

Ambiente de aprendizaje para la enseñanza de Estadística II: un enfoque en teoría, prácticas y
autoevaluación

Jorge Eduardo Suárez Cortés

Daniel Alejandro Sánchez Rodríguez

Trabajo de Grado para optar al título de Ingeniero de Sistemas

Director

Andrés Leonardo González Gómez

PhD (c). Ciencias de la Computación

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2025

Tabla de Contenido

Introducción	12
1. Objetivos	13
1.1. Objetivo General	13
1.2. Objetivos Especificos	13
2. Marco Teorico	15
2.1. Ambientes de aprendizaje	15
2.1.1. Concepto de ambiente de aprendizaje	15
2.1.2. Ambientes virtuales de aprendizaje (Moodle como LMS y su importancia en educación superior)	16
2.2. Tecnologías implementadas en el proyecto	17
2.2.1. Node.js y Express.js	17
2.2.2. React.js	17
2.2.3. Bases de datos relacionales (MariaDB)	18
2.2.4. Virtualización y contenedores (Docker)	18
2.2.5. Moodle como LMS	19
2.2.6. Python y R	19
2.2.7. Google Colab	19
2.2.8. Graders y autoevaluación automatizada	20
2.2.9. Playit.gg y tunelización de servicios	20

3. Estado del Arte	21
3.1. Antecedentes	21
3.2. Desarrollo Actual	22
4. Marco de Referencia	24
4.1. Automatización Robótica de Procesos (RPA)	24
4.2. Plataformas de Desarrollo de Bajo Código (LCDP's)	28
4.3. Tratamiento de datos (Data Wrangling)	32
4.4. Seguridad	33
4.4.1. Encriptación de Datos Sensibles	34
4.4.2. Conexiones Seguras a Bases de Datos	34
4.4.3. Seguridad en la plataforma Microsoft Power Platform	35
5. Metodología	37
5.1. Script de Python	37
5.2. Automatización Robótica de Procesos	39
5.3. Interfaz Gráfica de Usuario	41
5.4. Implementación de seguridad	41
5.4.1. Roles y Permisos en Microsoft Power Platform	42
5.4.1.1. Administrador de entorno	42
5.4.1.2. Creador de aplicaciones	42
5.4.1.3. Usuario básico	43
5.4.2. Conexión de Servicios	43
5.4.3. Encriptación de credenciales	44

AMBIENTE DE APRENDIZAJE PARA LA ENSEÑANZA DE ESTADÍSTICA II	4
5.4.3.1. Generación de clave	44
5.4.3.2. Almacenamiento seguro	45
5.4.3.3. Encriptación de credenciales	45
5.4.3.4. Desencriptación	45
5.4.4. Conexión entre máquinas	45
5.5. Pruebas y Conclusiones	46
6. Desarrollo	48
6.1. Paquete Python	48
6.2. Automatización Robótica de Procesos	51
6.3. Interfaz Gráfica de Usuario	52
6.4. Implementación de seguridad	56
6.5. Despliegue y Estabilización	57
6.6. Pruebas	61
7. Conclusiones	63
7.1. Trabajo Futuro	65
Referencias Bibliográficas	66

Lista de Figuras

Figura 1.	Arquitectura LUISA.	28
Figura 2.	Estructura LUISA.	29
Figura 3.	Plataforma virtual Universidad McGill.	31
Figura 4.	Plataforma virtual Universidad McGill.	32
Figura 5.	Modo Thin.	38
Figura 6.	Modo Thick.	38
Figura 7.	DataFrame Pandas.	39
Figura 8.	Ejecución Script con Power Automate	40
Figura 9.	Ejemplo Form Power Apps	41
Figura 10.	Botón que ejecuta un flujo	42
Figura 11.	Gestión de entornos en Power Platform	43
Figura 12.	Cifrado de datos en Power Platform	44
Figura 13.	Encriptación Simétrica	46

Figura 14.	Arquitectura	48
Figura 15.	Gráfico de la estructura del Paquete Python.	51
Figura 16.	Pantalla inicial GUI. Form CGN.	52
Figura 17.	Acción Botón "Digitar Parámetros". Form CGN.	53
Figura 18.	Pantalla de diligenciamiento de parámetros Form CGN.	53
Figura 19.	Acción del botón de confirmar ejecución. Form CGN.	54
Figura 20.	Pantalla emergente de re confirmación de generación. Form CGN.	54
Figura 21.	Acción del botón Generar Reporte. Form CGN.	55
Figura 22.	Flujo desencadenado por el botón "Generar Reporte". Form CGN.	56
Figura 23.	Acción del botón Envío a Entidades. Form CGN.	57
Figura 24.	Flujo desencadenado por el botón "Envío a entidades". Form CGN.	57
Figura 25.	Componentes de la solución. RPA_ReporteCGN.	58
Figura 26.	Exportación de la solución. RPA_ReporteCGN.	58
Figura 27.	Paquete de la solución descargado. RPA_ReporteCGN.	59
Figura 28.	Importación de la solución. RPA_ReporteCGN.	59

Figura 29.	Solución en producción. RPA_ReporteCGN.	60
Figura 30.	Pruebas flujo de nube que desencadena la solución	61
Figura 31.	Pruebas flujo de nube hospedado en cuenta técnica	61
Figura 32.	Pruebas flujo de escritorio	62
Figura 33.	Verificación reportes ejecutados	62
Figura 34.	Verificación reporte resultante	63
Figura 35.	Correo con detalle enviado.	64

Lista de Tablas

Glosario

Automatización: Uso de tecnologías para realizar tareas o procesos sin intervención humana, con el objetivo de aumentar la eficiencia y reducir errores manuales.

Microsoft Power Platform: Conjunto de herramientas de Microsoft, que incluye Power Automate, Power Apps y Power BI, que permite la creación de soluciones empresariales con poco o ningún código.

Python: Lenguaje de programación utilizado para crear scripts que permiten la conexión, extracción y tratamiento de información de bases de datos.

RPA (Automatización Robótica de Procesos): Tecnología que permite automatizar tareas repetitivas utilizando robots de software.

SAC (Sistema de Administración Comercial): Sistema utilizado para la administración de la información comercial en las empresas.

JD Edwards: Sistema de planificación de recursos empresariales (ERP) que gestiona finanzas, proyectos y otros procesos comerciales.

GUI (Interfaz Gráfica de Usuario): Interfaz que permite la interacción del usuario con la solución de software de manera visual e intuitiva, facilitando la recopilación y ejecución de parámetros.

Resumen

Título: Ambiente de aprendizaje para la enseñanza de Estadística II: un enfoque en teoría, prácticas y autoevaluación.

*

Autores: Jorge Eduardo Suárez Cortés y Daniel Alejandro Sánchez Rodríguez. **

Palabras Clave:

Descripción:

* Trabajo de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Andrés Leonardo Gonzáles Gómez

Abstract

Title: Learning Environment for the Teaching of Statistics II: A Focus on Theory, Practice, and Self-Assessment. *

Authors: Jorge Eduardo Suárez Cortés y Daniel Alejandro Sánchez Rodríguez. **

Key words:

Description:

* Degree Work

** Faculty of Physicomechanical Engineering. School of Systems and Computer Engineering.
Director: Andrés Leonardo Gonzáles Gómez

Introducción

Este trabajo presenta una propuesta para solventar este problema, que comprende la automatización de la generación de reportes contables, cuya arquitectura integra herramientas de Microsoft Power Platform y Python. Según el trabajo de (Narayn, 2023a) donde propone el uso de herramientas de Microsoft Power Platform, donde su integración genera un gran valor añadido a sus capacidades (Narayn, 2023a). Microsoft Power Automate ofrece gran facilidad al orquestar flujos de trabajo automatizados (Vartiainen, 2024a), el uso de Scripts de Python capaces de hacer conexión a los diferentes orígenes de datos así como su tratamiento (Rattenbury et al., 2017), y Microsoft Power Apps para la mejora en la interacción y control del usuario sobre la solución (Simon, 2022). Se explora la implementación técnica de esta solución, evaluando el impacto de la integración de estas tecnologías por medio del desarrollo de servicios, partiendo de la flexibilidad que ofrece Python a la hora de tratar datos, en la eficiencia operativa de la gestión contable. (R Core Team, 2020)

1. Objetivos

1.1. Objetivo General

- Diseñar y desarrollar un entorno virtual para la asignatura de Estadística II, de la Escuela de Ingeniería de Sistemas e Informática de la UIS, que facilite la ejecución de un plan de aula propuesto para la práctica, el aprendizaje y la autoevaluación del contenido de la asignatura, mediante herramientas de Python y R.

1.2. Objetivos Específicos

- Diseñar un plan de aula modular para la asignatura Estadística II que organice los contenidos teóricos y prácticos en unidades interactivas basadas en Colab Notebooks, Python y R, e incluya calificadores automáticos y ejercicios con retroalimentación instantánea.
- Implementar un ambiente de aprendizaje virtual en la plataforma Moodle que integre las tecnologías definidas (Colab Notebooks, Python y R) y organice los contenidos teóricos y prácticos junto con materiales didácticos de apoyo, consolidando una experiencia formativa práctica y accesible para la asignatura Estadística II.
- Establecer instrumentos de medición para las actividades prácticas y autoevaluativas en Moodle (desarrolladas con Python y R), que permitan medir el logro de los resultados de aprendizaje durante el desarrollo del proyecto en el curso de Estadística II.
- Pilotear el ambiente virtual de aprendizaje diseñado, con sus contenidos interactivos e instrumentos de evaluación, en por lo menos dos grupos de Estadística II de la Escuela de Inge-

nería de Sistemas e Informática de la UIS, con el fin de validar su usabilidad, satisfacción y efectividad en el logro de los resultados de aprendizaje.

2. Marco Teorico

2.1. Ambientes de aprendizaje

2.1.1. Concepto de ambiente de aprendizaje

El concepto de ambiente de aprendizaje se refiere al conjunto de condiciones, espacios, interacciones y recursos que permiten y favorecen el desarrollo de procesos educativos efectivos. Este término alude a un escenario dinámico en el que los individuos desarrollan capacidades, competencias, habilidades y valores. Es decir, no se trata de un espacio fijo ni neutral, sino de un entorno que debe transformarse en función de las necesidades de los estudiantes y de las innovaciones educativas (Castro, 2019).

En este sentido, un ambiente de aprendizaje no se limita al aula física, sino que constituye una construcción pedagógica en la que intervienen aspectos sociales, culturales, tecnológicos y metodológicos. De acuerdo con Castro (2019), un ambiente de aprendizaje debe ser:

- Flexible y adaptable a diferentes contextos y tecnologías.
- Fomentar la interacción social y la construcción colectiva del conocimiento.
- Proporcionar recursos adecuados, incluidos los tecnológicos, que potencien las competencias estudiantiles.
- Promover el rol activo del docente como diseñador, facilitador y mediador del aprendizaje.

Desde esta perspectiva, los ambientes de aprendizaje se conciben como espacios dinámicos

e integrales, que van más allá de la transmisión de información y buscan estimular la participación activa, la autonomía y la construcción colaborativa del conocimiento.

2.1.2. Ambientes virtuales de aprendizaje (Moodle como LMS y su importancia en educación superior)

El avance de las tecnologías digitales ha posibilitado la transición de los entornos tradicionales hacia los Ambientes Virtuales de Aprendizaje (AVA). Estos se definen como plataformas tecnológicas que integran recursos didácticos, actividades de aprendizaje, espacios de interacción y mecanismos de evaluación para facilitar procesos formativos en entornos digitales (Salinas, 2004).

En coherencia con las características planteadas por Castro (2019), los AVA son flexibles y adaptables a diferentes contextos, fomentan la interacción social y colaborativa, proporcionan recursos tecnológicos que amplían las competencias estudiantiles y posicionan al docente como un mediador y facilitador del aprendizaje.

Uno de los sistemas más representativos en este campo es Moodle, un Learning Management System (LMS) de código abierto que ha sido adoptado globalmente en educación superior debido a su enfoque pedagógico constructivista, su escalabilidad y su facilidad de integración con otras herramientas (Dougiamas & Taylor, 2003). Moodle permite gestionar cursos, estudiantes, actividades y evaluaciones, convirtiéndose en un eje articulador del proceso formativo.

En el contexto del presente proyecto, Moodle constituye el espacio institucional de referencia en la UIS, donde los estudiantes acceden a materiales de teoría, actividades interactivas y prácticas en Google Colab, además de las autoevaluaciones conectadas con el backend. Su im-

portancia radica en que integra de manera armónica la teoría, la práctica y la retroalimentación automática, consolidando un entorno virtual de aprendizaje efectivo.

2.2. Tecnologías implementadas en el proyecto

2.2.1. *Node.js* y *Express.js*

Node.js es un entorno de ejecución basado en el motor V8 de Google Chrome que permite ejecutar JavaScript en el lado del servidor, diseñado bajo un modelo de I/O no bloqueante. Esto lo convierte en una herramienta adecuada para aplicaciones que requieren manejar múltiples solicitudes concurrentes, como plataformas de aprendizaje en línea (Tilkov & Vinoski, 2010).

Express.js es un framework minimalista para Node.js que facilita la construcción de APIs y aplicaciones web mediante la organización de rutas, middleware y controladores (Brown, 2019). En el proyecto, Node.js y Express.js cumplen el rol de backend principal, gestionando la validación de respuestas de estudiantes, el control de intentos y la comunicación con la base de datos MariaDB.

2.2.2. *React.js*

React.js es una biblioteca de JavaScript desarrollada por Facebook que permite construir interfaces de usuario basadas en componentes reutilizables (Banks & Porcello, 2017). Su arquitectura con virtual DOM optimiza la actualización de vistas, mejorando la eficiencia y la experiencia de usuario.

El frontend, desarrollado en React.js, centraliza la gestión de calificaciones y el control de intentos de los estudiantes. Su principal ventaja es la transparencia: ante cualquier discrepancia con

la calificación calculada por el backend (Node.js), un instructor puede revisar tanto el código enviado por el estudiante como la nota obtenida. Esto facilita una verificación rápida y fundamentada, garantizando la equidad en la evaluación.

2.2.3. Bases de datos relacionales (MariaDB)

MariaDB es un sistema de gestión de bases de datos relacional de código abierto, derivado de MySQL, que organiza los datos en tablas relacionadas bajo el modelo relacional propuesto por Codd (1970).

Su función en el proyecto es ser el repositorio central de información, donde se almacenan los registros de estudiantes, intentos de ejercicios, calificaciones y respuestas. Gracias a sus características de integridad y consistencia, MariaDB asegura la fiabilidad en el almacenamiento de los resultados generados en los procesos de autoevaluación.

2.2.4. Virtualización y contenedores (Docker)

Docker es una plataforma de contenedores ligeros que empaqueta aplicaciones y dependencias en un entorno aislado, lo que garantiza portabilidad y reproducibilidad (Merkel, 2014).

En el proyecto, Docker permite desplegar servicios como el backend, frontend, playit.gg y la base de datos en un entorno controlado, asegurando que las pruebas piloto y los despliegues sean consistentes sin importar el sistema operativo. Su uso facilita la escalabilidad y reduce los problemas de compatibilidad entre entornos de desarrollo y producción.

2.2.5. Moodle como LMS

Moodle es un sistema de gestión de aprendizaje (LMS) de código abierto ampliamente utilizado en la educación superior. Ofrece herramientas para gestionar cursos, usuarios, recursos y actividades de evaluación (Dougiamas & Taylor, 2003).

En este proyecto, Moodle funciona como el entorno de acceso institucional provisto por la UIS, donde los estudiantes encuentran los enlaces a los notebooks de Google Colab, las actividades prácticas y los recursos teóricos. Moodle sirve como el punto de integración pedagógica que articula los recursos tecnológicos desarrollados.

2.2.6. Python y R

Python y R son lenguajes de programación consolidados en el campo del análisis estadístico y la ciencia de datos. Python, gracias a bibliotecas como NumPy, SciPy y Pandas, se ha convertido en una herramienta versátil para la enseñanza y aplicación de métodos estadísticos (McKinney, 2017). R, por su parte, es un lenguaje especializado en estadística y visualización de datos, ampliamente usado en contextos académicos (R Core Team, 2023).

En el proyecto, para el desarrollo de los graders se emplearán ambos lenguajes de programación, lo que permitirá dotarlos de una mayor versatilidad en su uso.

2.2.7. Google Colab

Google Colab es una plataforma en la nube que permite ejecutar notebooks de Python sin necesidad de instalar software localmente. Combina teoría, código ejecutable y resultados en un

único entorno interactivo (Bisong, 2019).

Su integración en el proyecto permite a los estudiantes resolver ejercicios en tiempo real, ejecutar simulaciones y enviar sus respuestas al backend para validación. Además, Colab facilita el aprendizaje activo y autónomo.

2.2.8. Graders y autoevaluación automatizada

Los graders son sistemas de evaluación automática que comparan respuestas de estudiantes con soluciones esperadas, otorgando calificaciones y retroalimentación inmediata (Kurnia et al., 2001). Herramientas como nbgrader o Otter-Grader han demostrado su efectividad en la enseñanza de programación y estadística.

En este proyecto, los graders están implementados en el backend desarrollado con Node.js y conectados a los notebooks de Colab. De esta forma, los estudiantes reciben retroalimentación instantánea, lo que fomenta el aprendizaje autónomo y reduce la carga de corrección manual para los docentes.

2.2.9. Playit.gg y tunelización de servicios

Playit.gg es una herramienta que permite crear túneles seguros entre una máquina local y la web pública, facilitando la exposición de servicios locales sin necesidad de configuración compleja de red.

Durante el desarrollo y pruebas piloto del proyecto, Playit.gg se utiliza para exponer el backend y el frontend alojados en contenedores Docker hacia los estudiantes y notebooks de Colab. Esta estrategia permite realizar pilotos de manera controlada.

3. Estado del Arte

3.1. Antecedentes

La Automatización Robótica de Procesos (RPA, por sus siglas en inglés) ha ganado popularidad en diversas industrias debido a su capacidad para automatizar tareas repetitivas y basadas en reglas, incrementando la eficiencia operativa y reduciendo costos. En el trabajo de (Cooper et al., 2019), se destaca cómo las firmas de contabilidad han adoptado RPA para automatizar la generación de reportes financieros y otras tareas rutinarias, mejorando la precisión y reduciendo los errores humanos. Este tipo de tecnologías permite a las empresas liberar a los empleados de tareas manuales para que puedan enfocarse en actividades de mayor valor añadido.

Un aspecto clave de la RPA es su capacidad para integrarse sin necesidad de modificar los sistemas subyacentes (Aalst et al., 2018), lo que facilita su implementación en una amplia variedad de entornos empresariales. Sin embargo, como mencionan (Syed et al., 2020), los desafíos actuales incluyen la necesidad de integrar capacidades cognitivas, como el aprendizaje profundo, para mejorar la adaptabilidad de los bots en entornos cambiantes. En este sentido, las plataformas de RPA han comenzado a evolucionar hacia la Automatización de Procesos Inteligentes (IPA, por sus siglas en inglés), que combina RPA con Inteligencia Artificial (IA) y Machine Learning (ML), ofreciendo soluciones más robustas (A. Gami y Singh, 2019).

3.2. Desarrollo Actual

La plataforma Microsoft Power Automate es una herramienta clave en la automatización de procesos de negocios, especialmente en entornos corporativos que ya utilizan la suite de Microsoft 365. Su capacidad para automatizar flujos de trabajo a través de aplicaciones como SharePoint, OneDrive y Outlook permite a las organizaciones integrar datos de múltiples fuentes y realizar tareas complejas sin intervención humana (Microsoft, 2024g). Además, Power Automate ofrece dos tipos principales de automatización: asistida y no asistida. En la automatización asistida, los bots replican acciones humanas en tiempo real, mientras que la automatización no asistida ejecuta tareas programadas sin supervisión (Xerox, 2023). Esto es particularmente útil para la automatización de procesos contables que requieren la generación y envío de reportes de manera recurrente.

Python, por su parte, ha sido ampliamente utilizado para la manipulación y tratamiento de datos en procesos de automatización. Herramientas como Pandas y NumPy permiten limpiar, transformar y analizar grandes volúmenes de datos, lo que resulta esencial para asegurar la calidad de los reportes generados automáticamente (Rattenbury et al., 2017). Python también ofrece librerías especializadas para la automatización de tareas de escritorio, como 'pywinauto' y 'pyautogui', las cuales facilitan la interacción automatizada con aplicaciones en sistemas Windows (Team, 2024). Estas bibliotecas permiten combinar el poder de la automatización robótica con las capacidades de procesamiento de datos de Python, ofreciendo una solución integral para la automatización de procesos contables.

En empresas como Xerox, Power Automate se ha utilizado para generar reportes de facturación automatizados, eliminando la necesidad de generación manual y reduciendo significativamente los

tiempos de procesamiento (Xerox, 2023). La integración de estas herramientas con tecnologías de Microsoft como Azure Key Vault y SharePoint asegura la seguridad y confiabilidad de los procesos, al utilizar encriptación de 256 bits para la protección de credenciales y datos sensibles.

En resumen, la Automatización Robótica de Procesos y las plataformas de bajo código como Microsoft Power Automate, combinadas con el poder de Python para el tratamiento de datos, representan una solución robusta y eficaz para la optimización de procesos contables. Estas tecnologías no solo reducen el esfuerzo humano y el riesgo de errores, sino que también mejoran la eficiencia operativa y facilitan el cumplimiento de normativas mediante la automatización de tareas repetitivas y la generación de reportes automáticos.

4. Marco de Referencia

La optimización de procesos es crucial para la competitividad de las organizaciones en el mercado, ya que busca minimizar costos y maximizar el rendimiento, la productividad y la eficiencia. Con el avance de la revolución tecnológica, las empresas han comenzado a implementar la transformación digital para optimizar los procesos en la cadena de valor. Este proyecto está enfocado en el desarrollo de soluciones digitales con el uso de tecnologías de Automatización Robótica de Procesos (RPA), Plataformas de Desarrollo de Bajo Código (LCDPs) y Análisis de Datos (Data Analysis).

4.1. Automatización Robótica de Procesos (RPA)

La Automatización Robótica de Procesos (RPA) en los últimos años se ha posicionado como una herramienta esencial para la optimización de proceso, debido a paradigmas de automatización de tareas repetitivas en diferentes industrias (Doguc, 2020). Su arquitectura es un software que imita las acciones humanas al usar interfaces de usuario de diferentes aplicaciones, permitiendo así una integración sin modificar sistemas subyacentes. En el trabajo de (Doguc, 2020) se describe a RPA como una tecnología clave que reduce costos y mejora la eficiencia operativa al automatizar tareas basadas en reglas que anteriormente realizaban los humanos. Esta tecnología no solo aumenta la precisión, sino que también tiene la gran capacidad de liberar a los empleados de tareas manuales, para que se centren en tareas más estratégicas y de mayor valor añadido. Sin embargo, en lo propuesto por (Syed et al., 2020) se analizan los temas contemporáneos y desafíos a los que se enfrenta RPA como nueva tecnología en auge, destacando así la integración de capacidades cognitivas y de aprendizaje profundo. Estos avances no solo permiten a los “bots” ir más allá de la

ejecución de tareas repetitivas, sino también la capacidad de adaptarse al entorno cambiante. Por otro lado, menciona la relevancia de la gobernanza centralizada y la preservación del conocimiento del proceso para asegurar la efectividad de RPA.

Según (Aalst et al., 2018) el enfoque “de afuera hacia adentro” de RPA lo diferencia de otros métodos de automatización. Este enfoque permite que los sistemas de información existentes permanezcan en el tiempo sin cambios, de la misma manera que los “bots” asumen las tareas humanas. Esto contrasta con enfoques tradicionales como el Procesamiento Directo (STP), que requerían modificaciones en los sistemas subyacentes. En su investigación (Madakam et al., 2019) exploran cómo RPA está transformando la fuerza laboral digital del futuro. La automatización de tareas rutinarias permite que los empleados inviertan su tiempo laboral en actividades que aporten mayor valor añadido a las empresas. Además de esto, se discuten los impactos en la estructura organizacional y la necesidad de gestionar el cambio para asegurar la implementación exitosa de estas tecnologías.

Finalmente, (M. Gami et al., 2019) ofrecen una revisión exhaustiva del estado del arte de la implementación de la tecnología RPA y su impacto en las organizaciones empresariales. Destacan los beneficios de su implementación como la reducción de costos y la mejora en la calidad del trabajo, también analizan el encaminamiento de estas tecnologías con dirección a la Automatización de Procesos Inteligentes (IPA), que combina RPA con tecnologías avanzadas como la Inteligencia Artificial (IA) y el Aprendizaje Automático para ofrecer soluciones más robustas y eficaces.

La adopción de la Automatización Robótica de Procesos representa una evolución crucial en la manera en que las organizaciones optimizan sus procesos, en aras de la reducción de costo y la

mejora constante en la calidad del trabajo. Entre las herramientas de RPA, Microsoft Power Automate se destaca por su capacidad para automatizar flujos de trabajo entre aplicaciones y servicios, permitiendo a las empresas aumentar su eficiencia y productividad mediante la simplificación de tareas diarias.

El software de automatización Microsoft Power Automate es un servicio de automatización basado en la nube y un conjunto de herramientas de desarrollo de bajo código, diseñado para facilitar la creación de flujos de trabajo automatizados, flujos que a su vez automatizan procesos empresariales. Esta herramienta de bajo código que hace parte de la suite de Microsoft Power Platform, tiene como objetivo optimizar las operaciones empresariales al reducir las herramientas manuales y aumenta la eficiencia. Según (Najjar, 2023) Power Automate facilita la programación autodidacta para usuarios empresariales y no desarrolladores, permitiéndoles construir automatizaciones funcionales si necesidad de conocimientos avanzados de programación. La herramienta soporta varios tipos de automatización, incluyendo flujos en la nube, flujos de escritorio y flujos de procesos de negocio, cada uno adaptado a diferentes necesidades de automatización. Añadido a esto, Power Automate cuenta con más de 400 conectores que permiten integrar servicios como SharePoint, Onedrive, Outlook y Teams, lo que maximiza su utilidad para las organizaciones que ya disponen de herramientas ofrecidas por Microsoft 365.

En este contexto, (Vartiainen, 2024b) enfatiza cómo Power Automate puede mejorar la experiencia de los empleados mediante RPA. Afirma que la plataforma facilita la automatización de flujos de trabajo completos, integrando datos de diversas fuentes y mejorando la productividad y eficiencia operativa. No solo es útil para automatizar tareas rutinarias, puesto que también es una herramienta poderosa para la integración de datos que mejora la toma de decisiones y la gestión de procesos em-

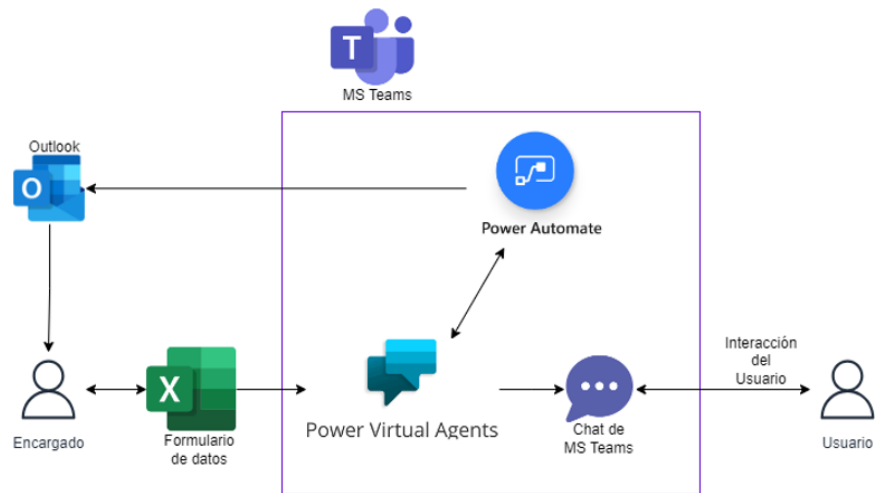
presariales. Esta herramienta con su capacidad para manejar flujos en la nube y flujos de escritorio puede adaptarse a una amplia gama de escenarios empresariales, desde la automatización de procesos simples hasta la gestión de flujos de trabajo complejos que involucren múltiples aplicaciones y servicios.

El asistente virtual interno LUISA de la ESSA, nace de la necesidad de agilizar el acceso a información clave por parte de los trabajadores. Tiene el objetivo de guiar a la Gente ESSA en la resolución de dudas sobre cesantías, vacaciones, reemplazos, pensiones, nómina, permisos, seguridad social y beneficios convencionales. De esta manera mejorar la eficiencia y productividad de los empleados al simplificar la obtención de datos relevantes proporcionando una herramienta integral para la gestión eficaz de temas laborales y prestaciones en todos los niveles y departamentos de la organización. El asistente virtual LUISA está implementado en Microsoft Teams, el cual es un medio de comunicación interno de la empresa. Este asistente virtual fue diseñado con la integración de tecnologías como Microsoft Power Automate, Microsoft Copilot Studio, Microsoft Teams, Microsoft Outlook y Microsoft Excel. (ver Figura 1)

La infraestructura de LUISA usa Microsoft Copilot Studio como plataforma principal para diseñar las conversaciones, definir flujos de trabajo y configurar respuestas. Microsoft Teams actúa como el canal de interacción principal, brindando a los usuarios un espacio familiar para acceder a las capacidades del asistente virtual de manera eficiente. En la integración con Microsoft Power Automate usa palabras desencadenadoras, las cuales al ser accionadas accionan un flujo creado previamente el cual tomará una serie de respuestas o acciones de envío de correos para dar soluciones al usuario. La estructura de los temas hallados en LUISA (ver Figura 2). está ramificado según el sindicato, pues el tipo de solicitud y sus respuestas son diferentes en cada una.

Figura 1.

Arquitectura LUISA.



Nota: Arquitectura de Luisa, RPA exitoso en la ESSA. León, J. (2024). Diseño propio.

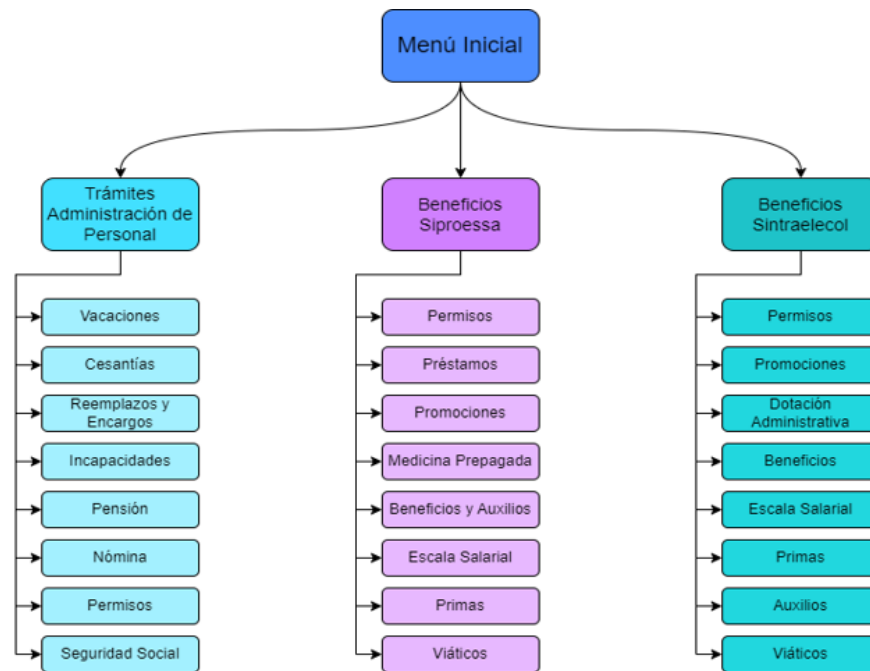
4.2. Plataformas de Desarrollo de Bajo Código (LCDP's)

Las plataformas de desarrollo de bajo código (LCDPs) han emergido como una solución innovadora para enfrentar la creciente demanda de aplicaciones. En el trabajo de (Alsaadi et al., 2021) se indica que las LCDPs permiten a las organizaciones crear aplicaciones rápida y fácilmente por medio de interfaces visuales y componente preconstruidos, disminuyendo significativamente la necesidad de codificación manual. Este enfoque libera el desarrollo de software, pues permite a empleados participar en desarrollos de aplicaciones sin tener un estudio técnico avanzado previo, a ellos se les conoce como desarrolladores ciudadanos. En consecuencia, (Prinz et al., 2021) destacan el aumento en la relevancia de las LCDP y cómo están transformando las operaciones y la estructura organizativa al agilizar el desarrollo, consecuente a esto la reducción de los costos asociados a la contratación y formación de personal en el cargo de desarrolladores especializados.

Las LCDPs ofrecen varios beneficios importantes para las empresas, (Sharma y Arora, 2023) indi-

Figura 2.

Estructura LUISA.



Nota: Estructura de Luisa, RPA exitoso en la ESSA. León, J. (2024). Diseño propio.

can que entre estos beneficios se encuentran la rapidez y eficiencia en el desarrollo, lo que permite desplegar aplicaciones en tiempos verdaderamente despreciables comparándolos con métodos que se realizaban anteriormente. Además, la facilidad de integración con diferentes orígenes de datos y sistemas, mejorando así la cooperación operativa en las organizaciones. No obstante, Käss, Strahring y Westner et al. (2023) afirman que la accesibilidad y facilidad de uso de estas plataformas permiten a los desarrolladores ciudadanos crear soluciones personalizadas que atienden necesidades específicas del negocio, promoviendo una mayor innovación interna. Sin embargo, la adopción de LCDP también enfrenta desafíos.

Finalmente, (Alsaadi et al., 2021) mencionan la necesidad establecer estructuras claras de gobernanza y medidas de seguridad robustas para así garantizas en gran parte la calidad y protección de

los datos.

En conclusión, las plataformas de desarrollo de bajo código están reestructurando el entorno del desarrollo de software empresarial, permitiendo a las organizaciones generar más agilidad e innovación en su cadena de valor. El aumento de la inclusión de estas plataformas en las empresas trae consigo la necesidad de implementar procesos de gobernanza y seguridad en aras de la maximización de su potencial. Una de las herramientas más relevantes en este contexto es Microsoft Power Apps, pues ofrece una robusta plataforma de bajo código, lo cual permite a los usuarios empresariales no solo desarrollar aplicaciones rápidamente, sino también integrar soluciones en su ecosistema.

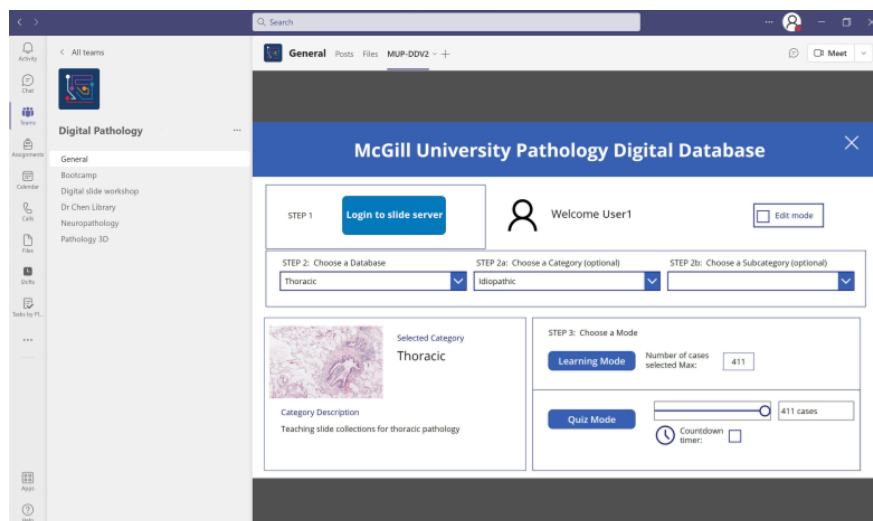
La herramienta Microsoft Power Apps no deja de revolucionar cada vez más el desarrollo de aplicaciones por medio de su plataforma de bajo código, la cual permite a los usuarios crear aplicaciones empresariales sin la necesidad de conocimientos avanzados en programación por parte de los empleados. Según (Pandit, 2024), Power Apps destaca en el contexto de LCDPs por su interfaz intuitiva la cual se trata de arrastrar y soltar, sus conectores preconstruidos y plantilla, y su gran capacidad de integración con diversas fuentes de datos. Esta plataforma no solo facilita a los usuarios la creación rápida de aplicaciones personalizadas, sino que también se destaca por fomentar la innovación tecnológica dentro de las organizaciones. En este sentido (Narayn, 2023b) destaca la importancia de Power Apps en el entorno del trabajo moderno, integrándolo con herramientas como SharePoint Online, Power Automate y Microsoft Teams para con ello ofrecer soluciones completas de colaboración y gestión de contenido. Por otro lado, (Di Ruscio et al., 2022) subraya cómo esta plataforma, en combinación con la ingeniería dirigida de modelos (MDE), proporciona un enfoque eficiente y flexible para el desarrollo de software, abordando tanto los aspectos técnicos

como organizativos de la creación de aplicaciones.

La Universidad McGill es un notable caso de éxito en su implementación de Microsoft Power Apps y Microsoft Teams, para el desarrollo de una plataforma educativa virtual de la disciplina patología. Este caso ha sido descrito por (Rajaram et al., 2022). El departamento de Patología Universidad McGill en respuesta a la pandemia de COVID-19, implementó Microsoft Power Apps y Microsoft Teams (ver Figura 3) para continuar la educación de los estudiantes de manera remota. Esta plataforma personalizada permitió a los residentes acceder a imágenes digitalizadas de diapositivas completas, participar en sesiones educativas y realizar exámenes a distancia (ver Figura 4).

Figura 3.

Plataforma virtual Universidad McGill.



Nota: Visión de la aplicación Patología Digital. Rajaram et al., 2022. Adaptación.

La integración con Microsoft Teams facilitó la organización y el acceso a todos los recursos educativos, mejorando así significativamente la continuidad y calidad de la educación durante esta etapa. Esta implementación demuestra el éxito de Power Apps, y cómo puede ser usado para desarrollar

soluciones educativas innovadoras y eficiente, incluso fuera de este contexto. La gran capacidad de adaptación a las necesidades cambiantes y con ello asegurando la continuidad educativa en tiempos de crisis.

Figura 4.

Plataforma virtual Universidad McGill.



Nota: Aplicación de Patología digital que muestra: a) Lista de imágenes de portaobjetos completos con sus diagnósticos que se pueden lanzar. b) Modo de cuestionario dentro de la aplicación. c) Visualización de las respuestas evaluadas por los usuarios. Rajaram et al., 2022. Adaptación.

4.3. Tratamiento de datos (Data Wrangling)

El proceso de limpieza al que se someten los datos, conocido como Data Wrangling, es parte importante en el ciclo de vida de los datos, puesto que este proceso asegura que los conjuntos de datos sean precisos, consistentes y compactos, esto los vuelve utilizables. Según (Rattenbury et al., 2017) la identificación y corrección de errores, la eliminación de duplicados y el tratamiento de valores atípicos y faltantes, es lo que implica este proceso. No obstante, (Kazil y Jarmul, 2016) aseguran que la implementación efectiva permite que los análisis se basen en datos confiables, lo cual es crucial para obtener resultados significativos en su utilización. Las técnicas más utilizadas en el Data Wrangling son la atribución de valores faltantes, la normalización de datos numéri-

cos, y la codificación de variables categóricas. Estas técnicas ayudan a mejorar la calidad de los datos, asegurando decisiones empresariales basadas en estos datos cohesionados, sean precisas y confiables.

En este contexto, Python es una de las herramientas más usadas en la limpieza de datos debido a su versatilidad y robustez del compendio de bibliotecas. Herramientas como Pandas, NumPy tienen una colección de funciones avanzadas que permiten la manipulación y transformación de los datos, consiguiendo facilitar tareas como la eliminación de duplicados, la gestión de valores faltantes, irregularidades en los registros y normalización de características numéricas (Van Der Aalst et al., 2018) , (Jafari, 2024). La biblioteca Pandas, permite la lectura, transformación y limpieza de grandes conjuntos de datos, que hace llamar “DataFrame”, de manera eficiente, lo cual es parte fundamental para la preparación de los datos a análisis con mayor complejidad y modelado predictivo (Provost y Fawcett, 2013). Además, Python es un lenguaje de programación con facilidad de uso, y su comunidad de usuarios la hacen la opción más accesible con un sector objetivo de oscila entre principiantes y profesionales experimentados en ciencia de datos. En conclusión, el uso de Python no solo optimiza el proceso de cleaning, sino que también ofrecen una base sólida para realizar análisis de datos más profundos y precisos.

4.4. Seguridad

En el contexto de las soluciones digitales, la seguridad en la protección de datos y credenciales es primordial. Por ello, aquellos trabajos que pretenden manejar información sensible deben implementar mecanismos sólidos de seguridad para evitar brechas que expongas información con alto riesgo, como credenciales o incluso información de terceros involucrados. Teniendo en cuenta esto,

existen diversas herramientas y tecnologías cuyas características de seguridad integradas permiten la gestión y protección de la información durante su ciclo de vida útil. Algunos métodos de estos son:

4.4.1. Encriptación de Datos Sensibles

Uno de los más importantes métodos para proteger los datos sensibles, es la encriptación de estos. La encriptación simétrica, especialmente mediante el algoritmo AES-256, se ha consolidado como uno de los métodos más seguros y eficientes para proteger datos en reposo. Este tipo de encriptación utiliza una única clave para cifrar y descifrar la información, garantizando así que solo se pueda acceder a los datos protegidos por medio de autorización. Los archivos de configuración, como aquellos que contienen credenciales de acceso a bases de datos, deben ser cifrados antes de ser almacenados para protegerse de accesos no autorizados (Cryptography, 2024).

Un punto importante parte de la necesidad de que la clave de encriptación debe estar gestionada de manera correcta, evitando que sea vulnerada. Existen herramientas de gestión de secretos como Azure Key Vault, que gestiona claves en entornos locales. Estas herramientas permiten almacenar y tener control sobre el acceso a estas claves de manera segura (Microsoft, 2024a).

4.4.2. Conexiones Seguras a Bases de Datos

En proyectos que dependen de consultas y operaciones sobre bases de datos, establecer conexiones seguras es crucial para proteger la información durante su transmisión. Bases de datos comerciales, como Oracle y SQL Server, ofrecen soporte para conexiones cifradas a través de protocolos SSL/TLS. Estos protocolos garantizan que los datos se transmitan de forma segura, evitando que

sean interceptados o modificados por terceros no autorizados.

En la práctica, el uso de la librería Python-oracledb en modo Thick, por ejemplo, facilita la configuración de conexiones seguras entre la solución y la fuente de datos, aprovechando las librerías cliente de Oracle para implementar Capa de Sockets Seguros/Seguridad de la capa de transporte (SSL/TLS) (Oracle, 2024a). Este proceso asegura el tránsito de información entre el servidor y las bases de datos, para solidificar aún más la solución, algunos trabajos refuerzan la seguridad haciendo uso de redes privadas virtuales (VPN) y firewalls para restringir el acceso a la base de datos únicamente a usuarios y direcciones IP autorizadas, brindando una capa adicional de seguridad (Microsoft, 2024f).

4.4.3. Seguridad en la plataforma Microsoft Power Platform

Microsoft Power Platform, que incluye herramientas como Power Apps y Power Automate, contiene prácticas de seguridad integradas para controlar los accesos a las soluciones creadas y garantizar la protección de los datos usados en los entornos. La plataforma implementa un modelo de control de acceso basado en roles (RBAC), que permite asignar permisos específicos a los usuarios en función de su rol. Esta segmentación asegura que solo los usuarios autorizados puedan acceder y manipular aplicaciones, flujos o datos sensibles dentro del entorno (Microsoft, 2024e).

Adicional a esto, contiene capacidades de cifrado de datos en tránsito y en reposo, utilizando protocolos como TLS para la transmisión de información y AES-256 para el almacenamiento de datos sensibles (Microsoft, 2024c). Garantizando que las conexiones y los datos almacenados en la nube estén protegidos de accesos no autorizados y manipulaciones.

Basándonos en esta base conceptual, se debe profundizar en la metodología de investigación.

5. Metodología

En esta sección se aborda de qué manera se van a llevar a cabo los objetivos planteados inicialmente, en la cual se notarán las configuraciones que se deben realizar en los softwares, las tecnologías que se van a usar, y qué se va a realizar con ellas. Consta de 4 etapas, las cuales son la conexión a las bases de datos y extracción de la información por medio de Python, la automatización de este proceso con Power Automate, la realización de la GUI con PowerApps y finalmente, la implementación de pruebas, documentación y manuales técnicos y de soporte.

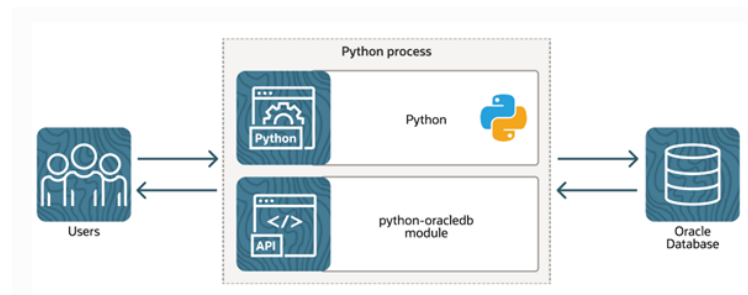
5.1. Script de Python

Se propone crear un script mediante el cual se utiliza la librería de Python para conectar con las bases de datos usadas en los sistemas de información. Estas bases de datos están alojadas en Oracle, por lo que se requiere llevar a cabo la conexión a través de la librería Python-oracledb. Esta librería, en particular, tiene dos modos de conexión: la arquitectura del modo "Thin"(ver Figura 5) permite la conexión directa a la base de datos, la cual puede estar en la misma máquina o puede ser remota (Oracle, 2024a).

El otro modo es el "Thick" (ver Figura 6). Python-oracledb opera en este modo cuando se enlaza con las librerías de Oracle Client, lo que le proporciona funcionalidad adicional. Dado que las bases de datos con las que se requiere establecer la conexión son de versiones antiguas, se propone utilizar Python-oracledb en modo Thick para garantizar compatibilidad y aprovechar sus capacidades avanzadas (Oracle, 2024a).

Figura 5.

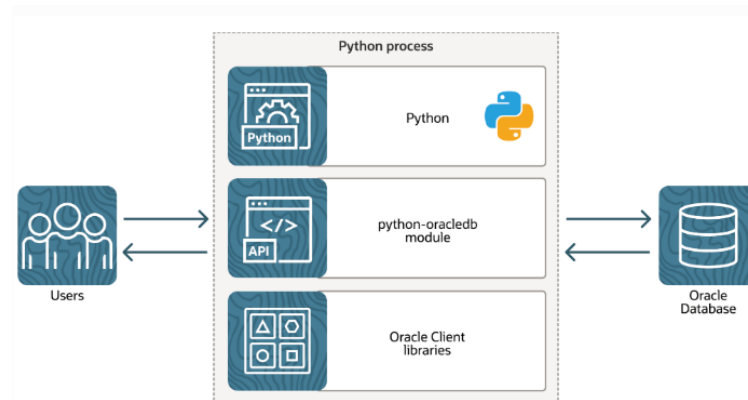
Modo Thin.



Nota: Arquitectura del controlador python-oracledb en modo Thin. (Oracle, 2024a). Adaptación.

Figura 6.

Modo Thick.



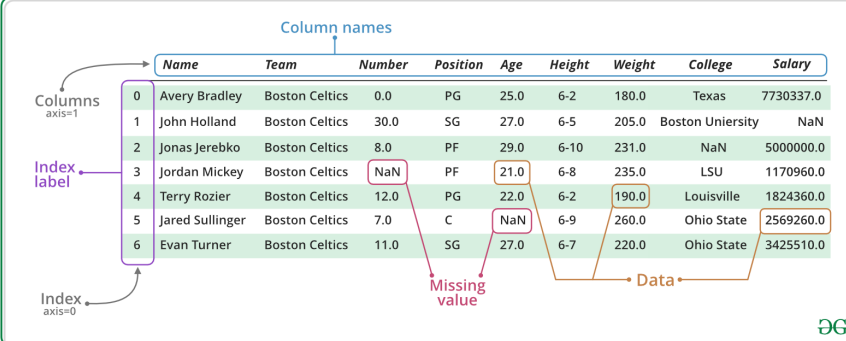
Nota: Arquitectura del controlador Python-oracledb en modo Thick. (Oracle, 2024a). Adaptación.

Posterior a la conexión, se hace una consulta a la base de datos, por medio de un query, normalmente está hospedado en un archivo con formato sql, esta consulta va a traer aquella información que se especifique, y para mayor facilidad en el siguiente proceso, se convertirá a un formato DataFrame (ver Figura 7). Este formato permite a la librería Pandas de Python limpiar y procesar la información con todas las herramientas que posee. Hay que hacer procesos de limpieza: eliminar duplicados, eliminar nulos, cruzar información con archivos de Excel, cambiar los valores de una columna en específico y cambiar el tipo de dato para operarlo con otros. Esta librería permite obtener un reporte con el formato establecido, y el tipo de archivo necesario (.txt) (pandas development

team, 2024).

Figura 7.

DataFrame Pandas.



The diagram illustrates the structure of a Pandas DataFrame. It shows a table with 9 columns and 7 rows. The columns are labeled 'Name', 'Team', 'Number', 'Position', 'Age', 'Height', 'Weight', 'College', and 'Salary'. The rows are indexed from 0 to 6. The 'Number' column contains values 0.0, 30.0, 8.0, NaN, 12.0, 7.0, and 11.0. The 'Age' column contains values 25.0, 27.0, 29.0, 21.0, 22.0, NaN, and 27.0. The 'Height' column contains values 6-2, 6-5, 6-10, 6-8, 6-2, 6-9, and 6-7. The 'Weight' column contains values 180.0, 205.0, 231.0, 235.0, 190.0, 260.0, and 220.0. The 'College' column contains values Texas, Boston University, NaN, LSU, Louisville, Ohio State, and Ohio State. The 'Salary' column contains values 7730337.0, NaN, 5000000.0, 1170960.0, 1824360.0, 2569260.0, and 3425510.0. The 'NaN' values in the 'Number', 'Age', and 'College' columns are highlighted with red boxes and labeled 'Missing value'. The 'Index label' is shown as a purple box around the row index (0-6). The 'Column names' are shown as a blue box around the column headers. The 'Data' is shown as an orange box around the cell values. The 'Index axis=0' and 'Columns axis=1' are also indicated.

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Nota: Estructura de un DataFrame de Pandas. (pandas development team, 2024). Adaptación.

Finalmente, este Script de Python resulta en un archivo plano que contiene la información lista para reportar.

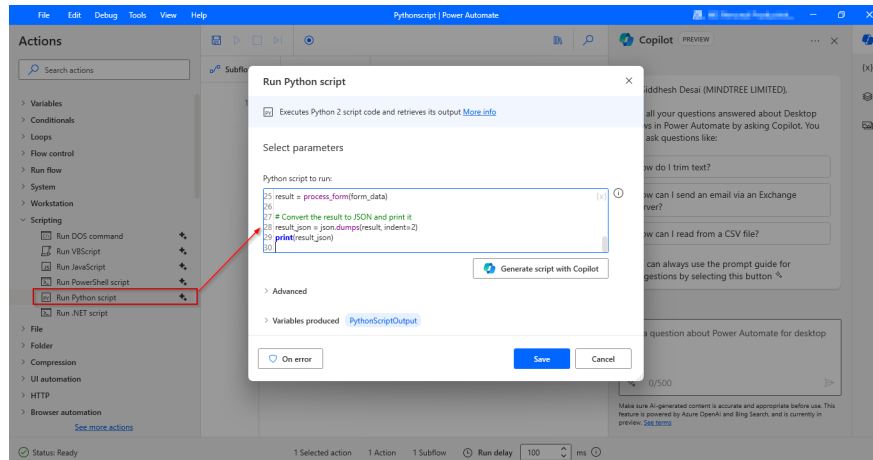
5.2. Automatización Robótica de Procesos

Como consiguiente de este proceso, alguien debe tener la responsabilidad de ejecutar este Script, y aquí se introduce el papel de la Automatización Robótica de Procesos (RPA) por medio de la herramienta Microsoft Power Automate, se crean flujos de nube, cuyo desencadenador es la captura de parámetros por medio de una interfaz. A partir de esto, la herramienta low-code proporciona gran facilidad para la creación de los flujos, para este caso en particular se requiere crear un flujo que ejecute el Script de Python por medio de Power Automate Desktop (ver Figura 8). Este flujo pondrá los parámetros capturados en el momento que se desencadena, y los pondrá en el Script, para seguir con la ejecución de este, ya que este Script da como resultado un archivo plano, el siguiente flujo buscará este archivo plano en la ruta donde se hospedó, para cargarlo en la página web requerida, para ello tendrá que validar con el usuario, por ello enviará un flujo de aprobación

(Stack Overflow, 2024).

Figura 8.

Ejecución Script con Power Automate



Nota: Ejemplo de la ejecución de un Script de Python desde un flujo de Power Automate Desktop. (Stack Overflow, 2024). Adaptación.

Siendo la aprobación efectiva, se construye el flujo que carga la información, por lo que irá a obtener las credenciales en un archivo cuya ruta conoce y hará el respectivo cargue. Una vez hecho lo anterior, se enviará un correo masivo a las entidades por medio de Outlook, este se enviará mediante un conector de Power Automate, cuyo contenido incluirá una tabla HTML dinámica, reflejando la información reportada a cada entidad.

Siendo la aprobación efectiva, se construye el flujo que carga la información, entonces irá a obtener las credenciales en un archivo cuya ruta conoce, y hará el respectivo cargue. Hubo hecho lo anterior, este enviará un correo masivo a las entidades por medio de Outlook, este es un conector de Power Automate, cuyo contenido tendrá una tabla HTML dinámica, reflejando la información reportada a cada entidad.

5.3. Interfaz Gráfica de Usuario

Para finalizar el desarrollo, los parámetros necesarios para la creación y cargue del reporte, se deben capturar por medio de una interfaz gráfica de usuario (GUI), para ello se utiliza la herramienta de Microsoft, Power Apps. Gracias a la facilidad que esta herramienta presta para la creación de interfaces se propone crear una de manera que sea un formulario (ver Figura 9) (tapanm-MSFT, 2024). Este formulario dará la posibilidad al usuario de digitar los parámetros requeridos para el reporte, y un botón cuya funcionalidad será ser el desencadenador de los flujos posteriores a estos, los cuales mantienen la lógica del proceso (ver Figura 10) (tapanm-MSFT, 2024).

Figura 9.

Ejemplo Form Power Apps

The screenshot shows a Power Apps form titled "Sales Order SO004". The form is organized into several sections with labels in blue. The top section includes fields for "Order date" (2/4/2016), "Order status" (Invoice), and "Customer purchase order reference". Below this, there are fields for "Name" (Lynn Haney) and "Description" (Tricia Hess). The next section contains "Total amount" (350), "Currency of Total amount" (USD), "Freight terms" (FOB), and "Shipping method" (AirBorne). The "Delivery address" section includes a text field for the address (123 Gray Rd) and a text field for the apartment (APT 723). The bottom section contains fields for "City" (Colorado), "State" (CO), "Postal code" (80001), and "Country/region" (US). The form uses a combination of text boxes, dropdown menus, and date pickers for data entry.

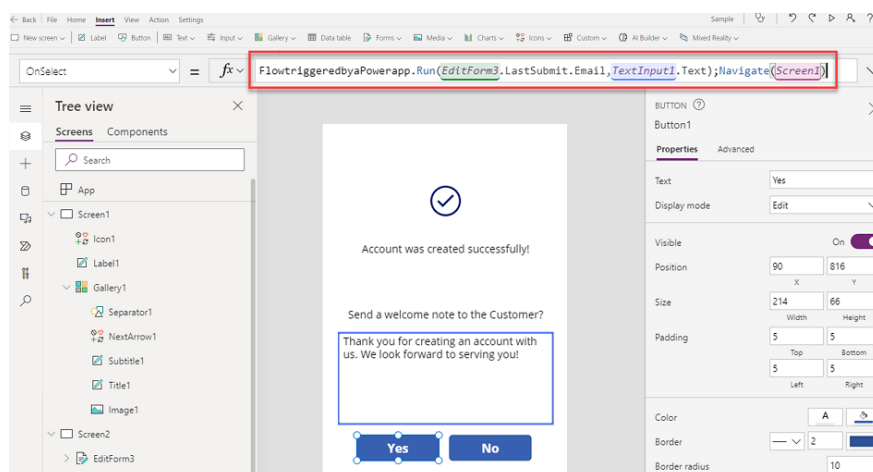
Nota: Ejemplo de la creación de un formulario para la captura de parámetros en Power Apps (tapanm-MSFT, 2024). Adaptación.

5.4. Implementación de seguridad

Con el fin de proporcionar una seguridad eficiente al proyecto y proteger tanto las credenciales como las conexiones entre la máquina y las bases de datos, se investigó las capacidades de seguridad

Figura 10.

Botón que ejecuta un flujo



Nota: Ejemplo de la configuración de un botón como desencadenador de un flujo de Power Automate (tapanm-MSFT, 2024). Adaptación.

proporcionadas por Microsoft Power Platform y la implementación de encriptación de credenciales mediante librería en Python.

5.4.1. Roles y Permisos en Microsoft Power Platform

Microsoft Power Platform utiliza un sistema de roles basado en permisos que indica qué acciones puede realizar un usuario en cada entorno (Microsoft, 2024d). Los entornos son creados para fines de desarrollo, pruebas o producción (Ver Figura 11). Estos roles se dividen en 3 principales categorías:

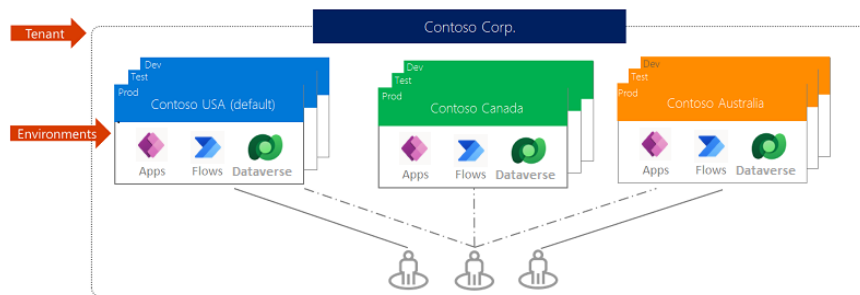
5.4.1.1. Administrador de entorno. Este rol tiene acceso completo al entorno, lo que incluye capacidad para gestionar usuario, flujos, aplicaciones e incluso configuraciones de seguridad.

5.4.1.2. Creador de aplicaciones. Un creador de aplicaciones puede crear y personalizar aplicaciones en el entorno, pero su capacidad, pero su alcance en otros aspectos del entorno es limitado.

5.4.1.3. Usuario básico. Tienen permisos restringidos y solo pueden utilizar las aplicaciones y flujos a los que se les haya otorgado acceso, un administrador de entorno.

Figura 11.

Gestión de entornos en Power Platform



Nota: Implementación de seguridad por medio de entornos administrados en Power Platform (Microsoft, 2024d). Adaptación.

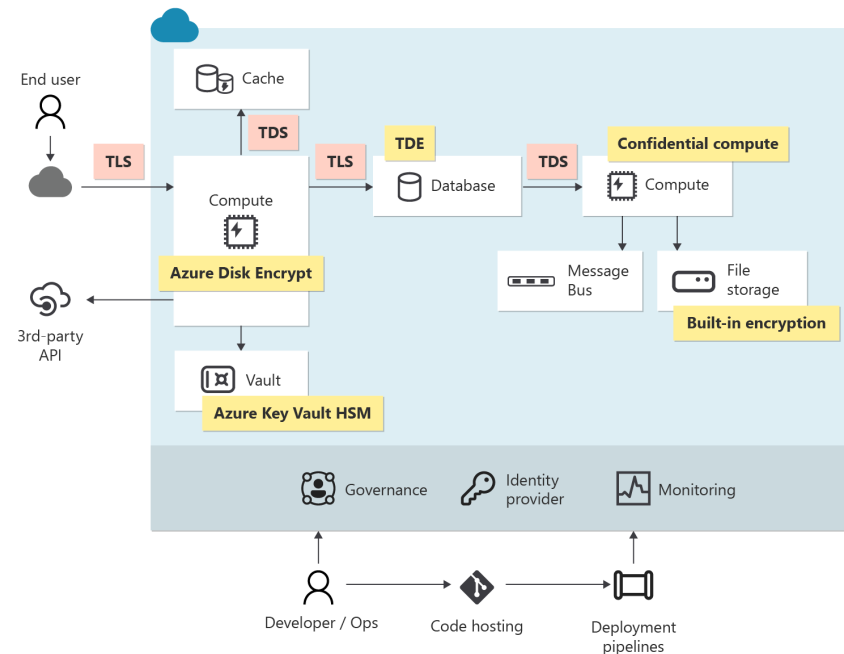
5.4.2. Conexión de Servicios

La comunicación entre los servicios de Microsoft, como Power Automate, Power Apps, Outlook o Sharepoint está protegida por medio de cifrado de datos en tránsito (TLS) (Ver Figura 12) y datos en reposo (AES-256), garantizando así que la información se mantenga protegida frente a accesos no autorizados en la transmisión y almacenamiento de esta (Microsoft, 2024c).

Por ello, es vital configurar de manera correcta los roles de los usuarios, con el fin de evitar la exposición no deseada de datos sensibles. En entorno de desarrollo y pruebas, es común tener un mayor acceso, pero en entornos de producción es recomendable restringir los permisos solo a usuarios esenciales (Microsoft, 2024d).

Figura 12.

Cifrado de datos en Power Platform



Nota: Cifrado de datos en tránsito (TLS) usado por Microsoft Power Platform (Microsoft, 2024c). Adaptación.

5.4.3. Encriptación de credenciales

Para asegurar las credenciales de acceso a las bases de datos como SAC o JD Edwards, se propone un sistema de encriptación basado en Python que utiliza cifrado simétrico (AES-256) a través de la biblioteca cryptography (Ver Figura 13). Este enfoque asegura que las credenciales se almacenen en un archivo JSON encriptado, y que únicamente sean desencriptadas durante la ejecución del Script (Microsoft, 2023).

5.4.3.1. Generación de clave. Una clave de encriptación se genera una única vez y se almacena en un archivo protegido (Microsoft, 2023).

5.4.3.2. Almacenamiento seguro. El archivo que contiene la clave, debe ser guardado en una ubicación segura, cuya autorización será restringida para su acceso (Microsoft, 2023).

5.4.3.3. Encriptación de credenciales. Durante el desarrollo del proyecto, las credenciales se encriptan usando esta clave previamente gestionada y se almacenan en un archivo JSON encriptado (Microsoft, 2023) y (Foundation, 2024).

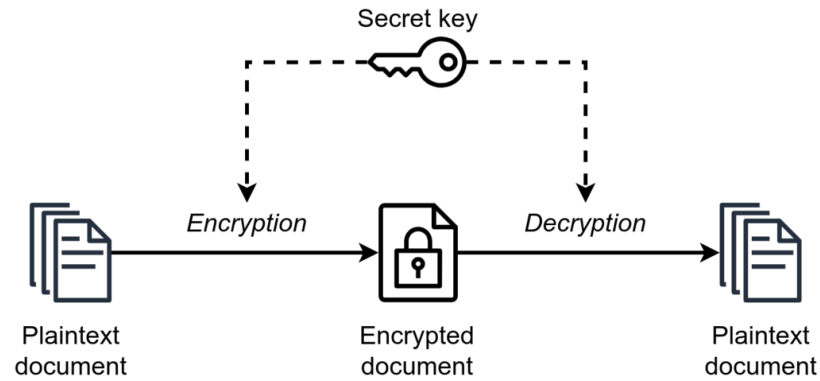
5.4.3.4. Desencriptación. Cuando se ejecuta el módulo del Script que requiere las credenciales, el archivo de claves es utilizado para desencriptar el contenido del archivo JSON (Microsoft, 2023) y (Foundation, 2024).

Es fundamental que el archivo de claves esté protegido con permisos adecuados. Solo el servidor que ejecuta el Script debe tener acceso a estos archivos, lo cual puede gestionarse mediante permisos del sistema de archivos o almacenamiento de secretos locales (Oracle, 2024b).

Como alternativa, se propone el uso de variables de entorno con el fin de gestionar la clave de encriptación evitando almacenar la clave en un archivo. Esto reduce los riesgos de exposición en sistemas compartidos. Las variables de entorno se configuran al nivel del sistema operativo, de modo que las claves nunca se almacenan directamente en los archivos de código (Microsoft, 2024b).

5.4.4. Conexión entre máquinas

Con el fin de garantizar la seguridad de la conexión entre el servidor donde será hospedada la solución y las bases de datos (SAC y JD Edwards) como se definió anteriormente, el uso de la librería Python-oracledb en modo Thick que soporta el protocolo SSL/TLS para el cifrado de las

Figura 13.**Encriptación Simétrica**

Nota: Encriptación simétrica usada por la librería Cryptography Fernet (Cryptography, 2024).
Adaptación.

conexiones (Microsoft, 2024a).

Debido a que la solución será desplegada en un entorno corporativo, el cual usa redes privadas virtuales (VPN) o túneles seguros para la conexión a bases de datos, especialmente en entornos remotos. Y el uso de firewalls para limitar qué IPs pueden acceder a la base de datos, consolidar la seguridad eficiente de la conexión (Open Web Application Security Project (OWASP), 2024) y (Real Python, 2024).

5.5. Pruebas y Conclusiones

Finalmente, teniendo la estructura planteada, se harán pruebas funcionales, verificando información con reportes contraídos y cargados anteriormente, atrapando excepciones y posibles errores que los flujos, o el Script puedan presentar, esto con el fin de que el mantenimiento de esta solución a futuro sea simple. Teniendo un testeo perfecto, se propone buscar técnicas para obtener un despliegue eficaz junto con la estabilización del proyecto en producción, dando lugar al usuario

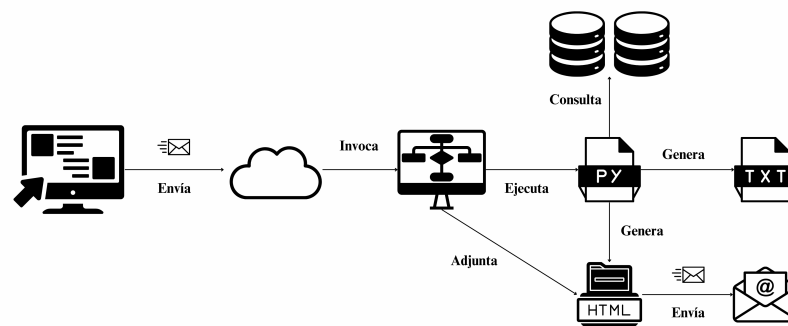
generar los reportes y sus respectivas verificaciones. Como trayecto final, con el proyecto en producción, la creación de un manual técnico de uso es vital ya que esto se procurará en el tiempo, permitiendo un mantenimiento preventivo útil dejando la puerta abierta para futuros cambios.

6. Desarrollo

La arquitectura que se llevó a cabo para la realización de la solución fue la siguiente:

Figura 14.

Arquitectura



Nota: Arquitectura completa de la solución desarrollada. León, J. (2024). Diseño propio.

Para el desarrollo de este trabajo se realizó una investigación y validación a fondo con el fin de garantizar la integración entre las herramientas y su correcto uso, teniendo en cuenta impedimentos a nivel empresarial, técnico y basándose en las reglas de negocio previamente socializadas con el usuario y su necesidad. Esta investigación da cabida a la integración del proyecto cuyo punto de partida es la lógica de la información, iniciando así con la construcción y diseño del Paquete Python.

6.1. Paquete Python

Inicialmente se tuvo planteado la creación de un Script de Python que llevara la lógica completa, sin embargo, en aras de la organización, captura de errores y buenas prácticas, se desarrolla un

Paquete Python (Beazley, 2013; Lutz, 2013). Se refiere a una colección o conjuntos de Scripts que están estructuralmente conectados. Algunos términos que permiten describir la estructura son:

- **Proyecto Python:** Se refiere al conjunto de Scripts o módulos que, al trabajar integrados, conforman una aplicación o sistema. Puede incluir archivos de Python `.py` con archivos individuales encargados de otras funcionalidades, como archivos de configuración, conexión, documentación y otros recursos (Martelli, 2006).
- **Módulo:** Cada Script `.py` es técnicamente hablando, un módulo en Python. Un módulo es cualquier archivo de código Python, que define clases, funciones y variables, y puede ser importado por otros scripts para su uso (Pilgrim, 2004).
- **Paquete:** Un paquete en Python es una colección de módulos organizados en directorios, que incluyen el archivo `init.py` para que Python los trate como un paquete importable. Los paquetes permiten estructurar proyectos de manera que las funcionalidades estén bien organizadas (Beazley, 2009).
- **Biblioteca (Library):** Se refiere a un conjunto de módulos o paquetes que proporcionan funcionalidades específicas y están diseñados para ser reutilizables en otros proyectos (van Rossum, 2001).

Un paquete Python ofrece las siguientes ventajas sobre un Script monolítico:

1. Mantenibilidad

- **Código más organizado:** Dividir la lógica en módulos hace que el código sea más fácil de entender y mantener. Cada módulo tiene una responsabilidad específica, lo que

facilita realizar modificaciones o correcciones sin afectar otras partes del sistema (Lutz, 2013).

- **Modularización:** Al tener diferentes funcionalidades separadas, es más sencillo localizar errores o añadir nuevas funcionalidades sin necesidad de revisar o modificar todo el código (Beazley, 2013).

2. Reutilización de código

- **Modularidad:** Los módulos pueden reutilizarse en otros proyectos; solo se requiere exportación, evitando la duplicidad de código y esfuerzos (Pilgrim, 2004).
- **Paquetes compartidos:** Los paquetes tienen la posibilidad de ser compartidos con otros desarrolladores o bien distribuirse públicamente, lo que facilita la creación de librerías utilizables (van Rossum, 2001).

3. Escalabilidad

- **Facilidad de expansión:** A medida que el proyecto crece en tamaño y complejidad, es más fácil agregar nuevas funcionalidades creando nuevos módulos en lugar de aumentar la longitud de un único Script (Beazley, 2013).
- **Evitar sobrecarga de funciones:** Con un solo Script, las funciones acaban por ser bastante largas y difíciles de gestionar, incluso llegando al punto de recrear funciones que ya estaban diseñadas (Lutz, 2013).

4. Pruebas y depuración

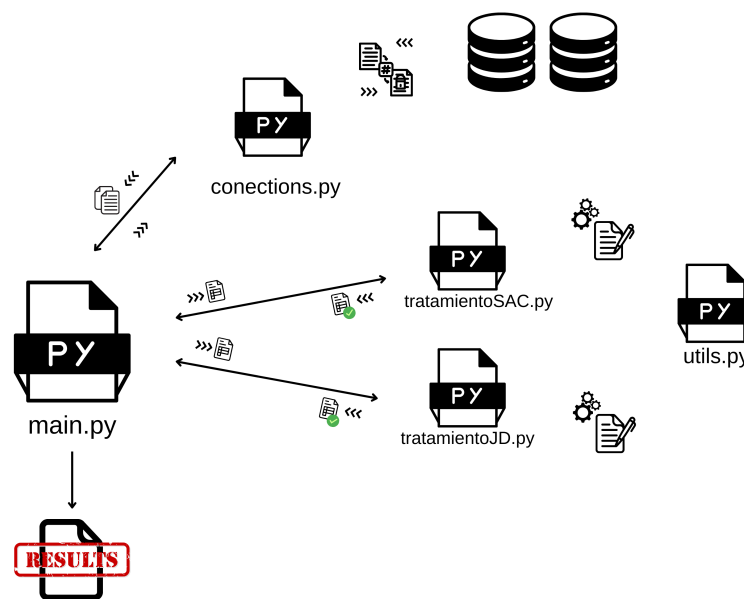
- **Pruebas unitarias:** Es más sencillo probar individualmente cada módulo y sus funcionalidades en un paquete modulado (Pilgrim, 2004).

- **Depuración sencilla:** Llegado el caso en que suceda un error, localizar el problema es mucho más sencillo en un sistema modular. Es posible depurar un módulo específicamente en lugar de todo el sistema (Lutz, 2013).

Teniendo en cuenta la viabilidad de realizar la lógica del Script modularmente en un Paquete, se da por inicio la creación de cada uno de estos, cada módulo está ligado con el principal, cuyo módulo mantiene el flujo de trabajo principal. Es este quien llama a cada uno de los módulos consiguientes para procesar la información. (Ver Figura 15)

Figura 15.

Gráfico de la estructura del Paquete Python.



Nota: Gráfico estructural de la integración modular del paquete de Python. León, J. (2024).
Diseño propio.

6.2. Automatización Robótica de Procesos

Se desarrolla un flujo de trabajo con la lógica para ejecutar el Paquete Pytho, sin embargo, ¿cómo este flujo se va a disparar? Se propuso inicialmente que sea por medio de un correo electrónico vía

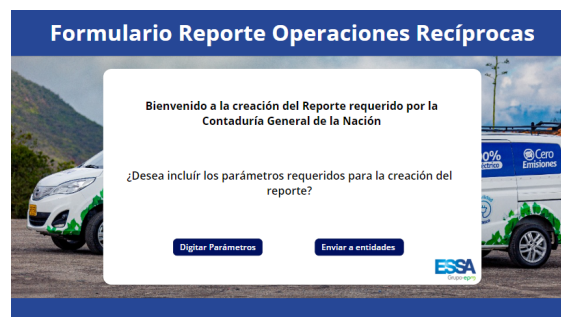
Outlook que es de por sí un conector de Microsoft Power Automate, el proceso de construcción de la lógica se divide en dos módulos de Automate, flujos de nube para el desencadenamiento de estos, ya que los flujos de escritorio no tienen trigger, y los correos se hace por medio de la nube de Microsoft, y por otro lado el flujo de escritorio que es el encargado de la ejecución directa en la línea de comandos del Paquete Python.

6.3. Interfaz Gráfica de Usuario

La interfaz fue creada en Microsoft Power Apps la cual consta de una sola pantalla, teniendo en cuenta que la interfaz es para efectos de generación de la lógica fuerte del proyecto, se decidió mantener pantallas emergente donde la lógica de visibilidad es basada en botones.

Figura 16.

Pantalla inicial GUI. **Form CGN.**



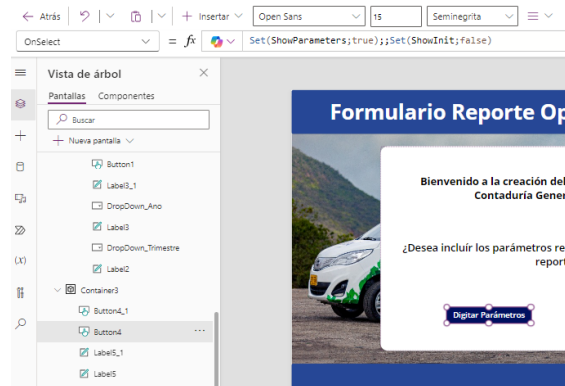
Nota: Visualización de la composición de la pantalla inicial de la interfaz gráfica. León, J. (2024). Diseño propio.

Fue desarrollada con parametrización a la hora de elegir los parámetros de ejecución, usando drop-box en lugar de cajas de texto propensas a errores de digitación e incluso una doble confirmación de ejecución para evitar mala diligencia de parámetros.

Inicialmente se tiene una primera pantalla con dos botones, uno pretende guiar al usuario a la digitación de los parámetros, y el otro al envío de las tablas HTML vía correo a cada entidad. (Ver

Figura 17.

Acción Botón "Digitar Parámetros". **Form CGN.**



Nota: Acción que realiza el botón "Digitar Parámetros". León, J. (2024). Diseño propio.

Figura 18.

Pantalla de diligenciamiento de parámetros **Form CGN.**

Nota: Pantalla dedicada al diligenciamiento de parámetros para la generación del reporte. León, J. (2024). Diseño propio.

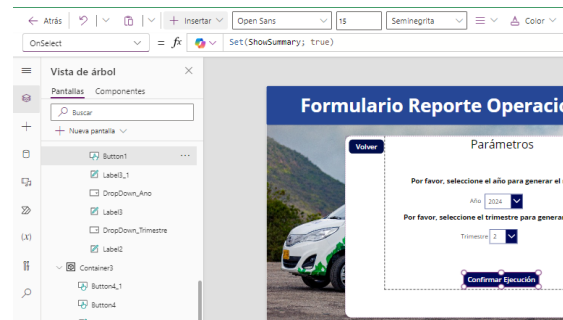
Figura 16)

La acción del botón **Digitar Parámetros**, es hacer visible el grupo de componentes que desarrollan la siguiente pantalla de digitalización, cambiando el valor booleano de las variables de visualización de cada componente. (Ver Figuras 17, 18)

En la pantalla de diligenciamiento de parámetros se tienen dos componentes **DropBox** o **Caja Desplegable** que permite al usuario elegir el año y el trimestre para el que requiere la generación

Figura 19.

Acción del botón de confirmar ejecución. **Form CGN.**



Nota: Acción que realiza el botón "Confirmar Ejecución". León, J. (2024). Diseño propio.

Figura 20.

Pantalla emergente de re confirmación de generación. **Form CGN.**



Nota: Pantalla emergente que resume los parámetros escogidos y re confirma la ejecución. León, J. (2024). Diseño propio.

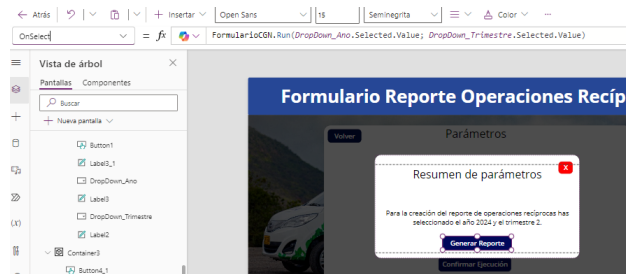
del reporte.

Posterior a esta acción, el usuario confirma la ejecución por medio del botón **Confirmar ejecución**, que manteniendo la lógica del anterior botón, hace visible una pantalla emergente que resume los parámetros escogidos y re confirma la generación del reporte. Ver figuras (19, 20)

Este botón de **Generar Reporte** tiene en su propiedad **OnSelect** la ejecución de un flujo de nube que tiene como parámetros los valores de las cajas desplegables, y las envía por correo a la cuenta técnica que contiene la solución. (Ver Figuras 21,22)

Figura 21.

Acción del botón Generar Reporte.**Form CGN.**

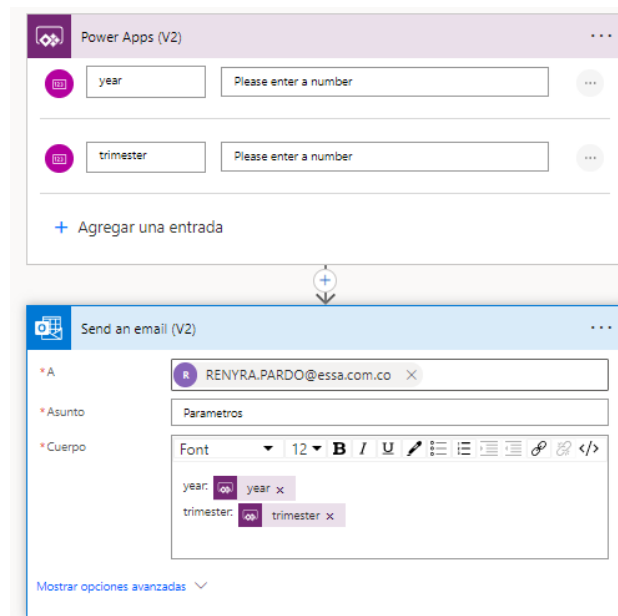


Nota: Acción que realiza el botón "Generar Reporte". León, J. (2024). Diseño propio.

Adicional a esto la acción y el flujo de envío de las tablas HTML, son generador por el botón **Envío a entidades**, cuya acción es desencadenar un flujo que envía una señal vía correo a la cuenta técnica que hospeda el flujo con tal lógica. (Ver Figuras 23,24)

Figura 22.

Flujo desencadenado por el botón "Generar Reporte". **Form CGN.**



Nota: Flujo cuyo trigger es el botón de la GUI de Power Apps, recibe parámetros y los envía por correo. León, J. (2024). Diseño propio.

6.4. Implementación de seguridad

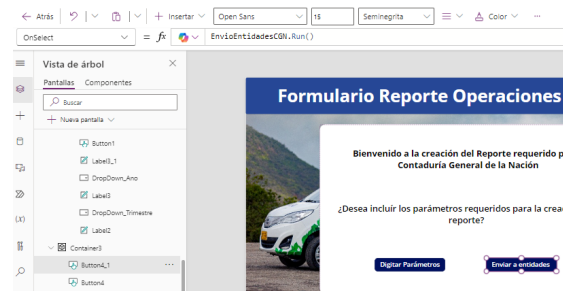
La implementación de seguridad se realizó inicialmente en la encriptación de credenciales para realizar la conexión a las bases de datos, por medio de un módulo del paquete de python, presentado en sesiones anteriores, donde su realización es de la siguiente manera:

También se realizar implementación de seguridad en la conexión a las bases de datos, ya que la librería Python-oracledb la cual hace enlace mediante el protocolo SSL/TLS para el cifrado de las conexiones, por otro lado la organización maneja el uso de firewalls y VPN como seguridad en las conexiones.

Por otra parte la implementación de la seguridad en los servicios explicada en sesiones anteriores,

Figura 23.

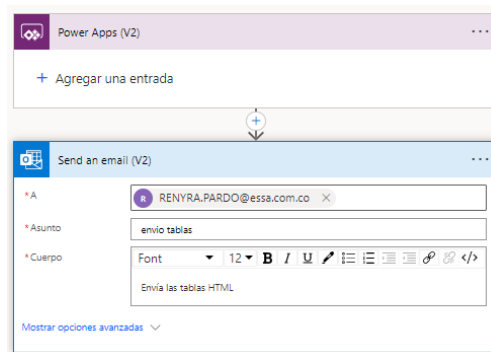
Acción del botón Envío a Entidades.**Form CGN.**



Nota: Acción que realiza el botón .Envío a entidades". León, J. (2024). Diseño propio.

Figura 24.

Flujo desencadenado por el botón .Envío a entidades". **Form CGN.**



Nota: Flujo cuyo trigger es el botón de la GUI de Power Apps y envía una señal de ejecución por correo. León, J. (2024). Diseño propio.

se hace uso de esta implementación otorgando permisos a los usuarios debido y manteniendo una gestión del flujo completo de la solución.

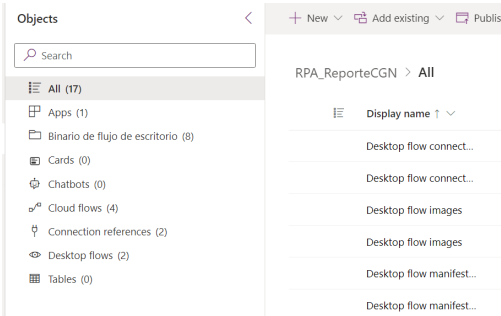
6.5. Despliegue y Estabilización

El despliegue de esta solución se realizó transfiriendo el Paquete Python al servidor donde se va a hospedar. Incluyendo requerimientos como la instalación de las librerías:

- pandas

Figura 25.

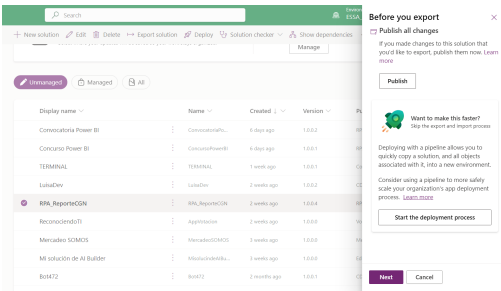
Componentes de la solución. **RPA_ReporteCGN**.



Nota: Componentes integrados en la solución RPA_ReporteCGN. León, J. (2024). Diseño propio.

Figura 26.

Exportación de la solución. **RPA_ReporteCGN**.



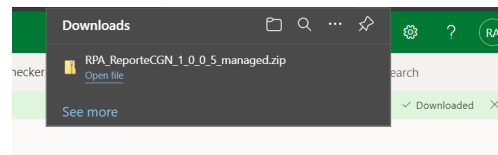
Nota: Exportación de la solución RPA_ReporteCGN en modo managed. León, J. (2024). Diseño propio.

- Python-Oracledb
- json
- cryptography.Fernet
- Oracle Instant Client

Ya que esta solución requiere de un lugar donde hospedar archivos, bien sea los resultantes, o los necesario para la ejecución, como lo son las conciliaciones y excepciones. Estos archivos se hospedan en una carpeta local compartida, que tiene acceso restringido, por ello tuvo que tramitarse

Figura 27.

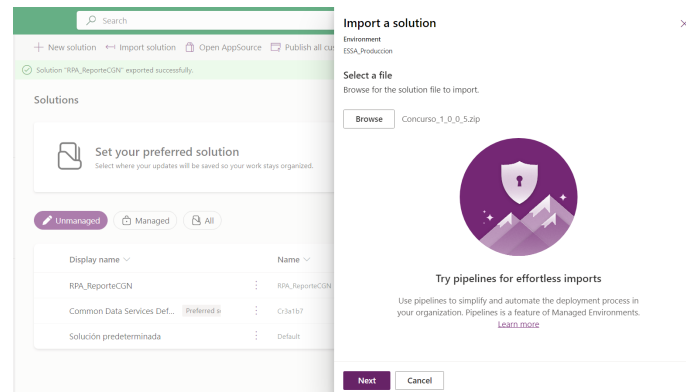
Paquete de la solución descargado. **RPA_ReporteCGN**.



Nota: Descarga de la carpeta compartida de los componentes de la solución RPA_ReporteCGN. León, J. (2024). Diseño propio.

Figura 28.

Importación de la solución. **RPA_ReporteCGN**.



Nota: Importación de la solución RPA_ReporteCGN en el entorno de producción. León, J. (2024). Diseño propio.

los permisos únicamente del usuario del servidor donde está hospedada la solución, y el usuario que debe cargar los archivos u obtener los resultantes.

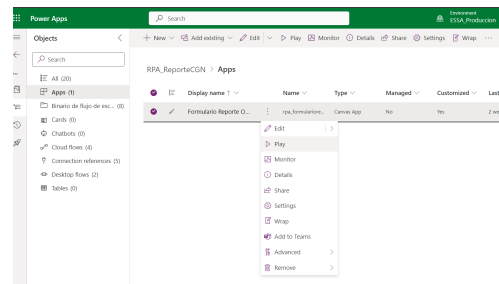
Se realiza despliegue a las conexión de las bases de datos, cambiando el ambiente al que se hace conexión, cambiando el host al que se apunta, el puerto y el sid o service name.

Para el despliegue de la parte de la solución desarrollada por herramientas de Microsoft, se debe crear una solución, integrando todos los elementos (Ver Figura 25):

- aplicación

Figura 29.

Solución en producción. **RPA_ReporteCGN.**



Nota: Solución RPA_ReporteCGN ejecutada en producción. León, J. (2024). Diseño propio.

- flujos de nube
- flujos de escritorio
- referencias de conexión
- conectores

Teniendo la solución creada en el entorno de desarrollo, se procede a exportarla en modo **Managed** por buenas prácticas. (Ver Figura 26)

Esta solución exporta todos los componentes en una carpeta comprimida (.zip). (Ver Figura 26)

Posterior a la descarga se realiza el cambio de entorno a producción y se importa la solución (Ver Figura 28)

Finalmente se ejecuta la solución en producción (Ver Figura 29)

Figura 30.

Pruebas flujo de nube que desencadena la solución

FormularioCGN > Run history			
Start time	Duration	Add column	Status
Sep 18, 10:50 AM (1 wk ago)	0:07 min		Successful
Sep 18, 10:53 AM (1 wk ago)	0:00:00		Successful
Sep 18, 04:10 PM (1 wk ago)	1:00 min		Successful
Sep 18, 04:09 PM (1 wk ago)	0:00:00		Successful
Sep 6, 04:50 PM (2 wk ago)	0:00:00		Successful

Nota: Prueba de las ejecuciones por parte del flujo desencadenado por la interfaz gráfica y desencadenador del flujo que invoca el flujo de escritorio. León, J. (2024). Diseño propio.

Figura 31.

Pruebas flujo de nube hospedado en cuenta técnica

GetParametersCGN > Run history			
Start time	Duration	Add column	Status
Sep 18, 10:59 AM (1 wk ago)	0:01:40		Successful
Sep 18, 10:34 AM (1 wk ago)	0:01:06		Successful
Sep 18, 04:17 PM (1 wk ago)	0:01:01		Successful
Sep 18, 04:09 PM (1 wk ago)	0:00:00		Successful
Sep 6, 05:15 PM (2 wk ago)	0:00:00		Failed
Sep 6, 05:13 PM (2 wk ago)	0:00:00		Failed
Sep 6, 05:07 PM (2 wk ago)	0:00:00		Failed
Sep 6, 04:59 PM (2 wk ago)	0:00:00		Failed

Nota: Prueba de ejecución del flujo que recibe los parámetros e invoca el flujo de escritorio. León, J. (2024). Diseño propio.

6.6. Pruebas

Se realizaron pruebas desde el punto más bajo de la solución (el paquete Python), hasta la solución en conjunto, desde la interfaz gráfica para el reporte del primer y segundo trimestre del presente año.

1. Se ejecutan los parámetros desde la aplicación y se probaron las ejecuciones del flujo desencadenador que enviará los parámetros vía correo (Ver Figura 30)
2. Con la ejecución del flujo, se estudia la ejecución del flujo que recibe los parámetros y ejecuta el flujo de escritorio a cargo del Paquete Python. (Ver Figura 31)
3. Ahora bien, teniendo pruebas se revisan las ejecuciones del flujo de escritorio que ejecuta el Paquete Python (Ver Figura 32)

Figura 32.

Pruebas flujo de escritorio

RunPythonScript > Run history

Requested	Duration	Status
Sep 16, 10:09 AM (1 wk ago)	00:14:29	Succeeded
Sep 16, 10:34 AM (1 wk ago)	00:19:33	Succeeded
Sep 11, 04:17 PM (1 wk ago)	00:12:04	Succeeded
Sep 11, 04:10 PM (1 wk ago)	00:01:29	Succeeded
Sep 6, 05:15 PM (2 wk ago)	00:24:53	Succeeded
Sep 6, 05:13 PM (2 wk ago)	00:00:42	Failed
Sep 6, 05:07 PM (2 wk ago)	00:00:39	Failed
Sep 6, 04:59 PM (2 wk ago)	00:00:53	Failed

Nota: Prueba de ejecución del flujo de escritorio que ejecuta el Paquete Python. León, J. (2024).
Diseño propio.

Figura 33.

Verificación reportes ejecutados

ARCHIVOS SCRIPT - REPORTE CGN > Reportes

Name	Date modified	Type	Size
1T OPERACIONES RECIPROCAS CHIP 2024.txt	9/18/2024 9:27 AM	Text Document	28 KB
2T OPERACIONES RECIPROCAS CHIP 2024.txt	9/26/2024 4:13 PM	Text Document	31 KB

Nota: Verificación de la creación del reporte en la ruta específica. León, J. (2024). Diseño propio.

4. Se hacen verificaciones de la creación del reporte y se verifican sus valores los cuales coinciden, ya que se hace comparativa con los reportes ya realizados (Ver Figuras 33,34)
5. Para el canal de envío del detalle (tablas HTML) a cada entidad se realiza de la misma manera, y fue exitoso el envío (Ver Figura 35)

Finalmente con cada ejecución se puede hacer una proyección de las ejecuciones y una comparativa referente al gatos previo a la implementación de la automatización. Los resultados de ejecución tienen un promedio de 20 minutos, esta ejecución es necesaria únicamente 4 veces al año, ya que su reporte se maneja con frecuencia trimestral. (Ver Figura ??)

Figura 34.

Verificación reporte resultante

2f OPERACIONES RECIPROCAS CHIP 2024.txt - Notepad

S	389000000	246	2024	CGN2015_002_OPERACIONES_RECIPROCAS_CONVERGENCIA
S	1.1.10.05	69600000	1866555847.17	0.0
D	1.1.10.06	69600000	1327040597.0	0.0
D	1.3.17.15	236105001	15116169.79	0.0
D	1.3.17.90	37217000	20249717.11	0.0
D	1.3.17.90	39363000	26541826.1	0.0
D	1.3.84.10	38541000	11660728.0	0.0
D	1.3.84.10	89600000	1339612.0	0.0
D	1.3.84.39	218768307	4121499.2	0.0
D	1.3.84.39	923272569	4037123.79	0.0
D	1.3.84.90	215568755	450000000.0	0.0
D	1.3.87.03	218168001	4244204821.0	0.0
D	1.3.87.03	211120011	178412940.0	0.0
D	1.3.87.03	211388012	5523623.0	0.0
D	1.3.87.03	211420616	741702782.0	0.0
D	1.3.87.03	211868318	2548770.0	0.0
D	1.3.87.03	212066020	15481291.0	0.0
D	1.3.87.03	212166121	2474898.0	0.0
D	1.3.87.03	212568425	2675499.0	0.0
D	1.3.87.03	214468344	3652291.0	0.0
D	1.3.87.03	214768547	2143743651.0	0.0
D	1.3.87.03	215068250	15115192.0	0.0
D	1.3.87.03	215268152	6640085.0	0.0
D	1.3.87.03	216011100	2063349375.0	0.0
D	1.3.87.03	216068160	32340076.0	0.0
D	1.3.87.03	216468464	1034106.0	0.0
D	1.3.87.03	216668266	13862543.0	0.0
D	1.3.87.03	216668468	10595000.0	0.0
D	1.3.87.03	216968169	35075444.0	0.0
D	1.3.87.03	217013670	49821878.0	0.0
D	1.3.87.03	217068370	32501673.0	0.0

Nota: Verificación del formato del reporte y los valores. León, J. (2024). Diseño propio.

7. Conclusiones

La integración de Python en entornos de Microsoft ha demostrado ser una solución eficiente y adaptable para la automatización de procesos complejos. Su capacidad para manejar grandes volúmenes de datos, junto con su amplia gama de bibliotecas especializadas como Pandas, ha facilitado el tratamiento y análisis de datos que, de otro modo, serían costosos y difíciles de gestionar utilizando únicamente herramientas nativas de Microsoft. La facilidad de integración con bases de datos permite realizar operaciones de manera rápida y eficaz, optimizando significativamente el tiempo de procesamiento de datos.

Como se muestra en la Figura ?? (a), el tratamiento manual de datos, que tradicionalmente llevaba alrededor de 540 horas al año, ha sido reducido en un 99.7 % mediante la automatización, disminuyendo el tiempo operativo a solo 80 minutos al año (Figura ?? (b)). Este ahorro de tiempo implica una liberación considerable de recursos humanos y tecnológicos, que ahora pueden ser re-

orientados hacia actividades de mayor valor añadido para la empresa, impulsando la productividad y la eficiencia operativa. Además, este enfoque refuerza la escalabilidad y flexibilidad de las soluciones basadas en Python en entornos corporativos, abriendo nuevas oportunidades para la mejora continua en la gestión de datos y la automatización de tareas.

Figura 35.

Correo con detalle enviado.



Nota: Correo de llegada enviado por la cuenta técnica con la tabla HTML, éxito de ejecución.
León, J. (2024). Diseño propio.

7.1. Trabajo Futuro

A futuro, la automatización de procesos contables en la Electrificadora de Santander S.A. (ESSA) podría beneficiarse de la integración de Inteligencia Artificial (IA) y Machine Learning (ML), lo que permitiría que los bots adaptaran su comportamiento a cambios inesperados en los datos o procesos, mejorando la toma de decisiones y evolucionando hacia la Automatización de Procesos Inteligentes (IPA). Asimismo, la incorporación de tecnologías de Procesamiento de Lenguaje Natural (NLP) facilitaría la interpretación de información no estructurada, reduciendo la intervención humana.

Otra área de mejora es la seguridad de los datos. Aunque ya se emplean herramientas como Azure Key Vault, el uso de criptografía homomórfica permitiría procesar datos cifrados sin necesidad de descifrarlos, aumentando así la protección de la información sensible. Además, la metodología desarrollada puede extenderse a otros departamentos de la empresa, como recursos humanos o finanzas, optimizando la eficiencia en varias áreas.

Finalmente, explorar otras plataformas de bajo código como Appsmith o Retool ofrecería mayor flexibilidad en los flujos de trabajo, y un análisis del impacto en la sostenibilidad podría alinear estos esfuerzos de automatización con los objetivos medioambientales de la empresa.

Bibliografía

- Aalst, W. M. P. v. d., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of Business Process Management*. Springer. <https://doi.org/10.1007/978-3-662-56509-4>
- Alsaadi, H., Radain, D., Alzahrani, M., Alshammari, W., Alahmadi, D., & Fakieh, B. (2021). Factors that affect the utilization of low-code development platforms: Survey study. *Revista Română de Informatică Şi Automatică*, 31, 123-140. <https://doi.org/10.33436/v31i3y202110>
- Banks, A., & Porcello, E. (2017). *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux*. O'Reilly Media.
- Beazley, D. (2009). *Python Essential Reference*. Addison-Wesley Professional.
- Beazley, D. (2013). *Python Cookbook: Recipes for Mastering Python 3*. O'Reilly Media.
- Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Apress.
- Brown, E. (2019). *Web Development with Node and Express: Leveraging the JavaScript Stack*. O'Reilly Media.
- Castro, M. C. (2019). Ambientes de aprendizaje. *Sophia*, 15(2), 123-134. <https://revistas.ugca.edu.co/index.php/sophia/article/view/827/1473>
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387. <https://doi.org/10.1145/362384.362685>

- Cooper, L. A., Holderness, D. K., Sorensen, T. L., & Wood, D. A. (2019). Robotic Process Automation in Public Accounting. *Accounting Horizons*, 33(4), 15-35. <https://doi.org/10.2308/acch-52466>
- Cryptography. (2024). *Fernet - symmetric encryption*. <https://cryptography.io/en/latest/fernet/>
- Di Ruscio, D., Kolovos, D., Lara, J., Pierantonio, A., Tisi, M., & Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin? *Software and Systems Modeling*, 21. <https://doi.org/10.1007/s10270-021-00970-2>
- Doguc, O. (2020). Robot Process Automation (RPA) and Its Future. En *Advances in e-business research series* (pp. 469-492). <https://doi.org/10.4018/978-1-7998-1125-1.ch021>
- Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2003(1), 171-178.
- Foundation, P. S. (2024). *Cryptography documentation*. <https://cryptography.io/en/latest/>
- Gami, A., & Singh, R. (2019). Intelligent Process Automation (IPA) and its Impact on Business Efficiency. *Journal of Business Automation*, 5(3), 45-60. <https://doi.org/10.1017/jba.2019.12>
- Gami, M., Jetly, P., Mehta, N., & Patil, S. (2019). Robotic Process Automation–Future of Business Organizations: A Review. <https://doi.org/10.2139/ssrn.3370211>
- Jafari, R. (2024). *Hands-On Data Preprocessing in Python*. O'Reilly Media.
- Kazil, J., & Jarmul, K. (2016). *Data Wrangling with Python: Tips and Tools to Make Your Life Easier*. O'Reilly Media.
- Kurnia, A., Lim, A., & Cheang, B. (2001). Online judge. *Computers & Education*, 36(4), 299-315. [https://doi.org/10.1016/S0360-1315\(01\)00013-4](https://doi.org/10.1016/S0360-1315(01)00013-4)

Lutz, M. (2013). *Learning Python*. O'Reilly Media.

Madakam, S., Holmukhe, R., & Jaiswal, D. (2019). The future digital workforce: Robotic Process Automation (RPA). *Journal Of Information Systems And Technology Management*, 16, 1-17. <https://doi.org/10.4301/s1807-1775201916001>

Martelli, A. (2006). *Python in a Nutshell*. O'Reilly Media.

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

Merkel, D. (2014). Docker: lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239), 2.

Microsoft. (2023). *Security and governance in Power*. <https://learn.microsoft.com/en-us/power-platform/admin/security>

Microsoft. (2024a). *Azure Key Vault - secure key management*. <https://learn.microsoft.com/en-us/azure/key-vault/general/basic-concepts>

Microsoft. (2024b). *Best practices for securely storing API keys and credentials in python*. <https://docs.microsoft.com/en-us/azure/key-vault/general/best-practices>

Microsoft. (2024c). *Encryption in Power Platform*. <https://learn.microsoft.com/en-us/power-platform/admin/encryption>

Microsoft. (2024d). *Entornos: Descripción general*. <https://learn.microsoft.com/es-es/power-platform/admin/environments-overview>

Microsoft. (2024e). *Security roles in Power Platform - Microsoft Learn*. <https://learn.microsoft.com/en-us/power-platform/admin/security>

Microsoft. (2024f). *Virtual Private Network (VPN) Overview*. <https://learn.microsoft.com/en-us/security/zero-trust/network-security/vpn-overview>

- Microsoft. (2024g). *What is RPA (Robotic Process Automation)?* [Accessed: 2024-09-26]. <https://www.microsoft.com/en-us/solutions/robotic-process-automation>
- Najjar, A. (2023). *Microsoft Power Automate Cookbook*. O'Reilly Media, Inc.
- Narayn, H. (2023a). *Building the Modern Workplace with SharePoint Online: Solutions with SPFx, JSON Formatting, Power Automate, Power Apps, Teams, and PVA*. APRESS.
- Narayn, H. (2023b). *Building the Modern Workplace with SharePoint Online: Solutions with SPFx, JSON Formatting, Power Automate, Power Apps, Teams, and PVA*. Apress.
- Open Web Application Security Project (OWASP). (2024). Cryptographic Storage Cheat Sheet. https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html
- Oracle. (2024a). *Introduction to the Python driver for Oracle Database* [https://pythonoracledb.readthedocs.io/en/latest/user_guide/introduction.html]. Oracle.
- Oracle. (2024b). *Introduction to the Python driver for Oracle Database*. https://pythonoracledb.readthedocs.io/en/latest/user_guide/introduction.html
- pandas development team, T. (2024). *pandas documentation* [<https://pandas.pydata.org/docs/index.html>]. pandas 2.2.2 documentation.
- Pandit, A. (2024). *Exploring Low-Code and No-Code Development with Power Apps*.
- Pilgrim, M. (2004). *Dive Into Python*. Apress.
- Prinz, N., Rentrop, C., & Huber, M. (2021). Low-Code Development Platforms: A Literature Review. *AMCIS*.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.
- R Core Team. (2020). *R: A Language and Environment for Statistical Computing* [Accedido el 3 de septiembre de 2025]. R Foundation for Statistical Computing. <https://www.R-project.org/>

- R Core Team. (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. <https://www.r-project.org/>
- Rajaram, A., Olory, C., Leduc, V., Evaristo, G., Coté, K., Isenberg, J., & Fiset, P. (2022). An integrated virtual pathology education platform developed using Microsoft Power Apps and Microsoft Teams. *Journal of Pathology Informatics*, 13, 100117. <https://doi.org/10.1016/j.jpi.2022.100117>
- Rattenbury, T., et al. (2017). *Principles of Data Wrangling: Practical Techniques for Data Preparation*. O'Reilly Media.
- Real Python. (2024). How to securely store and access secrets in Python applications. <https://realpython.com/python-environment-variables/>
- Salinas, J. (2004). Innovación docente y uso de las TIC en la enseñanza universitaria. *Revista de Universidad y Sociedad del Conocimiento*, 1(1), 1-16. <https://rusc.uoc.edu/rusc/ca/index.php/rusc/article/view/v1n1-salinas.html>
- Sharma, R., & Arora, R. (2023). *Low-code development with appsmith: Building internal tools and business applications* (1st). Apress.
- Simon, P. (2022). *Low-Code/No-Code: Citizen Developers and the Surprising Future of Business Applications*. Racket Publishing.
- Stack Overflow. (2024). Running Python Scripts in Microsoft Power Automate Cloud [Accessed: 2024-09-27]. <https://stackoverflow.com/questions/77751515/running-python-scripts-in-microsoft-power-automate-cloud>
- Syed, S., Ahmed, A., & Iqbal, M. (2020). Towards a Method for Automated Testing in Robotic Process Automation Projects. *14th International Workshop on Automation of Software Test (AST)*, 42-47. <https://doi.org/10.1109/AST.2019.00012>

- tapanm-MSFT. (2024). *Documentación oficial de Microsoft Power Apps - Power Apps*. <https://learn.microsoft.com/es-es/power-apps/>
- Team, P. (2024). *Pywinauto Documentation* [Accessed: 2024-09-26]. <https://pywinauto.readthedocs.io/en/latest/>
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83. <https://doi.org/10.1109/MIC.2010.145>
- Van Der Aalst, W., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, 60(4), 269-272. <https://doi.org/10.1007/s12599-018-0542-4>
- van Rossum, G. (2001). *Python Reference Manual* [Release 2.1]. Python Software Foundation.
- Vartiainen, H. (2024a). Lean Management Empowerment: Elevating Employee Experience through RPA Implementation with Microsoft Power Automate.
- Vartiainen, H. (2024b). *Lean Management Empowerment: Elevating Employee Experience through RPA Implementation with Microsoft Power Automate*.
- Xerox. (2023). *Robotic Process Automation with Power Automate | How we built it: Xerox* [Accessed: 2024-09-26]. <https://techcommunity.microsoft.com/t5/roboprocessautomation/rpa-xerox/ba-p/3653921>