ISBN: 978-628-96613-0-9. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LEIRD2024.1.1.709

Generating Dashboards with Python on the RPA process for optimizing the assignment of gardening tasks

Generación de Dashboards con Python sobre el proceso RPA de optimización de la asignación de tareas de jardinería

Aradiel Castañeda Hilario, Doctor[1], Acosta de la Cruz Pedro Raúl, Msc[1], Gerónimo Vásquez Alfonso Herminio, Doctor1 , Flores Salinas, José Alberto[1],Universidad Nacional de Ingeniería, Perú, haradiel@uni.edu.pe, pacosta@uni.edu.pe, ageronimov@uni.edu.pe, Jflores@uni.edu.pe

***Abstract:** This article presents the results of a research focused on the development and implementation of dashboards for the visualization of Key Performance Indicators (KPIs) in the context of the automation of the assignment of gardening tasks through Robotic Process Automation (RPA). The research was carried out under a mixed methodological approach, using the quantitative method for the collection and analysis of operational data, as well as the qualitative method through interviews and observations to evaluate the perception of end users on the effectiveness of the solution.*

The research design was based on the Design Thinking methodology for the identification of needs and the development of iterative prototypes, complemented by the Lean Six Sigma methodology to optimize processes and reduce inefficiencies. During the development phase, critical KPIs such as response time, service level and percentage of tasks correctly assigned were selected. Based on these indicators, interactive dashboards were designed and built using advanced visualization tools, such as Tableau and Power BI, and integrated with an RPA system set up in a controlled environment.

The results indicated significant improvements in three key areas: (1) a 40% increase in operational efficiency through the automation of repetitive tasks; (2) a 25% decrease in the number of errors in task assignment; and (3) the ability to monitor process performance in real time, facilitating informed decision-making. In addition, a high level of satisfaction was observed among users, who highlighted the clarity and usefulness of the dashboards to visualize the workflow and manage resources proactively.

In the discussion, the positive impact of combining RPA with KPI visualization to optimize operational processes is highlighted, and areas for improvement are identified, such as the need to customize dashboards for different user profiles. Finally, recommendations are offered for large-scale implementation in other service areas, highlighting the importance of continuous adaptation and real-time impact assessment to ensure sustained performance.

This study provides new insights into the strategic use of Robotic Process Automation and dashboards as complementary tools for operational process improvement and effective resource management in service-oriented industries.

Keywords–Python, RPA, UNI, Task Assignment, Landscaping, Dashboard, KPI..

***Resumen:** Este artículo presenta los resultados de una investigación centrada en el desarrollo e implementación de dashboards para la visualización de Indicadores Clave de Desempeño (Key Performance Indicators, KPIs) en el contexto de la automatización de la asignación de tareas de jardinería mediante Automatización Robótica de Procesos (Robotic Process Automation, RPA). La investigación se llevó a cabo bajo un enfoque metodológico mixto, empleando el método cuantitativo para la recolección y análisis de datos operativos, así como el método cualitativo a través de entrevistas y observaciones para evaluar la percepción de los usuarios finales sobre la efectividad de la solución.*

El diseño de la investigación se basó en la metodología Design Thinking para la identificación de necesidades y el desarrollo de prototipos iterativos, complementada con la metodología Lean Six Sigma para optimizar los procesos y reducir las ineficiencias. Durante la fase de desarrollo, se seleccionaron KPIs críticos como tiempo de respuesta, nivel de servicio y porcentaje de tareas asignadas correctamente. A partir de estos indicadores, se diseñaron y construyeron dashboards interactivos utilizando herramientas de visualización avanzadas, como Tableau y Power BI, y se integraron con un sistema de RPA configurado en un entorno controlado.

Los resultados indicaron mejoras significativas en tres áreas clave: (1) un incremento del 40% en la eficiencia operativa a través de la automatización de tareas repetitivas; (2) una disminución del 25% en el número de errores en la asignación de tareas; y (3) la capacidad de monitorear en tiempo real el desempeño del proceso, facilitando la toma de decisiones informadas. Además, se observó un alto nivel de satisfacción por parte de los usuarios, quienes resaltaron la claridad y utilidad de los dashboards para visualizar el flujo de trabajo y gestionar los recursos de manera proactiva.

En la discusión, se resalta el impacto positivo de combinar RPA con visualización de KPIs para optimizar procesos operativos, y se identifican áreas de mejora como la necesidad de personalizar los dashboards para diferentes perfiles de usuario. Finalmente, se ofrecen recomendaciones para la implementación a gran escala en otras áreas de servicios, subrayando la importancia de la adaptación continua y la evaluación de impacto en tiempo real para asegurar un rendimiento sostenido.

Este estudio aporta nuevas perspectivas sobre el uso estratégico de Automatización Robótica de Procesos y dashboards como herramientas complementarias para la mejora de procesos

operativos y la gestión efectiva de recursos en industrias orientadas a servicios.

Palabras clave–Python, RPA, UNI, Asignación de tareas, Jardinería, Dashboard, KPI..

I. Introduction

En un mundo donde la tecnología avanza a pasos agigantados, la automatización de procesos se ha convertido en un elemento crucial para mejorar la eficiencia y la productividad en diversas industrias. En particular, en el sector de servicios de jardinería, la optimización de la gestión de tareas es fundamental para garantizar un mantenimiento paisajístico eficiente y de alta calidad. En este contexto, la automatización de asignación de tareas de jardinería mediante Robotic Process Automation (RPA) emerge como una solución innovadora y prometedora.

Esta investigación se centra en el desarrollo de dashboards para la visualización de Key Performance Indicators (KPIs) en el proceso de automatización de asignación de tareas de jardinería mediante RPA. La implementación de dashboards efectivos permite monitorear en tiempo real el rendimiento del sistema automatizado, facilitando la toma de decisiones informadas y la identificación de áreas de mejora continua.

En esta introducción, se establece el contexto de la investigación y se delinean los objetivos y la importancia del estudio. Se discute brevemente la relevancia de la automatización mediante RPA en el sector de servicios de jardinería, así como la necesidad de desarrollar dashboards para la visualización de KPIs como herramienta clave para mejorar la gestión de tareas. Además, se presenta un resumen de la estructura del artículo, destacando los principales temas que se abordarán en las secciones siguientes.

A través de esta investigación, se espera contribuir al conocimiento en el campo de la automatización de procesos en el sector de servicios de jardinería, proporcionando una guía práctica y teórica para el desarrollo e implementación efectiva de dashboards para la visualización de KPIs en entornos de RPA.

Con la creciente demanda de eficiencia operativa y reducción de costos, las organizaciones de servicios de jardinería enfrentan el desafío de optimizar sus procesos de asignación de tareas. Sin embargo, la naturaleza manual de estos procesos presenta varias limitaciones:

1. Ineficiencia en la Asignación de Tareas: La asignación manual de tareas puede llevar a una distribución desigual del trabajo, donde algunos empleados están sobrecargados mientras que otros están subutilizados.

2. Errores Humanos: La intervención humana en la asignación de tareas puede resultar en errores, como la asignación de tareas incorrectas o duplicadas, lo que puede afectar negativamente la productividad y la satisfacción del cliente.

3. Falta de Monitoreo y Medición: Sin herramientas adecuadas para monitorear y medir el desempeño, es difícil evaluar la efectividad de las tareas realizadas y realizar ajustes en tiempo real.

4. Visibilidad Limitada: La falta de visibilidad sobre el progreso de las tareas y el rendimiento del equipo puede dificultar la toma de decisiones informadas y la planificación a largo plazo.

Planteamos como objetivo general de nuestro trabajo de investigación

• Desarrollar dashboards efectivos usando lenguaje Python para la visualización de KPIs de valor en el proceso de automatización de la asignación de tareas de jardinería mediante RPA, con el fin de mejorar la eficiencia operativa, la precisión en la asignación de tareas y la capacidad de monitoreo y evaluación del desempeño.

De la misma manera nuestros Objetivos Específicos son los siguientes:

•1 Determinar los indicadores de rendimiento más relevantes para evaluar la eficiencia y efectividad del proceso automatizado de asignación de tareas de jardinería.

•2 Crear dashboards intuitivos y visualmente atractivos que permitan una interpretación rápida y precisa de los KPIs.

•3 Analizar las métricas que pueden proporcionar una visión integral del desempeño del equipo y la calidad del servicio.

Además, para constatar nuestro trabajo planteamos las siguientes preguntas de investigación

• ¿Cómo puede la implementación de un sistema automatizado de RPA mejorar la eficiencia en la asignación de tareas de mantenimiento de áreas verdes en el campus universitario?

• ¿Qué impacto tiene la automatización de la asignación de tareas en la planificación y ejecución del mantenimiento de áreas verdes?

• ¿Cuáles son los beneficios específicos de monitorear el rendimiento del proceso RPA usando dashboards?.

II. Estado del Arte

Con el avance tecnológico de los últimos años, cada día se encuentra más máquinas o dispositivos que facilitan la realización de tareas manuales, hasta el punto de que son capaces de hacerlas por completo sin ninguna supervisión o intervención humana. En este punto es dónde RPA (Robotic Process Automation) tiene su impacto, pues con esta tecnología se puede crear, implementar y administrar robots de software [5]. Cabe resaltar que estos robots, también conocidos como asistentes digitales o bots, no tienen como objetivo reemplazar a las personas sino apoyarlas en sus actividades, ya que son programas informáticos que emulan o reproducen las acciones humanas en los sistemas digitales, de manera más rápida, sin errores y sin necesidad de tomar un descanso [5]. El mantenimiento de áreas verdes es fundamental para la conservación del medio ambiente y la mejora de la calidad de vida, especialmente en contextos universitarios. Sin embargo, la gestión de estas tareas se enfrenta a varios desafíos operativos que afectan su eficiencia. Tradicionalmente, estas tareas implican asignaciones manuales, coordinación entre diferentes equipos y seguimiento continuo, lo cual puede resultar en ineficiencias y errores humanos.

La automatización se puede aplicar a cualquier proceso, pero su diseño difiere significativamente. Ya que es necesario considerar múltiples factores, tales como: la estabilidad del proceso, numero de posibles puntos de fallo y error, plataforma o tecnología que permita su despliegue, de tal manera que la estrategia de la automatización pueda tener un gran impacto en el costo total del proyecto debido a los altos requisitos de mantenimiento. [2] La aplicación de la Robotic Process Automation (RPA) en entornos empresariales ha demostrado ser una herramienta eficaz para mejorar la eficiencia operativa y la productividad. La integración de RPA en una organización requiere identificar necesidades específicas que puedan ser automatizadas de manera eficiente. En este contexto, se destaca la importancia de evaluar la viabilidad de soluciones de RPA antes de su implementación, considerando procesos de alto volumen y repetitivos que sean susceptibles de automatización. [7]

El proyecto de implementación de RPA en la UNI se centra en la automatización del flujo de trabajo relacionado con la coordinación de tareas de mantenimiento de parques. La propuesta consiste en desarrollar un sistema automatizado que registre las tareas pendientes y las asigne eficientemente a los trabajadores. Esta automatización no solo optimizará los procesos operativos, sino que también liberará recursos humanos para actividades más estratégicas y de mayor valor agregado. En una empresa fácilmente se pueden encontrar procesos o subprocesos que son repetitivos, de baja remuneración, que generan poco valor, tediosos y que no dan satisfacción laboral a los empleados. La solución puede encontrarse en la configuración de un EPR u otro sistema que realice el proceso de forma automática, pero resulta ser muy costoso, su implementación puede requerir cambios estructurales y además son proyectos que tardan mucho tiempo. Por lo cual las empresas optan por mantener el proceso como esta o subcontratar alguna empresa que se haga cargo de dicho proceso. [4]

La automatización de procesos aplica tecnologías robóticas y cognitivas para automatizar tareas rutinarias, operadas manualmente y estandarizadas en beneficio de los trabajadores reconocidos de una empresa. El software robótico elimina la dependencia humana al liberar a los empleados humanos de estas tareas rutinarias para brindarles la oportunidad de concentrarse en los objetivos y operaciones comerciales centrales. La automatización ofrece muchos beneficios convincentes para el lugar de trabajo y una gestión eficaz y estratégica de los recursos humanos. [3] Uno de los principales desafíos en la gestión de áreas verdes en la UNI es la asignación manual de tareas, que consume mucho tiempo y es propensa a errores. Además, la comunicación ineficiente entre administradores y trabajadores y la subutilización de las

3

habilidades de los empleados resultan en un uso ineficiente de los recursos humanos y naturales.

En la actualidad existen muchos proveedores de tecnología que implementan RPA, ya que al elegir la plataforma adecuada es uno de los pasos clave en el éxito general de la implementación de RPA. [2] La implementación de RPA abordará estos problemas mediante la creación de un proceso automatizado para la gestión de tareas repetitivas. El sistema desarrollado en Python permitirá asignar tareas automáticamente a los jardineros y enviar las asignaciones por correo electrónico. Con esta automatización, se espera mejorar significativamente la eficiencia operativa y reducir la carga administrativa.

Los RPA son una tecnología enfocada a emular las acciones de una persona en un sistema informático y tiene como fin aumentar la eficiencia en los procesos de negocio, reduciendo tiempos y errores humanos; en esta tecnología se usan robots de software que pueden adaptarse a procesos existentes dentro de una organización. Por lo antes mencionado, distintos sectores han reaccionado ante el surgimiento de esta herramienta, el sector BPO es un ejemplo. Las empresas BPO, se especializan en la mejora de la eficiencia global corporativa de organizaciones de gran variedad de sectores, encargándose de la gestión de áreas puntuales dentro de la compañía; uno de sus campos o áreas de acción es la atención al cliente, en donde comúnmente se utiliza un Contact Center y en donde, debido a los procesos que se manejan, los RPA tiene gran campo de acción. En la Figura 1 se puede observar cómo funciona una BPO. [2]

En el ámbito de la seguridad física, se identifica la necesidad de implementar RPA debido a la presencia de procesos que cumplen con los requisitos para ser automatizados, lo que podría beneficiarse de una automatización desatendida, eliminando la necesidad de intervención humana en ciertas tareas. [1]

Plataforma de automatización robótica de procesos (RPA) con múltiples funciones e intuitiva, diseñada para ayudar a los analistas y administradores de negocios a automatizar los procesos dentro de sus negocios. Este software proporciona a las empresas ventajas de automatización global, especialmente en centros de llamadas, administración de documentos, finanzas, atención médica, habilitación de APIs, automatización de procesos, extracción y migración de documentos, verticales de tercerización de procesos e integración de aplicaciones. [2] Este enfoque innovador puede servir como modelo para otras instituciones educativas que enfrentan desafíos similares en la gestión de sus áreas verdes

III. METODOLOGIA

Para el diseño y elaboración del RPA se siguió la metodología para el uso de RPA del artículo “Optimización de ingreso de ordenes de servicio mediante RPA” (A41), se eligió esta metodología ya que se asemeja al desarrollo que se está haciendo, además presenta buena base teórica

Fig 1: Metodología para el desarrollo de un RPA

Donde presenta las siguientes fases

-Fase 1 o Construcción perspectiva teórica: En esta fase se realiza un estudio o investigación general del RPA en cuanto a su funcionamiento, herramientas y más para tener referencias o base en la planeación de la fase 2, es decir, para obtener un RPA que lea, registre y coordine las tareas de mantenimiento de las áreas verdes en lenguaje Python.

-Fase 2 o Desarrollo de flujo: Esta fase consiste en el desarrollo en python del RPA en el entorno de prueba Visual Studio haciendo uso de las librerías adecuadas para automatizar el registro y coordinación de tareas de mantenimiento de las áreas verdes.

-Fase 3 o Implementación de Prueba piloto: En esta etapa se obtiene la ruta en la que se encuentran la programación de tareas asignadas y se prueba en ellas el RPA para obtener el envío de correos adjuntando dichas tareas a los jardineros.

-Fase 4 o Comparación: En esta etapa se produce una comparación entre el tiempo de asignación de tareas de forma manual y automatizada, para detectar posibles mejoras en el sistema para los jardineros.

A. Arquitectura del RPA

Para expresar la arquitectura del RPA se presenta las funcionalidades del sistema de la siguiente manera:

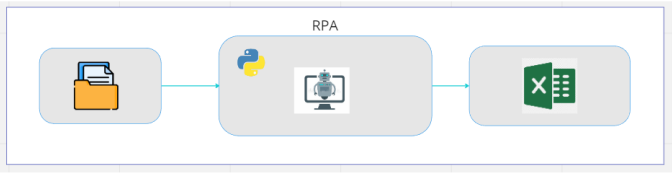

Fig 2: Arquitectura del RPA

- Los archivos que se encuentran en el repositorio drive contienen los datos de los trabajadores y las tareas por asignar, en excel,son el recurso al que el RPA accede para su respectivo registro y coordinación de tareas.

- El RPA se desarrolla con el lenguaje Python y un conjunto de librerías para realizar la captura de los datos de los datos de cada jardinero, luego los evalúa para asignarle tareas semanales y compara los tiempos que se hizo esta tarea programada con una hecha de manera manual.

- El resultado de las asignaciones por tarea será exportados de forma automática a un archivo de texto y se convertirá en pdf en el que se ordenan los nombres de los jardineros con sus respectivas tareas asignadas de cada día en la semana.

B. Técnicas e Instrumentos de recolección de información

Para poder desarrollar un caso de uso de RPA aplicando al registro y coordinación de tareas de mantenimiento de áreas verdes del campus, primero se tuvo que consultar la información que en el archivo de indicaciones ya se nos presenta, posteriormente es la revisión de los papers, como se pudo ver se reconoció una metodología de acuerdo con el trabajo,

Cabe mencionar que este será el uso del responsable en programación de tareas, el responsable podrá preparar el envío de un correo adjuntando archivos pdf y así el programa podrá asignar tareas y como salida tendrá el envío del archivo pdf sobre la asignación de tareas y si se desea buscar el rendimiento de esta automatización, será otra función que posea, en base a los archivos ofrecidos por la dirección que establece las tareas manualmente, se estará trabajando así debido a que son los únicos archivos que se posee

Continuando con los papers, para la base teórica se utilizó el paper “Impact of RPA Technologies on Accounting Systems", donde nos ofrece una explicación del RPA así como los beneficios que nos ofrecen, también los papers “Tecnologías RPA en el sector logístico”, “Propuesta para utilizar RPA en la resolución de incidencias del área de sistemas de una empresa de telefonía”, entre otros más.

Un caso de uso interesante de RPA es del siguiente paper “Diseño e implementación de asistente RPA para la facturación electrónica en la clínica somer y somer incare “, donde se explica que se usó RPA para administrar la descarga y registro de facturas a un correo de facturación en la Clínica Somer y Somer Incare, para darle solución a problemas como pérdida de facturas, duplicidad, confusión al asignarlas, demora a la hora de diligenciar los archivos o la eliminación de estos por parte de errores del personal encargado, además del almacenamiento del archivo XML junto a su respectiva factura en formato PDF.

Otros casos de uso son del paper “Diseño e implementación de asistente RPA para la facturación electrónica en la clínica somer y somer incare” donde se hace uso de un RPA para administrar la descarga y registro de las facturas entrantes a un correo de facturación en la Clínica Somer y Somer Incare, para darle solución a problemas como pérdida de facturas, duplicidad, confusión al asignarlas, demora a la hora de diligenciar los archivos o la eliminación de estos

por parte de errores del personal encargado, además del almacenamiento del archivo XML junto a su respectiva factura en formato PDF.

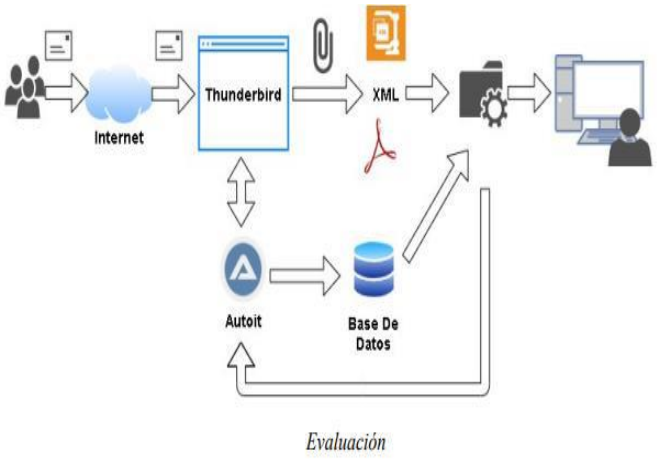

Fig 3: Proceso de automatización del proceso de la facturación

Otro caso de uso es del artículo Título: Apoyo en la gestión de proyectos de consultoría en tecnología con la aplicación RPA (Robotic Process Automation) (A15). Este artículo muestra el uso del RPA para el proceso de liquidación y su conexión con SAP de la siguiente manera:

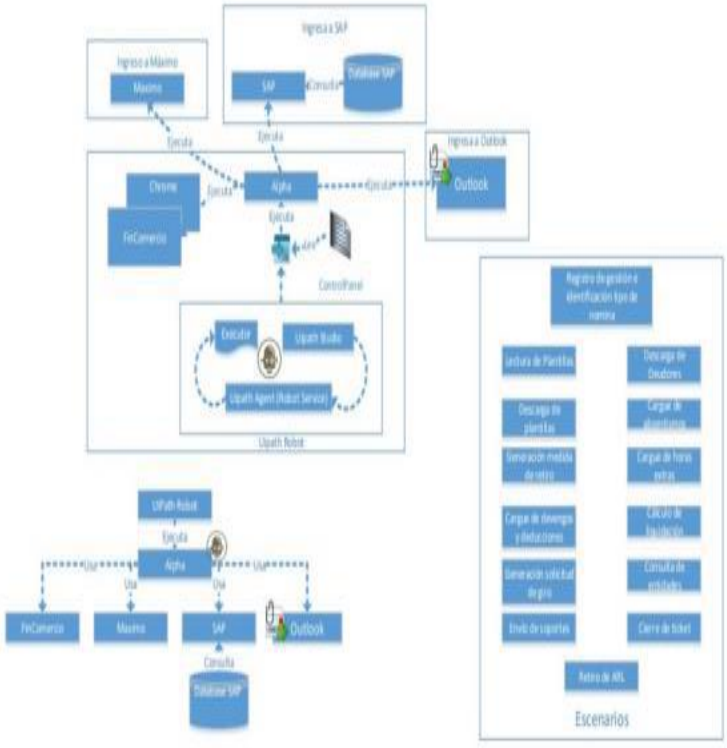

Fig 4: Proceso de automatización del proceso de liquidación

Algo similar pasó en los artículos “Proceso de desarrollo en RPA: Robots de software como apoyo a la realización de tareas repetitivas” (A8) y “Desarrollo e implementación de un RPA en los procesos de descarga de informes, segmentación y cargue de información a contacto” (A16). Ambos papers se centran en mejorar y automatizar la atención de PQRS (peticiones, quejas, reclamos y sugerencias) y por los resultados obtenidos ambos logran tener una optimización en dicho proceso. El artículo 8 expresa las tareas del RPA de la siguiente manera:

- Leer correos electrónicos desde una bandeja que cumplan con cierto asunto para obtener todos los casos a procesar.
- Por cada caso a procesar, se va a descargar y leer sus archivos adjuntos, para obtener el número del caso.
- Utilizar el componente para iniciar sesión en el sistema interno.
- Buscar el caso en el sistema interno.
- Subir el correo enviado en el sistema interno.
- Guardar en un archivo específico el registro de que al caso procesado se le ha subido su prueba (correo enviado).
- Notificar por correo electrónico que se ha completado la subida de la información

Mientras que el artículo 16 lo expresa en un flujo representado a continuación:

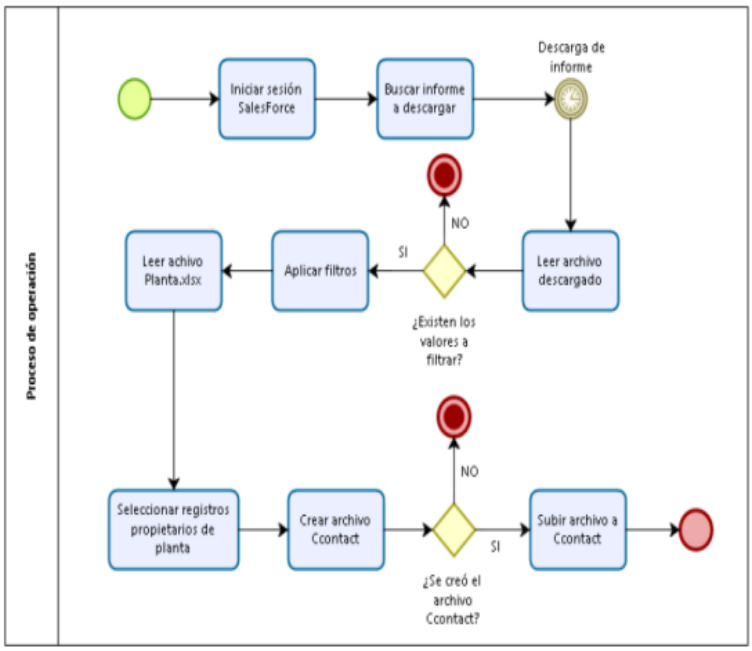

Fig 5: Proceso automatizado de la atención de PQRS

Algo sumamente importante los criterios a evaluar para ofrecer una automatización y gracias al paper “Propuesta de implementación de herramientas RPA en una empresa del sector BPO & Contact Center y su impacto en la productividad”

donde nos ofrece un listado de criterios para el desarrollo de RPA en un proceso o actividad.

Tabla 1: Criterios de automatización

	Autores				
Criterios	Fung	Deloitte	Capgemini Consulting	Murdoch	Moreno Zuluaga
Tareas manuales y repetitivas	X	X	X	X	X
Procesos maduros con reglas de negocio claras	X	X	X	X	X
Probabilidad de error humano	X	X	X	X	X
Alto volumen de interacciones	X	-	X	X	X
Procesos que no requieren análisis	X	X	X	X	X
Procesos sin situaciones imprevistas	X	X	X	X	-
Manejo de varios sistemas de información o aplicativos.	X	X	X	-	X

Para la elaboración y el diseño de los dashboards, se siguió la siguiente metodología:

• Fase 1 – Revisión de literatura: Realizar una revisión exhaustiva de la literatura existente sobre RPA, KPIs y dashboards, así como estudios previos en el ámbito de la automatización de tareas de jardinería, a partir de esto, se deben de identificar teorías, modelos y prácticas relevantes que informen el desarrollo y la implementación de los dashboards ligados a sus respectivos KPIs, que deben de proporcionar información sustancial sobre el rendimiento del RPA.

• Fase 2 – Elaboración de la data de prueba: Se contaba con una data de prueba utilizada previamente para el diseño del RPA, sin embargo, para fines prácticos para poder visualizar de una manera más adecuada los KPIs en los dashboards, se aumentó sustancialmente el número de trabajadores y de tareas de jardinería a realizar, por lo que se realizó una nueva data de prueba.

• Fase 3 – Implementación del Sistema RPA: Esta fase consiste en la implementación del Sistema RPA previamente diseñado, en el entorno de prueba seleccionado, en este caso Google Colab, haciendo uso de las librerías adecuadas para automatizar el registro y coordinación de tareas de mantenimiento de las áreas verdes.

• Fase 4 – Definición de los KPIs: En esta etapa se seleccionan los KPIs más relevantes identificados para el contexto de la automatización de tareas de jardinería, que además se adecuen a la información de la data de prueba.

KPIs a a analizar: 1. Número de tareas asignadas por trabajador (general), 2. Número de horas trabajadas por habilidad (general), 3. Número de horas trabajadas por trabajador y número de tareas por trabajador (comparativa uno x uno), 4. Horas trabajadas por día, 5. Número de tareas por habilidad, 6. Número de horas por prioridad de tarea, 7. Frecuencia de tareas no asignadas.

•

Fase 5 – Diseño de Dashboards: En esta etapa se diseñan y desarrollan los dashboards utilizando lenguaje Python, en este caso en particular, se utilizó en entorno Google Colab.

•

Fase 6 – Análisis de Datos: Analizar los datos cualitativos para identificar temas y patrones emergentes relacionados con la adopción y uso del sistema y analizar los datos cuantitativos utilizando técnicas estadísticas para evaluar el impacto del sistema.

C. Materiales

1) Python: Es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes.

2) OS: Es una librería de Python que proporciona una interfaz para interactuar con el sistema operativo. Permite realizar operaciones relacionadas con la manipulación de archivos y directorios, administración de rutas de archivo, control del entorno del programa y otras tareas relacionadas con el sistema operativo.

3) EmailMessage: El paquete email es una biblioteca para administrar mensajes de correo electrónico. Específicamente no está diseñado para realizar cualquier envío de mensajes de correo electrónico a SMTP (RFC 2821), NNTP u otros servidores; esas son funciones de módulos como smtplib y nntplib. El paquete email intenta ser lo más compatible con RFC, admitiendo RFC 5322 y RFC 6532, así como RFC relacionados con MIME como RFC 2045, RFC 2046, RFC 2047, RFC 2183 y RFC 2231.

4) SMTPLIB: El módulo smtplib define un objeto de sesión de cliente SMTP que se puede utilizar para enviar correo a cualquier máquina de Internet con un demonio de escucha SMTP o ESMTP. Para obtener detalles sobre el funcionamiento de SMTP y ESMTP, consulte RFC 821 (Protocolo simple de transferencia de correo) y RFC 1869 (Extensiones de servicio SMTP).

4) FPDF: PyFPDF es una biblioteca para la generación de documentos PDF en Python, portada desde PHP (consulte FPDF “Free”-PDF, un conocido reemplazo de la extensión PDFlib con muchos ejemplos, scripts y derivados). En comparación con otras bibliotecas de PDF, PyFPDF es simple, pequeña y versátil, con capacidades avanzadas y fácil de aprender, ampliar y mantener.

5) Google Colab: Google Colaboratory, también conocido como Colab, es un servicio gratuito en la nube que ofrece un entorno de desarrollo interactivo basado en Jupyter Notebooks. Permite a los usuarios escribir y ejecutar código Python sin necesidad de configurar ni instalar ningún software, aprovechando la potencia de los servidores de Google.

6) Librería Dash: Framework de código abierto para crear aplicaciones web analíticas (dashboards). Proporciona la estructura básica del dashboard, incluyendo componentes interactivos como gráficos, mapas, tablas, etc.

7) Librería Dash Bootstrap Components: Librería que extiende Dash con componentes basados en Bootstrap. Proporciona componentes prediseñados con estilos Bootstrap para mejorar la apariencia y usabilidad del dashboard.

Fases del Proyecto

Para poder implementar los dashboards de una manera efectiva, y que sobretodo, realicen un aporte sustancial de información para la toma de decisiones entorno al proceso RPA, se siguieron las siguientes fases.

A. Preparación de Datos

Para la realización de una mayor cantidad de datos con las pruebas del RPA, se preparó un Dataframe con los datos de prueba. Este Dataframe consta de dos hojas:

• Datos de Trabajadores: Contiene la información de los trabajadores, incluyendo identificadores únicos y detalles de contacto.

• Datos de Tareas: Contiene la lista de tareas a realizar, incluyendo descripciones y requisitos específicos.

Tabla 2: tares especificas

B. Importación de Librerías

Se instalaron e importaron las librerías necesarias para la implementación del RPA y la posterior realización de los dashboards en la plataforma Google Colab. Estas librerías incluyeron herramientas para la manipulación de datos, generación de reportes y envío de correos electrónicos.

```
[ ]  !pip install dash
     !pip install dash-bootstrap-components
     !pip install dash dash-bootstrap-components plotly pandas

[ ]  import dash
     from dash import dcc, html
     from dash.dependencies import Input, Output
     import plotly.express as px
     import dash_bootstrap_components as dbc
     import pandas as pd
```

Fig 6:Librería implementación RPA

C. Ejecución del RPA

Se ejecutó el código del proceso RPA previamente diseñado e implementado

```
def asignar_tareas_semanal(trabajadores, tareas):
    # Ordenar tareas por prioridad
    tareas.sort(key=lambda x: x.prioridad)

    # Crear una lista con los días de la semana (1=Lunes, 5=Viernes)
    dias_semana = list(range(1, 6))

    for tarea in tareas:
        # Repartir la frecuencia de la tarea en los días de la semana
        dias_asignados = 0
        for dia in dias_semana:
            if dias_asignados < tarea.frecuencia:
                # Filtrar trabajadores que tienen la habilidad necesaria, disponibilidad y experiencia
                candidatos = [t for t in trabajadores if tarea.habilidad_necesaria in t.habilidades and t.horas_disponibles >= tarea.duracion]

                if candidatos:
                    # Ordenar candidatos por la mayor experiencia y luego por la menor cantidad de horas asignadas
                    candidatos.sort(key=lambda x: (-x.experiencia, sum(len(x.tareas_asignadas[d]) for d in x.tareas_asignadas)))

                    # Asignar la tarea al primer candidato
                    trabajador_asignado = candidatos[0]
                    trabajador_asignado.tareas_asignadas[dia].append(tarea.nombre)
                    trabajador_asignado.horas_disponibles -= tarea.duracion
                    dias_asignados += 1
                else:
                    print(f"No hay trabajadores disponibles para la tarea: {tarea.nombre} en el día {dia}")

asignar_tareas_semanal(trabajadores, tareas)
```

```
No hay trabajadores disponibles para la tarea: Aplicar pesticidas en el día 5
No hay trabajadores disponibles para la tarea: Reparar cercas en el día 3
No hay trabajadores disponibles para la tarea: Reparar cercas en el día 4
No hay trabajadores disponibles para la tarea: Reparar cercas en el día 5
```

```
def generar_reporte_semanal(trabajadores):
    dias_semana = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes"]
    reporte = "Reporte de tareas semanal\n"

    for trabajador in trabajadores:
        reporte += f"{trabajador.nombre} {trabajador.apellido} tiene esta semana las siguientes tareas:\n"
        for dia, tareas in trabajador.tareas_asignadas.items():
            reporte += f"  {dias_semana[dia-1]}: {', '.join(tareas) if tareas else 'No hay tareas asignadas'}\n"
    return reporte
```

Fig 7: código del RPA

D. Diseño de funciones

Según lo que requiera el dashboard a realizar según cada KPI, se elabora la función que nos permitirá obtener la información sustancial que compondrá al dashboard.

```
[44] # Función para contar las tareas asignadas a cada trabajador y obtener los trabajadores según el número de tareas
     def contar_tareas(trabajadores):
         tareas_contadas = {}
         trabajadores_por_tareas = {}
         for trabajador in trabajadores:
             num_tareas = sum(len(tareas) for tareas in trabajador.tareas_asignadas.values())
             if num_tareas in tareas_contadas:
                 tareas_contadas[num_tareas] += 1
                 trabajadores_por_tareas[num_tareas].append(trabajador)
             else:
                 tareas_contadas[num_tareas] = 1
                 trabajadores_por_tareas[num_tareas] = [trabajador]
         return tareas_contadas, trabajadores_por_tareas


     # Contar las tareas asignadas
     tareas_contadas, trabajadores_por_tareas = contar_tareas(trabajadores)
```

Fig 8: Proceso de automatización RPA

E. Creación de dataframe

Se crea uno o más dataframes para el almacenamiento de datos para la elaboración de dashboards.

F. Crear el dashboard

Se inicializa la aplicación Dash para la elaboración del dashboard y se asigna la información contenida en el dataframe.

```
[57] # Inicializar la aplicación Dash
app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])

app.layout = html.Div([
    html.H1("Dashboard de Asignación de Tareas"),
    dcc.Graph(
        id='tareas-asignadas',
        figure=px.bar(df_tareas, x='Número de Tareas', y='Número de Trabajadores', title='Número de Tareas Asignadas a Trabajador
    ),
    html.H2("Trabajadores"),
    html.Div(id='trabajadores-lista')
])
```

Fig 9: Creacion del dashboard

G. Implementar el callback en los casos necesarios

Cuando el dashboard lo requiera, luego de haber usado Dash, se implementará el callback para poder hacer el gráfico más interactivo dependiendo de la necesidad del KPI.

```
# Callback para actualizar la lista de trabajadores al hacer clic en una barra
@app.callback(
    Output('trabajadores-lista', 'children'),
    Input('tareas-asignadas', 'clickData')
)
def actualizar_lista_trabajadores(clickData):
    if clickData:
        num_tareas = clickData['points'][0]['x']
        trabajadores_seleccionados = trabajadores_por_tareas[num_tareas]
        lista_trabajadores = html.Ul([html.Li(f"{t.nombre} {t.apellido} ({t.correo})") for t in trabajadores_seleccionado
    else:
        lista_trabajadores = html.P("Haga clic en una barra para ver los trabajadores.")
    return lista_trabajadores
```

Fig 10: Implementación del Callback

H. Ejecutar la aplicación para invocar el dashboard

Se ejecuta la aplicación invocandola para poder ejecutar el dashboard y visualizarlo.

```
[59] # Ejecutar la aplicación
     if __name__ == '__main__':
         app.run_server(debug=True)
```

Fig 11: ejecución de aplicación

I. Repetir el proceso base para los demás KPIs

El proceso ilustrado es la estructura base de la creación de los distintos dashboards creados para cada uno de los KPIs estudiados para evaluar el rendimiento del proceso RPA.

DASHBOARD 6: Número de horas por prioridad de la tarea

```
[ ] import dash
    import dash_bootstrap_components as dbc
    from dash import dcc, html
    from dash.dependencies import Input, Output
    import plotly.express as px
    import pandas as pd
```

```
# Función para calcular horas asignadas por prioridad
def calcular_horas_por_prioridad(trabajadores):
    horas_por_prioridad = {}
    for trabajador in trabajadores:
        for dia, tareas_asignadas in trabajador.tareas_asignadas.items():
            for tarea_nombre in tareas_asignadas:
                tarea = next(t for t in tareas if t.nombre == tarea_nombre)
                if tarea.prioridad not in horas_por_prioridad:
                    horas_por_prioridad[tarea.prioridad] = 0
                horas_por_prioridad[tarea.prioridad] += tarea.duracion
    return horas_por_prioridad

# Calcular horas por prioridad
horas_por_prioridad = calcular_horas_por_prioridad(trabajadores)
df_horas_prioridad = pd.DataFrame(list(horas_por_prioridad.items()), columns=['Prioridad', 'Horas Asignadas'])
```

```
[161] # Inicializar la aplicación Dash
app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])

app.layout = html.Div([
    html.H1("Dashboard de Horas Asignadas por Prioridad de Tarea"),
    dcc.Graph(
        id='horas-por-prioridad',
        figure=px.bar(df_horas_prioridad, x='Prioridad', y='Horas Asignadas', title='Horas Asignadas por Prioridad',
                      labels={'Prioridad': 'Prioridad de Tarea', 'Horas Asignadas': 'Horas Asignadas'})
    )
])
```

```
[162] # Ejecutar la aplicación
    if __name__ == '__main__':
        app.run_server(debug=True)
```

Fig 12: KPIS del rpa

IV. RESULTADOS

Al ejecutar el código en la plataforma Colab, se pudieron visualizar cada uno de los dashboards designados para los KPIs:

1. Número de tareas asignadas por trabajador (general)

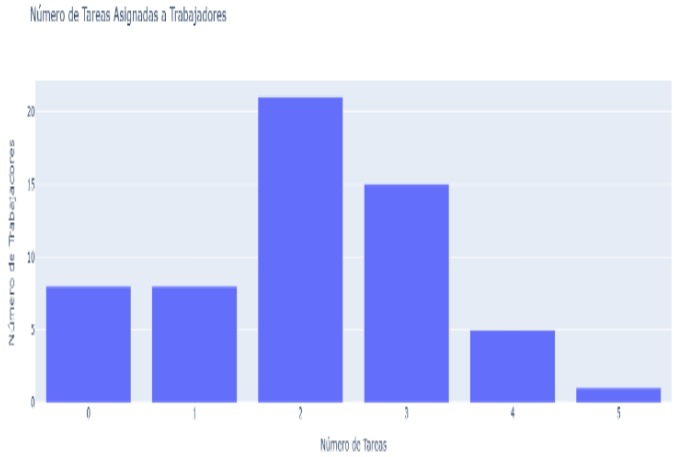

Fig 13: asignación de tareas

2. Número de horas trabajadas por habilidad (general)

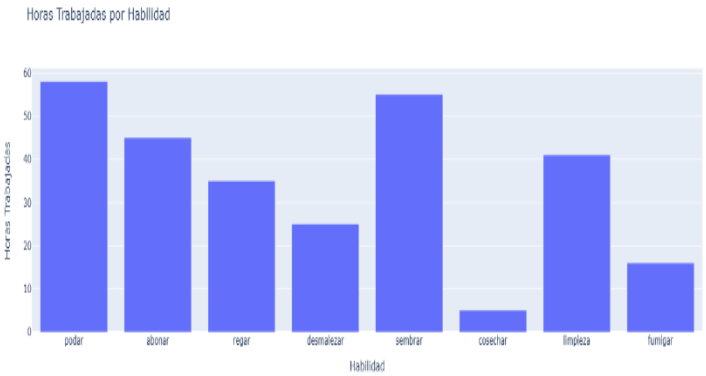

Fig 14: Tareas trabajadas

3. Número de horas trabajadas por trabajador y número de tareas por trabajador (comparativa uno x uno)

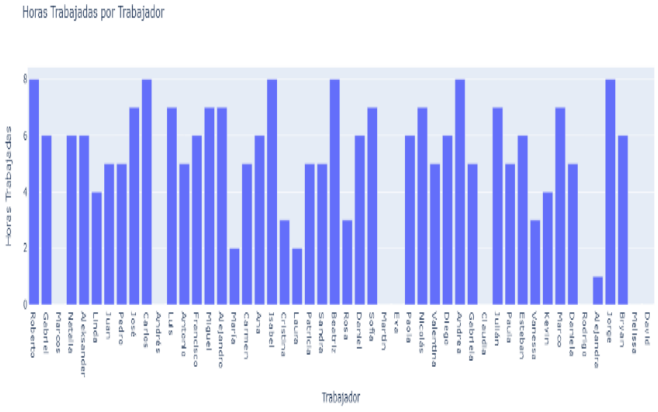

Fig 15: horas trabajas por trabajador

4. Horas trabajadas por día

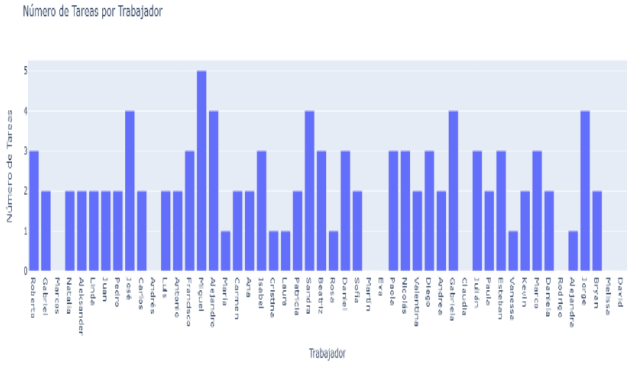

Fig 16: Horas trabajadas por dia

5. Número de tareas por habilidad

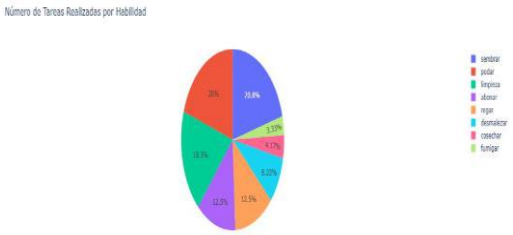

Fig 17: numero de tareas por habilidad

6. Número de horas por prioridad de tarea

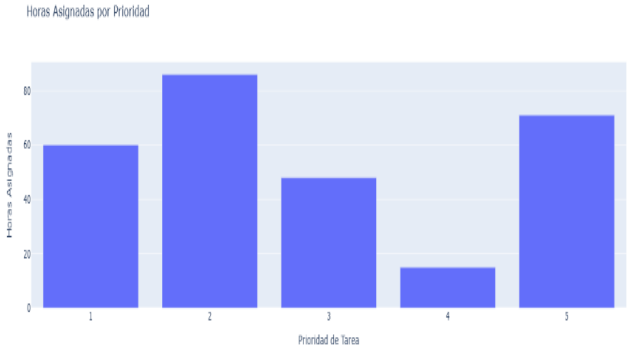

Fig 18: Tareas por prioridad

7. Frecuencia de tareas no asignadas

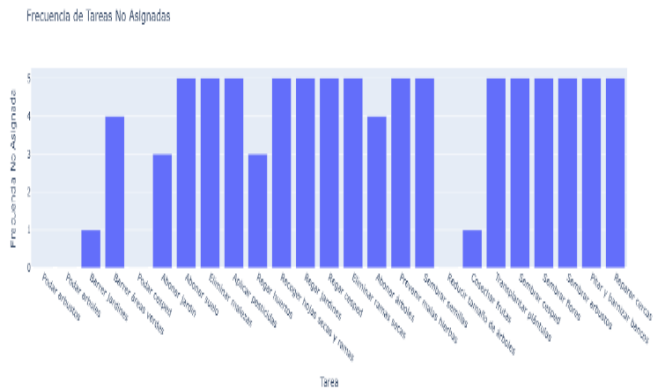

Fig 19: frecuencia de tareas no asignadas

V. CONCLUSIONES

- La implementación de la Automatización Robótica de Procesos (RPA) en la asignación de tareas de jardinería ha demostrado ser una estrategia altamente efectiva para incrementar la eficiencia operativa, reducir la incidencia de

errores y optimizar la utilización de los recursos humanos disponibles. Un factor determinante en esta mejora es el uso de Indicadores Clave de Desempeño (KPIs), los cuales permiten medir el rendimiento del proceso automatizado de manera objetiva, identificar puntos críticos y orientar la mejora continua. La creación de dashboards específicos para la visualización de estos KPIs facilita la supervisión en tiempo real del sistema RPA, al proporcionar representaciones claras y comprensibles de los datos, que permiten a los responsables de la toma de decisiones detectar patrones, anticiparse a problemas y aplicar soluciones correctivas de manera ágil y precisa. En conjunto, la integración de RPA con dashboards optimiza la precisión en la asignación de tareas, disminuye significativamente los tiempos de respuesta y maximiza el uso de los recursos disponibles, consolidándose como una herramienta estratégica para la gestión eficiente de procesos operativos complejos y para la mejora continua de la calidad del servicio en el ámbito de la jardinería.

REFERENCAS

[1] Plantillas de manuscritos para actas de conferencias, IEEE. http://www.ieee.org/conferences_events/conferences/publishing/templates.html.

[2] M. King, B. Zhu y S. Tang, “Planificación óptima de caminos”, Mobile Robots, vol. 8, no. 2, pp. 520-531, marzo de 2001.

[3] H. Simpson, Robots tontos, 3.ª ed., Springfield: UOS Press, 2004, pp. 6-9.

[4] M. King y B. Zhu, “Estrategias de juego”, en Planificación de Caminos hacia el Oeste, vol. II, S. Tang y M. King, Eds. Xi'an: Jiacda Press, 1998, pp. 158-176.

[5] B. Simpson, et al., “Título del artículo si se conoce aquí”, inédito.

[6] J.-G. Lu, “Título del artículo con solo la primera palabra en mayúscula”, J. Name Stand. Abbrev., en prensa.

[7] Y. Yorozu, M. Hirano, K. Oka y Y. Tagawa, “Estudios de espectroscopia electrónica en medios magnético-ópticos y en interfaces de sustratos plásticos”, IEEE Translated J. Magn. Japan, vol. 2, pp. 740-741, agosto de 1987 [Digest 9.º Simposio Anual de Conf. Magnética de Japón, 1982].

[8] M. Young, El manual del escritor técnico, Mill Valley, CA: University Science, 1989.