



Detección de cambios en interfaces web para procesos RPA utilizando Inteligencia Artificial

TRAZABILIDAD DEL DOCUMENTO

Versión	Preparado Por	Fecha documento	Revisado por	Descripción
1	Andrés Martín Cantos Rivadeneira María Paola Mendoza Mendieta	18-09-25	PhD. Gladys Villegas	Versión Inicial

CONTENIDO

TRAZABILIDAD DEL DOCUMENTO	1
CONTENIDO	3
1. MAPA DEL ESTADO DEL ARTE	5
1.1. Tabla Comparativa	5
1.2. Timeline de avances en el área	8
1.3. Identificación del gap que abordará el proyecto	8
2. ANÁLISIS DE DATASETS DISPONIBLES	10
2.1. Capturas de pantalla de dataset	10
2.2. Limitaciones y Sesgos Potenciales	12
2.3. Plan de Adquisición y Preprocesamiento	13
2.4. Comparativo de Ventajas/Desventajas	15
2.5. Recomendación	16
3. DEFINICIÓN DE MÉTRICAS DE ÉXITO	17
3.1. Métricas Técnicas	17
3.2. Métricas de Impacto	18
3.3. Umbrales de Éxito	19
4. ANÁLISIS DE STAKEHOLDERS	20
4.1. Stakeholders Primarios	20
4.2. Stakeholders Secundarios	21
4.3. Stakeholders Clave	22
4.4. Matriz de Influencia / Interés	23
5. DOCUMENTO DE ALCANCE DEL PROYECTO	24
5.1. OBJETIVO GENERAL	24
5.2. OBJETIVOS ESPECÍFICOS	24
5.3. ALCANCE INCLUIDO	24
5.4. ALCANCE EXCLUIDO	25
5.5. CRITERIOS DE ACEPTACIÓN	25
6. CRONOGRAMA CON METODOLOGÍA ÁGIL	26
6.1 Product Backlog con historias de usuario	27
6.2 Sprints	29
7. PLAN DE RECURSOS	33
7.1. Recursos Humanos	33
7.2. Recursos Técnicos	34
7.3. Recursos Financieros	35
7.4. Presupuesto total estimado:	36
7.5. Plan de Adquisición y Gestión de Recursos	36
8. HITOS Y ENTREGABLES	36

Bibliografía	40
ANEXOS	43
Dataset- SRI	43
Diagrama de arquitectura	43
Carta de permiso de licencia de RPA- Electroneek	44

1. MAPA DEL ESTADO DEL ARTE

La automatización de procesos robóticos (RPA) ha transformado la forma en que las empresas gestionan tareas repetitivas y basadas en reglas, mejorando la eficiencia y reduciendo errores humanos.

El funcionamiento fundamental de estos sistemas automatizados se basa en la capacidad de replicar fielmente las acciones que realizaría un operador humano al interactuar con diversas plataformas digitales. Estos agentes virtuales desarrollan sus actividades principalmente a través de la manipulación directa de elementos presentes en las interfaces gráficas de aplicaciones web, ejecutando procesos de navegación, captura y procesamiento de información de manera autónoma.

No obstante, esta metodología de trabajo presenta una fragilidad considerable que representa uno de los principales retos técnicos del sector. La estrecha vinculación entre el funcionamiento de estos sistemas y la estabilidad de las interfaces de usuario genera puntos críticos de vulnerabilidad.

Las modificaciones en la estructura visual de las aplicaciones web, ya sean transformaciones menores como el reposicionamiento de elementos interactivos, modificaciones en los identificadores únicos o clases de componentes HTML, o modificaciones integrales del esquema de presentación, pueden provocar interrupciones imprevistas en la ejecución de los procesos automatizados, comprometiendo la continuidad.

La inteligencia Artificial (IA) ofrece una oportunidad para abordar este desafío, permitiendo a los bots de RPA no solo detectar los cambios, sino también comprender su naturaleza e, idealmente, adaptarse a ellos de manera autónoma.

El objetivo de este capítulo es obtener un conocimiento profundo de la investigación existente y justificar la relevancia de la propuesta

1.1. Tabla Comparativa

Metodología			
-------------	--	--	--

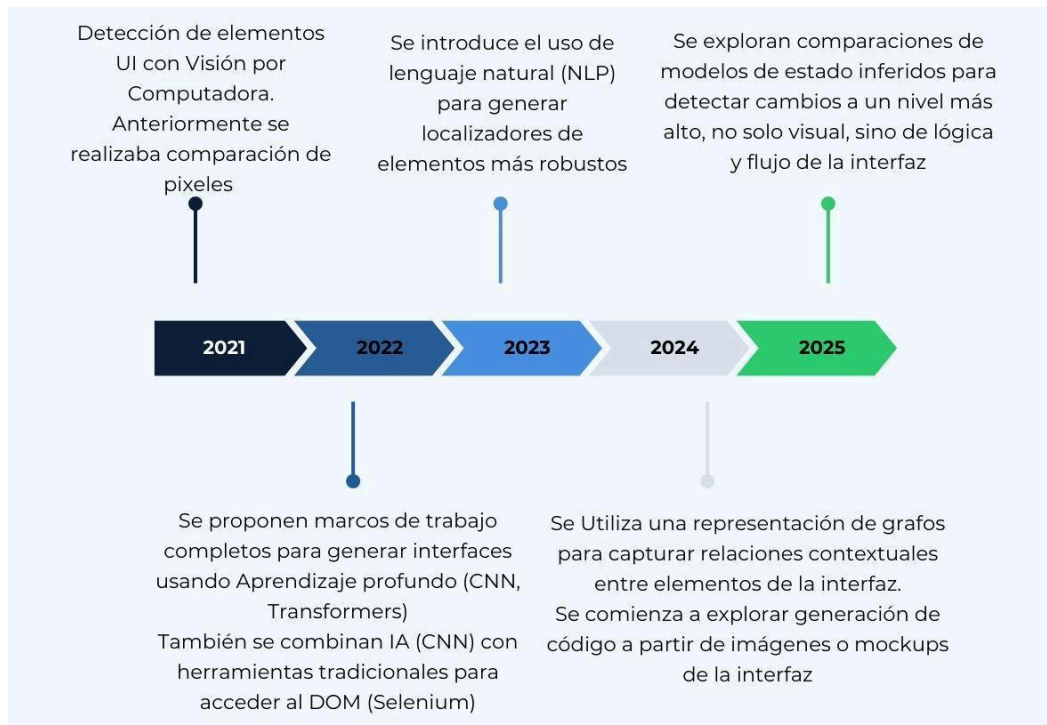
	Detección Basada en Visión por Computadora (CNN)	Detección basada en Análisis Estructural y de Grafo	Detección y Reparación con Modelos de Lenguaje Natural (NLP)
Enfoque principal	<p>Analiza capturas de pantalla a nivel visual como si fuera una persona</p> <p>También puede ser a nivel de píxeles</p> <p>El objetivo es identificar las diferencias visuales y que tan relevantes son</p>	<p>Representa la interfaz de usuario como un grafo para analizar relaciones entre los elementos</p>	<p>Utiliza lenguaje natural y modelos de lenguaje (LLMs) para crear localizadores de elementos más robustos</p>
Técnicas o enfoque con IA	<p>Modelos de detección de objetos (Faster R-CNN, YOLO)</p> <p>Segmentación de imágenes (U-Net)</p> <p>Redes Siamese</p>	<p>Grafos de conocimiento</p> <p>Algoritmos de Grafos</p>	<p>Procesamiento de Lenguaje Natural (NLP)</p> <p>Modelos de lenguaje Grandes (LLMs)</p>
Ventajas	<p>Detecta cambios visuales que no alteran el código</p> <p>Permite identificar ubicación precisa de cambios</p> <p>Puede llegar a detectar cambios</p>	<p>Proporciona un contexto semántico y relacional de los elementos de la interfaz gráfica</p> <p>Permite distinguir cambios visuales de cambios funcionales</p>	<p>Los localizadores inteligentes son menos propensos a romperse con cambios menores en el DOM.</p> <p>Permite una descripción más humana de los elementos a automatizar</p>

	visuales que no afecten la funcionalidad del RAP (ej. Publicidad dinámica)		
Desafíos o Limitaciones	<p>No distingue cambios funcionales</p> <p>Requiere grandes dataset de imágenes etiquetadas.</p> <p>Puede ser que sea necesario entrenarlo continuamente dependiendo del proceso RPA</p>	<p>Puede ser computacionalmente pesado o costoso, La creación del grafo puede ser compleja en interfaces muy dinámicas o con mucho contenido visual</p>	<p>Mucho mayor costo computacional, los Grandes modelos de lenguaje requieren mayor procesamiento (GPU)</p> <p>Probar Varios LLMs, no todos los modelos de lenguaje son óptimos para la detección de cambios visuales</p>
Referencias	[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]	[12] [13] [14]	[15] [16]

Los estudios demuestran que las técnicas de IA resultan muy útiles para resolver este tipo de problemas y evitar que los RPA fallen cuando la interfaz web sufre cambios. El verdadero desafío consiste en integrar estas metodologías de manera práctica y eficiente, de modo que un Bot de RPA pueda adaptarse de forma autónoma a dichas modificaciones.

Incluso para lograr una solución integral se podría combinar las técnicas mencionadas anteriormente (Visión por Computadora, Grafos y NLP), Esta integración permitiría una mayor precisión al aprovechar las fortalezas de cada enfoque, aunque también implicaría una mayor complejidad en su implementación y mantenimiento.

1.2. Timeline de avances en el área



Basado en la revisión bibliográfica, se observa un claro progreso desde las técnicas básicas de comparación de píxeles hacia enfoques más inteligentes y robustos. Las tendencias actuales se centran en el uso de visión por computadora y la comprensión semántica de las interfaces usando grafos [12] [13] [14] o grandes modelos de lenguaje [9].

El objetivo final ya no es solo detectar cambios, sino que los bots sean capaces de entender, adaptarse y recuperarse de manera autónoma ante cambios de la interfaz.

1.3. Identificación del gap que abordará el proyecto

A pesar de los avances en la automatización de procesos robóticos (RPA) y la aplicación de la inteligencia artificial, persiste una brecha crítica. Las soluciones actuales de RPA dependen en gran medida de localizadores de elementos de

interfaz de usuario (cómo selectores css o ids), lo que las hace muy vulnerables a los cambios en el diseño o la estructura de las páginas web.

Si bien existen investigaciones que proponen la detección de cambios utilizando visión por computadora, no abordan directamente el problema de detección en tiempo real de los cambios para los bots de RPA. El gap identificado es la falta de un módulo basado únicamente en la visión por computadora (CNN), que permita a un Bot de RPA detectar cambios visuales y operar sin depender de los localizadores tradicionales, mejorando así su autonomía y fiabilidad.

El proyecto se centrará en los siguientes puntos:

- **Desarrollo del Modelo de Detección:** Se diseñará y entrenará un modelo utilizando exclusivamente la visión por computadora con Redes Neuronales Convolucionales (CNN) para comparar capturas de pantalla de una interfaz web en dos momentos distintos y generar idealmente un mapa de calor que resalte las regiones de cambio significativo y un valor de porcentaje de similitud que nos ayude a distinguir que tan crítico es el cambio.
- **Integración con RPA:** El sistema de visión se integrará con un Bot de RPA, utilizando la plataforma Electroneek. La salida del modelo (los cambios detectados) guiará al Bot para validar si su flujo de trabajo original aún es viable, sin necesidad de reprogramación manual.
- **Validación y Pruebas:** El sistema propuesto será validado en escenarios de la vida real, como cambios en el diseño o la reubicación de botones y campos en páginas web públicas. Se comparará el desempeño del Bot con la integración de IA con el de un Bot tradicional, evaluando métricas como la tasa de éxito de la automatización y el tiempo de respuesta ante cambios.

Este proyecto busca demostrar que un Bot de RPA junto la IA puede “ver” los cambios en la interfaz, lo que permite adaptarse y mantener su funcionalidad en lugar de fallar y requerir intervención humana.

2. ANÁLISIS DE DATASETS DISPONIBLES

Dado que la solución propuesta combina tecnología RPA e Inteligencia Artificial, se ha definido como primer parámetro de entrada la recolección de datos provenientes de las páginas web que se planea automatizar. Para este caso, se han seleccionado cuatro portales públicos del Ecuador a los cuales se tiene acceso. Entonces el dataset a construir estará conformado únicamente por capturas de pantalla (imágenes) de las páginas públicas seleccionadas (SRI, MSP, SENESCYT, fiscalía general del Estado)

2.1. Capturas de pantalla de dataset

2.1.1. SRI

Descripción técnica:

Tamaño: depende de la frecuencia de captura. Por ejemplo, una imagen del login puede ocupar aproximadamente 242 MB; este valor debe multiplicarse por el número de pantallas que se requiera automatizar.

Formato: PNG/JPEG.

Estructura: carpeta /SRI/con las imágenes etiquetadas por categoría.

Procedencia: portal público del SRI (acceso libre).

Calidad: alta resolución visual, aunque con elementos dinámicos (banners, anuncios) que generan ruido.

Idoneidad: muy alta para detectar cambios visuales en procesos tributarios.

Accesibilidad: páginas públicas, sin costo; restricciones mínimas.

2.1.2. MSP

Descripción técnica:

Tamaño similar al del SRI, depende de la frecuencia de captura.

Formato: PNG/JPEG.

Estructura: carpeta /MSP/con las imágenes etiquetadas por categoría.

Procedencia: portal público del MSP.

Calidad: buena, aunque los portales suelen actualizar banners por campañas sanitarias → más cambios frecuentes.

Idoneidad: muy alta para probar robustez del modelo frente a cambios frecuentes.

Accesibilidad: libre, aunque partes del sitio pueden tener contenido embebido externo.

2.1.3. SENESCYT

Descripción técnica:

Tamaño: depende de la frecuencia de scraping.

Formato: PNG/JPEG.

Estructura: carpeta /SENESCYT/con las imágenes etiquetadas por categoría.

Procedencia: sitio público de SENESCYT.

Calidad: cambios frecuentes en convocatorias y formularios en línea.

Idoneidad: alta, pues refleja alteraciones en menús, banners de becas, formularios.

Accesibilidad: público, sin costo; debe verificarse que la captura automática no bloquee por medidas anti-bot.

2.1.4. Fiscalía General del Estado

Descripción técnica:

Tamaño: mismo patrón de captura.

Formato: PNG/JPEG.

Estructura: carpeta /Fiscalia/con las imágenes etiquetadas por categoría.

Procedencia: portal público de la Fiscalía.

Calidad: generalmente más estático; cambios menores pero relevantes (comunicados, banners, accesos a trámites).

Idoneidad: media/alta (menos dinámico, pero útil para contrastar).

Accesibilidad: público, libre acceso.

2.2. Limitaciones y Sesgos Potenciales

Temporalidad de los datos: las capturas iniciales manuales pueden no coincidir con los estados posteriores registrados automáticamente por el bot, generando diferencias en la línea de tiempo.

Sesgo de actualización: si el bot solo notifica y actualiza cuando detecta cambios, la información intermedia podría perderse, lo que afecta la trazabilidad.

Inconsistencia en formatos: las capturas manuales podrían diferir en resolución, tamaño o calidad respecto a las capturas automatizadas, afectando la homogeneidad del dataset.

Dependencia tecnológica: eventuales fallas del bot al detectar cambios pueden dejar períodos sin registro actualizado.

2.3. Plan de Adquisición y Preprocesamiento

Etapa	Actividad	Detalle	Entregables	Responsable
Adquisición	Identificación de fuentes	Selección de 4 portales públicos (SRI, SENESCYT, MSP y otro portal definido).	Lista de fuentes validadas	Analista de datos
	Captura inicial manual	Recolección de pantallas clave para establecer línea base.	Set de capturas iniciales	Equipo RPA
	Automatización de captura	Configuración del bot para registrar datos y detectar cambios periódicamente.	Bot configurado y operativo	Equipo RPA
	Definición de frecuencia	Establecer periodicidad (diaria/semanal/mensual) según dinámica de cada portal.	Cronograma de actualizaciones	PM / Analista
	Almacenamiento	Guardar capturas en repositorio estructurado, con estándares definidos (ej. PNG/JPG).	Repositorio inicial	Desarrolladores RPA / IA
Preprocesamiento	Normalización	Unificación de resoluciones, tamaños y extensiones de imágenes.	Dataset homogéneo	Desarrolladores RPA / IA

	Optimización	Compresión sin pérdida y recorte de secciones irrelevantes.	Imágenes optimizadas	Desarrolladores RPA / IA
	Extracción de información	Uso de OCR y Computer Vision para transformar imágenes en texto estructurado.	Dataset en formato CSV/JSON	Desarrolladores RPA / IA
	Limpieza y validación	Eliminación de duplicados, verificación de campos incompletos o inconsistentes.	Dataset depurado	Analista de datos
	Anonimización	Enmascaramiento/pseudonimización de datos sensibles si aplica.	Dataset seguro	Desarrolladores RPA / IA
	Estructuración	Organización en tablas, CSV o JSON para consumo en modelos de IA.	Dataset estructurado final	Desarrolladores RPA / IA
Trazabilidad	Registro de versiones	Control de fecha y hora de cada captura y actualización.	Bitácora de dataset	PM / Desarrolladores RPA / IA
	Monitoreo de cambios	Alertas automáticas del bot ante inconsistencias o fallos.	Reportes de monitoreo	Desarrolladores RPA / IA

2.4. Comparativo de Ventajas/Desventajas

Aspecto	Ventajas	Desventajas
Cobertura y procedencia	Todos los datos provienen de portales oficiales públicos (SRI, MSP, SENESCYT, Fiscalía), confiables y accesibles.	Sesgo hacia la información formal; no refleja economía informal ni procesos internos no visibles en la web.
Calidad visual	Capturas de alta resolución, útiles para detección de cambios visuales mediante IA y RPA.	Elementos dinámicos (banners, anuncios) generan ruido; inconsistencia si se mezcla captura manual y automatizada.
Relevancia analítica	SRI es altamente relevante para procesos contables y tributarios; MSP y SENESCYT permiten probar robustez ante cambios frecuentes.	Algunas páginas como la Fiscalía son menos dinámicas, limitando la variedad de casos para el modelo.
Accesibilidad	Páginas públicas, sin costo y con restricciones mínimas; bot puede automatizar capturas.	Posibles bloqueos por medidas anti-bot o contenido embebido externo (SENESCYT, MSP).
Frecuencia y temporalidad	Capturas periódicas mediante bot permiten mantener el dataset	Las capturas iniciales manuales pueden diferir temporalmente de las posteriores automatizadas;

	actualizado y rastrear cambios.	riesgo de pérdida de información intermedia.
Tamaño y almacenamiento	Formato estándar (PNG/JPEG), con organización estructurada por carpeta y categoría.	Cada imagen puede ser pesada (ej. 242 MB por login), generando alto consumo de almacenamiento si hay muchas pantallas o frecuencia alta.
Escalabilidad y preprocesamiento	Posibilidad de añadir nuevas fuentes; OCR y Computer Vision permiten estructurar datos para análisis.	Necesidad de preprocesamiento intensivo (normalización, compresión, limpieza) para homogeneizar dataset y evitar errores de IA.
Trazabilidad y monitoreo	Registro de versiones y alertas automáticas permiten mantener control sobre cambios y calidad de datos.	Dependencia tecnológica: fallas del bot pueden dejar períodos sin captura; inconsistencia de formatos manual/automático.

2.5. Recomendación

Recomendación de Dataset Principal: SRI

Se selecciona el Servicio de Rentas Internas (SRI) como dataset principal por su cobertura confiable, estructura estandarizada y relevancia analítica y contable, dado que sus datos son utilizados en todos los procesos contables de las empresas. Además, el portal se actualiza periódicamente y su información puede complementarse con otros datasets (SENESCYT, MSP) para enriquecer el análisis.

Recomendación operativa Plan B: MSP

El MSP se elige como respaldo en caso de que la pagina del SRI no esté disponible

3. DEFINICIÓN DE MÉTRICAS DE ÉXITO

3.1. Métricas Técnicas

Para evaluar el desempeño del modelo de visión por computadora en la detección de cambios, se utilizarán las siguientes métricas estándar que miden su precisión y eficiencia:

3.1.1. Métricas primarias

- **Accuracy de detección de cambios:** proporción de interfaces evaluadas en las que el sistema identifica correctamente los cambios relevantes. Es la métrica más general, pero puede ser engañosa en casos con desequilibrio de datos (ej., una página con muy pocos cambios).
- **F1-score por tipo de elemento (botones, formularios, menús, inputs):** balance entre precisión y exhaustividad en la detección de cambios en cada clase. Es la métrica más importante para nuestro proyecto, ya que nos da una visión equilibrada del rendimiento.
- **RMSE (Root Mean Square Error):** En un contexto de segmentación el RMSE podría medir la diferencia entre el mapa de calor predicho y la ubicación real del cambio. Un valor bajo indicaría que el modelo localiza los cambios con gran precisión.

3.1.2. Métricas secundarias

- **Precisión:** porcentaje de cambios detectados que son realmente relevantes, ¿cuántas fueron realmente cambios? Un valor alto de precisión es crítico para evitar que el Bot detenga un proceso innecesariamente.
- **Recall (Exhaustividad):** De todos los cambios que realmente ocurrieron en la interfaz, ¿cuántos fue capaz de detectar el modelo? Un valor alto de exhaustividad es vital para la resiliencia del Bot, asegurando que no pase por alto un cambio crítico que podría romper el proceso.

- **AUC-ROC (Área bajo la curva):** Mide la capacidad de tu modelo para distinguir entre las clases (para nuestro proyecto son "cambio" y "no cambio") a través de varios umbrales. Es una métrica robusta que evalúa el rendimiento general del clasificador y la discriminación del modelo entre un cambio real o ruido visual (ej. Publicidad)

3.1.3. Métricas de eficiencia

- **Tiempo promedio de detección por interfaz (segundos):** El tiempo que le toma al modelo procesar una nueva captura de pantalla y emitir una predicción. Importante para integración en pipelines con el RPA, este tiempo debe ser mínimo para evitar retrasos en el proceso.
- **Uso de Recursos de Hardware:** Cantidad de memoria RAM y potencia de CPU/GPU que el modelo consume para funcionar. Se buscará un modelo eficiente que pueda ejecutarse en un hardware modesto, para que sea más accesible para diferentes entornos de implementación de RPA.

3.2. Métricas de Impacto

Para demostrar cómo la solución genera un valor real se definen las siguientes métricas:

3.2.1. KPIs de negocio cuantificables

- **Reducción del Tiempo de Inactividad del Bot:** Se medirá la disminución en la cantidad de tiempo que un Bot de RPA está inactivo debido a fallas de la interfaz. Esto se puede calcular como: *Tiempo de inactividad reducido = Tiempo de inactividad actual – Tiempo de inactividad con IA*
- **Tasa de Éxito de la Automatización:** Porcentaje de procesos de RPA completados con éxito sin intervención manual, en un entorno con cambios de interfaz. El objetivo es que la solución de IA incremente este porcentaje.

3.2.2. Métricas de adopción/uso

- **Frecuencia de Uso:** Cuántas veces por día/semana/mes se invoca el módulo de detección de cambios de IA. Un uso frecuente indica que la solución es vista como valiosa y necesaria.

- **Disminución de la Intervención Humana:** Mide la reducción en la cantidad de horas-hombre dedicadas a la reparación manual de los bots de RPA. Un Bot resiliente requiere menos supervisión y mantenimiento.

3.2.3. Indicadores de satisfacción de usuarios

- **Tasa de Errores Críticos:** Número de errores graves o fallas catastróficas del Bot que el sistema de IA no logró prevenir o solucionar. *Tasa de errores críticos = (Número de errores críticos detectados / Total de ejecuciones evaluadas) x 100*
- **Calificación de Satisfacción:** Encuestas (escala 1 -5) dirigidas a los desarrolladores y operadores de RPA sobre la usabilidad, fiabilidad y valor de la solución.

3.3. Umbrales de Éxito

Para convertir las métricas en metas claras, se definen los siguientes umbrales de éxito:

3.3.1. Valores mínimos aceptables para aprobación

- F1-score ≥ 0.75 en detección de cambios relevantes.
- Tiempo promedio de detección ≤ 5 s por interfaz
- Reducción del 25% en el tiempo de inactividad del Bot debido a fallas
- Tasa de errores críticos $\leq 5\%$
- Satisfacción de usuarios $\geq 3.5/5$ en encuestas iniciales.

3.3.2. Objetivos realistas para excelencia

- F1-score ≥ 0.90
- Tiempo de detección ≤ 2 s por interfaz en hardware estándar.
- Reducción del 75% en el tiempo de inactividad del Bot debido a fallas
- Tasa de errores críticos $\leq 1\%$
- Satisfacción de usuarios $\geq 4.5/5$ en encuestas iniciales.

3.3.3. Comparación con benchmarks industriales

La tasa de errores críticos será la métrica principal para evaluar el éxito de la solución y para realizar un benchmark frente a las herramientas existentes.

Esta métrica es crucial porque las metodologías de detección de cambios de la industria a menudo se basan en una simple comparación de píxeles o del DOM. Como resultado, generan un alto número de falsos positivos, identificando como cambios eventos que son visualmente inofensivos para un Bot de RPA, como un cambio de color o publicidad.

Los trabajos académicos validan esta problemática. Por ejemplo, el reporte técnico de Himanshu Pathak “Computer Vision for UI Testing” [14], y los artículos de la industria como el de InfoQ [17], discuten las limitaciones de los enfoques tradicionales (pixel-diff). Nuestra solución que utilizará una red neuronal convolucional, aborda esta brecha al ir más allá de la comparación de píxeles. En lugar de simplemente detectar diferencias, nuestro modelo aprenderá a interpretar el contexto visual del cambio. Este enfoque está respaldado por investigaciones como la de “Artificial intelligence for context-aware visual change detection in software test automation” [9], que propone el uso de grafos para capturar relaciones contextuales entre elementos. Aunque nuestro proyecto se centrará en CNN, el principio es similar, dotar al sistema con la inteligencia para distinguir un cambio de diseño y categorizarlo como un cambio irrelevante, funcional o crítico.

Por lo tanto, el valor añadido de nuestro proyecto se medirá directamente en la capacidad para reducir drásticamente la tasa errores críticos. Al hacerlo no solo mejoramos la precisión técnica, sino que también generamos un impacto al negocio disminuyendo la necesidad de intervención humana.

4. ANÁLISIS DE STAKEHOLDERS

4.1. Stakeholders Primarios

Stakeholder	Rol e Influencia	Necesidades y Expectativas	Potenciales Barreras / Resistencias	Estrategia
-------------	------------------	----------------------------------	---	------------

Cientes (empresas que usarán la automatización)	Usarán la solución en procesos contables/tributarios	Datos confiables, reducción de errores, facilidad de uso	Desconfianza en la precisión, resistencia al cambio	Demostraciones prácticas, documentación clara, capacitaciones básicas
Empresas desarrolladoras de bots (proveedores de RPA/IA)	Brindan soporte, tecnología y mejores prácticas	Compatibilidad, estabilidad, posibilidad de escalabilidad	Incompatibilidad de versiones, costos de licencias, tiempos de soporte	Pruebas de integración, comunicación técnica constante, acuerdos de servicio claros

4.2. Stakeholders Secundarios

Stakeholder	Rol e Influencia	Necesidad/ Expectativa	Potenciales Barreras / Resistencias	Estrategia
María Paola Mendoza Mendieta	Implementadora técnica / Desarrolladora y PM	Titulación, resultados prácticos, aprendizaje técnico y metodológico	Sobrecarga de tareas, limitación de tiempo y recursos	Planificación clara, comunicación diaria, roles definidos, coordinación con Andrés
Andrés Martín	Implementador técnico /	Desarrollo de RPA,	Sobrecarga de tareas,	Coordinación con PM,

Cantos Rivadeneira	Desarrollador y Analista de datos	preprocesamiento de datos, validación de datasets	limitación de tiempo y recursos	revisiones periódicas de calidad de datos
PhD Gladys María Villegas	Valida y aprueba el proyecto académico	Resultados claros, metodológicamente sólidos	Diferencias entre enfoque académico y limitaciones técnicas	Reuniones periódicas, avances parciales, retroalimentación continua

4.3. Stakeholders Clave

Stakeholder	Rol e Influencia	Necesidad/ Expectativa	Potenciales Barreras / Resistencias	Estrategia
PhD Gladys María Villegas	Sponsor académico	Cumplimiento de objetivos académicos, calidad metodológica, innovación	Expectativas altas de resultados, ajustes de cronograma	Reuniones de seguimiento, entregas parciales, reportes de avance
Paola Mendoza	Asesora técnica y valida resultados (Usuario experto /	Relevancia práctica y validez técnica de resultados	Diferencias entre enfoque práctico y académico	Sesiones de asesoría, revisión de metodología, feedback técnico

	Product owner)			
--	----------------	--	--	--

4.4. Matriz de Influencia / Interés

Nombre	Stakeholder	Influencia	Interés	Prioridad Comunicación
PhD Gladys María Villegas	Sponsor	Alta	Alta	Alta
Clientes (empresas que usan bots)	Usarán la solución en procesos contables/tributarios	Media	Alta	Alta
Empresas desarrolladoras de bots	Beneficiarios del sistema	Media	Media	Media
María Paola Mendoza Mendieta	Implementadora técnica / Desarrolladora y PM	Alta	Alta	Alta
Andrés Martin Cantos Rivadeneira	Implementador técnico / Desarrollador y Analista de datos	Alta	Alta	Alta
Paola Mendoza	Usuario experto/ Product owner	Alta	Media	Alta

5. DOCUMENTO DE ALCANCE DEL PROYECTO

5.1. OBJETIVO GENERAL

Desarrollar un sistema inteligente de detección de cambios en interfaces web para procesos RPA, utilizando técnicas de Visión por Computadora y Redes Neuronales Convolucionales, con el fin de identificar y alertar variaciones visuales mínimas que puedan afectar la ejecución de bots, garantizando la continuidad operativa y reduciendo los tiempos de mantenimiento.

5.2. OBJETIVOS ESPECÍFICOS

- Diseñar e implementar un modelo de detección de cambios visuales mediante CNNs, capaz de identificar variaciones en banners, botones, colores, íconos y disposición de elementos.
- Integrar un componente OCR para reconocer y comparar textos en la interfaz.
- Desarrollar un sistema de alertas preventivas en tiempo real que notifique automáticamente los cambios visuales de la página.
- Validar el sistema en 1 o 2 aplicaciones web piloto mediante la ejecución de bots en ElectroNeek, con el fin de comprobar su efectividad en la detección de cambios y su impacto en la reducción de fallos operativos.

5.3. ALCANCE INCLUIDO

5.3.1. Funcionalidades técnicas específicas:

- Implementación de modelos de visión por computadora con CNNs para la detección de cambios visuales en interfaces web.
- Uso de OCR para el reconocimiento de textos en botones, etiquetas y secciones de la interfaz.
- Procesamiento de capturas de pantalla generadas por bots de ElectroNeek.

- Comparación automática de versiones “antes y después” de la interfaz.
- Generación de notificaciones inmediatas con las evidencias de los cambios.

5.3.2. Tipos de datos a procesar:

- Capturas de pantalla de interfaces web (datos visuales).
- Textos extraídos de las imágenes mediante OCR (datos textuales).

5.3.3. Usuarios/casos de uso cubiertos:

- Desarrolladores de RPA que mantienen y ajustan bots en ElectroNeek.
- **Validación en 1 a 2 aplicaciones web piloto de procesos RPA.**

5.4. ALCANCE EXCLUIDO

- Análisis estructural del DOM y validación semántica avanzada (considerado como mejora futura).
- Detección de cambios en aplicaciones de escritorio o móviles.
- Integración con otras plataformas RPA distintas a ElectroNeek.
- Escalabilidad a múltiples aplicaciones en paralelo (se limita a pilotos).

5.5. CRITERIOS DE ACEPTACIÓN

Nº	Criterio de Aceptación	Métrica / Estándar	Procedimiento de Validación
1	Detección de cambios visuales mediante CNNs	≥ 75% de precisión en identificación de banners, botones, colores y disposición	Comparación automática de capturas de pantalla antes/después en los casos piloto
2	Detección de cambios textuales mediante OCR	≥ 75% de precisión en reconocimiento de etiquetas y textos	Validación con cambios intencionales de textos en botones y secciones de la interfaz

3	Generación de alertas automáticas	Tiempo de respuesta < 1 minuto tras ejecución del bot en ElectroNeek	Medición del tiempo entre la ejecución del bot y la recepción de la alerta
4	Reducción de fallos operativos en bots	≥ 75% menos incidentes en ejecución de bots en los casos piloto	Registro de incidentes antes y después de aplicar el sistema en los bots piloto
5	Validación por desarrolladores de RPA	Al menos 2 desarrolladores confirman utilidad y aplicabilidad	Sesión de prueba con desarrolladores, recopilación de feedback formal
6	Entregables finales completos	Prototipo funcional + reporte de métricas	Revisión del prototipo y reportes

6. CRONOGRAMA CON METODOLOGÍA ÁGIL

Nombre del Proyecto	Detección de Cambios en Interfaces web para RPA usando IA
Sprints	4
Metodología ágil	<p>Scrum</p> <p>Product Owner: Paola Mendoza</p> <p>Scrum Master: Andrés Cantos</p> <p>Development Team: Andrés Cantos y Paola Mendoza</p>
Ceremonias	<p>Sprint Planning: Al inicio de cada sprint para definir el objetivo y seleccionar las user stories.</p> <p>Daily Scrum: Reunión diaria de 15 minutos para sincronización.</p>

	<p>Sprint Review: Al final de cada sprint para inspeccionar el incremento y obtener feedback.</p> <p>Sprint Retrospective: Al final de cada sprint para reflexionar sobre lo que funcionó y lo que se puede mejorar.</p>
--	--

6.1 Product Backlog con historias de usuario

ID	Prioridad	Historia	Sprint asignado
US_01	Alta	Como investigador, quiero analizar al menos 20 papers recientes (estado del arte) sobre detección de cambios en interfaces para fundamentar la arquitectura de la solución a proponer	1
US-02	Media	Como equipo, queremos definir las métricas de éxito para evaluar los resultados	1
US_03	Media	Como desarrollador, quiero configurar el repositorio GitHub básico y el entorno de desarrollo.	1
US_04	Alta	Como desarrollador, quiero investigar las arquitecturas de CNN existentes para la detección de cambios visuales, para poder seleccionar la más adecuada	1
US_05	Alta	Como desarrollador, quiero preparar el dataset inicial con cambios controlados y etiquetados para entrenamiento	1

US_06	Alta	Como desarrollador, quiero implementar el modelo CNN seleccionado, para que pueda procesar pares de capturas de pantalla (entrenamiento y pruebas)	2
US_07	Alta	Como desarrollador quiero implementar un script (Pipeline) de integración con Electroneek , para que el Bot pueda interactuar con el modelo IA	2
US_08	Alta	Como desarrollador, quiero realizar pruebas de integración entre el modelo y el script de Electroneek , para validar que la comunicación sea correcta	2
US_09	Alta	Como desarrollador, quiero realizar el ajuste de hiperparámetros del modelo CNN, para optimizar su precisión	3
US_10	Alta	Como desarrollador, quiero validar el modelo contra diferentes casos y dataset de pruebas para validar las métricas definidas	3
US_11	Alta	Como desarrollador, quiero realizar pruebas de extremo a extremo para asegurar el sistema completo (Bot RPA + IA) funciona sin errores en un entorno controlado	4
US_12	Media	Como equipo, queremos documentar todo lo realizado tanto la arquitectura como el código y de ser necesario manual de usuario, todo subido en el repositorio GitHub	4

US_13	Alta	Como equipo, queremos documentar todo lo realizado y los resultados obtenidos para realizar una presentación concisa y comunicar el valor y resultados del proyecto	4
-------	------	--	---

6.2 Sprints

Sprint 1: Investigación y Diseño			
Duración	Semana 1 - 2		
Objetivo	Realizar una investigación exhaustiva y definir la arquitectura y alcance de la solución, dejando el entorno de desarrollo listo para la implementación		
Alcance	Historia de usuario resumida	Responsable	Story points
	US_01 Analizar 20 papers (estado del arte)	Andrés	5
	US_02 Definir métricas de éxito	Andrés	2
	US_03 Configurar repositorio github y entorno de desarrollo	Andrés Paola	2
	US_04 Investigar las arquitecturas de CNN existentes	Andrés Paola	5
	US_05 Preparar el dataset inicial	Paola	8
Definition of done	<p>Encontrar 3 metodologías y seleccionar la metodología a utilizar en el proyecto</p> <p>Entorno de desarrollo funcional y versionado en un repositorio GitHub</p>		

	Tener seleccionada las páginas a utilizar y el dataset inicial para entrenamiento
Riesgos y mitigación	<p>Riesgo: La investigación puede extenderse y no lograr definir una arquitectura</p> <p>Mitigación: Se asigna un tiempo fijo para definir la arquitectura y se realiza una sesión para tomar una decisión final al inicio de la segunda semana para continuar</p>

Sprint 2: Desarrollo Core			
Duración	Semana 3 – Semana 4		
Objetivo	Implementar el algoritmo principal de detección de cambios y los componentes necesarios para la integración con el RPA		
Alcance	Historia de usuario resumida	Responsable	Story points
	US_06 Implementar el Modelo CNN y entrenamiento	Andrés Paola	13
	US_07 Implementar un script (pipeline) de integración con Electroneek	Andrés Paola	8
	US_08 Realizar pruebas de integración entre el modelo y el script	Andrés Paola	5
Definition of done	Modelo de CNN entrenado con el dataset de las páginas web a validar y con un F1-score > 0.75		

	Un script funcional (pipeline) de integración con Electroneek que tome capturas de pantalla y las envíe al modelo para su procesamiento
Riesgos y mitigación	<p>Riesgo: El modelo CNN no logra un rendimiento aceptable en las primeras iteraciones</p> <p>Mitigación: Se preparan scripts para el pre-procesamiento de datos y se establecen puntos de control para el ajuste de hiperparámetros</p>

Sprint 3: Optimización y Validación			
Duración	Semana 5		
Objetivo	Optimizar el rendimiento del modelo, validar los resultados con las métricas definidas y exportar el modelo para comenzar la integración completa con el Electroneek		
Alcance	Historia de usuario resumida	Responsable	Story points
	US_09 Realizar el ajuste de hiperparámetros	Andrés Paola	8
	US_10 Validar el modelo contra diferentes casos y dataset de pruebas	Andrés Paola	5
Definition of done	El modelo debe alcanzar los umbrales de éxito técnicos definidos en el punto 3.3 del documento de presentación		

	Exportar el Modelo funcional que muestre un rendimiento consistente con el dataset de pruebas, para que se integre con el Electroneek
Riesgos y mitigación	<p>Riesgo: Los resultados de las pruebas de validación no son satisfactorios.</p> <p>Mitigación: Se planifican iteraciones o entrenamientos adicionales para ajustar el modelo</p>

Sprint 4: Finalización y Documentación			
Duración	Semana 6		
Objetivo	Realizar pruebas integrales, documentar el sistema y preparar la presentación final		
Alcance	Historia de usuario resumida	Responsable	Story points
	US_11 Realizar pruebas integrales de extremo a extremo	Andrés Paola	5
	US_12 Documentación técnica y versionado en Github	Andrés Paola	3
	US_13 Presentación final	Andrés Paola	2
Definition of done	El sistema automatizado completo con éxito de 2 casos de uso (sitios web a validar) Código y Documentación técnica completa subida el repositorio GitHub Presentación Completa		

Riesgos y mitigación	Riesgo: Surgen bugs críticos durante las pruebas finales Mitigación: 1 integrante del equipo prioriza la solución de bugs mientras el segundo integrante realiza la presentación
-----------------------------	---

7. PLAN DE RECURSOS

7.1. Recursos Humanos

Rol / Puesto	Responsabilidades	Horas estimadas por sprint	Disponibilidad
María Paola Mendoza Mendieta	Desarrollo RPA, preprocesamiento de datos, implementación IA		Integrante del equipo, responsable core técnico
Andrés Martin Cantos Rivadeneira	Desarrollo RPA, integración de datos, pruebas		Integrante del equipo, responsable core técnico
Andrés Martin Cantos Rivadeneira	Limpieza, normalización, extracción de información		Asumido por los desarrolladores si es necesario
María Paola Mendoza Mendieta	Planificación de sprints, seguimiento de hitos		Uno de los integrantes asume rol de PM

PhD Gladys María Villegas	Asesoría en IA y visión por computadora	Según necesidad	Se solicita de forma puntual según complejidad
------------------------------	--	--------------------	--

7.2. Recursos Técnicos

Recurso	Detalle	Disponibilidad
Computadora de desarrollo – María Paola Mendoza	Lenovo IdeaPad L340-15IRH Gaming (LENOVO_MT_81LK_BU)	Equipo personal, disponible para todo el desarrollo y pruebas RPA/IA
	• CPU: Intel Core i7-9750H (6 núcleos, 12 hilos, 2.6 GHz base / 4.5 GHz turbo)	
	• RAM: 16 GB DDR4	
	• Almacenamiento: SSD 512 GB	
	• GPU: NVIDIA GeForce GTX 1650 4GB	
	• Sistema operativo: Windows 10	
Computadora de desarrollo – Andrés Cantos	HP Pavilion	Equipo personal, disponible para desarrollo y pruebas de preprocesamiento y bot
	• CPU: AMD Ryzen 7 5700U (8 núcleos, 16 hilos, 1.8 GHz base / 4.3 GHz turbo)	
	• RAM: 16 GB DDR4	
	• Almacenamiento: SSD 1 TB	
	• GPU integrada AMD Radeon	
	• Sistema operativo: Windows 11	

GPU / Procesamiento IA	Uso de las GPUs disponibles en las estaciones de trabajo (NVIDIA GTX 1650 en Lenovo y GPU integrada en HP) para entrenar modelos ligeros de IA	Uso compartido, escalable con servicios cloud si se requiere mayor potencia
Software / Licencias	Licencia ElectroNeek para RPA (costo anual; actualmente prestada por empresa con carta de autorización como anexo)	Disponibles sin costo adicional
	Librerías open-source para IA: Python, OpenCV, Tesseract OCR, Pandas	
Datos	Capturas de pantalla de portales públicos: SRI, MSP, SENESCYT, Fiscalía. Dataset estructurado y preprocesado para análisis de IA	Acceso libre; capturas automatizadas mediante bot

7.3. Recursos Financieros

Categoría	Estimación / Comentarios
Licencias software	ElectroNeek: préstamo temporal; costo anual según proveedor (anexo carta)
Hardware y almacenamiento	Uso de equipos existentes; no se requiere inversión adicional
Consultoría externa	Honorarios por sesiones puntuales (estimado según necesidad)

Contingencias (15-20%)	Considerar para licencias, imprevistos técnicos o expansión del proyecto
------------------------	--

7.4. Presupuesto total estimado:

- Se mantiene en \$0, dado que la mayoría de recursos son internos y la licencia de ElectroNeek es prestada.
- Contingencia incluida para cualquier eventualidad o necesidad de compra de licencias adicionales.

7.5. Plan de Adquisición y Gestión de Recursos

Recursos humanos:

- Ambos integrantes asumirán roles core (desarrollo, análisis y PM).
- La consultoría externa se solicitará según las necesidades del proyecto, específicamente para la validación técnica en temas de Inteligencia Artificial. Este rol podría ser desempeñado por la PhD. Gladys Villegas.

Recursos técnicos:

- Configurar estaciones de trabajo con librerías y entornos necesarios.
- Implementar almacenamiento estructurado para capturas de pantalla y dataset.
- Confirmar disponibilidad y autorización de ElectroNeek (carta como anexo).

Recursos financieros:

- Control de gastos de licencias externas y posibles servicios en la nube.
- Mantener registro de contingencias y justificar cualquier gasto adicional.

8. HITOS Y ENTREGABLES

A continuación, se detallan los 8 hitos claves para asegurar el progreso y calidad.

ID	Detalles del Hito	
H1	<p>Descripción: Diseño de Arquitectura y Selección del Modelo</p> <p>Fecha de finalización: Semana 1 (Día 7)</p>	<p>Entregable: Documento pdf con análisis del proyecto y Diseño</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> Arquitectura claramente definida Selección del modelo <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Presentación formal a los stakeholders, se requiere aprobación verbal para continuar con el proyecto</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> Riesgo: La arquitectura propuesta es demasiado compleja para el tiempo asignado Contingencia: Reducir el alcance a un prototipo más simple
H2	<p>Descripción: Preparación del Entorno de desarrollo</p> <p>Fecha de finalización: Semana 2 (Día 14)</p>	<p>Entregable: Repositorio en GitHub configurado y dataset sintético inicialmente cargado</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> Scripts Dataset Sintético con al menos 10 Capturas de 2 páginas debidamente etiquetadas <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Verificar dataset cargado en el github: ancantos99/proyectointegrador_ia_grupo8</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> Riesgo: Dataset insuficiente Contingencia: Inicialmente tenemos pensado crear nuestro propio dataset, pero se puede analizar utilizar algún dataset como el de huggingface (https://huggingface.co/datasets/YashJain/UI-Elements-Detection-Dataset)
H3	<p>Descripción:</p>	<p>Entregable: Prototipo funcional del algoritmo (modelo) con el código documentado</p> <p>Criterio de Aceptación:</p>

	<p>Implementación del algoritmo Base</p> <p>Fecha de finalización: Semana 3 (Día 21)</p>	<ul style="list-style-type: none"> El algoritmo o Modelo se ejecuta sobre el dataset y genera detección de cambios Precisión mínima ≥ 0.65 en pruebas iniciales <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Pruebas documentadas en GitHub</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> Riesgo: Bajo desempeño del modelo Contingencia: Volver al hito 2 para revisar el dataset de entrenamiento
H4	<p>Descripción: Pipeline en Electroneek</p> <p>Fecha de finalización: Semana 3 (Día 21)</p>	<p>Entregable: Un Bot de Electroneek con el Pipeline creado y listo para ser integrado con la IA</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> El Script o Pipeline que se usará en el Bot que inicialmente capture la pantalla sobre la que está trabajando <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: El Bot ejecuta correctamente la tarea y captura de pantalla</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> Riesgo: El Bot no captura la pantalla Contingencia: Probar la ejecución en diferentes entornos, validar permisos de usuario, y tener un script alternativo de captura (ej. con librerías nativas de Python/PowerShell)
H5	<p>Descripción: Integración de Bot con IA</p> <p>Fecha de finalización: Semana 4 (Día 28)</p>	<p>Entregable: Un Bot de Electroneek integrado (rpa)</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> El Bot ejecuta correctamente la tarea para la que fue diseñado y captura la pantalla El modelo IA fue exportado y cargado junto con el Bot <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Pruebas de integración demostrables</p> <p>Riesgos y contingencia:</p>

		<ul style="list-style-type: none"> • Riesgo: Problemas de compatibilidad o comunicación entre Electroneek y el modelo IA exportado • Contingencia: implementar un Middleware o Api simple para gestionar la comunicación entre el Bot y el modelo IA
H6	<p>Descripción: Optimización y Validación del modelo</p> <p>Fecha de finalización: Semana 5 (Día 35)</p>	<p>Entregable: Informe de Validación, pdf con métricas</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> • F1-score ≥ 0.75 en detección de cambios relevantes • Tiempo de detección ≤ 5 s por interfaz • Tasa de errores críticos $\leq 5\%$ <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Las pruebas deben estar debidamente documentadas</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> • Riesgo: Métricas debajo del Umbral • Contingencia: Ajustar hiper parámetros, analizar la posibilidad de obtener más datos o realizar un re-entrenamiento del modelo con técnicas de aumento de datos (data augmentation)
H7	<p>Descripción: Pruebas de Sistema con Casos de Uso reales</p> <p>Fecha de finalización: Semana 5 (Día 35)</p>	<p>Entregable: Reporte de pruebas del sistema completo</p> <p>Criterio de Aceptación:</p> <ul style="list-style-type: none"> • El sistema completo (Bot + IA) pasa al menos el 95% de los casos de pruebas definidos • Las pruebas deben ser reproducibles <p>Responsable: Andrés y Paola</p> <p>Procedimiento de Revisión: Reproducción de las pruebas</p> <p>Riesgos y contingencia:</p> <ul style="list-style-type: none"> • Riesgo: Fallos inesperados en los casos de prueba finales • Contingencia: Se realizan sesiones de depuración intensivas para resolver los

		errores y se ajusta el plan de pruebas si es necesario
H8	Descripción: Documentación y Presentación Fecha de finalización: Semana 6 (Día 42)	Entregable: Documentación técnica en repositorio github, Presentación final (Slides con Objetivos y logros alcanzados) Criterio de Aceptación: <ul style="list-style-type: none"> La documentación técnica es completa y debe describir todos los componentes de la solución Responsable: Andrés y Paola Procedimiento de Revisión: Defensa final Riesgos y contingencia: <ul style="list-style-type: none"> Riesgo: Falta de tiempo o falla durante la demostración final en vivo Contingencia: Prepara

Bibliografía

- [1] J. Wu, S. Wang, S. Shen, Y.-H. Peng, J. Nichols y J. P. Bigham, «WebUI: A Dataset for Enhancing Visual UI Understanding with,» *arXiv (Cornell University)*, 30 Enero 2023.
- [2] J. Vyskočil y P. Lukáš, «Improving web user interface element detection using Faster R-CNN,» *Proceedings of the Working Notes of CLEF 2021 – Conference and Labs of the Evaluation Forum*, pp. 1375-1386, 2021.
- [3] R. Khankhoje, «WEB PAGE ELEMENT IDENTIFICATION USING SELENIUM AND CNN: A NOVEL APPROACH,» *Journal of Software Quality Assurance (JSQA)* , vol. 1, pp. 1-17, 1 10 2023.
- [4] A. A. J. Cizotto, R. C. T. De Souza, V. C. Mariani y L. D. S. Coelho, «Web pages from mockup design based on convolutional neural network and class activation mapping,» *Multimedia Tools and Applications*, vol. 82, p. 38771–38797, 2023.

- [5] R. Zhang, H. Zhang, X. Ning, X. Huang, J. Wang y W. Cui, «Global-aware siamese network for change detection on remote sensing images,» *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 199, pp. 61-72, Mayo 2023.
- [6] X. Gao y G. Zheng, «SMILE: Siamese Multi-scale Interactive-representation LEarning for Hierarchical Diffeomorphic Deformable image registration,» *Computerized Medical Imaging and Graphics*, vol. 111, p. 102322, Enero 2024.
- [7] I. E. Livieris, E. Pintelas, N. Kiriakidou y P. Pintelas, «Explainable Image Similarity: Integrating Siamese Networks and Grad-CAM,» *Journal of Imaging*, vol. 9, nº 10, p. 224, 14 Octubre 2023.
- [8] Z. Wang, M. Xu, Z. Wang, Q. Guo y Q. Zhang, «ScribbleCDNet: Change detection on high-resolution remote sensing imagery with scribble interaction,» *International Journal of Applied Earth Observation and Geoinformation*, vol. 128, p. 103761, 12 Marzo 2024.
- [9] I. Corley, C. Robinson y A. Ortiz , «A change detection reality check,» *arXiv preprint arXiv:2402.06994v2*, Febrero 2024.
- [10] H. Pathak y R. Kapoor, «Computer Vision for UI Testing: Leveraging Image,» *TechRxiv*, 4 Abril 2025.
- [11] Y. Li, T. Zhang y C. L. P. Chen, «A Survey on Siamese Network: Methodologies, Applications and Opportunities,» *IEEE Transactions on Artificial Intelligence*, vol. 3, nº 6, pp. 994-1014, 15 Septiembre 2022.
- [12] M. Moradi, K. Yan y R. Asgari, «Artificial intelligence for context-aware visual change detection in software test automation,» *arXiv (Cornell University)*, 1 Mayo 2024.
- [13] J. Cho, J. Kim, D. Bae, J. Choo, G. Gwon y Y.-D. Kwon, «CAAP: Context-Aware Action Planning Prompting to Solve Computer Tasks with Front-End UI Only,» *arXiv*, 12 Junio 2024.
- [14] B. Marín, F. P. Ricós, T. E. Vos, R. Neeft y P. Aho, «Delta GUI change detection using inferred models,» *Computer Standards & Interfaces*, vol. 92, p. 103925, 1 Marzo 2025.

- [15] M. Ayli, Y. Bakouny, N. Jalloul y R. Kilany, «Enhancing the Resiliency of Automated Web Tests with Natural Language,» *Association for Computing Machinery*, pp. 63-69, 27 Septiembre 2024.
- [16] Z. Khaliq, S. U. Farooq y D. A. Khan, «A deep learning-based automated framework for functional User Interface testing,» *Information and Software Technology*, vol. 150, p. 106969, 7 Junio 2022.
- [17] S. Dirnstorfer, «Spotting Image Differences in Visual Software Testing with AI,» *InfoQ*, 11 Junio 2025.

ANEXOS

Dataset- SRI

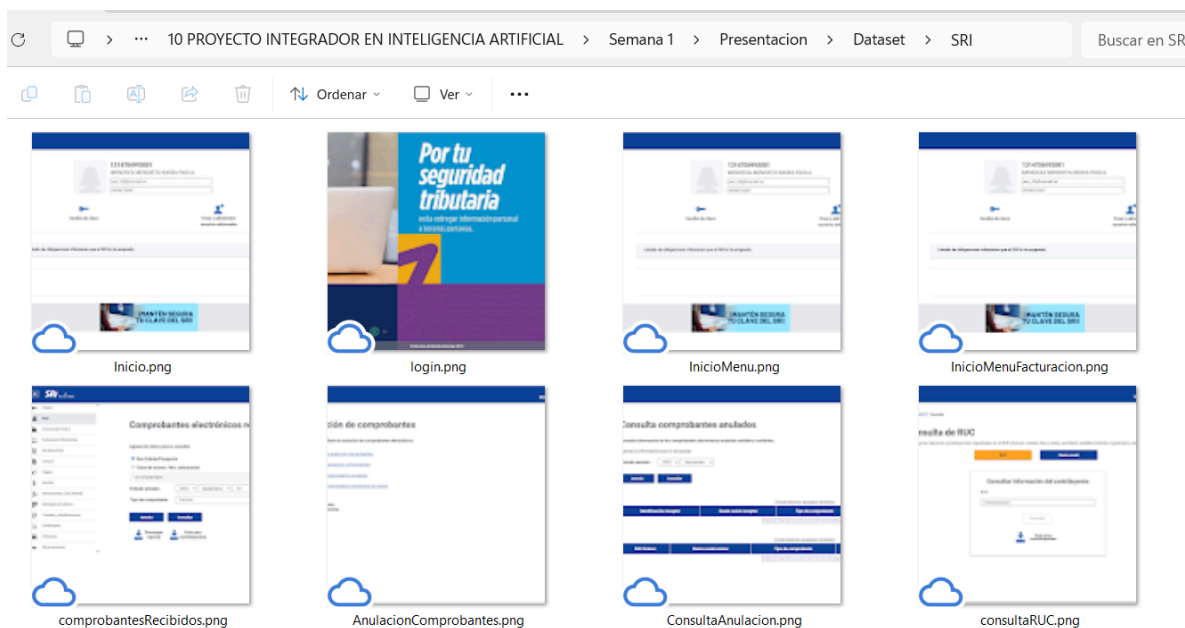
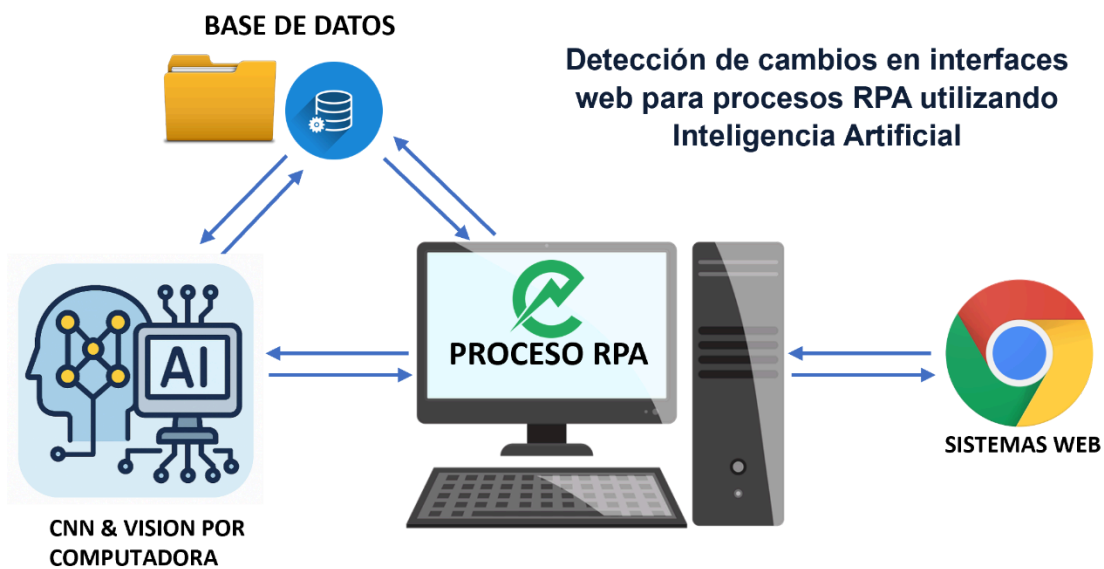


Diagrama de arquitectura



Carta de permiso de licencia de RPA- Electroneek



Guayaquil, 18 de septiembre del 2025

A quien corresponda:

Por la presente, RECREAMARKETING S.A., con RUC:0992958456001, declara que autoriza al equipo de maestría de Inteligencia Artificial de la **Universidad de Especialidades Espíritu Santo (UEES)**, conformado por:

- **María Paola Mendoza Mendieta**
- **Andrés Martin Cantos Rivadeneira**

a utilizar la licencia del software **ElectroNeek**, propiedad de RECREAMARKETING S.A, para fines exclusivos del desarrollo del proyecto integrador titulado:

"Detección de cambios en interfaces web para procesos RPA utilizando Inteligencia Artificial"

La presente autorización es temporal y de carácter **no comercial**, destinada únicamente al cumplimiento de los objetivos académicos del proyecto de maestría.

Para constancia de lo anterior, suscriben la presente:

Atentamente,



Luis Miguel Patiño
Gerente General
RECREAMARKETING S.A.