz de desarrollo, maximizar la vista en la interfaz de desarrollo, o en una nueva pestaña, para visualizar el comportamiento de la interfaz del aplicativo web.

Figura 40

Captura de la interfaz de desarrollo del aplicativo web, el botón Run utilizado para la simulación.

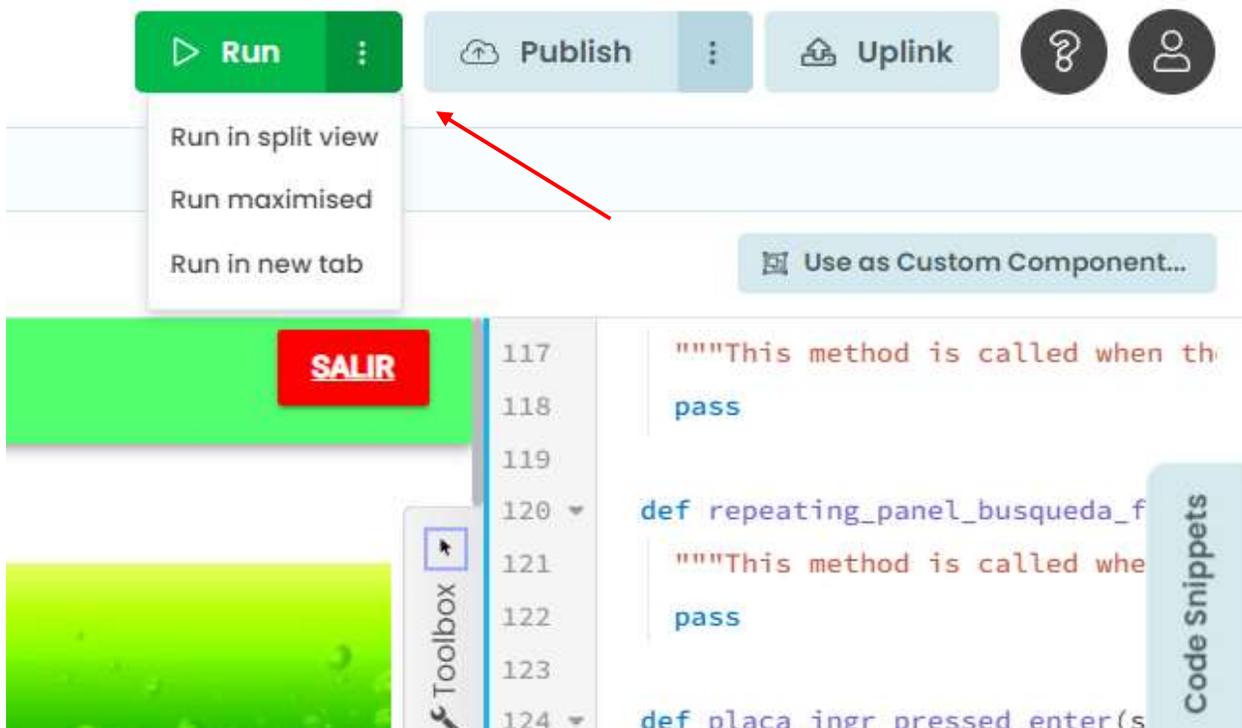


Figura 41

Captura de la interfaz de desarrollo del aplicativo web, presentado en pantalla dividida para la simulación.

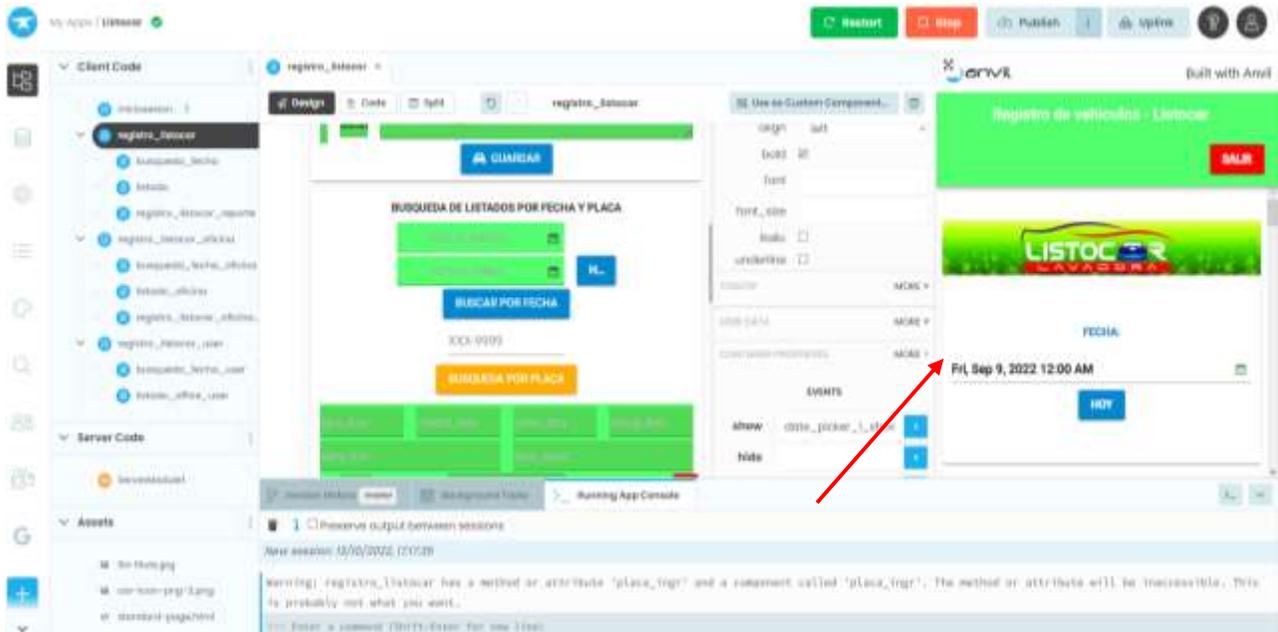


Figura 42

Captura de la interfaz de desarrollo del aplicativo web, maximizada para la simulación.

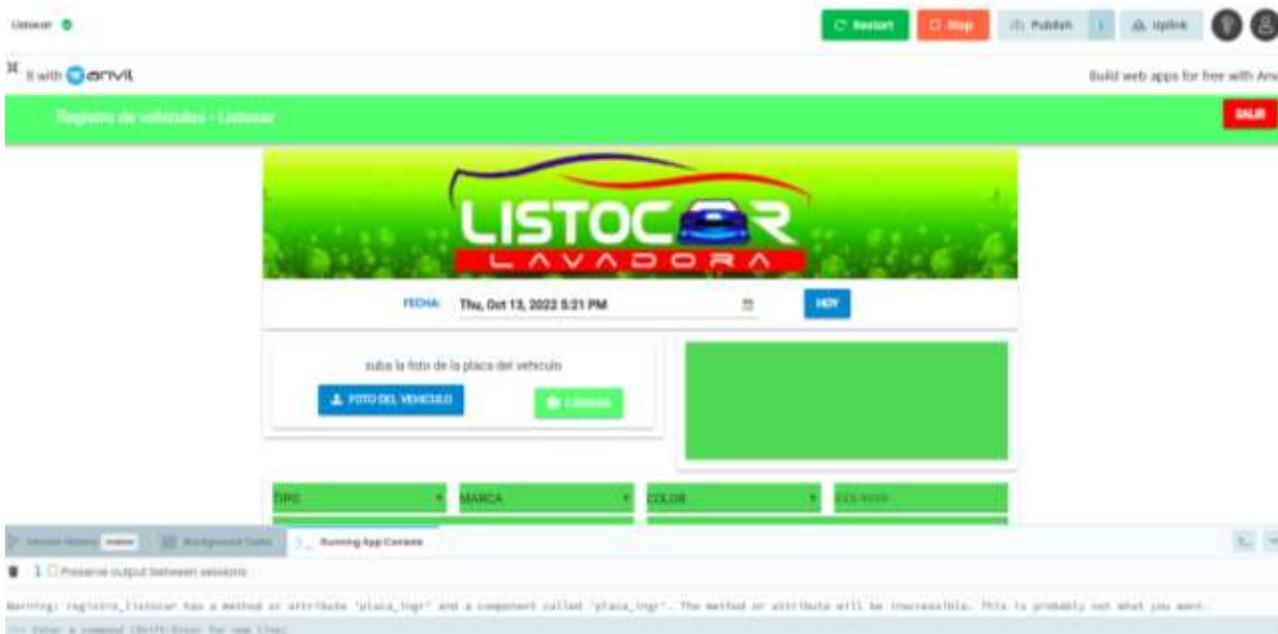


Figura 43

Captura de la interfaz de desarrollo del aplicativo web, presentado en una nueva pestaña para la simulación.



El aplicativo web es accesible a través de cualquier dispositivo utilizando el navegador de internet de preferencia, utilizando la URL personalizable. Al momento de su lanzamiento la URL en uso para el registro es <https://LISTOCAR-registro.anvil.app> que permite el acceso al registro en cualquier momento.

3.4.4.1.2. Instalación

Anvil permite que, al acceder a la URL del registro, en las opciones del navegador esté disponible la instalación del aplicativo web en el dispositivo de manera simple y rápida. En el caso de los dispositivos móviles, utilizamos para ejemplificar la instalación el navegador Chrome, como se puede visualizar a continuación.

Figura 44

Captura de pantalla del navegador y descripción de los pasos 1

	<ul style="list-style-type: none">• En la vista del navegador del dispositivo usado en el establecimiento se puede apreciar el formulario de registro de vehículos abierto en el navegador Chrome del dispositivo móvil.
---	--

Figura 45

Captura de pantalla del navegador y descripción de los pasos 2

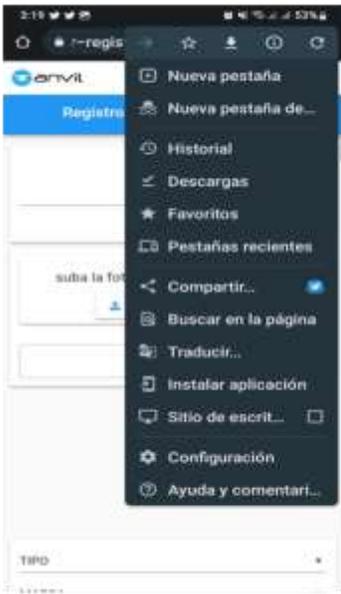
	<ul style="list-style-type: none">• Al desplegar las opciones del navegador, este da la opción “Instalar aplicación” disponible para seleccionar.
---	---

Figura 46

Captura de pantalla del navegador y descripción de los pasos 3

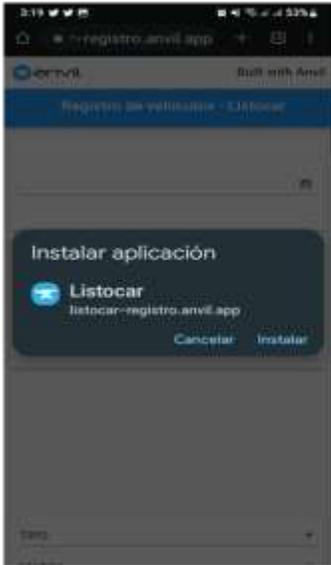
	<ul style="list-style-type: none">• Seleccionando la opción de “Instalar aplicación” se desplegará una ventana que solicitará la autorización para instalar la aplicación en el dispositivo .
---	---

Figura 47

Captura de pantalla del navegador y descripción de los pasos 4

	<ul style="list-style-type: none">• Tras unos momentos la aplicación es instalada en el dispositivo y esta disponible para su uso.
---	--

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Al realizar la investigación con respecto al estado del arte en visión artificial y reconocimiento de objetos y el reconocimiento óptico de caracteres, se han encontrado que existen diversas alternativas según la necesidad del usuario, tanto en cuanto alternativas comerciales como en software libre.
- El aplicativo web multiplataforma para el registro de vehículos y servicios ofrecidos en el local de lavado vehicular ha sido recibido con satisfacción por el gerente del establecimiento y su utilización ha sido inmediata.
- Se implementó un módulo de visión artificial para el reconocimiento de placas vehiculares cargadas desde los archivos del dispositivo o a través de fotografías en teléfonos móviles con cámara.
- Los registros de vehículos son accesibles por los usuarios a través de los diferentes dispositivos utilizados en el establecimiento.
- El aplicativo web utilizado con dispositivos móviles redujo los tiempos utilizados en el registro de vehículos, y agilizar la asignación de tareas al personal que labora en el establecimiento.
- Además de poderse usar con facilidad en las computadoras del establecimiento de forma sencilla y rápida, la administración del negocio de manera remota en conjunto con el sistema de cámaras de seguridad, la gerencia puede estar al tanto del manejo y estado del flujo de trabajo.

RECOMENDACIONES

- Al realizar pruebas y prototipos con códigos que ejecuten script, se debe ejecutar dando el permiso en la consola de administración de shell en el dispositivo donde son ejecutados, de lo contrario, los scripts podrían causar problemas en las pruebas.
- Se recomienda que para el procesamiento de las imágenes de las placas vehiculares las fotos sean tomadas con claridad antes de cargar la foto del vehículo al aplicativo de registro.
- Este proyecto puede ser utilizado para otros tipos de establecimientos, como estacionamientos de vehículos privados o públicos, patios de vehículos, etc. con pocos ajustes necesarios al proyecto presentado.

BIBLIOGRAFÍA

Reconocimiento Óptico de Caracteres (OCR) con EASYOCR. (2022, junio 26). OMES.

<https://omes-va.com/easyocr-python-opencv/>

What is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 documentation.

(s/f). Readthedocs.Io. Recuperado el 13 de agosto de 2022, de https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

Advantages and Disadvantages of Optical character Reader (OCR). (2020, diciembre 14).

GeeksforGeeks. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-optical-character-reader-ocr/>

Agile model. (s/f). [Www.javatpoint.com](http://www.javatpoint.com). Recuperado el 12 de agosto de 2022, de

<https://www.javatpoint.com/software-engineering-agile-model>

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J.,

Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>

Amin, F. (2022, Abril 4). *Optical Character Recognition using EasyOCR in Python.* Mlhive.com.

<https://mlhive.com/2022/04/optical-character-recognition-using-easyocr-in-python>

Arvind, N. (2022, Junio 6). *Text Detection using CRAFT text detector.* Analytics Vidhya.

<https://www.analyticsvidhya.com/blog/2022/06/text-detection-using-craft-text-detector/>

Ashtari, H. (2022, mayo 18). *What is a neural network? Definition, working, types, and applications in 2022*. Spiceworks. <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-a-neural-network/>

Bhattacharyya, J. (2020, octubre 12). *Hands-on tutorial on EasyOCR for scene text detection in images*. Analytics India Magazine. <https://analyticsindiamag.com/hands-on-tutorial-on-easyocr-for-scene-text-detection-in-images/>

Boesch, G. (2021, octubre 6). *VGG Very Deep Convolutional Networks (VGGNet) - What you need to know*. Viso.Ai. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>

Brownlee, J. (2019, enero 8). *A gentle introduction to the rectified linear unit (ReLU)*. Machine Learning Mastery. <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>

What is python? Opensource.com. Recuperado el 12 de agosto de 2022, de <https://opensource.com/resources/python>

Calvo, D. (2017, julio 20). *Red Neuronal Convolucional CNN*. Diego Calvo. <https://www.diegocalvo.es/red-neuronal-convolucional/>

Candido, T. (2020, mayo 26). *Reviewing deepnote — the new IDE for data scientists*. Towards Data Science. <https://towardsdatascience.com/reviewing-deepnote-the-new-ide-for-data-scientists-90c3464ebc5e>

Car logo classification. (2020, noviembre 17). Kaggle.com; Kaggle. <https://www.kaggle.com/code/jp0909/car-logo-classification/notebook>

- Coumar, N. (2020, Julio 19). *Optical Character Recognition(OCR) — Image, Opencv, pytesseract and easyocr*. Medium. <https://medium.com/@nandacoumar/optical-character-recognition-ocr-image-opencv-pytesseract-and-easyocr-62603ca4357>
shorturl.at/gjpN1
- Deep learning*. (2019, enero 8). Databricks. <https://www.databricks.com/glossary/deep-learning>
- Do, T. (2022, marzo 21). *Supervised and unsupervised machine learning - explained through real world examples*. Omdena | Building AI Solutions for Real-World Problems; Omdena. <https://omdena.com/blog/supervised-and-unsupervised-machine-learning/>
- Gebrehiwot, A., Hashemi-Beni, L., Thompson, G., Kordjamshidi, P., & Langan, T. E. (2019). Deep convolutional neural network for flood extent mapping using unmanned Aerial Vehicles data. *Sensors (Basel, Switzerland)*, 19(7), 1486. <https://doi.org/10.3390/s19071486>
- Gulati, A. P. (2021, Junio 22). *Text detection from images using EasyOCR: Hands-on guide*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/>
- IBM Cloud Education. (2020a, Junio 3). *What is Artificial Intelligence (AI)?* Ibm.com. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- IBM Cloud Education. (2020b, septiembre 14). *What are Recurrent Neural Networks?* Ibm.com. <https://www.ibm.com/cloud/learn/recurrent-neural-networks>

IBM Cloud Education. (2020c, octubre 20). *What are Convolutional Neural Networks?* Ibm.com.
<https://www.ibm.com/cloud/learn/convolutional-neural-networks>

Introduction to convolutional neural networks architecture. (2022, Junio 20). *ProjectPro*.
<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>

Jaided AI: EasyOCR tutorial. (s/f). Jaided.Ai. Recuperado el 12 de agosto de 2022, de
<https://www.jaided.ai/easyocr/tutorial/>

Jwalapuram, N. (2021, Abril 12). *PyTorch library*. Analytics Vidhya.
<https://www.analyticsvidhya.com/blog/2021/04/a-gentle-introduction-to-pytorch-library/>

Pianalytix-Sneha. (2020, diciembre 12). *Using EasyOCR library for text extraction*. Pianalytix -
Machine Learning. <https://pianalytix.com/using-easyocr-library-for-text-extraction/>

programador clic. (s/f). Programmerclick.com. Recuperado el 15 de agosto de 2022, de
<https://programmerclick.com/article/21771415339/>

Redes neuronales convolucionales son un tipo de redes neuronales. (2022, enero 19). Juan Barrios.
<https://www.juanbarrios.com/redes-neurales-convolucionales/>

Rosebrock, A. (2020, septiembre 14). *Getting started with EasyOCR for Optical Character Recognition*. PyImageSearch. <https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/>

Sami, T. (2021, Julio 15). *“tesseract” vs “keras-OCR” vs “EasyOCR” - MLearning.Ai - medium.*
MLearning.Ai. <https://medium.com/mlearning-ai/tesseract-vs-keras-ocr-vs-easyocr-ec8500b9455b>

Saxena, S. (2021, marzo 16). *LSTM.* Analytics Vidhya.
<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>

Shakhadri, S. A. G. (2021, junio 7). *What is ResNet.* Analytics Vidhya.
<https://www.analyticsvidhya.com/blog/2021/06/build-resnet-from-scratch-with-python/>

Simplilearn. (2021, diciembre 2). *What is PyTorch, and how does it work: All you need to know.*
Simplilearn.com; Simplilearn. <https://www.simplilearn.com/what-is-pytorch-article>

Solegaonkar, V. (2019, septiembre 20). *Introduction to PyTorch.* Towards Data Science.
<https://towardsdatascience.com/introduction-to-py-torch-13189fb30cb3>

SuperAnnotate. (2021, octubre 29). *What is Optical Character Recognition (OCR): Overview and use cases.* SuperAnnotate Blog. <https://blog.superannotate.com/ocr-overview-and-use-cases/>

What is intelligent document processing & how does it work? (s/f). ABBYY. Recuperado el 15 de agosto de 2022, de <https://www.abbyy.com/blog/intelligent-document-processing/>

What is OCR? - The Ultimate Guide to OCR 2022. (2022, abril 26). Klippa.
<https://www.klippa.com/en/blog/information/what-is-ocr/>

What is optical character recognition (OCR)? (2022, enero 5). Ibm.com.
<https://www.ibm.com/cloud/blog/optical-character-recognition>

Why unsupervised Machine Learning is the future of cybersecurity. (2021, septiembre 9).
TechNative. <https://technative.io/why-unsupervised-machine-learning-is-the-future-of-cybersecurity/>

Wikipedia contributors. (2022, agosto 9). *Project Jupyter*. Wikipedia, The Free Encyclopedia.
https://en.wikipedia.org/w/index.php?title=Project_Jupyter&oldid=1103334233

Zelic, F. (2022, agosto 9). *Tesseract OCR in Python with Pytesseract & OpenCV*. Nanonets AI &
Machine Learning Blog. <https://nanonets.com/blog/ocr-with-tesseract/>

(S/f-a). Researchgate.net. Recuperado el 12 de agosto de 2022, de
https://www.researchgate.net/publication/221322054_Mobile-D_An_Agile_Approach_for_Mobile_Application_Development

(S/f-b). Researchgate.net. Recuperado el 13 de agosto de 2022, de
https://www.researchgate.net/figure/Figura-10-Ciclo-de-desarrollo-Mobile-D-reproducido-de-15-La-fase-de-exploracion_fig9_267795011

ANEXOS.

Aplicativo Web de Registro Vehicular

LISTOCAR

Manual de Usuario

I. Introducción.

1. Introducción.

Otorgar soporte a los usuarios del establecimiento de lavado de vehículos LISTOCAR, teniendo un control de la información oportuna de los servicios solicitados por los clientes.

2. Requerimientos.

El uso de este aplicativo puede ser accedido desde cualquier dispositivo con acceso a internet y que disponga de un navegador de internet, siendo opcional la instalación del aplicativo en el dispositivo.

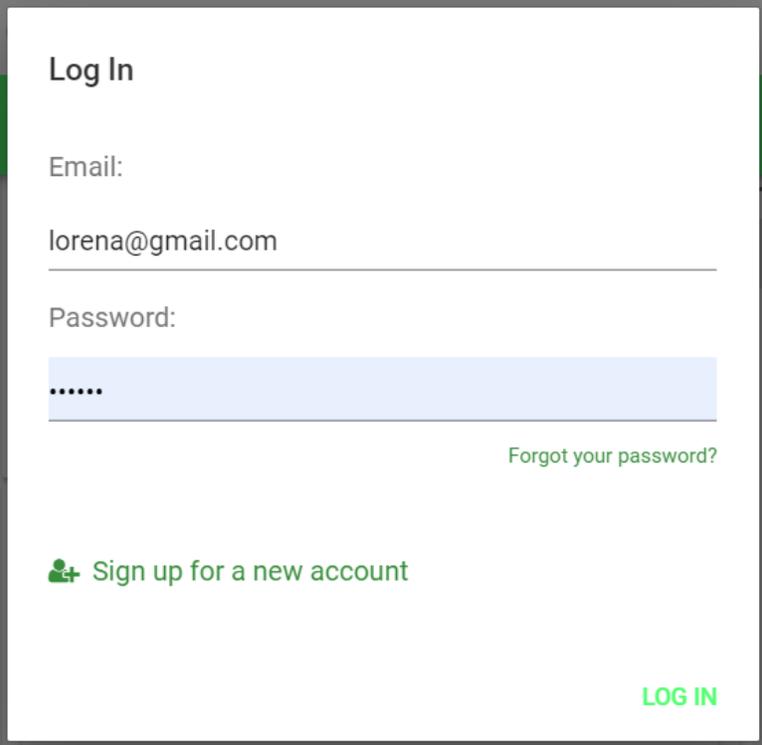
II. Opciones del Sistema

El presente Manual se organizó de acuerdo a la secuencia de ingreso a las pantallas del aplicativo web de la siguiente manera:

1. Ingreso al Aplicativo Web.
2. Registro de Usuarios.
3. Operaciones básicas

1. Ingreso al Sistema

En esta pantalla el usuario debe ingresar el correo registrado en el sistema y la clave de acceso, para luego hacer clic en el botón de ingreso **LOG IN**.

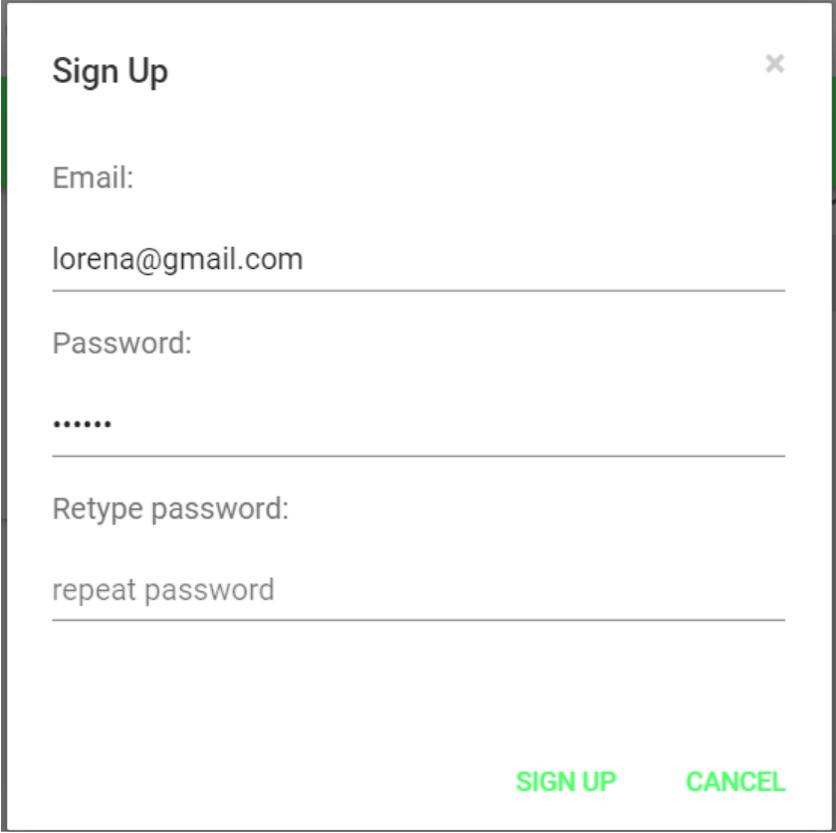


The screenshot displays a login interface with the following elements:

- Log In** title at the top.
- Email:** label above a text input field containing `lorena@gmail.com`.
- Password:** label above a password input field with masked characters (dots).
- Forgot your password?** link in green text to the right of the password field.
- Sign up for a new account** link in green text with a user icon, located below the password field.
- LOG IN** button in green text at the bottom right corner.

2. Registro de Usuario.

Si no está registrado, debe hacer clic en el botón  [Sign up for a new account](#) el cual despliega una ventana en la que se puede ingresar el correo a utilizar para el ingreso y la contraseña, bajo el área de ingreso de la contraseña, debe escribir nuevamente la contraseña. El usuario debe contactar con el administrador para activar su cuenta y pueda acceder a su sistema con las credenciales proporcionadas.



The image shows a 'Sign Up' dialog box with a close button (X) in the top right corner. It contains three input fields: 'Email' with the value 'lorena@gmail.com', 'Password' with masked characters '.....', and 'Retype password' with the value 'repeat password'. At the bottom right, there are two buttons: 'SIGN UP' and 'CANCEL'.

Sign Up ×

Email:
lorena@gmail.com

Password:
.....

Retype password:
repeat password

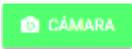
SIGN UP **CANCEL**

III. Operaciones básicas.

1. Ingreso de placa vehicular mediante un archivo en el dispositivo.

Al acceder al registro vehicular, se puede observar una casilla con el texto “suba la foto de la placa del vehículo” bajo este se encuentra el botón  , al hacer clic en el botón nos permitirá acceder a la siguiente ventana emergente:



Que permite ver los archivos recientes y el botón “Mostrar todos los archivos...” que permite acceder al explorador de archivos, el usuario puede seleccionar la imagen que desea procesar. Alternativamente se puede utilizar la cámara para tomar fotos de la placa a procesar al hacer clic en el botón  en el cual permite acceder a la cámara del dispositivo, después de tomar la foto, debe hacer clic en aceptar y la imagen será enviada para procesarse.



El resultado se mostrará en el espacio reservado para la placa del vehículo y se visualizará la imagen en el área verde junto a la opción de subir la foto de la placa.

suba la foto de la placa del vehiculo

1 FILE SELECTED CÁMARA



TIPO ▼ MARCA ▼ COLOR ▼ [PCP-5453]

SERVICIOS ▼ (NINGUNO) ▼

\$ 00.00 ANOTACIONES

GUARDAR

The form contains a file upload section with a blue button labeled "1 FILE SELECTED" and a green button labeled "CÁMARA". To the right is a green-bordered area containing a thumbnail image of a license plate that reads "PCP-5453". Below this is a form with several fields: "TIPO" and "MARCA" are dropdown menus; "COLOR" is a dropdown menu with the value "[PCP-5453]"; "SERVICIOS" is a dropdown menu with the value "(NINGUNO)"; a price field shows "\$ 00.00"; and an "ANOTACIONES" field. A blue "GUARDAR" button is at the bottom.

2. Registro de vehículos.

Para realizar el registro de un vehículo en el registro, se deberá ingresar las características del vehículo, acorde a las opciones y áreas de texto proporcionadas en el formulario presentado.

TIPO
BICICLETA
MOTO
AUTO
SUV
BUSETA
CAMIONETA
CAMION
OTRO

la placa del vehiculo

CÁMARA

AUTO ▼ FORD ▼ AZUL ▼ LBC-2578

LAV/ENCERADA ▼ GRAFITO ▼

\$ 18 Revisar el nivel del liquido de radiador

GUARDAR

Luego de ingresar los datos necesarios, se guardarán al momento de hacer clic en el botón guardar, el cual se mostrará al final del listado en el aplicativo.

AUTO FORD AZUL LBC-2578

LAV/ENCERADA GRAFITO

\$ 18 PAGADO Revisar el nivel del liquido de radiador 2022-10-17 00:45:33.469000-05:00

3. Búsqueda de vehículos por Fecha y Placa.

Bajo el formulario de registro del vehículo, se encuentra la opción de búsqueda de vehículos utilizando un rango de tiempo o indicando la fecha de inicio y buscará todos los vehículos ingresados hasta la fecha actual.



Como se muestra en la imagen, se ingresa el rango de tiempo que se desea consultar, y al hacer clic en el botón **BUSCAR POR FECHA** se desplegará el listado de los vehículos registrados en el espacio de tiempo proporcionado, si se desea buscar un vehículo en específico dentro del rango de tiempo se debe ingresar además la placa del vehículo, y hacer clic en el botón **BUSQUEDA POR PLACA** el resultado se visible debajo del formulario de búsqueda.



De necesitar realizar una nueva búsqueda, se debe hacer clic en el botón **LIMPIAR LISTA**

4. Emitir Reporte

Para realizar reportes se debe ingresar a la página de reportes la cual es accesible del botón **EMITIR REPORTE** en el formulario de búsqueda de listados por fecha y placa.

BUSQUEDA DE LISTADOS POR FECHA Y PLACA

Mon, Oct 17, 2022 12:00 AM

Mon, Oct 17, 2022 12:52 AM

HOY

BUSCAR POR FECHA

LBC-2578

BUSQUEDA POR PLACA

AUTO	FORD	AZUL	LBC-2578
LAV/ENCERADA			
\$ 18	<input type="checkbox"/> PAGADO	Revisar el nivel del liquido de radiador	2022-10-17 00:45:33.469000-05:00

LIMPIAR LISTA

EMITIR REPORTE

Al acceder al formulario de emisión de reportes, se encuentra un formulario similar al formulario de búsqueda por fecha y placa, al realizar la búsqueda ingresando espacio de tiempo para realizar la búsqueda y de ser necesario la placa de un vehículo en específico, al hacer la búsqueda del vehículo al hacer clic en el botón **BUSCAR POR FECHA** o de haberse ingresado una placa en específico al hacer clic en el botón **BUSQUEDA POR PLACA** se mostrará una lista bajo el formulario con los vehículos encontrados en el espacio de tiempo proporcionado y las coincidencias encontradas de haberse ingresado una placa especificada.



REPORTE DE REGISTRO VEHICULAR

BUSQUEDA DE LISTADOS POR FECHA Y PLACA

Mon, Oct 17, 2022 12:00 AM  HOY

FECHA FINAL 

BUSCAR POR FECHA EMITIR REPORTE

XXX-9999

BUSQUEDA POR PLACA EMITIR REPORTE

AUTO	FORD	AZUL	LBC-2578
LAV/ENCERADA			
\$ <u>18</u>	<input type="checkbox"/> PAGADO	Revisar el nivel del liquido 	2022-10-17 00:45:33.469000-05:00 

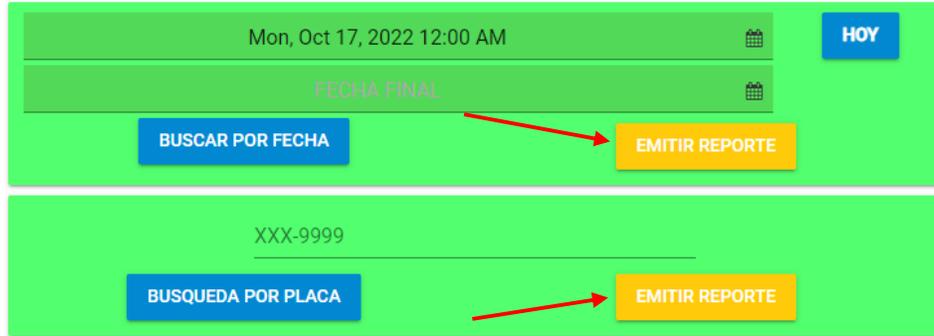
LIMPIAR LISTA

REGRESAR

De ser necesario se puede limpiar la lista de resultados obtenidos al hacer clic en el botón

LIMPIAR LISTA

bajo la lista, Para emitir el Reporte, se deberá hacer clic en el botón Emitir Reporte.



El reporte se emitirá como un documento en .pdf que se descargará automáticamente con el contenido de la búsqueda realizada, para regresar al formulario de registro se realiza clic en el botón **REGRESAR**.

5. Salir de la sesión.

Para salir del aplicativo, diríjase a la parte superior y haga clic en el botón **SALIR**.