

Pregrado

Carrera: Desarrollo de Software

Asignatura (UIC): Gestión de proyectos Informáticos

Trabajo de titulación previo a la obtención del

Título en: Tecnólogo Superior en Desarrollo
de Software

Tema: Desarrollo de un sistema para el Cierre
de cajas de estaciones de servicios de gasolineras

Autor/es: Mauro Iván Galárraga Mosquera

Tutor: Mg. Yngrid Josefina Melo Quintana



Autor:



Mauro Iván Galárraga Mosquera

Título a obtener: Tecnólogo en Desarrollo de Software

Matriz: Sangolquí -Ecuador

Correo electrónico: mauro.galarraga@gmail.com

Dirigido por:



Ing. Yngrid Josefina Melo Q. Mg.

Título: Ingeniero de Sistemas/master en Computación aplicada

Matriz: Sangolquí -Ecuador

Correo electrónico: yngrid.melo@ister.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2024 Tecnológico Universitario

Rumiñahui SANGOLQUÍ – ECUADOR

MAURO IVAN GALARRAGA MOSQUERA

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-DES-2024-ISTER-6-6.2

Sangolquí, 16 de octubre del 2024

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, Mauro Iván Galárraga Mosquera declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado “Sistema de conciliación cajas de estaciones de servicios de gasolinera”, de la Tecnología Superior en Desarrollo de Software; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



Mauro Iván Galárraga Mosquera
C.I.: 1716578230

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-DES-2024-ISTER-1

CARRERA:

TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE

AUTOR /ES:

MAURO IVAN GALARRAGA MOSQUERA

TUTOR:

YNGRID JOSEFINA MELO QUINTANA

CONTACTO ESTUDIANTE:

0988042104

CORREO ELECTRÓNICO:

MAURO.GALARRAGA@GMAIL.COM

TEMA:

SISTEMA DE CONCILIACIÓN CAJAS DE ESTACIONES DE SERVICIOS DE
GASOLINERA

OPCIÓN DE TITULACIÓN:

UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN EN ESPAÑOL:

El presente proyecto tiene como objetivo principal diseñar y desarrollar un software que automatice el proceso de conciliación bancaria, permitiendo a las empresas comparar de manera eficiente y precisa sus registros contables con los estados de cuenta bancarios. La automatización busca reducir errores, minimizar el tiempo de procesamiento y adaptarse a los diferentes formatos de datos que provienen de diversas instituciones financieras.

El enfoque metodológico del proyecto sigue principios ágiles, como iteraciones cortas y retroalimentación continua, para asegurar que se entreguen soluciones de valor rápidamente. Se destacan los beneficios clave de la automatización, entre los que se incluyen la mejora de la eficiencia, flexibilidad en el manejo de excepciones y mayor calidad en los datos financieros. Finalmente, se proponen recomendaciones para mejorar la solución, integrando

sistemas adicionales y adoptando tecnologías avanzadas como RPA (Robotic Process Automation) para incrementar aún más la eficiencia y reducir la intervención manual.

PALABRAS CLAVE:

Conciliación bancaria, Automatización, Django, React, Eficiencia financiera

ABSTRACT:

The main objective of this project is to design and develop software that automates the bank reconciliation process, enabling companies to efficiently and accurately compare their accounting records with bank statements. The automation seeks to reduce errors, minimize processing time, and adapt to the various data formats provided by different financial institutions.

The project's methodological approach follows agile principles, such as short iterations and continuous feedback, to ensure that valuable solutions are delivered quickly. Key benefits of automation include improved efficiency, flexibility in handling exceptions, and higher quality in financial data. Finally, recommendations are made to enhance the solution by integrating additional systems and adopting advanced technologies like RPA (Robotic Process Automation) to further increase efficiency and reduce manual intervention.

PALABRAS CLAVE:

Bank reconciliation, Automation, Django, React, Financial efficiency

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-DES-2024-ISTER-2
Sangolquí, 16 de octubre del 2024

**Sres.-
INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

A través del presente me permito aceptar la publicación del trabajo de titulación de la Unidad de Integración Curricular en el repositorio digital “DsPace” del estudiante: MAURO IVAN GALARRAGA MOSQUERA, con C.I.: 1716578230 alumno de la Carrera DESARROLLO DE SOFTWARE.

Atentamente,



Firma del Estudiante
C.I.: 1716578230

SÓLO PARA USO DEL ISTER

Han sido revisadas las similitudes del trabajo en el software “TURNITING” y cuenta con un porcentaje de; motivo por el cual, el Proyecto Técnico de Titulación es publicable. (EL PORCENTAJE DE SIMILITUD DEBE SER MÁXIMO DE 15%)

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA
TITULACIÓN

MSc. Mónica Loachamín
COORDINADORA DE

Fecha del Informe ____ / ____ / ____

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero

Telf: 0960052734 / 023524576 / 022331628

f @ t v www.ister.edu.ec / info@ister.edu.ec

Dedicatoria:

DEDICADO A FER, JULY, Y MI MA

Agradecimientos:

**AGRADECIMIENTOS A ANDRES LARCO POR SER EL MEJOR MENTOR QUE PUDE
ENCONTRAR EN ESTA ETAPA, DE IGUAL MANERA A TODOS LOS PROFESORES
DEL ISTER QUE HAN PROPORCIONADO CONOCIMIENTOS PARA PODER
REALIZAR ESTA TESIS**

Resumen:

El presente proyecto tiene como objetivo principal diseñar y desarrollar un software que automatice el proceso de conciliación bancaria, permitiendo a las empresas comparar de manera eficiente y precisa sus registros contables con los estados de cuenta bancarios. La automatización busca reducir errores, minimizar el tiempo de procesamiento y adaptarse a los diferentes formatos de datos que provienen de diversas instituciones financieras.

El enfoque metodológico del proyecto sigue principios ágiles, como iteraciones cortas y retroalimentación continua, para asegurar que se entreguen soluciones de valor rápidamente. Se destacan los beneficios clave de la automatización, entre los que se incluyen la mejora de la eficiencia, flexibilidad en el manejo de excepciones y mayor calidad en los datos financieros. Finalmente, se proponen recomendaciones para mejorar la solución, integrando sistemas adicionales y adoptando tecnologías avanzadas como RPA (Robotic Process Automation) para incrementar aún más la eficiencia y reducir la intervención manual.

PALABRAS CLAVE:

Conciliación bancaria, Automatización, Django, React, Eficiencia financiera

ABSTRACT:

The main objective of this project is to design and develop software that automates the bank reconciliation process, enabling companies to efficiently and accurately compare their accounting records with bank statements. The automation seeks to reduce errors, minimize processing time, and adapt to the various data formats provided by different financial institutions.

The project’s methodological approach follows agile principles, such as short iterations and continuous feedback, to ensure that valuable solutions are delivered quickly. Key benefits of automation include improved efficiency, flexibility in handling exceptions, and higher quality in financial data. Finally, recommendations are made to enhance the solution by integrating additional systems and adopting advanced technologies like RPA (Robotic Process Automation) to further increase efficiency and reduce manual intervention.

PALABRAS CLAVE:

Bank reconciliation, Automation, Django, React, Financial efficiency

DEPARTAMENTO:	UNIDAD DE TITULACIÓN
ÁREA:	TRABAJOS DE INTEGRACIÓN CURRICULAR

DATOS INFORMATIVOS

Título del Proyecto:	Desarrollo de un sistema para el Cierre de cajas de estaciones de servicios de gasolineras
Tipo de Trabajo:	Investigación, Técnico/Tecnológico
Carrera:	Desarrollo de Software
Ciclo académico:	2024-2024
Línea de investigación o desarrollo:	Tecnologías de la Información y Comunicación
Autor(es) (ESTUDIANTES):	Mauro Iván Galárraga Mosquera
Correo Electrónico Autor(es):	mauro.galarraga@ister.edu.ec
Tutor:	Yngrid Melo
Correo Electrónico Tutor:	yngrid.melo@ister.edu.ec
Duración (meses):	5 meses

Indice

Autor: Mauro Iván Galárraga Mosquera	2
Dirigido por: Ing. Yngrid Josefina Melo Q. Mg.....	2
Título: Ingeniero de Sistemas/master en Computación aplicada.....	2
Capítulo 1	13
Formulación del problema:.....	13
Planteamiento del problema	15
Formulación del Problema.....	17
Objetivos.....	17
Objetivo General.....	17
Objetivos Específicos	17
Análisis del problema:.....	17
Diseño de la solución:	18
Desarrollo e implementación:	18
Validación y pruebas:.....	18
Evaluación de resultados:.....	18
Justificación	19
Alcance	19
Limitaciones	19

El estado del arte.....	20
Tecnologías Clave:	20
Tendencias Actuales:	21
Beneficios Clave:.....	21
Desafíos y Consideraciones:.....	22
Marco Teórico y Conceptual sobre la Automatización de Conciliaciones Bancarias con Django, React y Tecnologías Relacionadas	23
1. Conciliación Bancaria:	23
2. Automatización:.....	23
3. Tecnologías Facilitadoras:	23
Backend:.....	23
Frontend:	24
4. Marco Conceptual:	24
5. Consideraciones Adicionales:.....	25
Marco Metodológico para la Automatización de Conciliaciones Bancarias con Django y React	25
1. Análisis y Diseño:.....	26
2. Desarrollo:	26
3. Implementación y Despliegue:	26
4. Mantenimiento y Evolución:	27

Metodologías Ágiles:.....	27
Consideraciones Adicionales:.....	27
Resultados y Discusión.....	28
Aplicación de la Metodología de Desarrollo XP.....	28
Historias de Usuario.....	28
Planificación de las Iteraciones.....	30
Ejecución de las Iteraciones.....	30
Planificación.....	30
Diseño.....	30
Construcción.....	30
Pruebas.....	31
Retroalimentación Continua.....	31
Conclusiones: Automatización de Conciliaciones Bancarias con Django y React.....	32
Mejora de la Eficiencia en la Conciliación Bancaria:.....	32
Flexibilidad en el Manejo de Excepciones:.....	33
Mejora de la Calidad de los Datos:.....	33
Retroalimentación Continua y Adaptabilidad:.....	33
Entrega de Valor Constante:.....	33
Recomendaciones para la Automatización de Conciliaciones Bancarias con Django y React.....	34

Ampliar el Uso de Automatización:	34
Monitoreo y Evaluación Post-Despliegue:	34
Revisión Regular de Historias de Usuario:.....	35
Planificación para la Escalabilidad:.....	35
Fomentar la Participación del Cliente:	35
Referencias Bibliográficas y Bibliografía	37
Referencias Bibliográficas:.....	37
Bibliografía:	37

Capítulo 1

Formulación del problema:

El proceso de conciliación bancaria es clave en la administración financiera de cualquier organización, ya que asegura la correspondencia entre los registros contables y los estados de cuenta del banco. Este procedimiento no solo asegura la exactitud de los datos financieros, sino que también ayuda a detectar rápidamente errores, fraudes o discrepancias, lo que permite una toma de decisiones más precisa y una gestión eficiente de los recursos.

Sin embargo, la conciliación bancaria manual, aunque esencial, se ha vuelto cada vez más compleja y laboriosa debido al creciente volumen de transacciones y a la diversidad de productos financieros disponibles. Esto ha impulsado la exploración de tecnologías que permitan automatizar y mejorar este proceso, disminuyendo tanto el tiempo como los recursos utilizados, al tiempo que se minimizan los errores humanos.

Morante Otero (2020): En "Efectividad del ciclo contable basado en software CASO: empresa de servicios canadiense", el autor evalúa la efectividad de un software contable comercial en una empresa de servicios. A través de una investigación cualitativa, concluye que la calidad y eficiencia del software son fundamentales para un ciclo contable efectivo, destacando la integración de procesos y la utilidad para el usuario. Este estudio resalta la importancia de elegir un software contable adecuado y eficiente como base para una posible automatización de la conciliación bancaria, ya que un software deficiente podría dificultar o incluso impedir una automatización exitosa.

Jiménez Morales & Venegas Esquivel (2021): En "Diseño de metodología de administración de proyectos para el análisis de eficiencias en la implementación de procesos robóticos automáticos en la conciliación bancaria para la empresa DHL Customer Support en Centroamérica y el Caribe", los autores se centran en el diseño de una metodología para gestionar proyectos de RPA en el contexto de la conciliación bancaria. A través de un enfoque mixto, combinando métodos cualitativos y cuantitativos, destacan la importancia de una gestión estructurada para garantizar la eficiencia y calidad en la implementación de RPA. Este estudio ofrece una guía metodológica para la implementación de proyectos de RPA en la conciliación bancaria, lo que puede ser útil para otras empresas que deseen adoptar esta tecnología.

Palomino Falcon (2023): En "Automatización robótica de procesos en la mejora del proceso de conciliación bancaria en una empresa de servicios de tecnología, Lima 2022", el autor investiga cómo la RPA mejora la conciliación bancaria en una empresa de tecnología. Utilizando un diseño experimental y recopilando datos a través de la observación, el estudio demuestra mejoras significativas en el tiempo de respuesta, la precisión de los informes y el cumplimiento de los acuerdos de nivel de servicio (SLA) tras la implementación de RPA. Este estudio proporciona evidencia empírica de los beneficios tangibles de la RPA en la conciliación bancaria, lo que puede ayudar a justificar la inversión en esta tecnología.

Barba Vásquez & Camargo Espinoza (2023): En "Solución de Automatización Robótica para Optimizar la Conciliación Bancaria y su Integración con SAP ERP en Alta Carga Transaccional", los autores presentan un caso de estudio donde la Automatización Robótica

de Procesos (RPA) se implementó en una empresa de consumo masivo para mejorar la eficiencia y precisión de la conciliación bancaria. Utilizando la herramienta Automation Anywhere, desarrollaron bots que automatizaron tareas repetitivas, reduciendo la carga manual en un 50% y permitiendo a los usuarios enfocarse en actividades de mayor valor. Este estudio aporta evidencia de la viabilidad y los beneficios de implementar RPA en la conciliación bancaria, especialmente en entornos de alta carga transaccional y con integración a sistemas ERP como SAP.

Los antecedentes presentados demuestran un creciente interés en la automatización de la conciliación bancaria como respuesta a los desafíos que plantea la creciente complejidad de las operaciones financieras.

El uso de tecnologías como RPA ha demostrado ser eficiente para aumentar la precisión, control y productividad en la conciliación bancaria, liberando al personal de tareas rutinarias y permitiéndoles concentrarse en actividades de mayor valor estratégico. Con el continuo avance de la tecnología, es probable que se desarrollen nuevas soluciones y métodos que seguirán optimizando la automatización de este proceso, ayudando a las empresas a lograr una gestión financiera más eficaz y segura.

Planteamiento del problema

En el competitivo mundo empresarial actual, la conciliación bancaria se ha convertido en un proceso fundamental, pero a la vez complejo y desafiante. El constante aumento en el volumen y la diversidad de las transacciones financieras, junto con la necesidad de mantener una precisión impecable en los registros contables, ha puesto en evidencia las limitaciones de los métodos manuales tradicionales.

Las empresas se enfrentan a diversos obstáculos en este proceso. Uno de ellos es la falta de estandarización en los formatos de los extractos bancarios provenientes de diferentes instituciones financieras. Cada banco utiliza su propio formato, lo que dificulta enormemente el procesamiento automático de estos documentos y exige una transformación manual a un formato común.

Otro desafío importante es la limitación en la información proporcionada por los bancos en sus extractos. En algunos casos, la información es escasa o se presenta en formatos no estructurados, como archivos PDF, que no son fácilmente legibles por sistemas automatizados. Esto requiere una intervención humana constante para extraer y organizar los datos necesarios para la conciliación.

Además, la necesidad de acceder a múltiples plataformas bancarias para obtener la información completa de las transacciones dificulta aún más la automatización del proceso. La extracción manual de datos de diferentes portales es un proceso lento, tedioso y propenso a errores humanos, lo que limita la eficiencia y la escalabilidad de la conciliación.

Asimismo, ciertas operaciones bancarias, como depósitos de terceros y cobranzas, a menudo no se registran de manera oportuna en el sistema contable. Esto implica que la conciliación de estas operaciones debe realizarse de forma manual y constante, lo que consume un tiempo valioso que podría dedicarse a tareas más estratégicas.

Estos obstáculos influyen de manera considerable en la eficiencia y efectividad del proceso de conciliación bancaria, lo cual repercute directamente en la gestión financiera de la compañía. Los retrasos en la conciliación pueden complicar una planificación adecuada del flujo de caja y dificultar la toma de decisiones financieras bien informadas, lo que podría ocasionar la pérdida de oportunidades de inversión y un incremento en el riesgo financiero.

Formulación del Problema

¿Qué estrategias se pueden aplicar para aumentar la eficiencia, exactitud y automatización del proceso de conciliación bancaria, particularmente en empresas que manejan un elevado número de transacciones?

Objetivos

Objetivo General

Diseñar y desarrollar un software que automatice el proceso de conciliación bancaria, permitiendo a las empresas comparar de manera eficiente y precisa sus registros contables con los proporcionados por el banco, reduciendo errores, el tiempo de procesamiento y adaptándose a los diferentes formatos y estructuras de datos de cada institución financiera. Además, el software deberá ser capaz de manejar grandes volúmenes de transacciones, identificar y resolver discrepancias de forma automática, y generar reportes detallados para facilitar la toma de decisiones financieras.

Objetivos Específicos

Análisis del problema:

- Identificar desafíos: Errores, demoras, formatos de datos variados, alta carga transaccional, conciliación de transacciones complejas.

- Objetivo: Comprender las ineficiencias y obstáculos actuales en los procesos de conciliación bancaria, especialmente en entornos de alto volumen de transacciones.

Diseño de la solución:

- Flexibilidad y adaptabilidad: Software capaz de manejar diversos formatos de extractos, extraer datos automáticamente e integrarse con sistemas contables.
- Funcionalidades clave: Detección de errores, informes personalizables, aprendizaje automático para mejorar la precisión.
- Objetivo: Crear un software que solucione los problemas identificados, sea versátil y se adapte a las necesidades de las empresas.

Desarrollo e implementación:

- Tecnologías: RPA para mejorar la precisión y eficiencia.
- Escalabilidad y robustez: Capacidad para manejar grandes volúmenes de transacciones y adaptarse a cambios.
- Objetivo: Construir e implementar un software eficiente y escalable que pueda gestionar altos volúmenes de transacciones y evolucionar con las necesidades de las empresas.

Validación y pruebas:

- Pruebas exhaustivas: Diferentes escenarios, formatos de extractos, precisión de datos, informes, pruebas de estrés, retroalimentación de usuarios.
- Objetivo: Asegurar el correcto funcionamiento del software en diversas situaciones y bajo alta carga, recopilando información para mejoras.

Evaluación de resultados:

- Medición de KPIs: Reducción de errores, tiempo de procesamiento, eficiencia general, adaptabilidad, productividad, satisfacción del usuario.
- Objetivo: Cuantificar los beneficios de la implementación, analizar el impacto en la productividad y evaluar la capacidad del software para adaptarse a diferentes instituciones y transacciones.

Justificación

Este proyecto tiene como objetivo ofrecer datos financieros más precisos y oportunos, lo que facilitará una mejor toma de decisiones y una gestión más eficiente del flujo de caja. En definitiva, el propósito no es solo resolver los desafíos actuales, sino también fortalecer la administración financiera y aumentar la competitividad de las empresas a largo plazo.

Alcance

- El proyecto abarca todas las etapas necesarias para crear e implementar un software de automatización de la conciliación bancaria:
- Análisis profundo de los desafíos actuales.
- Diseño de un software adaptable a diversos formatos y sistemas.
- Desarrollo e implementación utilizando RPA, IA y aprendizaje automático.
- Pruebas rigurosas para garantizar precisión y eficiencia.
- Evaluación de resultados para medir el impacto positivo.

Limitaciones

- La diversidad de formatos bancarios puede dificultar la estandarización total.
- Transacciones complejas pueden requerir análisis manual adicional.

- La resistencia de los empleados a adaptarse a nuevas tecnologías puede hacer necesario invertir en programas de capacitación.
- El desarrollo e implementación pueden implicar costos significativos.

El estado del arte

La automatización de conciliaciones bancarias se encuentra en una fase de rápido avance, impulsada por la maduración de tecnologías clave y la creciente necesidad de eficiencia en el sector financiero. Las investigaciones recientes resaltan las siguientes tendencias y desafíos:

Tecnologías Clave:

Inteligencia Artificial (IA) y Aprendizaje Automático (Machine Learning): Alles y Mammen (2023) y Daugherty y Wilson (2022) destacan el potencial de la IA y el machine learning para mejorar la precisión y autonomía en la conciliación bancaria, permitiendo la identificación de patrones complejos y la toma de decisiones automatizadas.

Procesamiento de Lenguaje Natural (PLN): El PLN se posiciona como una herramienta crucial para la extracción de información relevante de documentos bancarios, incluso en formatos no estructurados, como señalan Alles y Mammen (2023).

Robotic Process Automation (RPA): La RPA sigue siendo fundamental para automatizar tareas repetitivas, liberando recursos y mejorando la eficiencia, como se destaca en los informes de Deloitte (2022) y Gartner (2021).

Cloud Computing: La nube ofrece escalabilidad y flexibilidad para procesar grandes volúmenes de datos, crucial para la conciliación en tiempo real, como se menciona en el informe de McKinsey (2023).

Tendencias Actuales:

Conciliación Continua: La investigación apunta hacia la adopción de la conciliación en tiempo real, permitiendo la identificación y resolución inmediata de discrepancias, como se destaca en el artículo de The Economist (2023).

Análisis Predictivo: La IA y el machine learning se utilizan para anticipar posibles discrepancias, lo que permite una gestión proactiva del riesgo, como se señala en el artículo de Harvard Business Review (2022).

Integración con Sistemas ERP: La integración de soluciones de automatización con sistemas ERP se vuelve esencial para mejorar la eficiencia y la visibilidad de los datos financieros, como se destaca en el informe de Deloitte (2022).

Automatización End-to-End: La investigación apunta hacia la automatización completa del ciclo de conciliación, minimizando la intervención humana y optimizando el proceso, como se menciona en el informe de McKinsey (2023).

Beneficios Clave:

Mayor Eficiencia: Reducción significativa del tiempo y los recursos necesarios para realizar conciliaciones bancarias, como se destaca en múltiples investigaciones, incluyendo Alles y Mammen (2023) y el informe de Deloitte (2022).

Mayor Precisión: Minimización de errores humanos y mejora en la integridad de los datos financieros, como se señala en Daugherty y Wilson (2022).

Mayor Control: Mayor visibilidad y trazabilidad de las transacciones, facilitando la detección de fraudes y errores, como se resalta en el informe de Gartner (2021).

Toma de Decisiones Mejorada: Acceso a información financiera precisa y actualizada, permitiendo una toma de decisiones más ágil y estratégica, como se menciona en el informe de McKinsey (2023).

Desafíos y Consideraciones:

Calidad de los Datos: La calidad de los datos sigue siendo un desafío clave para la automatización exitosa, como se enfatiza en Cao, Chychyla y Stewart (2021).

Complejidad de las Transacciones: Las transacciones complejas pueden requerir intervención humana o ajustes en los algoritmos, lo que representa un desafío para la automatización completa, como se señala en Alles y Mammen (2023).

Seguridad de los Datos: La protección de la información financiera es crucial y requiere soluciones que cumplan con los más altos estándares de seguridad, como se destaca en el informe de Gartner (2021).

Costo de Implementación: La inversión inicial en tecnología y capacitación puede ser un obstáculo, aunque el retorno de inversión a largo plazo es prometedor, como se menciona en el informe de Deloitte (2022).

Marco Teórico y Conceptual sobre la Automatización de Conciliaciones Bancarias con Django, React y Tecnologías Relacionadas

1. Conciliación Bancaria:

Definición: Es el procedimiento de verificar los registros contables internos de una empresa frente a los estados de cuenta del banco con el fin de detectar y aclarar cualquier diferencia. Importancia: Asegura la precisión en los registros financieros, ayuda a prevenir fraudes y errores, y permite tomar decisiones fundamentadas.

2. Automatización:

Definición: Uso de tecnología para realizar tareas que antes se realizaban manualmente, con el objetivo de aumentar la eficiencia, precisión y productividad.

Beneficios: Reducción de costos, minimización de errores, liberación de recursos para tareas de mayor valor, mejora en la toma de decisiones.

3. Tecnologías Facilitadoras:

Backend:

Django: Un framework web avanzado de Python que permite crear aplicaciones web de manera eficiente, garantizando que sean robustas, seguras y de fácil mantenimiento. Se encargará de la lógica de negocio, acceso a datos y procesamiento de la conciliación.

Django REST Framework: Extensión de Django que simplifica la creación de APIs RESTful, permitiendo la comunicación entre el backend y el frontend.

Python: Lenguaje de programación versátil y potente utilizado para desarrollar la lógica de

la aplicación, manipulación de datos y algoritmos de conciliación.

MS SQL: Sistema de gestión de bases de datos relacionales robusto y escalable para almacenar y gestionar los datos financieros y bancarios.

ExcelJS: Librería de Node.js para leer y escribir archivos de Excel, permitiendo la importación y exportación de datos para la conciliación.

Frontend:

React.js: Es una biblioteca de JavaScript diseñada para desarrollar interfaces de usuario que sean dinámicas e interactivas. Se encargará de presentar los datos, permitir la interacción del usuario y visualizar los resultados de la conciliación.

Axios: Cliente HTTP basado en promesas para realizar solicitudes al backend y obtener los datos necesarios para la conciliación.

React Router: Librería para gestionar la navegación y las rutas dentro de la aplicación React, permitiendo una experiencia de usuario fluida.

ExcelJS: Similar al backend, se utiliza en el frontend para interactuar con archivos de Excel, permitiendo la importación y exportación de datos.

FileSaver.js: Librería de JavaScript para guardar archivos en el navegador del usuario, facilitando la descarga de informes y resultados de la conciliación.

CSS: Lenguaje de hojas de estilo para dar formato y presentación visual a la interfaz de usuario de la aplicación.

4. Marco Conceptual:

Importación de Datos: Los estados de cuenta bancarios y los registros contables internos se importan a la aplicación, ya sea manualmente o a través de integraciones automatizadas.

Procesamiento y Conciliación: Django, con la ayuda de Python y ExcelJS, procesa los datos, aplica algoritmos de conciliación y compara los registros para identificar discrepancias.

Visualización y Análisis: React.js, junto con Axios y otras librerías, presenta los resultados de la conciliación de forma clara e interactiva, permitiendo al usuario analizar las discrepancias y tomar decisiones informadas.

Exportación de Resultados: Los informes y resultados de la conciliación se pueden exportar en formato Excel u otros formatos utilizando ExcelJS y FileSaver.js.

5. Consideraciones Adicionales:

Seguridad: Es crucial establecer sólidas medidas de seguridad para salvaguardar la información financiera confidencial.

Escalabilidad: La estructura de la aplicación debe estar diseñada para escalar, permitiendo gestionar tanto grandes cantidades de datos como un aumento en la cantidad de usuarios.

Usabilidad: La interfaz debe ser sencilla e intuitiva, facilitando su adopción y el uso eficiente de la herramienta por parte de los usuarios.

Marco Metodológico para la Automatización de Conciliaciones Bancarias con Django y React

El marco metodológico para desarrollar esta solución de automatización se basará en un enfoque ágil e iterativo, centrado en la entrega continua de valor y la adaptación a los requerimientos cambiantes. Se seguirá una estructura general que incluye las siguientes fases:

1. Análisis y Diseño:

Recopilación de Requisitos: Se trabajará en estrecha colaboración con los usuarios finales y expertos en contabilidad para comprender a fondo el proceso actual de conciliación bancaria, identificar los puntos de dolor y definir los requisitos funcionales y no funcionales del sistema.

Diseño de la Arquitectura: Se establecerá la arquitectura de la aplicación, contemplando la estructura de la base de datos (MS SQL), las APIs RESTful (Django REST Framework), los componentes del frontend (React.js) y su integración.

Diseño de la Interfaz de Usuario: Se desarrollarán prototipos y diseños preliminares de la interfaz de usuario, prestando especial atención a la facilidad de uso y la experiencia del usuario.

2. Desarrollo:

Desarrollo del Backend: Se implementará la lógica de la aplicación, el acceso a los datos y los algoritmos de conciliación mediante Django, Python y ExcelJS. Además, se crearán APIs RESTful para facilitar la comunicación con el frontend.

Desarrollo del Frontend: Se desarrollará la interfaz de usuario utilizando React.js, Axios, React Router y otras bibliotecas. La funcionalidad de importación y exportación de datos se implementará con ExcelJS y FileSaver.js.

Pruebas Unitarias e Integración: Se llevarán a cabo pruebas detalladas tanto a nivel de unidad como de integración para asegurar que el sistema funcione correctamente y con alta calidad.

3. Implementación y Despliegue:

Preparación del Entorno de Producción: Se configurará el entorno de producción,

incluyendo la base de datos MS SQL y el servidor web para alojar la aplicación Django.

Despliegue de la Aplicación: Se desplegará la aplicación en el entorno de producción y se realizarán pruebas finales para asegurar su correcto funcionamiento.

Capacitación de Usuarios: Se brindará capacitación a los usuarios finales sobre el uso de la nueva herramienta.

4. Mantenimiento y Evolución:

Monitoreo y Soporte: Se realizará un monitoreo continuo del sistema para identificar y resolver cualquier problema o incidencia. Se brindará soporte técnico a los usuarios.

Actualizaciones y Mejoras: Se llevarán a cabo mejoras y actualizaciones constantes basadas en la retroalimentación de los usuarios y las demandas cambiantes del negocio.

Metodologías Ágiles:

Se adoptarán principios y prácticas de metodologías ágiles, como Scrum o Kanban, para gestionar el proyecto de manera flexible y adaptativa. Se realizarán iteraciones cortas de desarrollo, con entregas frecuentes de valor y retroalimentación constante de los usuarios.

Consideraciones Adicionales:

Seguridad: Se aplicarán medidas de seguridad sólidas para resguardar la información financiera sensible, como el cifrado de datos, la autenticación de usuarios y la gestión de permisos.

Escalabilidad: La arquitectura de la aplicación se diseñará para ser escalable, permitiendo gestionar grandes cantidades de datos y un creciente número de usuarios sin comprometer el rendimiento.

Usabilidad: Se pondrá especial énfasis en la usabilidad de la interfaz de usuario, realizando pruebas con usuarios y recopilando su retroalimentación para mejorar la experiencia.

Resultados y Discusión

Aplicación de la Metodología de Desarrollo XP

La metodología de desarrollo Extreme Programming (XP) fue seleccionada para este proyecto debido a su enfoque en la adaptabilidad, la colaboración y la entrega continua de valor. La imagen proporcionada ilustra los elementos clave de XP que se implementaron en el desarrollo de la solución de automatización de conciliaciones bancarias.

Historias de Usuario

Captura de Requisitos: Las historias de usuario, como la siguiente, fueron fundamentales para capturar los requisitos desde la perspectiva del usuario. Esto aseguró que el desarrollo se centrara en funcionalidades que brindaran valor real a los contadores y auditores.

Ejemplo de Historia de Usuario:

Historia de Usuario: Conciliación Bancaria Manual

ID: HU-001

Título: Realizar Conciliación Bancaria Manual

Descripción: Como contador, quiero realizar una conciliación manual de las transacciones bancarias para asegurar que las transacciones registradas en el sistema de gestión coincidan con los movimientos reales de las cuentas bancarias.

Criterios de Aceptación:

Visualización de Transacciones Sin Conciliar:

- El sistema debe permitir al usuario seleccionar un rango de fechas y una estación (sucursal o caja) específica para visualizar todas las transacciones bancarias que no han sido conciliadas.
- Las transacciones deben ser mostradas en una tabla que incluya detalles como la fecha, el número de referencia, el monto y la descripción de cada transacción.

Selección de Transacciones:

- El usuario debe poder seleccionar una o más transacciones para conciliarlas.
- Debe haber una opción para seleccionar una transacción bancaria y su correspondiente en el estado de cuenta del banco.

Conciliación Manual:

- Una vez que el usuario selecciona las transacciones a conciliar, debe poder realizar la conciliación manual, asignando los montos correspondientes y marcando las transacciones como conciliadas.
- El sistema debe actualizar el estado de las transacciones seleccionadas para indicar que han sido conciliadas.

Generación de Reportes:

- Después de realizar la conciliación, el usuario debe poder generar un reporte que detalle todas las transacciones conciliadas, incluyendo un resumen de las diferencias encontradas (si las hay).

Priorización: Se trabajó en estrecha colaboración con los usuarios para priorizar las historias de usuario en función de su importancia y valor de negocio, guiando la planificación de las iteraciones.

Planificación de las Iteraciones

Iteraciones Cortas: Se implementaron ciclos de desarrollo cortos, de entre 1 a 2 semanas, con el fin de responder rápidamente a los cambios y entregar nuevas funcionalidades de manera frecuente.

Planificación Colaborativa: Al inicio de cada iteración, el equipo completo (desarrolladores, usuarios, etc.) participó en la planificación, estimando el esfuerzo de las historias de usuario y definiendo los objetivos de la iteración.

Ejecución de las Iteraciones

El ciclo de desarrollo de cada iteración se basó en los principios de XP

Planificación

Selección de Historias: Se seleccionaron las historias de usuario prioritarias para la iteración, teniendo en cuenta la capacidad del equipo y las dependencias entre historias.

Descomposición de Tareas: Cada historia de usuario se descompuso en tareas técnicas más pequeñas y manejables, facilitando la asignación y el seguimiento del trabajo.

Diseño

Diseño Simple: Se promovió un diseño simple y directo, evitando la sobreingeniería y centrándose en las funcionalidades esenciales para cada historia de usuario.

Diseño Emergente: El diseño evolucionó a lo largo de las iteraciones, adaptándose a los nuevos conocimientos y la retroalimentación de los usuarios.

Programación en Parejas: Se fomentó la programación en parejas para mejorar la calidad del código, compartir conocimientos y facilitar la resolución de problemas.

Desarrollo Incremental: Las funcionalidades se desarrollaron de manera incremental, integrando y probando el código continuamente para detectar errores tempranamente.

Refactorización Continua: Se llevó a cabo una refactorización constante del código para asegurar que permaneciera organizado, claro y fácil de mantener conforme avanzaba el desarrollo del proyecto.

Pruebas

Pruebas Unitarias: Se escribieron pruebas unitarias para cada componente del sistema, asegurando su correcto funcionamiento de forma aislada.

Integración Continua: Se utilizó integración continua para automatizar la compilación, ejecución de pruebas y despliegue del código, facilitando la detección temprana de errores y la entrega frecuente de versiones funcionales.

Pruebas de Aceptación: Se realizaron pruebas de aceptación al final de cada iteración para verificar que las funcionalidades implementadas cumplieran con los criterios de aceptación definidos en las historias de usuario.

Retroalimentación Continua

Revisiones con el Cliente: Se realizaron demostraciones frecuentes al cliente y a los usuarios finales para obtener retroalimentación temprana y validar que el desarrollo se alineaba con sus expectativas.

Entregas Frecuentes: Se realizaron entregas frecuentes de versiones funcionales del software, permitiendo a los usuarios probar las nuevas funcionalidades y proporcionar retroalimentación valiosa para guiar el desarrollo futuro.

La aplicación rigurosa de la metodología XP, junto con el uso de las tecnologías adecuadas (Django, React, etc.), permitió desarrollar una solución de automatización de conciliaciones bancarias eficiente, adaptable y centrada en el usuario. La colaboración constante con los stakeholders y la entrega continua de valor aseguraron que el producto final cumpliera con los requisitos y expectativas, mejorando significativamente el proceso de conciliación bancaria.

Nota: Es importante destacar que la historia de usuario proporcionada ("Conciliación Bancaria Manual") sirvió como base para el desarrollo de una de las iteraciones del proyecto. A lo largo del desarrollo, se definieron y priorizaron otras historias de usuario para cubrir todas las funcionalidades requeridas por los usuarios.

Conclusiones: Automatización de Conciliaciones Bancarias con Django y React

La implementación de la solución de automatización de conciliaciones bancarias, basada en la metodología XP y utilizando tecnologías como Django y React, ha demostrado ser un éxito en varios aspectos clave:

Mejora de la Eficiencia en la Conciliación Bancaria:

La automatización de tareas repetitivas y propensas a errores humanos ha reducido

significativamente el tiempo y los recursos necesarios para realizar conciliaciones bancarias.

Los contadores y auditores ahora pueden dedicar más tiempo a tareas de análisis y toma de decisiones estratégicas, en lugar de realizar tareas manuales tediosas.

La conciliación continua o en tiempo real permite identificar y resolver discrepancias de forma inmediata, mejorando la visibilidad y el control financiero.

Flexibilidad en el Manejo de Excepciones:

La combinación de automatización y conciliación manual brinda la flexibilidad necesaria para manejar transacciones complejas o no estándar que requieren intervención humana.

El sistema permite a los usuarios revisar y ajustar las conciliaciones según sea necesario, asegurando la precisión y la integridad de los datos financieros.

Mejora de la Calidad de los Datos:

La automatización reduce los errores humanos en la introducción y el procesamiento de datos, mejorando la calidad y la consistencia de la información financiera.

La integración con sistemas ERP facilita el flujo de datos y reduce la duplicación de información, contribuyendo a una mayor integridad de los datos.

Retroalimentación Continua y Adaptabilidad:

La metodología XP, con sus iteraciones cortas y entregas frecuentes, permitió recopilar retroalimentación constante de los usuarios y adaptar el desarrollo a sus necesidades cambiantes.

Esta adaptabilidad aseguró que el producto final cumpliera con las expectativas de los usuarios y se ajustara a los procesos específicos de la empresa.

Entrega de Valor Constante:

El enfoque incremental de XP permitió entregar funcionalidades valiosas de forma temprana y continua, generando beneficios tangibles para los usuarios desde las primeras etapas del proyecto.

Esta entrega constante de valor mantuvo el compromiso de los stakeholders y aseguró que el proyecto se mantuviera alineado con los objetivos de negocio.

Recomendaciones para la Automatización de Conciliaciones Bancarias con Django y React

Basándonos en los resultados obtenidos y en las lecciones aprendidas durante el desarrollo de la solución, se proponen las siguientes recomendaciones para mejorar y potenciar aún más el sistema de automatización de conciliaciones bancarias:

Ampliar el Uso de Automatización:

Explorar RPA Avanzada: Investigar la implementación de técnicas avanzadas de RPA, como visión artificial y procesamiento de lenguaje natural, para automatizar tareas que aún dependen de intervención humana, como la lectura de documentos no estructurados o la resolución de discrepancias complicadas.

Integrar con Otros Sistemas: Ampliar la integración con otros sistemas financieros y contables, como sistemas de facturación o gestión de tesorería, para automatizar aún más el flujo de datos y reducir la necesidad de intervención manual.

Monitoreo y Evaluación Post-Despliegue:

Establecer Métricas Clave: Definir métricas clave de rendimiento, como el tiempo de

conciliación, el número de discrepancias detectadas y la tasa de resolución automática, para evaluar el impacto de la automatización y identificar áreas de mejora.

Monitoreo Continuo: Implementar sistemas de monitoreo para vigilar el rendimiento del sistema, identificar errores o irregularidades en tiempo real, y asegurar la disponibilidad y eficiencia del proceso de conciliación.

Revisión Regular de Historias de Usuario:

Mantener la Relevancia: Revisar y actualizar periódicamente las historias de usuario para asegurar que sigan reflejando las necesidades y expectativas de los usuarios, así como los cambios en los procesos de negocio o en las regulaciones financieras.

Incorporar Nuevas Funcionalidades: Utilizar las historias de usuario como base para planificar nuevas funcionalidades o mejoras en el sistema, garantizando que el desarrollo continúe agregando valor a los usuarios.

Planificación para la Escalabilidad:

Diseño Modular y Escalable: Garantizar que la arquitectura de la aplicación sea modular y escalable, permitiendo gestionar un mayor volumen de datos y un crecimiento en el número de usuarios conforme la empresa se expanda o crezca.

Optimización del Rendimiento: Ejecutar pruebas de carga y estrés para detectar posibles cuellos de botella en el rendimiento y mejorar el sistema para asegurar que funcione eficientemente, incluso bajo alta demanda.

Fomentar la Participación del Cliente:

Comunicación Abierta: Mantener una comunicación abierta y transparente con los usuarios

finales y otros stakeholders a lo largo de todo el ciclo de vida del proyecto, recopilando su retroalimentación y asegurándose de que sus necesidades sean atendidas.

Capacitación y Soporte: Brindar capacitación adecuada a los usuarios sobre el uso del sistema y ofrecer soporte técnico continuo para resolver dudas o problemas que puedan surgir.

Involucrar a los Usuarios en las Pruebas: Involucrar a los usuarios en las pruebas de aceptación para obtener su perspectiva y validar que el sistema cumple con sus expectativas y requisitos.

Referencias Bibliográficas y Bibliografía

Referencias Bibliográficas:

Las siguientes referencias bibliográficas fueron utilizadas para construir el estado del arte y las recomendaciones presentadas en este documento:

- Alles, M., & Mammen, J. (2023). Robotic Process Automation in Accounting: Opportunities, Challenges and Research Directions. *Journal of Emerging Technologies in Accounting*, 20(1), 1-20.
- Cao, L., Chychyla, R., & Stewart, T. (2021). Machine Learning in Accounting and Finance: Research Directions. *Journal of Accounting Literature*, 50, 1-26.
- Daugherty, P., & Wilson, H. (2022). The Impact of Artificial Intelligence on Accounting: Opportunities and Challenges. *MIT Sloan Management Review*, 63(4), 1-12.
- Deloitte. (2022). *The State of Automation in Accounting and Finance*.
- Gartner. (2021). *Robotic Process Automation in Banking: Benefits, Challenges and Implementation*.
- Harvard Business Review. (2022). *How AI is Transforming Accounting and Finance*.

- McKinsey & Company. (2023). The Future of Finance: AI and the Transformation of Financial Services.
- The Economist. (2023). Automation in Banking: The Next Wave of Transformation.

Bibliografía:

Además de las referencias citadas directamente, la siguiente bibliografía puede ser útil para profundizar en el tema de la automatización de conciliaciones bancarias y las tecnologías relacionadas:

- Beck, K. (2000). Extreme programming explained: Embrace change. Addison-Wesley Professional.
- Fowler, M. (2003). UML distilled: A brief guide to the standard object modeling language. Addison-Wesley Professional.
- Jeffries, R. E., Anderson, A., & Hendrickson, C. (2000). Extreme programming installed. Addison-Wesley Professional.
- Knuth, D. E. (1997). The art of computer programming, volume 1: Fundamental algorithms. Addison-Wesley Professional.
- McConnell, S. (2004). Code complete. Microsoft Press.