TECTZAPIC®
Revista Académico-Científica

Revista científica y tecnológica. ISSN: 2444-4944
Tecnológico Nacional de México/I.T. de Ciudad Valles

BaucherMatch: Herramienta de ingeniería de software como apoyo al proceso de conciliación financiera en una institución de educación superior

BaucherMatch: A software engineering tool to support the financial reconciliation process at a higher education institution.

Recibido: 11 de junio de 2026
Aceptado: 17 de agosto de 2026

Uriel González Sánchez
Instituto Tecnológico de Ciudad Valles
ORCID: 0009-0007-5161-8088
Nitgard Zapata Garay
Instituto Tecnológico de Ciudad Valles
ORCID: 0000-0002-4060-1826
Autor de correspondencia: nitgard.zg@cdvalles.tecnm.mx
Reyna Isabel Redondo González
Instituto Tecnológico de Ciudad Valles
ORCID: 0009-0007-7404-1808
Juan Manuel Banda Moreno
Instituto Tecnológico de Ciudad Valles
ORCID: 0009-0004-7204-857X

RESUMEN

La conciliación de ingresos es una función crucial para las organizaciones de educación superior, ya que vincula la gestión académica con la gestión financiera. Cuando los estudiantes se inscriben, reinscriben y en la prestación de servicios, el personal administrativo revisa grandes cantidades de transacciones bancarias, recibos y referencias estudiantiles. Cuando el personal hace esto manualmente, consume muchos días de trabajo, y el proceso puede llevar a actualizaciones tardías del estado financiero y errores en la entrada de datos. Presentamos el diseño, desarrollo y evaluación de BaucherMatch, una herramienta de escritorio que puede extraer y organizar información financiera en estados de cuenta bancarios en formato PDF. Este estudio se basa en la ingeniería de software y utiliza la lógica de investigación basada en diseño en la que creamos una tecnología para resolver un problema real y luego la validamos en el entorno operativo En el Instituto Tecnológico de Ciudad Valles se contabilizan 12,563 transacciones en 144 estados de cuenta para el año fiscal 2024. La arquitectura combina Python, FastAPI, React, TypeScript, Tauri, pdftotext y expresiones regulares parametrizadas. Los resultados muestran una precisión de más del 99.8% en la identificación de números de control, fechas y montos, y el tiempo total es de 3.46 segundos en comparación con un estimado de 158 días de trabajo manual.

Palabras clave: Ingeniería de software, automatización documental, conciliación financiera, extracción de información, PDF, expresiones regulares.

Abstract

Revenue reconciliation is a crucial function for higher education institutions, as it links academic management with financial management. During student enrollment, re-enrollment, and the provision of services, administrative staff must review large volumes of bank transactions, receipts, and student references. When performed manually, this process consumes numerous workdays and can lead to delayed financial status updates and data entry errors. We present the design, development, and evaluation of BaucherMatch, a desktop tool capable of extracting and organizing financial information from PDF bank statements. This study is grounded in software engineering and utilizes design science research methodology, wherein we create a technology to solve a real-world problem and subsequently validate it in an operational environment (Hevner et al., 2004; Peffers et al., 2007). We validated our work at the Instituto Tecnológico de Ciudad Valles, processing 12,563 transactions across 144 bank statements for the 2024 fiscal year. The architecture combines Python, FastAPI, React, TypeScript, Tauri, pdftotext, and parameterized regular expressions. The results demonstrate an accuracy of over 99.8% in identifying control numbers, dates, and amounts, with a total processing time of 3.46 seconds, compared to an estimated 158 days of manual labor.

Keywords: Software engineering, document automation, financial reconciliation, information extraction, PDF, regular expressions.

INTRODUCCIÓN

La transformación digital ha cambiado la forma en que las organizaciones registran, procesan y utilizan sus datos. Para las empresas del sector público, este cambio no siempre es a través de grandes plataformas empresariales, sino en algunos casos a través de soluciones específicas que resuelven cuellos de botella reales. La situación en las instituciones educativas es la misma cada semestre: están generando datos en sistemas académicos, bancarios y administrativos, pero aún deben pasar por revisiones manuales para cerrar algunos procesos financieros. Desde el punto de vista de los sistemas de información, el valor de la herramienta no solo está en lo sofisticada que es, sino también en la calidad, oportunidad y utilidad de los datos que ayudan en la toma de decisiones (Agarwal & Dhar, 2014; DeLone & McLean, 2003).

Por todo esto, la conciliación financiera es clave. Cada pago por cada estudiante debe realizarse con algún depósito, fecha, monto y referencia institucional. Cuando el área responsable revisa archivos PDF, recibos físicos o descripciones bancarias uno por uno, el proceso depende de la experiencia humana. La falta de una letra, una referencia deficiente o una entrada incorrecta en el formato puede bloquear la validación del pago. La literatura sobre gestión de procesos muestra que automatizar el trabajo repetitivo puede reducir el tiempo y aumentar la trazabilidad (Dumas et al., 2018; van der Aalst, 2016), dejando al personal más tiempo para analizar y monitorear el valor organizacional.

En el Instituto Tecnológico de Ciudad Valles, este era el trabajo del departamento de recursos financieros en épocas de alta actividad. El personal tenía que emparejar recibos de depósito y estados de cuenta institucionales para validar pagos relacionados con estudiantes y servicios. El problema no era la información digital, sino la falta de una herramienta para convertir la información bancaria en datos para ser revisados. Esto está en línea con lo que se ha descrito en la ingeniería de *software* como una oportunidad de diseño (Hevner et al., 2004; Wieringa, 2014): un problema operativo específico, un entorno institucional claro y la posibilidad de crear un artefacto que haga un proceso más eficiente.

Las soluciones empresariales comunes como los Programas de Recursos empresariales (ERP, por sus siglas en inglés) pueden usarse en procesos financieros complejos; pero no siempre se adaptan al mercado local. La institución pública enfrenta presiones presupuestarias, infraestructura heterogénea y reglas internas que dificultan el uso de plataformas de alto costo. En este caso, una herramienta ligera que pueda ejecutarse localmente y adaptarse a los escenarios reales de los estados de cuenta podría tener un impacto muy inmediato. La separación de funciones y las pruebas empíricas para medir el rendimiento son esenciales si el *software* se va a utilizar como una herramienta de utilidad, no solo como una herramienta, sino como una intervención de ingeniería medible (Parnas, 1972; Runeson & Höst, 2009).

Es por esto que BaucherMatch se generó como una herramienta que extrae texto de los estados de cuenta en PDF, identifica fechas, montos y números de control usando expresiones regulares, normaliza los datos y presenta los resultados en CSV o JSON para apoyar al proceso de conciliación dentro del departamento de recursos financieros. Este proyecto no es un sistema de inteligencia artificial en general, sino una solución de ingeniería de *software* basada en reglas simples y una arquitectura que permite su mantenimiento. Este enfoque está en línea con la investigación documental, donde la estructura de entrada es regular y la interpretabilidad del algoritmo es tan importante como la precisión (Thompson, 1968; Manning et al., 2008).

El objetivo del estudio es desarrollar, implementar y evaluar una herramienta de software para la extracción y estructuración de información financiera de estados de cuenta bancarios en

formato PDF con el fin de optimizar el proceso de conciliación en una institución de educación superior. Para lograr esto, el trabajo proporciona el contexto institucional, expone el enfoque de ingeniería de software, describe la arquitectura de BaucherMatch y evalúa la precisión del *software* basado en el proceso. Este enfoque permite presentar la contribución como un artefacto para su uso en el mundo real, como lo recomienda la ingeniería de *software* (Peffers et al., 2007; Wohlin et al., 2012).

MARCO TEÓRICO

Ingeniería de software aplicada a procesos administrativos

La ingeniería de *software* no solo se utiliza para construir programas, sino también para resolver problemas, diseñar soluciones, probar resultados y mantener estos sistemas en el mundo real. Este enfoque es útil en la administración para convertir tareas poco interesantes en procesos regidos por reglas, interfaces y datos verificables. La construcción de BaucherMatch se ve así (Shaw, 2003; Wieringa, 2014): un artefacto técnico diseñado para resolver un problema de operación financiera y que necesita evidencia de su efectividad.

La investigación basada en el diseño es muy apropiada para este tipo de trabajo, donde podemos conectar el conocimiento técnico con la implementación práctica. Hevner et al. (2004) argumentan que un artefacto debe evaluarse en relación con el problema que intenta resolver, mientras que Peffers et al. (2007) proponen un proceso que comienza con la identificación del problema, continúa con el diseño y termina con la comunicación de los resultados. Según esta lógica, BaucherMatch es valioso no solo por su código, sino también por el hecho de que la conciliación financiera ha mejorado significativamente (Hevner et al., 2004; Peffers et al., 2007).

La modularidad es también un elemento clave. Parnas (1972) argumentó que una buena descomposición del sistema permite entender, mantener y evolucionar el software. En BaucherMatch eso significa que podemos separar la interfaz gráfica, los servicios API, la lógica de extracción y la generación de salidas. Esta separación elimina dependencias innecesarias y permite cambiar patrones bancarios sin tener que reconstruir toda la aplicación. En proyectos pequeños, esta decisión puede marcar la diferencia entre una herramienta útil a corto plazo y una solución que la organización puede mantener a lo largo del tiempo (Parnas, 1972; Boehm, 1988).

Automatización documental y gestión de procesos

La automatización de documentos consiste en convertir datos en archivos, formularios o informes en datos que una organización puede consultar, validar y analizar. Este tipo de automatización funciona bien cuando una tarea consume muchas horas y tiene un proceso predecible. Los datos son los estados de cuenta digitales y el resultado final es en forma de algún tipo de registros tabulares con campos claros. La gestión de procesos empresariales es una excelente manera de identificar tales actividades, rediseñarlas y medir su progreso (Dumas et al., 2018; van der Aalst, 2016).

Las herramientas de automatización de procesos están ganando atención ya que permiten llevar a cabo procesos administrativos sin la necesidad de intervención humana. Syed et al. (2020) destacan que la automatización de procesos robóticos se ocupa de tareas repetitivas y basadas en reglas con entrada digital. Sin embargo, la automatización debe pensarse en términos de gobernanza, mantenimiento y control. BaucherMatch es similar en muchos aspectos, pero es un

sistema especializado que no imita acciones humanas en una interfaz externa, sino que procesa documentos bancarios para obtener datos estructurados (Syed et al., 2020; Dumas et al., 2018).

La minería de procesos es otra buena perspectiva para este trabajo. BaucherMatch no busca detectar modelos de procesos a partir de registros, sino que registra tiempos, volúmenes y resultados y puede monitorear el flujo de conciliación. van der Aalst (2016) afirma que los datos de eventos pueden usarse para monitorear comportamientos reales, identificar cuellos de botella y comparar variantes de ejecución. Más adelante, la herramienta puede ser una fuente de eventos para analizar la evolución de los procesos financieros institucionales (van der Aalst, 2016; Agarwal & Dhar, 2014).

Extracción de información en documentos PDF

Los estados de cuenta bancarios en formato PDF son una ventaja y un desafío. Por un lado, tienen datos digitales que pueden recuperarse sin transcripción. Pero pueden no proporcionar una estructura tabular clara para el *software*, especialmente si el diseño visual del documento separa los campos que el *software* necesita unir. La extracción de información resuelve precisamente este problema: localizar entidades relevantes en texto semiestructurado y convertirlas en datos reales (Manning et al., 2008; Yang et al., 2022).

Cuando el PDF contiene texto nativo, un enfoque determinista es suficiente. Cambiar a texto plano simplifica el problema y permite aplicar patrones a las cadenas de caracteres. Pero si el documento se escanea desde una pantalla, el sistema necesita reconocimiento óptico de caracteres y análisis de diseño visual. Smith (2007) muestra que los motores OCR pueden recuperar texto de imágenes, y Zhong et al. (2019) y Binmakhashen y Mahmoud (2019) explican que el análisis de diseño de documentos también requiere identificar bloques, tablas y relaciones espaciales. BaucherMatch se centra en el primer caso de PDFs digitales con texto recuperable (Smith, 2007; Zhong et al., 2019).

Esta decisión metodológica evita sobredimensionar la solución. Un modelo de aprendizaje profundo puede proporcionar flexibilidad en un documento muy diverso, pero también necesita ser etiquetado, entrenado, computado y monitoreado. En un entorno con estados de cuenta estables, las reglas son un enfoque económico, auditable y fácil de entender. La literatura sobre extracción de información reconoce que el método debe elegirse para el documento, la disponibilidad de datos y la interpretación de los datos por parte del usuario final (Manning et al., 2008; Yang et al., 2022).

Expresiones regulares y reconocimiento de patrones

Las expresiones regulares son una de las técnicas más antiguas y útiles para buscar patrones en texto. Thompson (1968) mostró cómo podían usarse para encontrar cadenas específicas sin gastar mucho tiempo y recursos; este método todavía se utiliza hoy en día en muchos lenguajes de programación. En BaucherMatch, las expresiones regulares se refieren a fechas, montos y números de control de información bancaria. El método es adecuado ya que los campos que nos interesan son formas esperadas (Thompson, 1968; Manning et al., 2008): dinero, fechas y claves institucionales.

La ventaja de este enfoque es que es interpretable en cierto sentido. Cuando se clasifica una transacción, el desarrollador puede ver exactamente qué patrón produjo la coincidencia. Esta transparencia ayuda a corregir falsos positivos, modificar la lógica a nuevos formatos y explicar cómo funciona el sistema al personal administrativo. En los procesos financieros, esta claridad

es operativa ya que los usuarios necesitan confiar en los datos antes de usarlos para validar pagos o emitir informes (DeLone & McLean, 2003; Petter et al., 2008).

METODOLOGÍA

Diseño del estudio

Para este estudio, adoptamos un enfoque tecnológico para analizar el mundo real y experimentar con los datos en tiempo real. El análisis se basó en el proceso de extracción y conciliación de las transacciones de bancos institucionales. El artefacto fue BaucherMatch, una aplicación de escritorio que transforma la información de cuentas en formato PDF hacia registros estructurados. Este diseño tiene sentido en estudios de caso en ingeniería de software donde el investigador está interesado en un fenómeno actual en el entorno de software e informa sobre su uso (Runeson & Höst, 2009; Wieringa, 2014).

El escenario de validación fue el Departamento de Recursos Financieros del Instituto Tecnológico de Ciudad Valles en San Luis Potosí, México. La muestra fue tipo censo, por supuesto, ya que se incluyeron todas las transacciones de crédito disponibles para 2024. El sistema procesó 144 estados de cuenta digitales y 12,563 transacciones financieras. La decisión de usar la herramienta con datos reales hasta ahora, con carga de trabajo tipo escolar y carga de trabajo en el área administrativa (Wohlin et al., 2012; Kitchenham et al., 2009), permitió evaluar el trabajo.

Las principales variables fueron el tiempo de procesamiento de documentos y la precisión de la extracción. El tiempo se midió en segundos de ejecución desde la carga del documento hasta la salida estructurada. La precisión se midió comparando los campos extraídos por el sistema con un conjunto de campos verificados manualmente. Las coincidencias correctas recibieron un valor de acierto y cualquier discrepancia fue un error o un caso de revisión. Este tipo de evaluación permite comparar el artefacto con un estándar de referencia y reduce la percepción subjetiva de la calidad de extracción (Wohlin et al., 2012; DeLone & McLean, 2003).

El desarrollo se realizó de manera incremental. Primero, se examinaron descripciones de cuentas y cuentas bancarias para identificar patrones comunes. Luego, se desarrollaron reglas de extracción para fechas, montos y números de control. Finalmente, se construyó la arquitectura del sistema, se implementó la interfaz de usuario y funciones de exportación. Cada vez que se actualizaba la lógica basada en los errores observados en las pruebas internas, una práctica común ya que es bueno ser ágil y trabajar en las cosas de manera temprana e iterativa (Dybå & Dingsøyr, 2008; Boehm, 1988). La Tabla 1 resume los campos usados para evaluar el sistema.

Tabla 1

Variables consideradas para la validación de BaucherMatch

Variable	Tipo	Indicador	Criterio de validación
Tiempo de procesamiento	Cuantitativa continua	Segundos por archivo y acumulado anual	Medición automática en ejecución
Precisión en número de control	Binaria	Acierto / error	Comparación con registro validado

Variable	Tipo	Indicador	Criterio de validación
Precisión en monto monetario	Binaria	Acierto / error	Coincidencia exacta del importe
Precisión en fecha	Binaria	Acierto / error	Coincidencia con estado de cuenta
Estatus de conciliación	Nominal	Coincidencia exacta / revisión manual	Clasificación final del sistema

La organización de la tabla 1 permitió analizar tanto el rendimiento temporal como la calidad de los datos extraídos. En estudios de ingeniería de software, la definición explícita de variables facilita repetir la evaluación, comparar resultados y comunicar con claridad el aporte del artefacto desarrollado (Wohlin et al., 2012; Shaw, 2003).

Diseño y desarrollo de BaucherMatch

BaucherMatch se desarrolló como una aplicación de escritorio para uso local. La decisión se tomó por dos razones. Primero, los trabajadores administrativos necesitaban una herramienta simple para cargar estados de cuenta y descargar resultados sin necesidad de cambiar servicios externos. Segundo, los documentos que se procesan son información financiera sensible, y por lo tanto la ejecución local reduce los riesgos de subir archivos a plataformas de terceros. Esta orientación es consistente con el diseño centrado en el contexto y la necesidad de adaptar el artefacto para trabajar en un contexto organizacional real (Hevner et al., 2004; Runeson & Höst, 2009).

La arquitectura combina cuatro funciones: presentación, servicios API, lógica de negocio y procesamiento. La capa de presentación maneja la interacción con el usuario. La capa API coordina solicitudes y respuestas. La lógica de negocio crea reglas de extracción, normalización y validación. Las utilidades de procesamiento convierten PDF a texto y generan las salidas. La separación de las dos partes del sistema se puede hacer sin afectar todo el sistema (Parnas, 1972; Boehm, 1988).

El *backend* se implementó con Python y FastAPI. Python se utilizó para manejar archivos, cadenas de texto y expresiones regulares, y FastAPI para estructurar *endpoints* ligeros para procesar documentos y devolver resultados en formatos comunes. El sistema llama a pdftotext para recuperar texto de PDFs digitales para evitar OCR cuando el texto escrito ya está presente. La elección técnica para nuestro sistema es un criterio de suficiencia (Manning et al., 2008; Wieringa, 2014): la forma más simple de resolver el problema con precisión aceptable.

El *frontend* fue construido en React y TypeScript. React permitió que la interfaz de usuario se organizara en componentes y TypeScript evitó errores comunes en las estructuras de datos. Tauri se utilizó para empaquetar la aplicación para usuarios de escritorio con un menor uso de recursos. El valor científico del proyecto no depende de la tecnología, sino que la arquitectura es mantenible, con separación de responsabilidades y una experiencia de usuario que es agradable para los equipos administrativos (Dybå & Dingsøyr, 2008; DeLone & McLean, 2003). La figura 1 muestra la arquitectura del software desarrollado.

Figura 1
Arquitectura general de BaucherMatch

La Figura 1 muestra el flujo lógico de la aplicación, comenzando con la carga del documento y luego los documentos completados pueden ser visualizados e integrados en procesos subsecuentes. Esto hace que la contribución de ingeniería sea clara: el sistema no solo convierte archivos, sino que organiza una secuencia de procesamiento que transforma un documento bancario en datos operativos para la reconciliación (Dumas et al., 2018; Thompson, 1968).

Extracción y normalización

La extracción utiliza reglas parametrizadas para identificar patrones de interés, como: las fechas bancarias, que se determinan de acuerdo con los formatos esperados de día, mes y año, las cantidades se identifican por la estructura decimal y los separadores de miles y centavos, los números de control de estudiantes se capturan mediante patrones que coinciden con la sintaxis institucional. Esta estrategia funciona porque los estados de cuenta son lo suficientemente regulares como para realizar búsquedas textuales sin utilizar modelos probabilísticos (Thompson, 1968; Manning et al., 2008).

La normalización convierte las coincidencias en registros homogéneos, por ejemplo, las cantidades se limpian para que los separadores numéricos sean los mismos, las fechas se convierten en el mismo formato y las referencias se almacenan como cadenas verificables. Esto evita que el área financiera tenga una salida difícil de usar. En los sistemas de información, la utilidad de los datos se trata tanto de su captura como de su presentación en un formato confiable y accionable (DeLone & McLean, 2003; Petter et al., 2008).

Los valores atípicos se envían para revisión manual. Esta decisión evita que el sistema obligue a coincidencias cuando la descripción bancaria contiene referencias incompletas, caracteres

dañados y formatos imprevistos. La automatización no elimina el juicio humano; más bien, se enfoca en situaciones donde realmente se necesita. Esta combinación de procesamiento automático y revisión enfocada es más adecuada para procesos administrativos sensibles que la automatización cerrada sin posibilidades de control (Syed et al., 2020; Wieringa, 2014). La Tabla 2 sintetiza los componentes tecnológicos.

Tabla 2

Componentes tecnológicos de BaucherMatch

Componente	Tecnología	Función principal	Criterio de selección
Backend	Python / FastAPI	Procesar documentos y exponer servicios internos	Rapidez de desarrollo y manejo de texto
Extracción textual	Pdftotext	Convertir PDF digital en texto plano	Bajo costo computacional
Motor de patrones	Expresiones regulares	Identificar fechas, montos y referencias	Interpretabilidad y precisión
Frontend	React / TypeScript	Gestionar carga, bitácora y descarga	Componentización y validación de tipos
Empaquetado	Tauri	Generar aplicación de escritorio	Ligereza y ejecución local
Salidas	CSV / JSON	Entregar datos interoperables	Compatibilidad con hojas de cálculo y sistemas

La combinación elegida, mostrada en la Tabla 2, evita una infraestructura compleja y permite que la herramienta pueda funcionar en computadoras de escritorio de las oficinas del departamento. Desde la ingeniería de *software*, esta es una elección lógica para una solución incremental y de bajo costo adaptada a un proceso específico sin comprometer los requisitos de mantenimiento y trazabilidad (Boehm, 1988; Parnas, 1972).

RESULTADOS Y DISCUSIÓN

La validación procesó 12,563 transacciones financieras a través de 144 estados de cuenta para el año fiscal 2024. El tiempo total de ejecución fue de 3.46 segundos (en comparación con el procedimiento manual del área que era de aproximadamente 14.4 días hábiles por mes y 158 días anuales). La diferencia observada evidencia una reducción del tiempo computacional necesario para estructurar la información; sin embargo, el estudio no realizó una evaluación económica del costo total del proceso ni cuantificó costos de desarrollo, mantenimiento, capacitación o gestión de excepciones (Dumas et al., 2018; Syed et al., 2020).

El tiempo promedio de procesamiento por archivo completo es de 0.31 segundos (ver la tabla 3). En donde, el tiempo más corto fue de 0.06 segundos en julio y el más largo fue de 1.16 segundos en agosto debido al volumen de 4,002 créditos. La diferencia en los meses muestra que el tiempo de ejecución aumenta con el número de transacciones, pero se mantiene dentro

de un rango operativo muy bajo. Desde un punto de vista experimental, estos datos nos permiten observar el comportamiento del sistema bajo diferentes cargas, no solo en la prueba aislada (Wohlin et al., 2012; Runeson & Höst, 2009).

Tabla 3

Comparativo de rendimiento temporal entre proceso manual y BaucherMatch

Métrica	Proceso manual	BaucherMatch	Interpretación
Media mensual	13.16 días	0.31 segundos por archivo	Reducción sustancial del tiempo operativo
Mes de menor volumen	4.50 días	0.06 segundos	Procesamiento inmediato en baja carga
Mes de mayor volumen	25 días	1.16 segundos	Respuesta estable en inscripción masiva
Acumulado anual	158 días	3.46 segundos	Reducción sustancial del tiempo operativo

En la Tabla 3 podemos ver que el principal beneficio está en los segundos ahorrados, pero también en la redistribución del trabajo humano. El personal puede dejar de transcribir y comenzar a revisar excepciones, manejar casos especiales y generar informes. Esta transición se alinea con la visión de la automatización como apoyo al trabajo administrativo, no como un reemplazo absoluto del juicio humano (Syed et al., 2020; Petter et al., 2008).

En términos de precisión, BaucherMatch tuvo un rendimiento del 100% para fechas, 99.9% para montos monetarios y 99.8% para números de control estudiantil. Los 25 casos de revisión para números de control podrían deberse a formatos no estandarizados o descripciones bancarias inexactas. Las 13 discrepancias encontradas en los montos estaban relacionadas con caracteres de puntuación atípicos. El enfoque basado en reglas es el más preciso cuando los documentos están en forma regular (Thompson, 1968; Manning et al., 2008).

El procesamiento computacional de los documentos evaluados requirió un tiempo acumulado de 3.46 segundos. Como referencia contextual, el área administrativa estimó que el procedimiento manual asociado con la revisión y preparación de esta información representa aproximadamente 158 días-persona de trabajo durante un año. Ambas magnitudes no constituyen mediciones directamente equivalentes: la primera representa exclusivamente tiempo de ejecución del software, mientras que la segunda incorpora actividades humanas de revisión y gestión. La comparación se presenta, por tanto, únicamente como referencia del potencial de automatización de la etapa de extracción y estructuración documental.

El resultado general también demuestra una clara ventaja: el sistema no necesita aprender de miles de ejemplos etiquetados para funcionar. La lógica se basa en reglas claras que pueden ser revisadas por el equipo técnico. Esto ayuda a ahorrar dinero en capacitación y auditoría. Para las instituciones públicas con pocos recursos, la simplicidad técnica puede ser una fortaleza siempre que el dominio del documento proporcione un patrón claro y la organización mantenga el control sobre los cambios en el formato bancario (Wieringa, 2014; Parnas, 1972). La Tabla 4 muestra cómo se comportan las variables evaluadas.

Tabla 4

Precisión de extracción por variable financiera evaluada

Variable evaluada	Aciertos	Errores / revisión	Tasa de efectividad
Número de control estudiantil	12,538	25	99.801 %
Monto monetario	12,550	13	99.897 %
Fecha de operación bancaria	12,563	0	100.000 %
Consolidado global de atributos	37,651	38	99.899 %

La tasa de coincidencia exacta fue de 99.801 % para el número de control (IC 95 % de Wilson: 99.706–99.865 %), 99.897 % para el monto monetario (IC 95 %: 99.823–99.940 %) y 100.000 % para la fecha de operación (IC 95 %: 99.969–100.000 %). Estos intervalos permiten expresar la incertidumbre asociada con las proporciones observadas y complementan las estimaciones puntuales originalmente reportadas.

La fecha bancaria resultó ser el campo más estable, mientras que el número de control dominó la mayoría de los casos de revisión. Esto era de esperar porque el campo de referencia depende de cómo el estudiante o el banco registren la descripción de la transacción. Por lo tanto, la herramienta no elimina la necesidad de políticas de captura, pero sí reduce significativamente el volumen de revisión manual (DeLone & McLean, 2003; Petter et al., 2008).

Ingeniería de software

La contribución de BaucherMatch es construir y validar una herramienta institucional. En términos de investigación en ingeniería de *software*, el trabajo reporta un artefacto, un contexto de uso, un procedimiento de evaluación y resultados cuantitativos. Shaw (2003) aconseja que los trabajos en esta área expliquen qué problema resuelven, por qué la solución es diferente o útil y qué evidencia respalda su trabajo. En este documento, seguimos esa lógica conectando el diseño con la implementación y la medición con el rendimiento (Shaw, 2003; Hevner et al., 2004).

La idea de las expresiones regulares parece y es bastante simple en comparación con las técnicas de aprendizaje automático, además de ser apropiada para este campo. Los modelos de aprendizaje profundo son de gran utilidad cuando la variación de documentos es grande o el sistema tiene que interpretar un diseño visual complejo. Sin embargo, los estados de cuenta digitales en este estudio ya son recuperables y relativamente estables. Una solución determinista es rápida, de bajo costo e interpretable, pero un modelo más complejo podría introducir dependencia de entrenamiento en el sistema, lo cual va en la dirección opuesta (Yang et al., 2022; Zhong et al., 2019).

El sistema también demuestra una muy buena conexión entre la calidad de los datos y la calidad del proceso. Si el estudiante captura una referencia incompleta, ningún algoritmo puede recuperar de manera confiable información que no existe en la descripción bancaria. Por lo tanto, la mejora tecnológica debe complementarse con requisitos administrativos sobre referencias de pago, formatos de depósito y revisión de excepciones. Como hemos visto en la literatura de sistemas de información, el éxito de cualquier herramienta depende de la calidad del sistema, la calidad de la información y el uso organizacional que se le dé (DeLone &

McLean, 2003; Petter et al., 2008).

La aplicación es local y no necesita cargar estados de cuenta a servicios externos. Reduce la exposición de datos financieros y permite a la institución mantener el control sobre archivos sensibles. En proyectos de automatización administrativa, la eficiencia no debe ser una entidad separada de la confidencialidad y la gobernanza. Syed et al. (2020) afirman que los proyectos de automatización deben tener mecanismos de control, monitoreo y mantenimiento para asegurar que serán valiosos con el tiempo (Syed et al., 2020; Wieringa, 2014).

La transferibilidad de BaucherMatch depende principalmente de la estructura de los documentos de entrada y de las reglas institucionales utilizadas para identificar las referencias de pago. En otro campus que utilice estados de cuenta PDF con texto nativo, la arquitectura general, el mecanismo de conversión mediante pdftotext, el flujo de normalización, la interfaz y los mecanismos de exportación pueden mantenerse sin modificaciones sustanciales.

Las adaptaciones se concentran principalmente en los patrones de extracción correspondientes a referencias institucionales, formatos de fecha, representación de montos y estructura textual de los estados de cuenta. Cuando se incorpora un banco o formato documental diferente, deben parametrizarse estas reglas y realizarse nuevamente una validación contra un conjunto de referencia. Para documentos escaneados sería necesario añadir una etapa de OCR y, cuando la estructura visual sea altamente variable, mecanismos adicionales de análisis de diseño documental. En este sentido, la contribución no se plantea como una solución universal, sino como una arquitectura transferible bajo condiciones documentales explícitas.

La escalabilidad del enfoque depende de dos factores: la estabilidad de los documentos y la capacidad de modificar reglas. Si otro campus de la red TecNM tiene estados de cuenta con la misma estructura, BaucherMatch puede ajustarse con cambios menores. Si el banco cambia drásticamente el formato o los documentos son escaneados, el sistema necesitaría módulos adicionales de OCR y análisis de diseño. Esta limitación no debilita el resultado; define precisamente el alcance del artefacto y abre líneas de evolución técnica consistentes con el ciclo de mejora del *software* (Smith, 2007; Binmakhashen & Mahmoud, 2019).

Prototipo BaucherMatch

El artefacto tecnológico resultante se consolidó en una aplicación de escritorio ejecutable nativa que opera localmente, eliminando riesgos asociados con el manejo de datos financieros sensibles en la nube. La interfaz gráfica desarrollada muestra un entorno intuitivo compuesto por un área de arrastre de archivos, un visor de registros y componentes interactivos para descargar transacciones normalizadas en formatos interoperables CSV y JSON. Las figuras 1 y 2 muestran la interfaz de la aplicación propuesta.

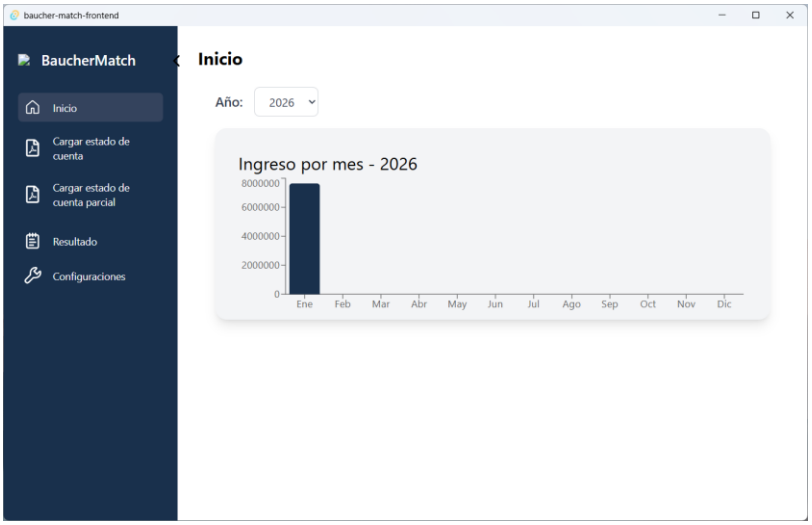

Figura 1

Pantalla de Inicio BaucherMatch

Nota: En la presente se muestra las opciones de inicio del sistema.

Fuente: Propia

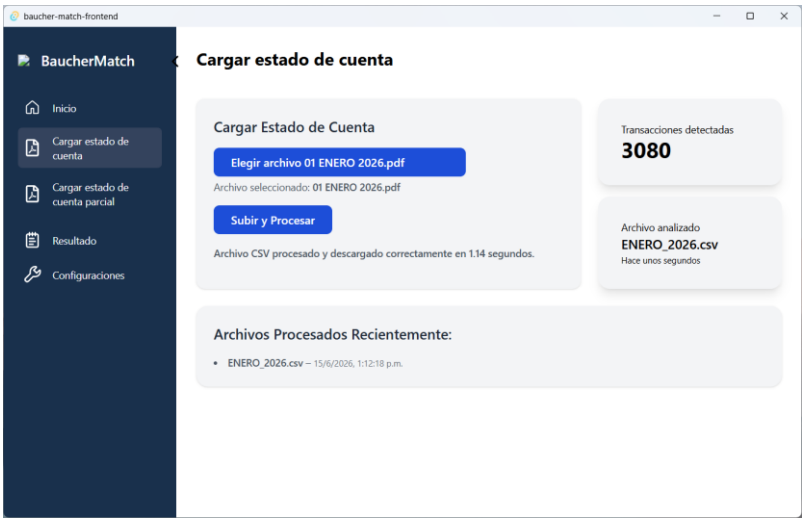

Figura 2

Pantalla de procesamiento de estado de cuenta

Nota: En la presente se muestra las opciones de Carga y Procesamiento de Estado de Cuenta.

Fuente: Propia

CONCLUSIONES

BaucherMatch fue diseñado para automatizar la extracción y organización de la información financiera contenida en los extractos bancarios en PDF. La herramienta procesó 12,563 transacciones reales del año fiscal 2024 con una precisión cercana al 99.9% y un tiempo total de 3.46 segundos. Demostramos que una solución bien establecida de ingeniería de *software* puede mejorar significativamente un proceso administrativo clave sin invertir en plataformas costosas o infraestructura complicada (Wohlin et al., 2012; Hevner et al., 2004).

La investigación sugiere que los enfoques deterministas basados en reglas aún son aplicables si los documentos son estables. En este caso, las expresiones regulares nos permitieron identificar fechas, montos y referencias institucionales con gran precisión. El método es interpretable y fue fácil detectar casos de revisión y sugerir mejoras. Esto es útil para que una institución pública lo use porque reduce costos, simplifica el mantenimiento y ayuda a hacer visible el sistema para los usuarios administrativos (Thompson, 1968; Parnas, 1972).

La principal contribución del artículo es que no solo creamos una aplicación, sino que ilustramos todo el proceso de diseño, implementación y evaluación en un contexto del mundo real. El artículo también proporciona pruebas de que los ingenieros de *software* pueden resolver problemas administrativos a través de artefactos específicos y medibles que se ajustan a los requisitos institucionales. Esta orientación fortalece la relación entre la investigación aplicada y la mejora organizacional (Peffers et al., 2007; Shaw, 2003).

Debido al texto nativo en los PDFs y la estabilidad sintáctica de los extractos, BaucherMatch aún no está disponible para documentos escaneados, encriptados o con diseños altamente variables. En futuras versiones, se podrían añadir módulos de OCR, análisis de diseño, registros históricos y aprendizaje automático para mejorar la cobertura del documento actual. Estas mejoras deberían mantener las propiedades de trazabilidad y control local que caracterizan la versión actual (Smith, 2007; Zhong et al., 2019).

Como acción de mejora institucional, el proyecto puede ser una herramienta con un repositorio histórico, indicadores y seguimiento de excepciones. Incluso puede ser utilizado por otros institutos que enfrentan procesos similares en los que solo obtienen estados de cuenta en formato PDF. Su valor radica en que la automatización no siempre requiere soluciones genéricas a gran escala; a veces, una herramienta enfocada, construida con criterios de ingeniería y evaluada con datos reales, aporta beneficios inmediatos y sostenibles (Dumas et al., 2018; van der Aalst, 2016).

REFERENCIAS

Agarwal, R., & Dhar, V. (2014). Editorial--Big data, data science, and analytics: The opportunity and challenge for IS research. Information Systems Research, 25(3), 443-448. https://doi.org/10.1287/isre.2014.0546

Binmakhashen, G. M., & Mahmoud, S. A. (2019). Document layout analysis: A comprehensive survey. ACM Computing Surveys, 52(6), Article 109. https://doi.org/10.1145/3355610

Boehm, B. W. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72. https://doi.org/10.1109/2.59

DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. Journal of Management Information Systems, 19(4), 9-30. https://doi.org/10.1080/07421222.2003.11045748

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of business process management (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-56509-4

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology, 50(9-10), 833-859. https://doi.org/10.1016/j.infsof.2008.01.006

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75-105. https://doi.org/10.2307/25148625

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. Information and Software Technology, 51(1), 7-15. https://doi.org/10.1016/j.infsof.2008.09.009

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. Communications of the ACM, 15(12), 1053-1058. https://doi.org/10.1145/361598.361623

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45-77. https://doi.org/10.2753/MIS0742-1222240302

Petter, S., DeLone, W., & McLean, E. (2008). Measuring information systems success: Models, dimensions, measures, and interrelationships. European Journal of Information Systems, 17(3), 236-263. https://doi.org/10.1057/ejis.2008.15

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14, 131-164. https://doi.org/10.1007/s10664-008-9102-8

Seddon, P. B. (1997). A respecification and extension of the DeLone and McLean model of IS success. Information Systems Research, 8(3), 240-253. https://doi.org/10.1287/isre.8.3.240

Shaw, M. (2003). Writing good software engineering research papers. Proceedings of the 25th International Conference on Software Engineering, 726-736. https://doi.org/10.1109/ICSE.2003.1201262

Smith, R. (2007). An overview of the Tesseract OCR engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition, 629-633. https://doi.org/10.1109/ICDAR.2007.4376991

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic process automation: Contemporary themes and challenges. Computers in Industry, 115, Article 103162. https://doi.org/10.1016/j.compind.2019.103162

Thompson, K. (1968). Programming techniques: Regular expression search algorithm. Communications of the ACM, 11(6), 419-422. https://doi.org/10.1145/363347.363387

van der Aalst, W. M. P. (2016). Process mining: Data science in action (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-49851-4

Wieringa, R. J. (2014). Design science methodology for information systems and software engineering. Springer. https://doi.org/10.1007/978-3-662-43839-8

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Springer. https://doi.org/10.1007/978-3-642-29044-2

Yang, Y., Chen, M., Wang, X., Liu, F., & Liu, X. (2022). A survey of information extraction based on deep learning. Applied Sciences, 12(19), Article 9691. https://doi.org/10.3390/app12199691

Zhong, X., Tang, J., & Jimeno Yepes, A. (2019). PubLayNet: Largest dataset ever for document layout analysis. Proceedings of the International Conference on Document Analysis and Recognition, 1015-1022. https://doi.org/10.1109/ICDAR.2019.00166