



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**TECNOLÓGICO
NACIONAL DE MÉXICO**

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco

Optimización de la Gestión Financiera y Operativa Mediante la Implementación de un Sistema Basado en NetSuite (Ciclo de Egresos)

Presenta: Jesus Cuahutle Cuatecontzi No. Control 20370951
IT GLOBAL KNOWLEDGE IN SYSTEM SOLUTIONS S.A.
DE C.V.

Asesor externo: Lic. Maricruz Vega Vásquez
Asesor Interno: M.C. Nicolás Alonzo Gutiérrez

Mayo 2025





Instituto Tecnológico de Apizaco
Sistemas y Computación

Apizaco, Tlaxcala, **30/junio/2025**

Oficio No. S.C./OPV/0251/25

ASUNTO: **Liberación de proyecto
para titulación integral**

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	CUAHUTLE CUATECONTZI JESÚS
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	20370951
NOMBRE DEL PROYECTO:	Optimización de la gestión financiera y operativa mediante la implementación de un sistema basado en Netsuite (Ciclo de Egresos)
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

A T E N T A M E N T E

*Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar®*

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN
INSTITUTO TECNOLÓGICO DE APIZACO
SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera
Expediente
MSP/mrol*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

NICOLAS ALONZO GUTIÉRREZ
ASESOR INTERNO



2025
Año de
**La Mujer
Indígena**

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491
Tel. (241) 4172010 Ext. 115, e-mail: sistemas@apizaco.tecnm.mx
tecnm.mx | apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala 05/septiembre/2025

CUAHUTLE CUATECONTZI JESÚS
EGRESADO(A) DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES.
CON NÚMERO DE CONTROL 20370951.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **OPTIMIZACIÓN DE LA GESTIÓN FINANCIERA Y OPERATIVA MEDIANTE LA IMPLEMENTACIÓN DE UN SISTEMA BASADO EN NETSUITE (CICLO DE EGRESOS)**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®

SECRETARÍA DE EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES


CECILIA YARELLY ARREGUIN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo

CYAL/rrc



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



AGRADECIMIENTOS

El desarrollo de este proyecto no habría sido posible sin el apoyo y la contribución de diversas personas que, de manera directa o indirecta, han sido fundamentales no solo en su realización, sino también en mi formación académica y profesional.

En primer lugar, quiero agradecer profundamente a mi familia, quienes han sido mi mayor fuente de fortaleza y motivación a lo largo de toda mi trayectoria universitaria. Su apoyo incondicional, sus palabras de aliento en los momentos difíciles, y su constante fe en mí han sido fundamentales para alcanzar este logro. Gracias por estar siempre presentes, por su paciencia, y por brindarme un hogar que me ha permitido desarrollarme plenamente. También quiero expresar mi gratitud a mis amigos y a todas las personas que conocí a lo largo de este viaje, quienes de alguna forma dejaron una huella en mí. Gracias por los momentos compartidos, las conversaciones sinceras, los ánimos en tiempos de presión y por ser una parte vital del equilibrio entre el esfuerzo y la alegría durante este proceso.

Agradezco sinceramente a IT-Global Knowledge In System Solutions S.A. de C.V., por darme la oportunidad de formar parte de su equipo y confiar en mis habilidades. Y de manera especial, extendiendo mi más sincero agradecimiento a mi líder Oscar Perez, por compartir su experiencia y conocimientos, por su paciencia ante cada pregunta, y por su accesibilidad constante durante cada fase del proyecto. Su guía clara, orientación técnica y compromiso con el desarrollo de este trabajo han sido claves para alcanzar los objetivos propuestos y para enriquecer mi experiencia profesional.

A cada una de estas personas y organizaciones, mi más profundo reconocimiento y gratitud. Este logro también les pertenece.

ÍNDICE DE CONTENIDOS

Introducción.....	10
Descripción de la empresa.....	11
Descripción del cargo del residente	12
Problemas a resolver	13
Objetivos	14
Justificación.....	15
Marco Teórico.....	16
¿Qué es un ERP?.....	16
¿Qué es un sistema ERP?	16
NetSuite ERP.....	17
Módulos funcionales	17
Ventajas y desventajas de NetSuite ERP.....	18
Comparativa con otros ERP en el mercado	19
SuiteCloud Platform.....	20
JavaScript.....	22
SuiteScript	23
Tipos de Script en SuiteScript	24
Módulos de SuiteScript	25
Bundles.....	27
Gestión de Egresos	28
¿Qué es un egreso?	28
Gestión de egresos en empresas sin un ERP.....	29
Ciclo de Egresos en NetSuite	29
Facturación Electrónica en México	31

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Evolución de la facturación electrónica en México	32
Objetivo principal.....	33
CFDI 4.0.....	33
Nodos principales en el CFDI 4.0.....	33
Procedimiento	36
Kick off.....	36
Personalización	38
Enviar Datos de Migración.....	77
Retroalimentación.....	79
Migración a entorno de producción.....	81
Conclusiones.....	85
Competencias Desarrolladas y Aplicadas	86
Fuentes de Información.....	88

ÍNDICE DE TABLAS

Tabla 1: Ventajas y desventajas de Netsuite ERP	19
Tabla 2: Comparativa ERPs en el mercado.....	20
Tabla 3: Tipos de Script en SuiteScript.....	25
Tabla 4: Módulos de SuiteScript.....	27
Tabla 5: Nodos de un CFDI 4.0	34

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Componentes principales de SuiteCloud	21
Ilustración 2: Ciclo de egresos en NetSuite.....	30
Ilustración 3: Formulario de Gastos.....	31
Ilustración 4: Ejemplo de archivo XML CFDI 4.0	35
Ilustración 5: Diagrama de carriles	39

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Ilustración 6: Diagrama de flujo de creación de búsqueda guardada	42
Ilustración 7: Interacción usuario final búsqueda guardada	43
Ilustración 8: Diagrama de flujo de creación de workbook	44
Ilustración 9: Interacción usuario final portlets. Fuente	45
Ilustración 10: Definición de versión API y tipo de script	46
Ilustración 11: Definición de módulos SuiteScript	46
Ilustración 12: Creación de objeto form	46
Ilustración 13: Creación del campo de ayuda	47
Ilustración 14: Obtención de empleado	47
Ilustración 15: Mostrar nombre de empleado	47
Ilustración 16: Campos de carga de archivo	48
Ilustración 17: Pantalla carga de archivos	48
Ilustración 18: Definición de versión API y tipo de script	49
Ilustración 19: Definición de módulo SuiteScript	49
Ilustración 20: Validación de archivos XML y PDF	49
Ilustración 21: Mensaje de error archivo XML	50
Ilustración 22: Mensaje de error archivo PDF	50
Ilustración 23: Definición de versión API y tipo de script	51
Ilustración 24: Validación del archivo XML	51
Ilustración 25: Procesamiento del contenido del archivo XML	51
Ilustración 26: Extracción de atributos del nodo Comprobante	51
Ilustración 27: Extracción de datos de los nodos Emisor y Receptor	52
Ilustración 28: Procesamiento de los Conceptos	52
Ilustración 29: Construcción del JSON final	53
Ilustración 30: Envío de datos al Suitelet	53
Ilustración 31: Manejo de errores extracción de datos	53
Ilustración 32: Pantalla de error XML incompleto	54
Ilustración 33: Pantalla de error XML inválido	54
Ilustración 34: Definición de API, tipo de script y módulos	54
Ilustración 35: Creación de instancia form	55
Ilustración 36: Recibir JSON enviado	55

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Ilustración 37: Validación de datos CFDI y generación de advertencia	55
Ilustración 38: Pantalla de error datos no recibidos.....	56
Ilustración 39: Fragmento de función validateCFDI.....	56
Ilustración 40: Función searchDuplicateUUID	57
Ilustración 41: Pantalla de error validación de campos	57
Ilustración 42: Llamada de función validate CFDI	58
Ilustración 43: Creación de grupos.....	58
Ilustración 44: Ejemplo de creación de campo y asignación de valor	59
Ilustración 45: Pantalla de visualización de datos (parte 1).....	60
Ilustración 46: Pantalla de visualización de datos (parte 2).....	60
Ilustración 47: Pantalla de visualización de datos (parte 3).....	61
Ilustración 48: Consulta del CFDI en el SAT	64
Ilustración 49: Validación de CFDI y creación de registro	64
Ilustración 50: Creación del Registro de Gasto	65
Ilustración 51: Función para la creación y validación de carpetas.....	66
Ilustración 52: Función para almacenamiento de archivos XML y PDF. Fuente. Elaboración propia fragmento de código.....	67
Ilustración 53. Gabinete de archivos con carpetas de empleados	67
Ilustración 54. Carpeta de empleado con carpeta de factura	68
Ilustración 55: Creación del Formulario de Confirmación	68
Ilustración 56: Recepción de Parámetros.....	69
Ilustración 57: Campo de mensaje de éxito en pantalla	69
Ilustración 58: Pantalla de confirmación	69
Ilustración 59: Pantalla de error.....	70
Ilustración 60: Selección de tipo de Registro.....	71
Ilustración 61: Criterios de búsqueda	71
Ilustración 62: Criterios de filtro	72
Ilustración 63: Columnas resultados	72
Ilustración 64: Configuración de permisos	73
Ilustración 65: Búsqueda guardada de gastos de empleados	73
Ilustración 66: Selección dataset.....	74

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Ilustración 67: Filtros dinámicos	75
Ilustración 68: Construcción de tabla de datos.....	75
Ilustración 69: Construcción de gráfico de datos.....	76
Ilustración 70: Portlet de gráfico de gastos de empleados en panel de inicio	76
Ilustración 71: Portlet de tabla de gastos de empleados en panel de inicio. Fuente. Captura de pantalla de entorno de pruebas NetSuite.	77
Ilustración 72: Carpeta del proyecto con archivos JS desarrollados	77
Ilustración 73: Creación de registro de script	78
Ilustración 74: Registro de script creado	78
Ilustración 75: Creación de sublistas.....	80
Ilustración 76: Sublista de conceptos	80
Ilustración 77: Sublista impuestos	80
Ilustración 78: Llenado de datos básicos	81
Ilustración 79: Selección de objetos del bundle.....	82
Ilustración 80: Objetos seleccionados para el bundle	82
Ilustración 81: Búsqueda de bundle	83
Ilustración 82: Resultado de búsqueda	83
Ilustración 83: Bundle Recepción CFDI.....	84
Ilustración 84: Instalación de bundle	84

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Introducción

En un entorno empresarial altamente competitivo, la eficiencia en la gestión financiera y operativa es un factor clave para el éxito. La optimización del ciclo de egresos permite un mejor control sobre las cuentas por pagar, la conciliación de pagos y la automatización de procesos financieros, reduciendo errores y mejorando la toma de decisiones estratégicas.

Este proyecto tiene como objetivo la personalización del ciclo de egresos de NetSuite ERP en IT Global Knowledge In System Solutions S.A. de C.V. A través del desarrollo de soluciones específicas, como la integración de facturas electrónicas, la validación, y la generación automatizada de reportes financieros, se busca mejorar la eficiencia y transparencia en la administración de los recursos financieros de la empresa.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Descripción de la empresa

IT GLOBAL KNOWLEDGE IN SYSTEM SOLUTIONS S.A. DE C.V.

IT Global ofrece servicios generales en consultoría de TI, principalmente sobre tecnologías Oracle. Con la prioridad de satisfacer las necesidades de cada uno de los clientes, con calidez de servicio y equipo de trabajo con experiencia en proyectos nacionales e internacionales.

Misión

Llevar a cabo servicios de consultoría de calidad para nuestros clientes que están en constante optimización de sus organizaciones, desde la perspectiva de mejora continua, innovación y gestión de tecnologías.

Visión

Incrementar nuestra presencia a nivel internacional, consolidándonos como socios de valor para nuestros clientes, superando sus expectativas con servicios de calidad. Siendo una empresa integradora de talento humano, promoviendo siempre el desarrollo profesional.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Descripción del cargo del residente

NetSuite ERP es una plataforma robusta que ofrece múltiples soluciones y módulos estándar para la gestión financiera, operativa y administrativa de las empresas. Sin embargo, cada organización tiene procesos específicos que requieren personalizaciones y configuraciones avanzadas para adaptarse a sus necesidades particulares. En IT Global Knowledge In System Solutions S.A. de C.V., se trabaja con diversas empresas que, aunque utilizan NetSuite como base, requieren desarrollos personalizados que optimicen su eficiencia y les permitan una integración más fluida con sus operaciones.

El Consultor Técnico NetSuite (Residente) en IT Global Knowledge In System Solutions S.A. de C.V., se encarga de la implementación, personalización y optimización de NetSuite ERP, enfocándose en el desarrollo de soluciones técnicas para mejorar los procesos financieros y operativos de la empresa. Las responsabilidades incluyen la automatización y personalización de procesos mediante SuiteCloud Platform Tools, la integración y validación de datos, la configuración de módulos, la optimización del rendimiento del ERP y la generación de reportes financieros automatizados. Además, brindar soporte técnico, documentando cada proceso para garantizar soluciones escalables y alineadas con las necesidades del negocio, contribuyendo así a la transformación digital y eficiencia financiera de la organización.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Problemas a resolver

1. Falta de automatización en el registro y de egresos: Actualmente, el registro y validación de gastos es un proceso manual, lo que genera retrasos en la aprobación y pagos, además de un mayor riesgo de errores humanos.
2. Dificultad en la conciliación de egresos con CFDI: La validación de facturas electrónicas (CFDI) no está integrada en NetSuite, lo que dificulta la conciliación de los egresos con los comprobantes fiscales, aumentando la posibilidad de pagos incorrectos o rechazo de deducciones fiscales.
3. Reportes financieros limitados y poco accesibles: La generación de reportes sobre los egresos no está automatizada ni personalizada, lo que dificulta el análisis en tiempo real y la toma de decisiones estratégicas para la optimización de los recursos financieros.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Objetivos

Desarrollar e implementar soluciones personalizadas mediante bundles en NetSuite ERP, que respondan a las necesidades del cliente en el ámbito financiero.

Objetivos específicos

1. Desarrollar bundles según los requerimientos del cliente.
2. Enviar las personalizaciones (bundles) para ser probadas y validadas.
3. Recibir retroalimentación de pruebas unitarias y hacer correcciones en caso de ser necesario.
4. Migrar las configuraciones finales para habilitar el entorno de producción.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Justificación

La personalización de NetSuite ERP es necesaria para adaptar la plataforma a los procesos financieros específicos del cliente. Aunque el sistema cuenta con diversas funcionalidades estándar, estas no cubren completamente los requerimientos particulares del cliente.

El desarrollo de bundles permitirá configurar NetSuite de acuerdo con las necesidades del cliente, asegurando que el sistema contemple sus flujos de trabajo y cumpla con sus exigencias dentro del ciclo financiero.

Estas personalizaciones ayudarán a que la empresa pueda utilizar el ERP de manera más adecuada a su operación diaria, facilitando tareas y mejorando la gestión de su información financiera. El desarrollo de estas soluciones garantiza que NetSuite se ajuste mejor a la realidad de la empresa y a sus procesos internos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Marco Teórico

¿Qué es un ERP?

Enterprise resource planning (ERP) es un software que las organizaciones utilizan para gestionar actividades diarias como la contabilidad, el aprovisionamiento, la gestión de proyectos, el cumplimiento, y las operaciones de la cadena de suministro. Una solución de ERP completa incluye herramientas de análisis de datos y gestión financiera para planificar, presupuestar y notificar resultados financieros.

Los sistemas de ERP facilitan el flujo de datos entre procesos empresariales, recopilando datos de transacciones de diversas fuentes, eliminando duplicados y proporcionando una única fuente de confianza. Los sistemas de ERP facilitan el flujo de datos entre procesos empresariales, recopilando datos de transacciones de diversas fuentes, eliminando duplicados y proporcionando una única fuente de confianza. Hoy en día, son esenciales para miles de empresas de todos los tamaños y sectores.

¿Qué es un sistema ERP?

Los sistemas de ERP gestionan actividades empresariales cotidianas como la contabilidad, las finanzas, la gestión de inventarios, la gestión de la cadena de suministro y la fabricación.

Los sistemas de ERP gestionan actividades empresariales cotidianas como la contabilidad, las finanzas, la gestión de inventarios, la gestión de la cadena de suministro y la fabricación. Estos sistemas pueden ser locales o en la nube, y respaldan la gestión financiera, los recursos humanos y la producción. Proporcionan transparencia en los procesos de negocio, supervisando aspectos de producción, logística y finanzas.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Proporcionan transparencia en los procesos de negocio, supervisando aspectos de producción, logística y finanzas. Estos sistemas integrados actúan como eje central de la empresa para la totalidad de flujos de trabajo y datos, brindando a diversos departamentos acceso a esta información.

Los sistemas de ERP soportan funciones empresariales variadas y se adaptan a diferentes sectores, mejorando la eficiencia y optimizando los recursos de la empresa.

NetSuite ERP

NetSuite ERP es una solución integral de gestión empresarial en la nube, basada en IA, que ayuda a las organizaciones a operar con mayor eficacia mediante la automatización de procesos clave y la visibilidad en tiempo real del rendimiento operativo y financiero. Gestiona fácilmente múltiples filiales, unidades de negocio y entidades legales con una única solución ERP. Con una única suite integrada para gestionar la contabilidad, los pedidos, el inventario, los proyectos, la producción, la cadena de suministro y las operaciones de almacén, NetSuite ERP ofrece a las empresas una visibilidad clara de sus datos y un control más preciso a negocios.

Módulos funcionales

NetSuite ofrece un conjunto robusto de módulos funcionales que cubren distintas áreas del negocio:

- **Contabilidad y Finanzas:** permite la gestión de cuentas, libros contables, conciliaciones bancarias, presupuestos, y más.
- **Gestión de Inventarios:** control de existencias, almacenes, niveles de stock, y órdenes de compra/venta.
- **CRM (Customer Relationship Management):** gestión de relaciones con clientes, oportunidades de ventas y seguimiento postventa.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

- **Gestión de Proyectos y Tiempos:** herramientas para la planificación y ejecución de proyectos empresariales.
- **RRHH y Nómina:** administración de personal, nóminas, permisos y beneficios.
- **E-commerce y Omnicanalidad:** integración con tiendas en línea, puntos de venta y comercio electrónico B2B/B2C.

Ventajas y desventajas de NetSuite ERP

En la tabla 1 se resume las principales ventajas y desventajas de implementar NetSuite ERP en una organización. Este análisis permite tener una visión equilibrada sobre los beneficios que ofrece esta solución en términos de eficiencia operativa, escalabilidad y automatización, así como los posibles retos que pueden surgir durante su adopción, como el costo o la curva de aprendizaje.

Ventajas de NetSuite ERP	Desventajas de NetSuite ERP
Plataforma 100% en la nube, accesible desde cualquier lugar con internet.	Costo elevado para pequeñas empresas, especialmente en licencias y personalización.
Escalable, ideal para empresas en crecimiento o con múltiples ubicaciones.	Curva de aprendizaje moderada, especialmente para usuarios nuevos o sin experiencia ERP.
Amplia cobertura funcional: finanzas, compras, ventas, CRM, inventario, RRHH, etc.	Dependencia de partners para implementaciones complejas o personalizaciones avanzadas.
Personalización potente con SuiteScript, SuiteFlow, y SuiteBuilder.	Limitada capacidad offline, ya que requiere conexión constante a internet.
Actualizaciones automáticas y frecuentes, gestionadas por Oracle.	Algunas funciones estándar requieren licencias adicionales o desarrollos específicos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Integración nativa con otras soluciones Oracle y gran capacidad de integración vía API.	Tiempo de implementación puede ser largo si no se gestiona adecuadamente.
Generación de reportes y analítica avanzada con Workbooks, Saved Searches y Dashboards.	Interfaz de usuario puede parecer menos moderna comparada con ERPs más recientes.

Tabla 1: Ventajas y desventajas de Netsuite ERP. Fuente. Elaboración Propia.

Comparativa con otros ERP en el mercado

En el competitivo entorno empresarial actual, elegir el sistema ERP adecuado es una decisión estratégica que puede impactar significativamente en la eficiencia operativa y en la capacidad de adaptación al crecimiento. Existen múltiples soluciones ERP en el mercado, cada una con características, enfoques y fortalezas distintas. En esta sección se realiza una comparativa entre NetSuite ERP y otros sistemas reconocidos como SAP, Microsoft Dynamics 365, Oracle ERP Cloud y Odoo, con el fin de identificar las diferencias clave en funcionalidad, flexibilidad, costos, escalabilidad y facilidad de implementación. Esta comparación busca proporcionar una visión clara y objetiva para facilitar la selección del ERP que mejor se ajuste a las necesidades específicas de una organización.

Criterio	NetSuite ERP	SAP Business One	MS Dynamics 365
Modelo de implementación	100% en la nube	Local / Nube (partner-hosted)	Nube / Local / Híbrido
Flexibilidad y personalización	Alta, con SuiteScript, Workflows y SuiteBuilder	Moderada, requiere consultores SAP	Alta, requiere conocimientos técnicos

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Facilidad de uso	Intuitiva, interfaz amigable	Interfaz menos intuitiva	Integración nativa con entorno Microsoft
Módulos incluidos	Finanzas, CRM, Inventarios, Proyectos, RRHH, etc.	Finanzas, CRM, Inventario, Compras, Producción	Finanzas, Ventas, Compras, Servicio al cliente
Actualizaciones	Automáticas, controladas por Oracle	Manuales (o mediante partner)	Automáticas en la nube
Escalabilidad	Muy alta, orientado a empresas medianas y grandes	Limitada para grandes empresas	Alta, especialmente en entornos Microsoft
Costo	Medio a alto (SaaS licencias por usuario)	Medio a alto (licencia + implementación)	Medio, por usuario y módulo
Ideal para	Empresas en crecimiento o multinacionales	PYMEs en crecimiento	Empresas con ecosistema Microsoft

Tabla 2: Comparativa ERPs en el mercado. Fuente. Elaboración Propia.

SuiteCloud Platform

La plataforma NetSuite SuiteCloud es un entorno de extensibilidad y personalización que permite a las empresas adaptar NetSuite a las necesidades únicas de sus negocios. Diseñada para profesionales de TI, analistas de negocio y desarrolladores de aplicaciones con diversas habilidades, SuiteCloud ofrece un conjunto flexible de soluciones unificadas para configurar, personalizar, integrar y ampliar programáticamente cualquier funcionalidad del producto principal.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Entre los casos de uso más comunes que aprovechan la potencia de SuiteCloud se encuentran la automatización de flujos de trabajo como la aprobación de egresos o pagos, la validación personalizada de datos al registrar transacciones, y la creación de pantallas interactivas o formularios específicos mediante Suitelets. Asimismo, las empresas utilizan esta plataforma para integrar NetSuite con sistemas externos como plataformas de comercio electrónico, CRMs o bancos, y para generar reportes y dashboards personalizados que respondan a las necesidades estratégicas de distintas áreas del negocio. Estas capacidades permiten que NetSuite evolucione junto con los procesos empresariales sin perder cohesión ni seguridad.

SuiteCloud está compuesto por varias herramientas clave presentadas en la Ilustración 1. Estas herramientas permiten extender las capacidades nativas de NetSuite mediante configuraciones sin código, integraciones con sistemas externos y desarrollo personalizado. En la siguiente sección se abordará específicamente SuiteScript, uno de los componentes más potentes de la plataforma, que permite crear flujos personalizados, automatizaciones y aplicaciones a medida utilizando JavaScript dentro del entorno de NetSuite.

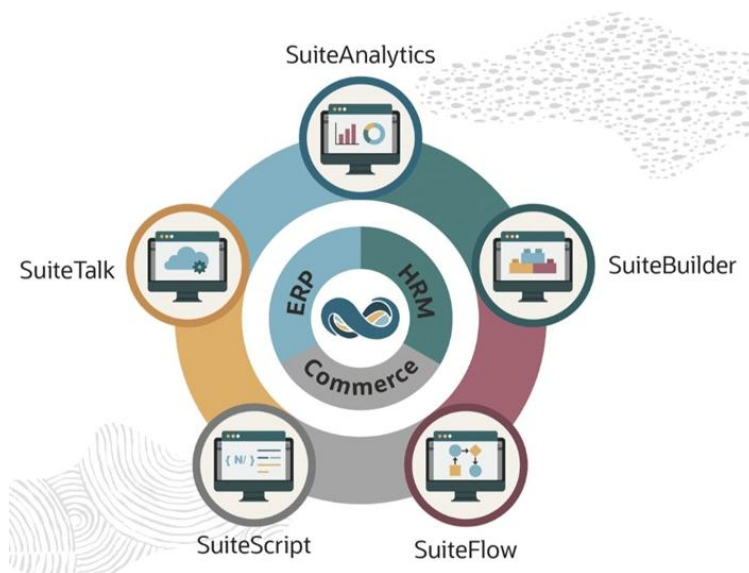


Ilustración 1: Componentes principales de SuiteCloud. Fuente. Oracle MyLearn. (2025). Componentes principales de SuiteCloud [imagen]. SuiteScript Foundations [curso]. <https://mylearn.oracle.com/netsuite/learning-path/suitescript-foundations/>

Optimización de la Gestión Financiera ... (Ciclo de egresos)

JavaScript

Antes de adentrarnos en el subtema de SuiteScript, es fundamental comprender las bases del lenguaje en el que está construido: JavaScript. Es la base sobre la cual se desarrollan los scripts personalizados que permiten automatizar procesos, validar datos y extender funcionalidades del sistema.

JavaScript es un lenguaje de programación interpretado, orientado a objetos y basado en prototipos, ampliamente utilizado en el desarrollo web para crear experiencias interactivas en el navegador. Fue desarrollado originalmente por Netscape en 1995 y desde entonces se ha consolidado como uno de los lenguajes más importantes en el mundo del desarrollo de software. A diferencia de HTML y CSS, que se encargan de la estructura y el estilo de una página web respectivamente, JavaScript se utiliza para añadir dinamismo, responder a eventos del usuario, manipular el DOM (Document Object Model), y comunicarse con servidores a través de tecnologías como AJAX.

Una de las grandes ventajas de JavaScript es su compatibilidad con múltiples plataformas y navegadores, lo cual lo convierte en una herramienta esencial no solo en el lado del cliente (frontend), sino también en el servidor (backend), gracias al uso de entornos como Node.js. JavaScript soporta programación funcional y orientada a objetos, lo que le brinda una gran flexibilidad para adaptarse a distintos estilos de codificación y requerimientos de proyectos.

Dominar los fundamentos de JavaScript es esencial para cualquier desarrollador que trabaje con NetSuite y desee aprovechar al máximo las posibilidades de personalización y automatización que ofrece la plataforma SuiteCloud.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

SuiteScript

SuiteScript es el framework de desarrollo basado en JavaScript que ofrece Oracle NetSuite para extender, personalizar y automatizar los procesos de negocio dentro de su plataforma ERP. Al estar construido sobre JavaScript, SuiteScript resulta familiar para muchos desarrolladores, lo que facilita el aprendizaje y la adopción de sus capacidades. Sin embargo, lo que diferencia a SuiteScript de JavaScript estándar es su conjunto de APIs específicas, diseñadas para interactuar directamente con los objetos, formularios, registros y flujos de trabajo del ecosistema NetSuite.

SuiteScript permite a los desarrolladores implementar lógica personalizada en diferentes momentos del ciclo de vida de un registro, mediante el uso de scripts del lado del cliente (Client Scripts), del lado del servidor (User Event Scripts, Suitelets, RESTlets), o scripts programados (Scheduled Scripts). Estas categorías permiten cubrir una amplia gama de casos de uso, como validaciones de datos en tiempo real, automatización de tareas complejas, integración con sistemas externos, o generación dinámica de interfaces para el usuario.

Entre sus características destacadas se encuentran:

- **Acceso a registros de NetSuite:** SuiteScript permite crear, leer, actualizar y eliminar registros dentro de la base de datos de NetSuite mediante funciones estandarizadas como `record.load()`, `record.create()`, entre otras.
- **Interacción con formularios personalizados:** A través de Suitelets, se pueden construir interfaces de usuario personalizadas para necesidades específicas como la carga de archivos, validaciones especiales o recopilación de datos adicionales.
- **Automatización de procesos:** Con Scheduled Scripts, los desarrolladores pueden programar tareas que se ejecutan automáticamente en horarios definidos, ideal para procesos nocturnos o recurrentes.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

- Integraciones externas: Mediante RESTlets, SuiteScript facilita la creación de endpoints personalizados para recibir o enviar datos a otros sistemas externos.

SuiteScript 2.x, la versión moderna del framework, ofrece una estructura modular basada en `define()` y `require()`, que mejora la organización del código, la reutilización de funciones y la depuración. También proporciona un enfoque más robusto y escalable que su versión predecesora (SuiteScript 1.0), promoviendo buenas prácticas de programación.

Tipos de Script en SuiteScript

NetSuite SuiteScript proporciona varios tipos de scripts que permiten ejecutar lógica personalizada en distintos contextos del sistema, adaptándose a los requerimientos específicos de cada proceso de negocio. Estos scripts están diseñados para actuar en momentos clave del ciclo de vida de los registros, así como en la interfaz del usuario, en tareas programadas o en integraciones con sistemas externos. La tabla 3 resume los principales tipos de scripts disponibles en SuiteScript, sus características y usos comunes.

Tipo de Script	Descripción	Uso Principal
Client Script	Se ejecuta en el navegador del usuario, directamente sobre formularios.	Validaciones en tiempo real, interactividad con campos, personalización de UI.
User Event Script	Se ejecuta en el servidor al crear, editar o eliminar registros.	Validaciones de negocio, lógica previa o posterior al guardado de registros.
Suitelet	Crea interfaces de usuario personalizadas dentro de NetSuite.	Formularios de carga, asistentes (wizard), reportes personalizados.
RESTlet	Expone funciones como servicios web REST personalizados.	Integraciones con sistemas externos mediante APIs.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Scheduled Script	Se ejecuta automáticamente en intervalos definidos.	Automatización de tareas recurrentes o procesos por lotes.
Map/Reduce Script	Diseñado para procesar grandes volúmenes de datos en paralelo.	Procesamiento masivo eficiente, análisis y limpieza de datos.
Portlet Script	Permite crear portlets personalizados en el dashboard.	Widgets, gráficos o indicadores de negocio a medida.
Mass Update Script	Permite aplicar lógica personalizada en actualizaciones masivas de registros.	Actualización masiva de campos con lógica específica.

Tabla 3: Tipos de Script en SuiteScript. Fuente. Elaboración Propia

Cada uno de estos scripts puede utilizarse individualmente o en conjunto para lograr un alto nivel de personalización, automatización e integración dentro de NetSuite. Elegir el tipo adecuado depende del momento en que se requiere ejecutar la lógica, del volumen de datos y del entorno de ejecución.

Módulos de SuiteScript

Para facilitar la organización del código y promover la reutilización de funciones, SuiteScript adopta una estructura modular basada en el modelo AMD (Asynchronous Module Definition). Este enfoque permite dividir la lógica en archivos independientes que se pueden importar y utilizar según sea necesario, mejorando la legibilidad, el mantenimiento y la eficiencia del desarrollo. A través del sistema de módulos, los scripts de SuiteScript pueden acceder de manera controlada a funcionalidades específicas proporcionadas por NetSuite, como manejo de registros, UI, búsquedas o integración, haciendo que el desarrollo sea más robusto y escalable.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Categoría	Módulo	Descripción
Registros y Datos	N/record	Creación, carga, edición y eliminación de registros.
	N/search	Realización de búsquedas guardadas y consultas avanzadas.
	N/query	Creación y ejecución de consultas tipo SQL.
Interfaz de Usuario	N/ui/serverWidget	Creación de formularios, listas y sublistas personalizadas.
	N/ui/dialog	Muestra alertas, confirmaciones y diálogos emergentes.
	N/ui/message	Muestra mensajes flotantes en la UI.
	N/ui/button	Creación y personalización de botones en formularios.
	N/ui/menu	Personalización de menús en la UI.
Automatización y Eventos	N/workflow	Manipulación y ejecución de flujos de trabajo.
	N/scheduledScript	Control de ejecución de scripts programados.
	N/mapReduce	Procesamiento masivo de datos con Map/Reduce.
Integraciones y API	N/https	Realización de llamadas HTTP (GET, POST, etc.).
	N/email	Envío de correos electrónicos.
	N/file	Manipulación de archivos en el File Cabinet.
	N/xml	Procesamiento y manipulación de archivos XML.
	N/encode	Codificación y decodificación de datos (Base64, UTF-8, etc.).
Seguridad y Usuarios	N/runtime	Obtención de información del usuario y script en ejecución.
	N/role	Manejo de roles de usuario.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Localización y Configuración	N/config	Obtención y actualización de configuraciones de NetSuite.
	N/translation	Gestión de traducciones y localización.
Depuración y Registros	N/log	Registro de mensajes en NetSuite para depuración.

Tabla 4: Módulos de SuiteScript. Fuente. Elaboración propia.

Bundles

En NetSuite, los Bundles son paquetes reutilizables de personalizaciones que permiten implementar, compartir y distribuir configuraciones, scripts, formularios personalizados, campos, roles, dashboards, búsquedas guardadas, entre otros elementos dentro del ecosistema de NetSuite. Esta funcionalidad es parte del Bundle Installation System (BIS), que permite a los desarrolladores y administradores de cuentas trasladar soluciones completas de un entorno a otro de manera eficiente y segura.

Los bundles son especialmente útiles para desarrolladores que trabajan en ambientes de desarrollo, prueba y producción, ya que permiten migrar soluciones completas sin necesidad de reconstruir configuraciones desde cero. Además, proporcionan una forma de versionar y mantener aplicaciones o personalizaciones a lo largo del tiempo.

Cada bundle puede incluir scripts desarrollados en SuiteScript (como Client Scripts, Suitelets, User Event Scripts, etc.), configuraciones de formularios, flujos de trabajo (workflows), archivos, listas personalizadas, entre otros. Una vez creado, un bundle puede compartirse de forma privada entre cuentas específicas o publicarse en el Bundle Repository de NetSuite para que otras empresas puedan instalarlo.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Entre las principales ventajas del uso de bundles se destacan:

- Reutilización y distribución: permite compartir soluciones personalizadas entre diferentes cuentas o subsidiarias.
- Facilidad de mantenimiento: con control de versiones, se pueden actualizar fácilmente los elementos del bundle.
- Automatización de implementación: evita errores manuales al replicar configuraciones entre ambientes.

Gestión de Egresos

¿Qué es un egreso?

Un egreso representa la salida de recursos económicos de una organización, generalmente como resultado del pago de obligaciones, adquisición de bienes o servicios, o reembolsos. En términos prácticos, los egresos reflejan el dinero que la empresa gasta para operar, mantener sus actividades o cumplir compromisos con terceros.

Dentro de los egresos, los gastos ocupan un lugar importante, ya que representan las salidas de dinero destinadas a cubrir los costos necesarios para el funcionamiento operativo y administrativo de una empresa. Los gastos, a diferencia de otros tipos de egresos como las inversiones, no están directamente orientados a generar ingresos futuros, sino que son esenciales para mantener las operaciones actuales del negocio.

Un buen control y análisis de los gastos permite a las empresas evitar gastos innecesarios, optimizar sus recursos y, en última instancia, mejorar su rentabilidad.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Gestión de egresos en empresas sin un ERP

En las empresas que no cuentan con un sistema ERP (Enterprise Resource Planning), la gestión de egresos se realiza mediante herramientas tradicionales o procesos manuales. Aunque estas alternativas pueden ser funcionales en empresas pequeñas o emergentes, presentan diversas limitaciones que afectan la eficiencia y la capacidad de respuesta ante los desafíos del mercado.

La gestión de egresos en empresas sin un ERP enfrenta diversos retos y limitaciones, como el alto riesgo de errores humanos debido al ingreso manual de datos, la falta de trazabilidad que dificulta rastrear cada egreso y su impacto en el presupuesto, y los procesos lentos y laboriosos que consumen tiempo y esfuerzo. Además, la ausencia de herramientas avanzadas limita la generación de reportes útiles para la toma de decisiones estratégicas, afectando la capacidad de controlar las finanzas de manera eficiente y de adaptarse al crecimiento o diversificación de operaciones. Estos factores, en conjunto, reducen la competitividad y eficiencia de la empresa.

Ciclo de Egresos en NetSuite

En NetSuite, el ciclo de egresos abarca el conjunto de procesos necesarios para gestionar eficientemente las salidas de dinero, desde la solicitud de un gasto hasta su validación y contabilización. Este ciclo puede incluir diferentes tipos de egresos como pagos a proveedores, servicios, viáticos o reembolsos, siendo este último especialmente relevante cuando se trata del registro de gastos de empleados.

En NetSuite, la gestión del ciclo de egresos relacionados con los gastos de empleados está diseñado para simplificar, centralizar y guiar el proceso, asegurando un manejo eficiente y trazable de los reembolsos y pagos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

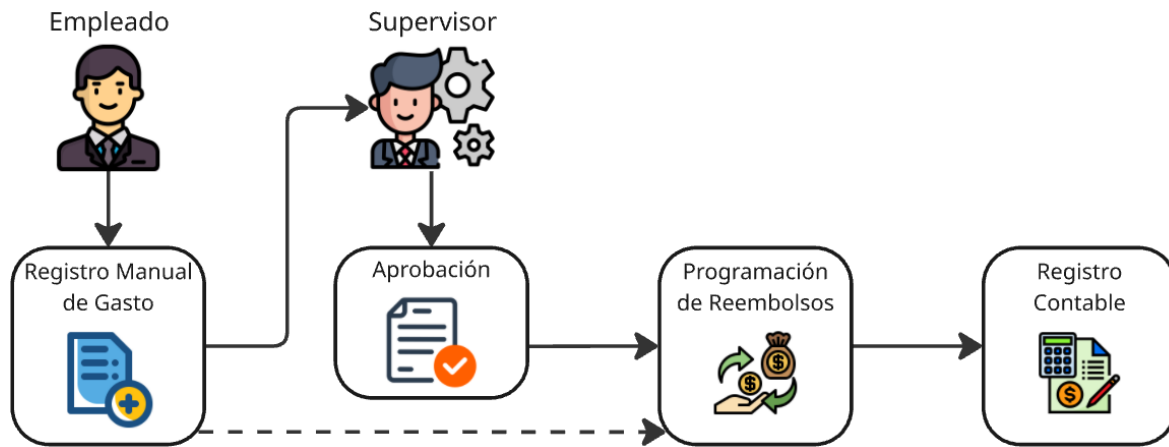


Ilustración 2: Ciclo de egresos en NetSuite. Fuente. Elaboración Propia.

Se detalla el proceso en la ilustración 2. Donde:

1. Registro de gasto: Los empleados registran sus gastos a través del formulario de gastos (ilustración 3) para empleados en NetSuite. Ingresan los detalles de los gastos (como fechas, categorías, montos). Una vez completado, se envía su reporte de gastos para aprobación al supervisor del empleado o va directo al otro paso (depende de la configuración y/o reglas de negocio).
2. Proceso de aprobación: NetSuite permite configurar flujos de aprobación automáticos basados en reglas de negocio. Se verifica automáticamente el cumplimiento de las políticas, como límites de gasto o categorías permitidas. Si se aprueba, el reporte avanza automáticamente al siguiente paso; si no, se devuelve con comentarios.
3. Programación de reembolsos: Una vez aprobado, NetSuite programa los reembolsos para los empleados. Los datos del gasto se integran automáticamente con las cuentas por pagar. El sistema calcula automáticamente los montos finales, incluyendo impuestos aplicables o deducciones.
4. Registro contable: Por último, se realiza el registro contable automáticamente una vez que se procesan los gastos. Se clasifican los gastos según las cuentas contables correspondientes del catálogo de cuentas configurado.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

The screenshot shows the Oracle NetSuite 'Expense Report' form. The interface includes a top navigation bar with 'Sandbox', 'ORACLE NetSuite', and 'Global' logos, along with a search bar and various menu items like 'Activities', 'Pages', 'Transactions', 'Links', 'Reports', 'Analytics', 'Documents', 'Setup', 'Customization', 'Commerce', 'SuiteApps', and 'Support'. The main form area is titled 'Expense Report' and contains several sections:

- Primary Information:** Includes fields for 'Custom Form' (set to 'Standard Expense Report'), 'Exp. Report #', 'To Be Generated', 'Employee', 'Account' (set to 'Informes de gastos no aprobados'), 'Currency', 'Exchange Rate', 'Advance to Apply', and 'Purpose'.
- Date and Period:** Includes 'Date' (set to '17/4/2025'), 'Posting Period' (set to 'Apr 2025'), and 'Date Due'.
- Approval Status:** Includes 'Approval Status' (set to 'Pending Approval') and 'Next Approver'.
- Summary:** A table showing financial totals:

Summary	
Expenses Total	0.00
Total in Base Currency	\$0.00
Reimbursable Expenses	0.00
Corporate Card	0.00
Tax Total	
Advance to Apply	0.00
Total Reimbursable Amount	0.00

Ilustración 3: Formulario de Gastos. Fuente. Captura de pantalla entorno de pruebas NetSuite.

Facturación Electrónica en México

La facturación electrónica es un sistema digital que sustituye el uso de documentos físicos para emitir, recibir y almacenar comprobantes fiscales. En México, se ha convertido en un pilar fundamental del sistema tributario, diseñado para promover la transparencia, combatir la evasión fiscal y facilitar el cumplimiento de las obligaciones fiscales de los contribuyentes.

La facturación electrónica consiste en la emisión de comprobantes fiscales digitales a través de Internet (CFDI), los cuales tienen la misma validez legal que las facturas tradicionales en papel. Este sistema se apoya en tecnologías avanzadas que garantizan la autenticidad y trazabilidad de las transacciones realizadas entre empresas, empleados y consumidores finales.

Hay un riguroso conjunto de procesos que los contribuyentes deben seguir cuando realicen la facturación electrónica en México.

- El archivo XML generado debe cumplir, tanto en formato como en sintaxis, con las especificaciones de la Resolución Miscelánea Fiscal vigente.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

- El documento debe pasar las reglas de validación establecidas tanto para su contenido como para el formato del archivo XML generado.

Es obligatorio que todos los documentos electrónicos, incluidas las facturas electrónicas, se envíen al PCCFDI (o al SAT en casos excepcionales) para su validación. La factura debe incluir determinada información, como se detalla más adelante en la página, para que se considere legalmente válida.

La ley mexicana de facturación no requiere que el destinatario confirme que ha recibido la factura electrónica, aunque debe almacenar los documentos de manera segura durante cinco años a partir del momento en que se presentó la declaración de impuestos correspondiente.

Evolución de la facturación electrónica en México

México es uno de los países pioneros en la implementación de la facturación electrónica en América Latina. El sistema se introdujo de manera gradual para facilitar la transición de los contribuyentes, y su evolución ha sido marcada por los siguientes hitos clave:

1. **2004:** El Servicio de Administración Tributaria (SAT) introduce por primera vez el modelo de factura electrónica como una opción voluntaria para ciertos contribuyentes.
2. **2011:** Se hace obligatorio el uso de comprobantes fiscales digitales (CFDI) para empresas grandes y medianas.
3. **2014:** La facturación electrónica se convierte en una obligación para todos los contribuyentes, incluyendo personas físicas con actividad empresarial.
4. **2017:** Se introduce el modelo CFDI versión 3.3, con nuevos requisitos para mejorar la calidad y precisión de la información contenida en las facturas.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

5. **2022:** Entrada en vigor de la versión 4.0 del CFDI, que incorpora nuevos campos de validación y medidas adicionales para fortalecer el control fiscal.

Objetivo principal

El principal propósito de la facturación electrónica en México es fortalecer la recaudación fiscal mediante un sistema más eficiente y transparente. Al digitalizar los procesos, el gobierno y las empresas obtienen diversos beneficios, como la reducción de costos administrativos, el acceso a información en tiempo real y la simplificación de auditorías fiscales.

CFDI 4.0

La versión más reciente del CFDI en México es 4.0, que ha actualizado las principales características del documento. Esto abarca el nuevo requisito de incluir los nombres del remitente y del destinatario, campos adicionales para los productos exportados y una sección que explica el motivo de la cancelación de la documentación.

Nodos principales en el CFDI 4.0

Si bien, todos los nodos son importantes para la correcta emisión del comprobante fiscal, algunos de los más importantes debido al tipo de información que incluyen:

Nodo	Descripción
Comprobante	Formato estándar del Comprobante Fiscal Digital por Internet.
InformacionGlobal	Expresa la información relacionada con el comprobante global de operaciones con el público en general.
Emisor	debe expresar la información del contribuyente que emite el comprobante fiscal.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Receptor	Expresa la información del contribuyente receptor del comprobante.
Conceptos	Debe expresar los conceptos descritos en el comprobante.
Concepto	Expresa la información detallada de un bien o servicio descrito en el comprobante.
ComplementoConcepto	En este nodo se puede expresar la información adicional específica de los conceptos registrados en la factura electrónica. Dichos Complementos Concepto se encuentran publicados en el Portal del SAT, de acuerdo con las disposiciones particulares para cada sector o actividad específica.
ACuentaTerceros	Puede expresar información del contribuyente tercero, a cuenta del que se realiza la operación. Conforme a la regla 2.7.1.3 de la Resolución Miscelánea Fiscal vigente

Tabla 5: Nodos de un CFDI 4.0. Fuente. Elaboración propia.

En la ilustración 4 se presenta un ejemplo de un archivo XML de un CFDI 4.0, donde se pueden observar los principales elementos que conforman la estructura del comprobante. Este formato XML es generado de manera automática y debe cumplir con las especificaciones establecidas por el Servicio de Administración Tributaria (SAT) para garantizar su validez. Además, el archivo XML contiene la información codificada de la factura electrónica, permitiendo su fácil transmisión y validación dentro de los sistemas de contabilidad y administración de las empresas.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

```
<?xml version="1.0" encoding=UTF-8"?>
<cfdi:Comprobante Fiscal Digital Version="4.0" xmlns:cfdi="http://www.sat.gob.mx/cfdi"
xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sat.gob.mx/cfdi http://www.sat.gob.mx/cfdi/4.0/cfdi.xsd
http://www.sat.gob.mx/TimbreFiscalDigital http://www.sat.gob.mx/TimbreFiscalDigital.xsd">
  <cfdi:Emisor>
    <cfdi:Nombre>Emisor SA de CV</cfdi:Nombre>
    <cfdi:RFC>EMIS123456789</cfdi:RFC>
    <cfdi:RegimenFiscal>601</cfdi:RegimenFiscal>
    <cfdi:LugarExpedicion>99999, Mexico</cfdi:LugarExpedicion>
  </cfdi:Emisor>
  <cfdi:Receptor>
    <cfdi:Nombre>Receptor SA de CV</cfdi:Nombre>
    <cfdi:RFC>RECEP987654321</cfdi:RFC>
    <cfdi:LugarExpedicion>99999, Mexico</cfdi:LugarExpedicion>
  </cfdi:Receptor>
  <cfdi:Impuestos TotalImpuestosRetenidos="0.00" TotalImpuestosTrasladados="16.00">
    <cfdi:Retenciones>
      <cfdi:Retencion Importe="0.00" TipoRetencion="ISR">
        <cfdi:BaseRetencion>0.00</cfdi:BaseRetencion>
      </cfdi:Retencion>
    </cfdi:Retenciones>
    <cfdi:Traslados>
      <cfdi:Traslado Importe="16.00" TipoImpuesto="IVA" TasaImpuesto="16.00">
        <cfdi:BaseImpuesto>100.00</cfdi:BaseImpuesto>
      </cfdi:Traslado>
    </cfdi:Traslados>
  </cfdi:Impuestos>
  <cfdi:TimbreFiscalDigital Version="1.1" xmlns:tfd="http://www.sat.gob.mx/TimbreFiscalDigital">
    <tfd:TimbreDigital>
      <tfd:Version>1.1</tfd:Version>
      <tfd:UUID>7E6492B9-43B7-418B-90C1-B174F2743242</tfd:UUID>
      <tfd:FechaTimbrado>2024-06-04T15:45:55Z</tfd:FechaTimbrado>
    </tfd:TimbreDigital>
  </cfdi:TimbreFiscalDigital>
</cfdi:Comprobante>
```

Ilustración 4: Ejemplo de archivo XML CFDI 4.0. Fuente. Elaboración Propia.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Procedimiento

Kick off

Se inició el proyecto con la asignación formal del equipo de trabajo por parte del líder de proyecto. Posteriormente, se asistió a la reunión grupal, donde se presentó el equipo y se discutieron los pormenores del proyecto.

En una segunda reunión, se presentó la lista de requerimientos funcionales y no funcionales y en base a estos se definieron las actividades específicas y la distribución de las tareas dentro del equipo. También se acordaron los medios de comunicación que se utilizarán a lo largo del proyecto.

Como parte del proceso de formación, se asignó un curso en línea proporcionado por Oracle, enfocado en el uso y desarrollo de SuiteScript, con el objetivo de familiarizarse con la herramienta y aplicarla en el desarrollo del proyecto.

Los cursos mencionados son:

- SuiteScript Foundations: Este curso introductorio de Oracle MyLearn enseña los fundamentos de SuiteScript, el lenguaje de scripting de NetSuite basado en JavaScript. Explora cómo crear scripts para automatizar procesos y personalizar funcionalidades dentro del sistema ERP, abordando conceptos clave como estructuras de código, tipos de scripts y mejores prácticas.
- SuiteScript Development: Diseñado para desarrolladores con conocimientos previos en SuiteScript, este curso profundiza en el desarrollo avanzado de scripts en NetSuite. Cubre temas como la interacción con datos mediante APIs, la automatización de flujos de trabajo y la optimización de código, proporcionando habilidades esenciales para personalizar y extender las capacidades del sistema ERP.

El líder del proyecto fue el encargado de asignar tareas, entregar requerimientos y proporcionar los cursos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Requerimientos del sistema

Los requerimientos representan las capacidades, reglas y comportamientos específicos que el sistema debe cumplir para satisfacer las necesidades del usuario final y garantizar el correcto funcionamiento del proceso de gestión de facturas electrónicas. En el desarrollo de la personalización, estos requerimientos definen con claridad qué funcionalidades debe ofrecer la solución implementada en NetSuite, tales como la carga de archivos CFDI, la validación de su contenido, la extracción automática de datos y la generación de registros de egresos. Son fundamentales porque aseguran que el sistema sea útil, eficiente y alineado con los objetivos operativos, permitiendo una automatización confiable del ciclo de egresos y un cumplimiento adecuado de las normativas fiscales vigentes. A continuación, se detallan los requerimientos establecidos para el desarrollo de esta solución.

1. Carga de Facturas Electrónicas (CFDI)

- Permitir la carga de archivos XML y PDF del CFDI mediante una pantalla Suitelet Assistant.
- Validar que el archivo XML cumple con el estándar CFDI 4.0 antes de su almacenamiento.

2. Identificación Automática del Empleado

- Determinar automáticamente el empleado en sesión para asociarlo con la factura a registrar.

3. Extracción de Datos del XML

- Leer y mostrar de forma automática todos los datos del CFDI.

4. Validación de Datos

- Verificar que el UUID no sea duplicado.
- Revisar que los RFC tenga una longitud válida.
- Validar que los campos de receptor y emisor no estén vacíos.
- Verificar que los datos de fecha sean consistentes y estén dentro de un rango válido, por ejemplo, que no correspondan a fechas futuras no permitidas.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

5. Notificaciones de estados

- Errores.
- Confirmaciones.

6. Almacenamiento de Documentos

- Guardar los archivos XML y PDF en el sistema, vinculándolos con el registro de egreso generado y del empleado.

7. Seguridad

- Garantizar que los documentos cargados sean inalterables tras su registro.

8. Creación Automática del Registro de Gasto

- Generar un registro de egreso en NetSuite con la información extraída del CFDI.
- Asociar la factura con el empleado y el centro de costos correspondiente.

9. Generación de Reportes

- Crear un **Reporte General de Egresos de Empleados** mediante búsquedas guardadas en NetSuite. Permitir el filtrado por empleado y otros criterios como fecha y supervisor.
- Desarrollar un **Workbook** con tablas y gráficos visuales que puedan ser agregados como portlets en el dashboard. Filtrando los gastos del mes.
- Configurar búsqueda y portlets y signar permisos a rol de contador.

Personalización

Como parte del desarrollo del bundle para la recepción y validación de facturas electrónicas de gastos en NetSuite, se realizó el diagramado del sistema. Este análisis gráfico permitió estructurar y visualizar las interacciones clave dentro del flujo de trabajo.

El primer diagrama realizado fue el diagrama de carriles que muestra claramente la distribución de responsabilidades entre los distintos actores involucrados: el empleado, la interfaz de NetSuite, el sistema interno y el servicio web del SAT. A través de este diagrama, se logra identificar cómo las acciones del empleado se apoyan en la interfaz para iniciar la carga del CFDI y recibir retroalimentación sobre errores o confirmaciones. Por su parte, el

Optimización de la Gestión Financiera ... (Ciclo de egresos)

sistema realiza una validación automatizada de archivos, extrae los datos, los valida técnicamente, y posteriormente presenta al usuario la información extraída para una revisión visual.

Finalmente, tras la verificación visual y técnica de los datos, se procede a autenticar el CFDI contra el servicio del SAT, y en caso de que todo sea correcto, se genera el registro del gasto y se almacenan los archivos correspondientes. Este modelo gráfico ha sido clave para garantizar la trazabilidad y robustez del flujo diseñado.

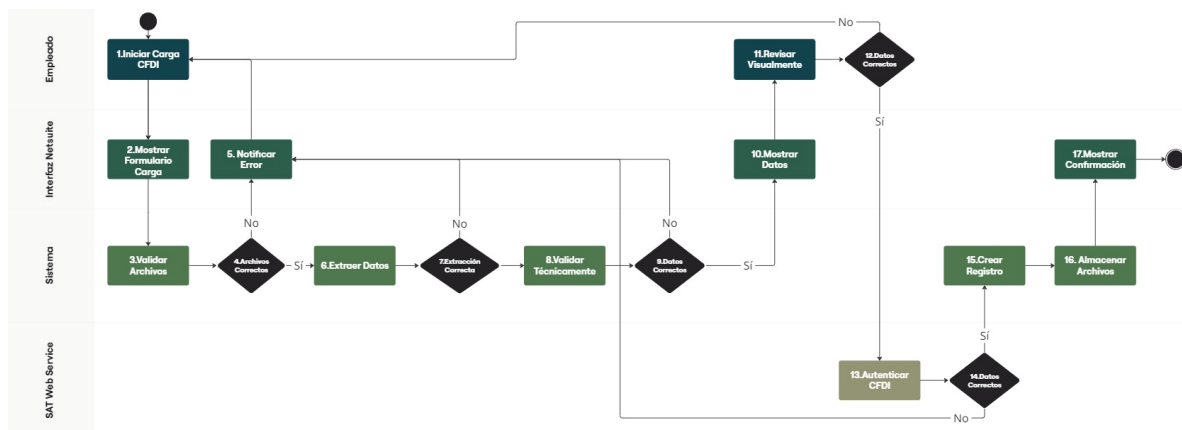


Ilustración 5: Diagrama de carriles. Fuente. Elaboración Propia.

A continuación, se realiza la descripción detallada de cada acción siguiendo el flujo de eventos:

- 1. Iniciar Carga CFDI.** Responsable: Empleado. El proceso comienza cuando el empleado accede a la interfaz de NetSuite e inicia el procedimiento para registrar un gasto, seleccionando y cargando un archivo CFDI (XML).
- 2. Mostrar Formulario Carga.** Responsable: Interfaz NetSuite. La interfaz despliega el formulario donde el empleado puede seleccionar los archivos XML y PDF relacionada con el gasto.
- 3. Validar Archivos.** Responsable: Sistema. Se realiza una primera validación técnica de los archivos cargados: se comprueba que correspondan al tipo de archivo esperado y que se pueda acceder a su contenido.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

4. **Archivos Correctos (Decisión).** Responsable: Sistema. Se determina si los archivos cumplen con los criterios de validez. Si no son correctos, se notifica el error al usuario.
5. **Notificar Error (si aplica).** Responsable: Interfaz NetSuite. En caso de que ocurra algún error, se recibirá el mensaje enviado desde la acción que lo desencadenó con indicando el error sucedido y se mostrará en pantalla con esta acción.
6. **Extraer Datos.** Responsable: Sistema. Si los archivos son válidos, el sistema procede a leer el contenido del CFDI y extraer los datos relevantes como: emisor, receptor, monto, impuestos, conceptos, etc.
7. **Extracción Correcta (Decisión).** Responsable: Sistema. Se valida si la extracción de datos fue exitosa y completa. Si no lo fue, se notifica al usuario el error.
8. **Validar Técnicamente.** Responsable: Sistema. Los datos extraídos se validan contra reglas técnicas y las validaciones establecidas en el requerimiento del sistema 4 (Validación de datos): UUID no duplicado, tipos de dato, campos obligatorios, estructura fiscal del RFC, formato de fecha y códigos.
9. **Datos Correctos (Sistema) (Decisión).** Responsable: Sistema. Se determina si los datos extraídos cumplen con los estándares técnicos internos. Si no son correctos, se informa el error.
10. **Mostrar Datos.** Responsable: Interfaz NetSuite. Si los datos pasan las validaciones técnicas, se presentan en pantalla para que el empleado los revise visualmente antes de continuar.
11. **Revisar Visualmente.** Responsable: Empleado. El empleado revisa los datos extraídos, asegurándose de que coincidan con la información de la factura original. Esto actúa como una segunda validación, ahora visual.
12. **Datos Correctos (Visual) (Decisión).** Responsable: Empleado. El empleado decide si los datos son correctos. Si no lo son, se reinicia el flujo para realizar correcciones.
13. **Autenticar CFDI.** Responsable: SAT Web Service. Se realiza una llamada al servicio web del SAT para verificar la validez del CFDI: existencia, vigencia, estado fiscal y que no haya sido cancelado.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

- 14. Datos Correctos (SAT) (Decisión).** Responsable: SAT Web Service. Se determina si el CFDI es válido según el SAT. Si no lo es, se notifica el error.
- 15. Crear Registro.** Responsable: Sistema. Si todas las validaciones anteriores han sido satisfactorias, se genera un nuevo registro del gasto en el sistema de NetSuite.
- 16. Almacenar Archivos.** Responsable: Sistema. Los archivos XML y PDF se almacenan y se vinculan al registro del gasto para su respaldo y futuras consultas.
- 17. Mostrar Confirmación.** Responsable: Interfaz NetSuite. Se muestra un mensaje final confirmando al empleado que el CFDI fue validado correctamente, registrado exitosamente en el sistema y los archivos almacenados.

Como parte del desarrollo del módulo de reportes para el seguimiento de egresos de empleados, se implementó la creación de búsquedas guardadas (Saved Searches) y workbooks como herramientas de análisis dentro de NetSuite. Esta solución permite transformar los datos del sistema en información visual y filtrable que apoya la toma de decisiones.

El proceso técnico puede dividirse en dos grandes componentes:

1. Búsquedas Guardadas (Saved Searches)

Las búsquedas guardadas se configuran desde el entorno de NetSuite. El proceso técnico realizado por el administrador incluye:

1. Seleccionar el tipo de registro principal (en este caso, “Transacción”).
2. Definir el criterio de búsqueda (en este caso tipo “Expense Report”).
3. Definir criterios de filtro: selección de campos como empleado, fecha y monto.
4. Establecer columnas de resultados: especificar los datos que se desean visualizar (por ejemplo, nombre del empleado, total del gasto, fecha, categoría).

Optimización de la Gestión Financiera ... (Ciclo de egresos)

5. Configurar permisos: habilitar la visibilidad para ciertos roles o usuarios, y marcar como disponible para usar en Workbooks.



Ilustración 6: Diagrama de flujo de creación de búsqueda guardada. Fuente. Elaboración propia.

De modo que la interacción del usuario final (rol de contador) es la siguiente:

1. Inicio de sesión: El usuario inicia sesión en la plataforma.
2. Abrir la búsqueda guardada: Ir a la ruta reportes -> Búsquedas Guardadas -> Seleccionar la búsqueda **“Reporte General de Egresos de Empleados”**.
3. Aplicar filtros: Aplicar los filtros que requiera.
4. Ver resultados: Ver la lista de los resultados de la búsqueda.

Optimización de la Gestión Financiera ... (Ciclo de egresos)



Ilustración 7: Interacción usuario final búsqueda guardada. Fuente. Elaboración propia.

2. Workbooks

Los workbooks permiten representar los datos extraídos mediante búsquedas guardadas de forma visual, interactiva y dinámica. El proceso técnico realizado por el administrador incluye:

1. Seleccionar dataset base: usar una búsqueda guardada como fuente o crear un nuevo dataset a partir de registros del sistema.
2. Agregar filtros dinámicos: habilitar el filtrado en tiempo real por mes.
3. Construcción de vistas: generar tablas, gráficos de barras, líneas o circulares para mostrar tendencias o comparaciones.
4. Publicar en dashboard: añadir las vistas creadas como portlets en dashboards personalizados de empleados o supervisores para análisis rápido.

Optimización de la Gestión Financiera ... (Ciclo de egresos)



Ilustración 8: Diagrama de flujo de creación de workbook. Fuente. Elaboración propia.

Este enfoque modular garantiza que los datos de egresos sean accesibles, analíticos y estén disponibles para los diferentes perfiles dentro de la empresa.

De modo que la interacción del usuario final (rol de contador) es la siguiente:

1. Inicio de sesión: El usuario inicia sesión en la plataforma.
2. Acceder a dashboard. El usuario va al panel de inicio con los portlets ya configurados.
3. Ver resultados: Ver los portlets gráficos o tabla de datos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)



Ilustración 9: Interacción usuario final portlets. Fuente. Elaboración propia.

En el desarrollo del módulo de reportes, el rol del administrador del sistema se limita a la creación, configuración y publicación de las herramientas analíticas, como las búsquedas guardadas y los workbooks. Sin embargo, el uso operativo posterior de estas herramientas como la exploración de datos, la aplicación de filtros dinámicos o la interpretación de gráficos queda en manos del usuario final, quien accede a esta información desde su dashboard para tomar decisiones o realizar seguimiento de los egresos. Esta separación de funciones garantiza una adecuada gobernanza de los datos, facilitando su disponibilidad sin comprometer la integridad ni la estructura base del reporte.

Con los diagramas en la ilustración 5, ilustración 6 e ilustración 8, se consiguió una visión más clara y estructurada del módulo desarrollado y de las herramientas a configurar. Este análisis permitió definir con mayor precisión los componentes y funcionalidades clave, garantizando una implementación adecuada en NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Proceso de desarrollo

El proceso de desarrollo dio inicio con la creación de la pantalla Suitelet, la cual está pensada como interfaz principal para la recepción y validación de facturas electrónicas de gastos. Esta decisión se basó en el requerimiento (Carga de Facturas Electrónicas), para permitir la carga de archivos XML y PDF del CFDI de forma guiada, validando que el archivo XML cumpla con el estándar CFDI 4.0 desde el inicio y cumpliendo con el requerimiento de usabilidad brindando un proceso intuitivo y guiado.

Subida de la Factura Electrónica (XML y PDF)

Este script fue desarrollado como un Suitelet utilizando la versión 2.1 del API de NetSuite, lo cual permite crear interfaces personalizadas para la interacción con el usuario. Es ideal para construir formularios dinámicos que guíen al usuario paso a paso.

```
/**
 * @NApiVersion 2.1
 * @NScriptType Suitelet
 */
```

Ilustración 10: Definición de versión API y tipo de script. Fuente. Elaboración propia fragmento de código.

De igual modo, se definieron los módulos de SuiteScript que se usarán en el script:

```
define(['N/ui/serverWidget', 'N/runtime'],
```

Ilustración 11: Definición de módulos SuiteScript. Fuente. Elaboración propia fragmento de código.

El Suitelet inicia creando una instancia del Form, con un título personalizado que indica el propósito del flujo.

```
var form = serverWidget.createForm({
  title: 'Facturación Electrónica para Empleados: Carga de Archivos'
});
```

Ilustración 12: Creación de objeto form. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

A continuación, se define el campo de ayuda indicando las instrucciones al usuario.

```
var helpField = form.addField({
  id: 'custpage_help_text',
  type: serverWidget.FieldType.HELP,
  label: 'Instrucciones: Por favor, sube tu CFDI en formato XML y PDF antes de continuar con la validación de los datos.'
});
helpField.updateLayoutType({ layoutType: serverWidget.FieldLayoutType.STARTROW });
```

Ilustración 13: Creación del campo de ayuda. Fuente. Elaboración propia fragmento de código.

En este paso, en cumplimiento del requerimiento (Identificación Automática del Empleado) el sistema determina automáticamente qué empleado está loggeado usando el método `runtime.getCurrentUser()` del módulo `N/runtime`:

```
// Obtener el ID del usuario loggeado
var currentUser = runtime.getCurrentUser();
var employeeId = currentUser.id;
```

Ilustración 14: Obtención de empleado. Fuente. Elaboración propia fragmento de código.

Luego, se agrega un campo tipo `SELECT` para mostrar al empleado actual, el cual está prellenado y se muestra como solo lectura.

```
// Mostrar el empleado facturando (obtenido automáticamente)
var empleadoField = assistant.addField({
  id: 'custpage_empleado',
  type: serverWidget.FieldType.SELECT,
  label: 'Empleado Facturando:',
  source: 'employee'
});
empleadoField.defaultValue = employeeId;
empleadoField.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
```

Ilustración 15: Mostrar nombre de empleado. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Por último, se habilitan los campos para la subida obligatoria del XML y del PDF, cumpliendo con el requerimiento (Carga de Facturas Electrónicas):

```
// Campos para subir archivos
var fileXMLField = assistant.addField({
  id: 'custpage_cfdi_file_xml',
  type: serverWidget.FieldType.FILE,
  label: 'Subir CFDI (XML)'
});
fileXMLField.isMandatory = true;

var filePDFField = assistant.addField({
  id: 'custpage_cfdi_file_pdf',
  type: serverWidget.FieldType.FILE,
  label: 'Subir CFDI (PDF)'
});
filePDFField.isMandatory = true;
```

Ilustración 16: Campos de carga de archivo. Fuente. Elaboración propia fragmento de código.

El resultado fue la pantalla siguiente:

Ilustración 17: Pantalla carga de archivos. Fuente. Captura de pantalla entorno de pruebas NetSuite.

A través de esta pantalla, se buscó facilitar un flujo estructurado e intuitivo que permitiera identificar automáticamente al empleado en sesión, subir los archivos del CFDI, y posteriormente guiar al usuario en la validación y registro automatizado del egreso, asegurando al mismo tiempo el almacenamiento adecuado de los archivos.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Además, cuenta con el manejo de errores y notificaciones al usuario. Realizándose mediante un script de cliente ligado al anterior suitlet.

```
/**
 * @NApiVersion 2.1
 * @NScriptType ClientScript
 */
```

Ilustración 18: Definición de versión API y tipo de script. Fuente. Elaboración propia fragmento de código.

De igual modo, se definió el módulo de SuiteScript que se usará en el script:

```
define(['N/ui/message'], function(message) {
```

Ilustración 19: Definición de módulo SuiteScript. Fuente. Elaboración propia fragmento de código.

A través de la función saveRecord, se reciben ambos archivos, se comprueba su extensión y se devuelve un valor booleano falso en caso de ser incorrectos, de lo contrario se devuelve el valor true:

```
function saveRecord(context) {
    var fileXML = context.currentRecord.getValue({ fieldId: 'custpage_cfdi_file_xml' });
    var filePDF = context.currentRecord.getValue({ fieldId: 'custpage_cfdi_file_pdf' });

    if (fileXML && !fileXML.toLowerCase().endsWith('.xml')) {
        showMessage('Error', 'Solo se permiten archivos XML en el campo de CFDI XML.', 'ERROR');
        return false;
    }

    if (filePDF && !filePDF.toLowerCase().endsWith('.pdf')) {
        showMessage('Error', 'Solo se permiten archivos PDF en el campo de CFDI PDF.', 'ERROR');
        return false;
    }

    return true;
}
```

Ilustración 20: Validación de archivos XML y PDF. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

The screenshot shows the NetSuite interface with a navigation bar at the top containing 'Sandbox', 'ORACLE NetSuite', 'ITGlobal', a search bar, and icons for Help, Feedback, and a user profile. Below the navigation bar is a menu with 'Activities', 'Pagos', 'Transactions', 'Lists', 'Reports', 'Analytics', 'Documents', 'Setup', 'Customization', 'Commerce', 'SuiteApps', and 'Support'. A red error banner at the top of the main content area reads: 'Error Solo se permiten archivos XML en el campo de CFDI XML.' The main content area is titled 'Facturación Electrónica para Empleados: Carga de Archivos' and includes a 'Cargar Archivos' button and a 'Cancelar' button. Below the buttons, instructions state: 'Instrucciones: Por favor, sube tu CFDI en formato XML y PDF antes de continuar con la validación de los datos.' There is a text input field labeled 'Empleado Facturando:' and a 'More' button with a question mark icon. On the right side, there are two file upload sections: 'Subir CFDI (XML) *' with a 'Seleccionar archivo' button and an empty text box, and 'Subir CFDI (PDF) *' with a 'Seleccionar archivo' button and the text 'Sin archivos seleccionados'.

Ilustración 21: Mensaje de error archivo XML. Fuente. Captura de pantalla entorno de pruebas NetSuite.

The screenshot shows the NetSuite interface with a navigation bar at the top containing 'Sandbox', 'ORACLE NetSuite', 'ITGlobal', a search bar, and icons for Help, Feedback, and a user profile. Below the navigation bar is a menu with 'Activities', 'Pagos', 'Transactions', 'Lists', 'Reports', 'Analytics', 'Documents', 'Setup', 'Customization', 'Commerce', 'SuiteApps', and 'Support'. A red error banner at the top of the main content area reads: 'Error Solo se permiten archivos PDF en el campo de CFDI PDF.' The main content area is titled 'Facturación Electrónica para Empleados: Carga de Archivos' and includes a 'Cargar Archivos' button and a 'Cancelar' button. Below the buttons, instructions state: 'Instrucciones: Por favor, sube tu CFDI en formato XML y PDF antes de continuar con la validación de los datos.' There is a text input field labeled 'Empleado Facturando:' and a 'More' button with a question mark icon. On the right side, there are two file upload sections: 'Subir CFDI (XML) *' with a 'Seleccionar archivo' button and the text 'Sin archivos seleccionados', and 'Subir CFDI (PDF) *' with a 'Seleccionar archivo' button and an empty text box.

Ilustración 22: Mensaje de error archivo PDF. Fuente. Captura de pantalla entorno de pruebas NetSuite.

Extracción de datos del XML

En cumplimiento del requerimiento (Extracción de Datos del XML), una vez validados los archivos se extraen los datos del XML para primero pasar por validaciones internas cumpliendo el requerimiento (Validación de Datos) y para que posteriormente el usuario los visualice y proceder al registro.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Para esto, se creó otro script de cliente que recibe el archivo XML, para luego extraer la información del CFDI y enviarla en formato JSON al Suitelet.

```
/**
 * @NApiVersion 2.1
 * @NScriptType ClientScript
 */
```

Ilustración 23: Definición de versión API y tipo de script. Fuente. Elaboración propia fragmento de código.

Se obtiene el campo de entrada del archivo XML y se valida si existe y contiene un archivo seleccionado. Si no, se muestra un mensaje de alerta al usuario.

```
const xmlField = document.getElementById('custpage_cfdi_file_xml');
if (!xmlField || !xmlField.files || xmlField.files.length === 0) {
    alert('Por favor selecciona un archivo XML válido.');
```

Ilustración 24: Validación del archivo XML. Fuente. Elaboración propia fragmento de código.

Se extrae el contenido del archivo XML como una cadena de texto, se convierte en un documento XML y se define el espacio de nombres (namespace) para interpretar correctamente las etiquetas del CFDI.

```
const file = xmlField.files[0];
const xmlString = await file.text();
const parser = new DOMParser();
const xmlDoc = parser.parseFromString(xmlString, 'text/xml');
const ns = "http://www.sat.gob.mx/cfd/4";
```

Ilustración 25: Procesamiento del contenido del archivo XML. Fuente. Elaboración propia fragmento de código.

Se obtiene la información del nodo Comprobante, extrayendo datos clave como la versión, fecha, total y tipo de comprobante.

```
// Comprobante
const comprobante = xmlDoc.getElementsByTagNameNS(ns, "Comprobante")[0];
const comprobanteData = {
    Version: getAttr(comprobante, "Version"),
    Fecha: getAttr(comprobante, "Fecha"),
    Moneda: getAttr(comprobante, "Moneda"),
    Total: getAttr(comprobante, "Total"),
    SubTotal: getAttr(comprobante, "SubTotal"),
    TipoDeComprobante: getAttr(comprobante, "TipoDeComprobante"),
    FormaPago: getAttr(comprobante, "FormaPago")
};
```

Ilustración 26: Extracción de atributos del nodo Comprobante. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Se extraen los datos del nodo Emisor y Receptor, tales como el RFC, el nombre y los regímenes fiscales correspondientes.

```
// Emisor
const emisor = xmlDoc.getElementsByTagNameNS(ns, "Emisor")[0];
const emisorData = {
  Rfc: getAttr(emisor, "Rfc"),
  Nombre: getAttr(emisor, "Nombre"),
  RegimenFiscal: getAttr(emisor, "RegimenFiscal")
};

// Receptor
const receptor = xmlDoc.getElementsByTagNameNS(ns, "Receptor")[0];
const receptorData = {
  Rfc: getAttr(receptor, "Rfc"),
  Nombre: getAttr(receptor, "Nombre"),
  DomicilioFiscalReceptor: getAttr(receptor, "DomicilioFiscalReceptor"),
  RegimenFiscalReceptor: getAttr(receptor, "RegimenFiscalReceptor"),
  UsoCFDI: getAttr(receptor, "UsoCFDI")
};
```

Ilustración 27: Extracción de datos de los nodos Emisor y Receptor. Fuente. Elaboración propia fragmento de código

Se procesa el nodo Concepto, obteniendo los atributos de cada concepto y sus impuestos asociados.

```
// Conceptos
const conceptos = Array.from(xmlDoc.getElementsByTagNameNS(ns, "Concepto")).map(concepto => {
  const impuestosNode = concepto.getElementsByTagNameNS(ns, "Impuestos")[0];
  const traslados = impuestosNode ?
    Array.from(impuestosNode.getElementsByTagNameNS(ns, "Traslado")).map(t => ({
      Base: getAttr(t, "Base"),
      Impuesto: getAttr(t, "Impuesto"),
      TipoFactor: getAttr(t, "TipoFactor"),
      TasaOCuota: getAttr(t, "TasaOCuota"),
      Importe: getAttr(t, "Importe")
    })) : [];

  return {
    ClaveProdServ: getAttr(concepto, "ClaveProdServ"),
    NoIdentificacion: getAttr(concepto, "NoIdentificacion"),
    Cantidad: getAttr(concepto, "Cantidad"),
    ClaveUnidad: getAttr(concepto, "ClaveUnidad"),
    Unidad: getAttr(concepto, "Unidad"),
    Descripcion: getAttr(concepto, "Descripcion"),
    ValorUnitario: getAttr(concepto, "ValorUnitario"),
    Importe: getAttr(concepto, "Importe"),
    Descuento: getAttr(concepto, "Descuento"),
    Impuestos: traslados
  };
});
```

Ilustración 28: Procesamiento de los Conceptos. Fuente. Elaboración propia fragmento de código

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Se organiza toda la información extraída en un objeto JSON que será enviado al servidor para su uso posterior.

```
// JSON Final
const cfdiJSON = {
  comprobante: comprobanteData,
  emisor: emisorData,
  receptor: receptorData,
  conceptos: conceptos,
  impuestosGenerales: impuestosGenerales,
  complemento: complemento
};
```

Ilustración 29: Construcción del JSON final. Fuente. Elaboración propia fragmento de código

Par enviar el JSON al Suitelet en Netsuite se utiliza una solicitud POST, que se encargará de procesarlo en el servidor.

```
// Enviar a Suitelet
await fetch('/app/site/hosting/scriptlet.nl?script=863&deploy=1', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify(cfdiJSON)
});
```

Ilustración 30: Envío de datos al Suitelet. Fuente. Elaboración propia fragmento de código

El proceso anterior está contenido dentro del bloque try, encargado del manejo de errores. Si todo se ejecuta correctamente, el JSON generado se enviará a la siguiente pantalla. Sin embargo, si ocurre un error, esto indica que el parser no pudo encontrar uno o más nodos esperados dentro del archivo XML. Esto sugiere que el usuario probablemente cargó un archivo XML incorrecto (que no corresponde al de una factura electrónica), incompleto o que contiene inconsistencias. En tal caso, se mostrará un mensaje de error en la misma pantalla, notificando al usuario sobre el problema. Permite recargar la página para volver a intentarlo.

```
} catch (error) {
  if (error.message.includes('nodos faltantes')) {
    showMessage(form, 'Error', 'El archivo XML está incompleto o no tiene los nodos esperados. Verifica que el archivo sea v');
  } else {
    showMessage(form, 'Error', 'El archivo proporcionado no es un XML válido de una factura electrónica.', 'ERROR');
  }
  console.error('Detalles del error:', error);
}
```

Ilustración 31: Manejo de errores extracción de datos. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

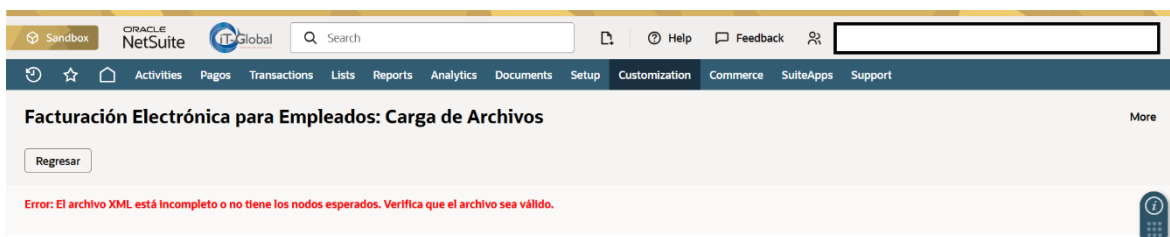


Ilustración 32: Pantalla de error XML incompleto. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

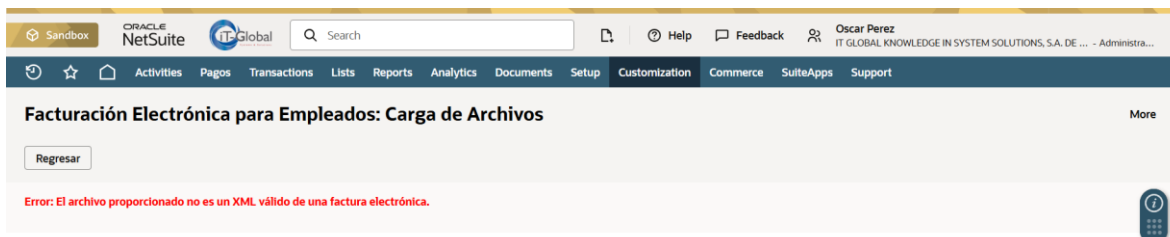


Ilustración 33: Pantalla de error XML inválido. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Recepción y validación de datos extraídos

Ahora tendremos que preparar la otra pantalla que recibirá el JSON con los datos extraídos. Primero definiendo la versión de API, tipo de script y módulos importados.

```
/**
 * @NApiVersion 2.1
 * @NScriptType Suitelet
 */
define(['N/ui/serverWidget', 'N/runtime', 'N/log'], (serverWidget, runtime, log) => {
```

Ilustración 34: Definición de API, tipo de script y módulos. Fuente elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Iniciamos creando una instancia de form que contendrá todos los campos con los datos extraídos del XML.

```
var form = serverWidget.createForm({
  title: 'Verifica los datos extraídos'
});
```

Ilustración 35: Creación de instancia form. Fuente elaboración propia fragmento de código.

Ahora recibiremos los datos enviados desde el client script, procesando solicitudes con el método POST. Luego, obtiene el cuerpo de la solicitud y lo registra para inspección. Si el cuerpo contiene datos, intenta convertirlo de formato JSON a un objeto JavaScript. Si ocurre un error durante el proceso de conversión, captura la excepción y registra un mensaje de error con los detalles del problema.

```
let cfdiData = {};

// Leer datos del POST
if (request.method === 'POST') {
  log.debug('Método POST recibido');
  try {
    const body = request.body;
    log.debug('Request Body:', body); // Mostrar el contenido recibido
    if (body) {
      cfdiData = JSON.parse(body);
      log.debug('Datos CFDI recibidos:', cfdiData);
    }
  } catch (error) {
    log.error('Error al parsear JSON:', error);
  }
}
```

Ilustración 36: Recibir JSON enviado. Fuente. Elaboración propia fragmento de código.

Luego se verifica si no se han recibido datos del CFDI y muestra un mensaje de advertencia al usuario en la interfaz. Se utiliza la función showMessage que contiene un mensaje indicando que no se ha proporcionado información del CFDI.

```
if (!cfdiData || !cfdiData.comprobante) {
  showMessage('Error', 'No se ha recibido información del CFDI.', 'ERROR');
}
```

Ilustración 37: Validación de datos CFDI y generación de advertencia. Fuente elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

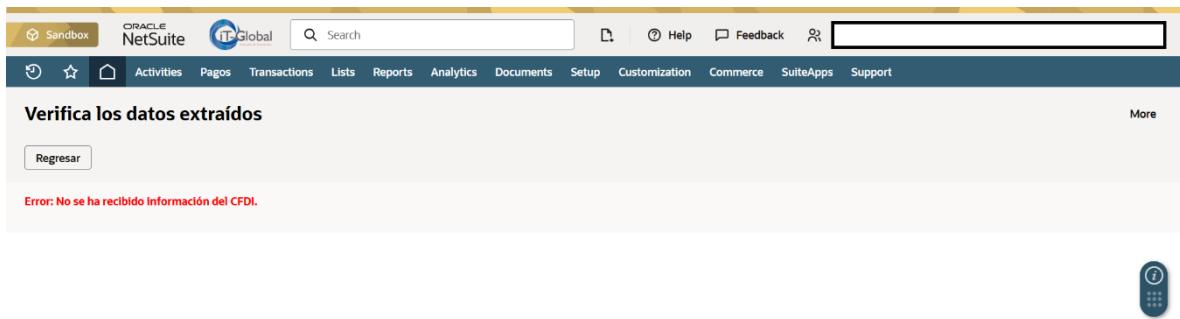


Ilustración 38: Pantalla de error datos no recibidos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Una vez recibidos los datos realizaremos las validaciones internas descritas en el requerimiento (Validación de Datos). Para eso, se crea la siguiente función recibirá el objeto con el objeto parseado.

Tiene como objetivo garantizar la validez e integridad de los datos CFDI proporcionados en formato JSON. Realiza validaciones específicas, incluyendo que el UUID sea único en los registros de NetSuite, utilizando una búsqueda de duplicados basada en el campo correspondiente. Además, se asegura de que los RFC del emisor y receptor cumplan con los formatos y longitudes esperados, que las fechas sean coherentes y no futuras, y que los importes como total y subtotal sean numéricos y coherentes. Si alguna validación falla, lanza un error con un mensaje claro para que sea identificado en la pantalla.

```
function validateCFDI(cfdiData) {  
  if (!cfdiData.comprobante || !cfdiData.emisor || !cfdiData.receptor) {  
    throw new Error('El JSON recibido no contiene los nodos esenciales: comprobante, emisor y receptor.');  }  
  
  // Validar UUID  
  const uuid = cfdiData.comprobante.UUID;  
  if (!uuid || uuid.length !== 36) {  
    throw new Error('El UUID proporcionado no es válido o está ausente.');  }  
  
  // Verificar que el UUID no sea duplicado en los registros de NetSuite  
  const duplicateUUID = searchDuplicateUUID(uuid);  
  if (duplicateUUID) {  
    throw new Error('Ya existe una factura registrada con el UUID: ${uuid}.');  }  
}
```

Ilustración 39: Fragmento de función validateCFDI. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Para la validación de UUID duplicados se usó la siguiente función. Utiliza una búsqueda con filtros para localizar coincidencias en el campo uuid, recuperando el uuid de los resultados. Si encuentra algún registro con el UUID proporcionado, devuelve true, indicando que el documento ya existe, evitando así duplicaciones. En caso de error durante la búsqueda, captura la excepción y lanza un mensaje de advertencia, garantizando la integridad de los datos antes de procesarlos.

```
function searchDuplicateUUID(uuid) {  
  try {  
    const searchResult = search.create({  
      type: 'expensereport',  
      filters: [  
        ['custbody_uuid', 'is', uuid]  
      ],  
      columns: ['internalid']  
    }).run().getRange({ start: 0, end: 1 });  
  
    return searchResult.length > 0;  
  } catch (error) {  
    log.error('Error al buscar UUID duplicado en gastos de empleados:', error);  
    throw new Error('Error al verificar duplicados en los registros de gastos de empleados en NetSuite.');  }  
}
```

Ilustración 40: Función searchDuplicateUUID. Fuente. Elaboración propia fragmento de código.

Cualquier error en algún campo se mostrará en una pantalla como la siguiente.

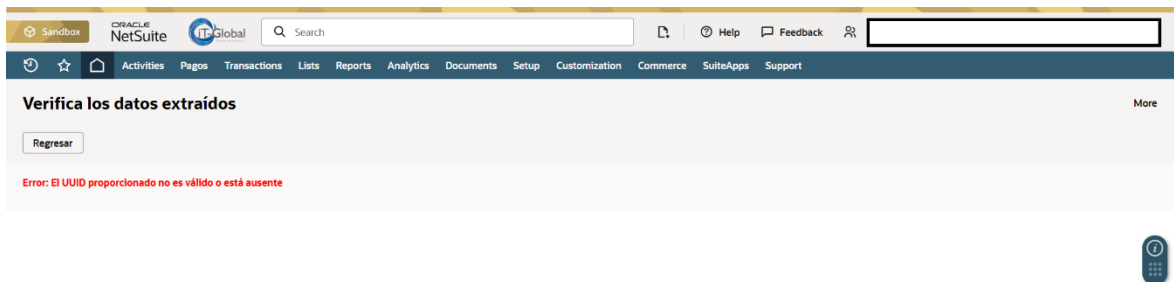


Ilustración 41: Pantalla de error validación de campos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Se llama la función `validateCFDI` dentro de un bloque `try catch` para el manejo de errores y se almacena en la variable `estadoValidacion` el valor booleano que retorna (`true`) en caso de pasar las validaciones.

```
//Validación interna de datos
try {
    estadoValidacion = validateCFDI(cfdiData);
} catch (error) {
    log.error('Error:', error);
    showError(context.response, error.message);
    return;
}
```

Ilustración 42: Llamada de función `validateCFDI`. Fuente. Elaboración propia fragmento de código.

Después, se comprueba el valor que tiene estado de validación, si es `true`, ha pasado la validación y ahora estamos listos para mostrarle los datos al usuario.

Visualización de datos extraídos

Con el fin de brindar una experiencia al usuario correcta y ordenada creamos grupos para los campos, estos nos servirán para organizarlos conforme a los nodos del XML del CFDI.

```
// Creación de grupos de datos
var groupComprobante = form.addFieldGroup({ id: 'custpage_grp_comprobante', label: 'Comprobante' });
var groupEmisor = form.addFieldGroup({ id: 'custpage_grp_emisor', label: 'Emisor' });
var groupReceptor = form.addFieldGroup({ id: 'custpage_grp_receptor', label: 'Receptor' });
var groupConceptos = form.addFieldGroup({ id: 'custpage_grp_conceptos', label: 'Conceptos' });
var groupImpuestos = form.addFieldGroup({ id: 'custpage_grp_impuestos', label: 'Impuestos' });
var groupComplemento = form.addFieldGroup({ id: 'custpage_grp_complemento', label: 'Complemento' });
```

Ilustración 43: Creación de grupos. Fuente elaboración propia fragmento de código.

Se generan los campos correspondientes para cada uno de los datos obtenidos de los nodos del XML y se les asigna el valor extraído, asegurando que la información sea procesada y visualizada correctamente en el formulario. Se muestra únicamente el ejemplo de la

Optimización de la Gestión Financiera ... (Ciclo de egresos)

creación de un campo y la asignación de su valor extraído del XML. Sin embargo, este mismo procedimiento se aplicó de forma consistente para todos los campos correspondientes a los datos extraídos de los nodos del XML, garantizando la asignación correcta de cada atributo en el formulario.

```
const versionField = form.addField({
  id: 'custpage_version',
  type: serverWidget.FieldType.TEXT,
  label: 'Versión',
  container: 'custpage_grp_comprobante'
});
versionField.defaultValue = cfdiData.comprobante.Version || '';
```

Ilustración 44: Ejemplo de creación de campo y asignación de valor. Fuente. Elaboración propia fragmento de código.

A continuación, se presentan capturas de pantalla que muestran el resultado final de la interfaz generada, en la cual se incluyen todos los campos creados con los datos extraídos del XML. En esta etapa, el usuario tiene la posibilidad de revisar visualmente la información antes de proceder. Para continuar con el proceso, se ha implementado un botón con la etiqueta "Confirmar", acompañado de un mensaje informativo que explica claramente la acción que se realizará: al hacer clic, se enviará la factura para su validación ante el SAT. Esta combinación de mensaje y acción busca brindar claridad, confianza y control al usuario final antes de confirmar la carga del CFDI.

Una vez que el empleado revise y confirme la información extraída del XML, el sistema procederá con la validación formal a través del web service del SAT para asegurarse de que el comprobante es auténtico y cumple con las regulaciones fiscales.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Verifica los datos extraídos

Confirmar Cancelar

Comprobante 1

Versión 1
Fecha 2
Moneda 3

Total 4
Subtotal 5

Tipo de Comprobante 6
Forma de Pago 7

Emisor 2

RFC Emisor 1
Nombre Emisor 2
Régimen Fiscal Emisor 3

Receptor 3

RFC Receptor 1
Nombre Receptor 2

Domicilio Fiscal Receptor 3
Régimen Fiscal Receptor 4

Uso CFDI 5

Ilustración 45: Pantalla de visualización de datos (parte 1). Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Conceptos 4

Clave ProdServ 1
No Identificación 2
Cantidad 3
Clave Unidad 4
Unidad 5

Descripción 6
Valor Unitario 7
Importe 8

Descuento 9
Base 10
Impuesto 11
Tipo Factor 12
Tasa o Cuota 13
Importe Impuesto 14

Ilustración 46: Pantalla de visualización de datos (parte 2). Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Impuestos 5

Total Impuestos Traslados **1**

Base (General) **2**

Impuesto (General) **3**

Tipo Factor (General) **4**

Tasa o Cuota (General) **5**

Importe (General) **6**

Complemento 6

Versión (Complemento) **1**

UUID **2**

Fecha Timbrado **3**

RFC Proveedor Certificado **4**

No. Certificado SAT **5**

Atención

Al hacer clic en **Confirmar**, se enviará la factura para su validación ante el SAT. Si el CFDI es válido, el sistema registrará el egreso y almacenará los archivos. Si no es válido, se te notificará con el motivo del error y no se realizará ningún registro.

Confirmar **Cancelar**

Ilustración 47: Pantalla de visualización de datos (parte 3). Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Descripción de campos:

1. Comprobante

1. Versión: Indica la versión del CFDI.
2. Fecha: Fecha de emisión del comprobante.
3. Moneda: Moneda utilizada en la transacción.
4. Total: Importe total del comprobante.
5. Subtotal: Importe antes de impuestos.
6. Tipo de Comprobante: Tipo de operación (Ingreso, Egreso, etc.).
7. Forma de Pago: Método de pago utilizado.

2. Emisor

1. RFC Emisor: Registro Federal de Contribuyentes del emisor.
2. Nombre Emisor: Nombre del emisor del comprobante.
3. Régimen Fiscal Emisor: Régimen fiscal del emisor.

3 Receptor

1. RFC Receptor: Registro Federal de Contribuyentes del receptor.
2. Nombre Receptor: Nombre del receptor del comprobante.
3. Domicilio Fiscal Receptor: Dirección fiscal del receptor.
4. Régimen Fiscal Receptor: Régimen fiscal del receptor.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

5. Uso CFDI: Uso que el receptor dará al CFDI.

4 Conceptos

1. Clave ProdServ: Identificador del producto o servicio.
2. No Identificación: Número de identificación del producto o servicio.
3. Cantidad: Cantidad de productos o servicios.
4. Clave Unidad: Clave de la unidad de medida.
5. Unidad: Descripción de la unidad de medida.
6. Descripción: Descripción del producto o servicio.
7. Valor Unitario: Precio unitario.
8. Importe: Importe total.
9. Descuento: Descuento aplicado.
10. Base: Importe base de impuestos.
11. Impuesto: Tipo de impuesto aplicado.
12. Tipo Factor: Factor del cálculo de impuestos.
13. Tasa o Cuota: Tasa aplicada al impuesto.
14. Importe Impuesto: Importe del impuesto.

5 Impuestos

1. Total Impuestos Traslados: Suma total de impuestos trasladados.
2. Base (General): Base imponible para impuestos generales.
3. Impuesto (General): Tipo de impuesto general aplicado.
4. Tipo Factor (General): Factor del cálculo para impuestos generales.
5. Tasa o Cuota (General): Tasa aplicada a impuestos generales.
6. Importe (General): Importe de impuestos generales.

6 Complemento

1. Versión (Complemento): Versión del complemento.
2. UUID: Identificador único del comprobante.
3. Fecha Timbrado: Fecha en que se timbró el CFDI.
4. RFC Proveedor Certificado: RFC del proveedor del certificado.
5. No. Certificado SAT: Número del certificado emitido por el SAT.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Validación ante el SAT

El Servicio de Consulta de CFDI proporcionado por el SAT permite verificar el estado de un comprobante fiscal digital a través de una conexión con su web service.

Este servicio utiliza una consulta estructurada en formato SOAP, enviando parámetros esenciales del CFDI como el UUID, RFC del Emisor, RFC del Receptor, Total del comprobante y una porción del sello digital. A partir de estos datos, el SAT responde con el estado del comprobante, que puede ser:

- Vigente: El comprobante es válido y está registrado correctamente.
 - Cancelado: La factura ha sido anulada por el emisor.
 - Cancelable con aceptación: Requiere aceptación del receptor para ser cancelada.
 - Cancelable sin aceptación: Puede ser cancelada sin aprobación del receptor.

Para la validación, se creó un client script responsable de validar el CFDI antes de registrarlo como un gasto de empleado. Primero, verifica el estado del CFDI a través del web service del SAT, asegurándose de que el UUID no esté duplicado y que la factura sea válida. Si el comprobante está vigente, se procede con la creación del registro en NetSuite; si está cancelado, se muestra un mensaje de error en pantalla.

Esta función envía los datos clave del CFDI (UUID, RFC del emisor y receptor, total) al web service del SAT, construyendo una petición SOAP. Luego, procesa la respuesta para determinar si el CFDI está Vigente o Cancelado, devolviendo el estado correspondiente.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

```
async function consultarCFDI(uuid, rfcEmisor, rfcReceptor, total) {
  const sello = total.slice(-8);
  const soapRequest = `
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/">
      <soapenv:Header/>
      <soapenv:Body>
        <tem:Consulta>
          <tem:expresionImpresa><![CDATA[?re=${rfcEmisor}&rr=${rfcReceptor}&tt=${total}&id=${uuid}&fe=${sello}]]></tem:expresionImpresa>
        </tem:Consulta>
      </soapenv:Body>
    </soapenv:Envelope>`;

  try {
    const response = await fetch("https://consultaqr.facturaelectronica.sat.gob.mx/ConsultaCFDIService.svc", {
      method: "POST",
      headers: {
        "Content-Type": "text/xml; charset=utf-8",
        "SOAPAction": "http://tempuri.org/IConsultaCFDIService/Consulta"
      },
      body: soapRequest
    });

    const textResponse = await response.text();
    if (textResponse.includes('<Estado>Vigente</Estado>')) return 'Vigente';
    if (textResponse.includes('<Estado>Cancelado</Estado>')) return 'Cancelado';
    return 'Desconocido';
  } catch (error) {
    console.error('Error en la consulta al SAT:', error);
    return 'Error';
  }
}
```

Ilustración 48: Consulta del CFDI en el SAT. Fuente. Elaboración propia fragmento de código.

La siguiente función es el núcleo del proceso de validación, ya que toma los datos del CFDI y consulta su estado en el web service del SAT. Si el comprobante está vigente, el sistema procede a registrar un gasto de empleado en NetSuite; si está cancelado, se interrumpe la operación y se informa al usuario.

```
async function validarCFDI(cfdiData) {
  try {
    const estadoCFDI = await consultarCFDI(cfdiData.comprobante.UUID, cfdiData.emisor.Rfc, cfdiData.receptor.Rfc, cfdiData.comprobante.Total);

    if (estadoCFDI === 'Cancelado') {
      log.error('CFDI Cancelado', `El CFDI con UUID ${cfdiData.comprobante.UUID} ha sido cancelado.`);
      return { success: false, message: `El CFDI con UUID ${cfdiData.comprobante.UUID} ha sido cancelado.` };
    } else if (estadoCFDI === 'Vigente') {
      const gastoId = crearRegistroGasto(cfdiData);
      return { success: true, message: `CFDI válido. Registro de gasto creado con ID: ${gastoId}` };
    } else {
      return { success: false, message: `Estado desconocido del CFDI: ${estadoCFDI}` };
    }
  } catch (error) {
    log.error('Error en validación CFDI', error);
    return { success: false, message: 'Error al validar el CFDI.' };
  }
}
```

Ilustración 49: Validación de CFDI y creación de registro. Fuente. Elaboración propia fragmento de código.

Creación de registro

Cuando el CFDI ha sido validado correctamente, y como parte del cumplimiento del requerimiento (Creación Automática del Registro de Gasto) este fragmento de código crea

Optimización de la Gestión Financiera ... (Ciclo de egresos)

un registro de gasto dentro de NetSuite. Los datos clave, como el UUID, RFC, moneda, total, etc., se asignan al registro antes de guardarlo. Si el proceso es exitoso, el sistema devuelve el ID del registro, permitiendo su rastreo en futuros reportes administrativos.

```
function crearRegistroGasto(cfdiData) {
  try {
    const gasto = record.create({
      type: 'expensereport',
      isDynamic: true
    });

    gasto.setValue({ fieldId: 'custbody_uuid', value: cfdiData.comprobante.UUID });
    gasto.setValue({ fieldId: 'entity', value: cfdiData.receptor.Rfc });
    gasto.setValue({ fieldId: 'currency', value: cfdiData.comprobante.Moneda });
    gasto.setValue({ fieldId: 'amount', value: cfdiData.comprobante.Total });

    const gastoId = gasto.save();
    log.debug('Registro de gasto creado', `ID: ${gastoId}`);
    return gastoId;
  } catch (error) {
    log.error('Error al crear registro de gasto', error);
    return null;
  }
}
```

Ilustración 50: Creación del Registro de Gasto. Fuente. Elaboración propia fragmento de código.

La función no se presenta completa, ya que solo incluye algunos campos. Sin embargo, el proceso de asignación de valores mediante se aplica de manera similar para todos los datos extraídos del XML, asegurando que cada atributo del CFDI se asigne correctamente en el registro de gasto en NetSuite.

Almacenamiento de archivos en gabinete de empleado

Una vez que los archivos han sido validados y el comprobante ha sido registrado correctamente, el sistema procede con el almacenamiento de archivos en el gabinete del empleado en cumplimiento del requerimiento (Almacenamiento de Documentos). Para ello, se busca o se crea una carpeta personalizada dentro del sistema de NetSuite, donde se guardan los archivos XML y PDF correspondientes a cada factura registrada. Esta organización permite que la información esté estructurada de forma clara y accesible, facilitando su consulta en futuros procesos administrativos y auditorías.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Los encargados de realizarlo son las siguientes funciones:

La función `buscarOCrearCarpeta` busca una carpeta dentro de NetSuite con el nombre del empleado. Si la carpeta ya existe, devuelve su ID; si no, la crea dentro de la carpeta general de los empleados. Este mecanismo garantiza que la información del empleado se almacene en una estructura organizada, evitando duplicaciones innecesarias y asegurando una gestión adecuada de los documentos.

```
function buscarOCrearCarpeta(nombreCarpeta, parentId) {
  try {
    // Buscar carpeta por nombre
    const searchResult = search.create({
      type: 'folder',
      filters: [['name', 'is', nombreCarpeta], parentId ? ['parent', 'is', parentId] : []],
      columns: ['internalid']
    }).run().getRange({ start: 0, end: 1 });

    // Si ya existe la carpeta, retornar su ID
    if (searchResult.length > 0) {
      return searchResult[0].getValue('internalid');
    }

    // Crear nueva carpeta
    const nuevaCarpeta = record.create({
      type: 'folder',
      isDynamic: true
    });

    nuevaCarpeta.setValue({ fieldId: 'name', value: nombreCarpeta });
    if (parentId) nuevaCarpeta.setValue({ fieldId: 'parent', value: parentId });

    const carpetaId = nuevaCarpeta.save();
    Log.debug('Carpeta creada', `Nombre: ${nombreCarpeta}, ID: ${carpetaId}`);
    return carpetaId;
  } catch (error) {
    Log.error('Error al buscar o crear carpeta', error);
    return null;
  }
}
```

Ilustración 51: Función para la creación y validación de carpetas. Fuente. Elaboración propia fragmento de código.

La siguiente función almacena los archivos XML y PDF en NetSuite dentro de una carpeta del empleado. Primero, busca o crea la carpeta del empleado. Luego, genera una subcarpeta exclusiva para la factura, donde se guardan los archivos correspondientes.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

```
function guardarArchivos(xmlFile, pdfFile, empleadoId, facturaUUID) {
  try {
    // Buscar o crear carpeta del empleado
    let empleadoFolderId = buscarOCrearCarpeta('Empleado ${empleadoId}', null);
    if (!empleadoFolderId) throw new Error('No se pudo obtener la carpeta del empleado.');
```

// Crear subcarpeta para la factura dentro de la carpeta del empleado

```
let facturaFolderId = buscarOCrearCarpeta('Factura ${facturaUUID}', empleadoFolderId);
if (!facturaFolderId) throw new Error('No se pudo crear la carpeta de la factura.');
```

// Guardar archivo XML

```
const xmlArchivo = file.create({
  name: xmlFile.name,
  fileType: file.Type.XMLDOC,
  contents: xmlFile.contents,
  folder: facturaFolderId
});
const xmlArchivoId = xmlArchivo.save();
log.debug('Archivo XML guardado', 'ID: ${xmlArchivoId}');
```

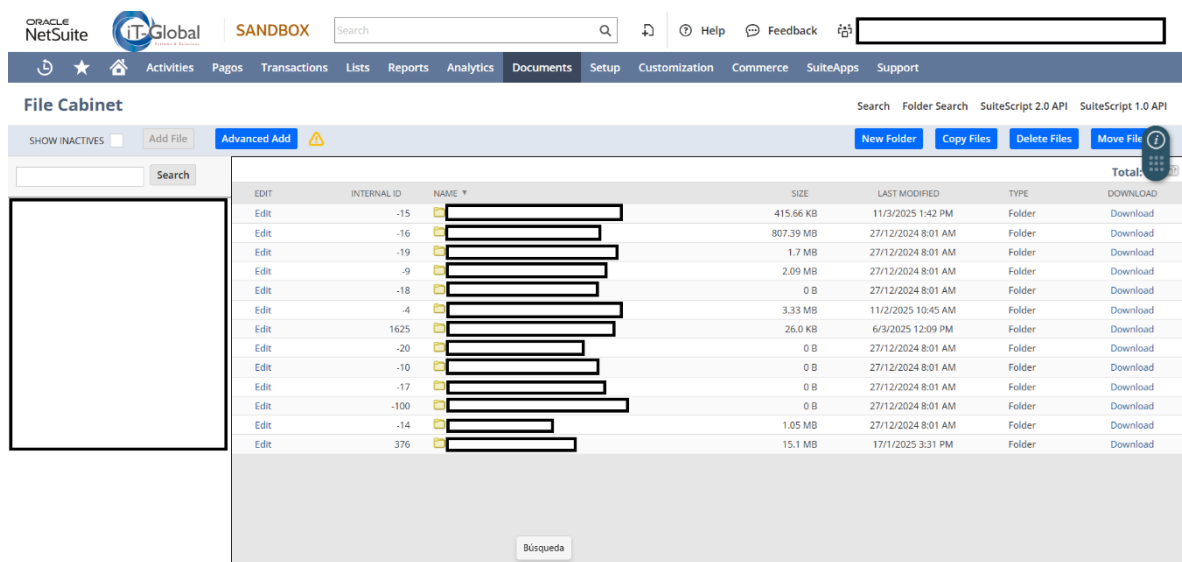
// Guardar archivo PDF

```
const pdfArchivo = file.create({
  name: pdfFile.name,
  fileType: file.Type.PDF,
  contents: pdfFile.contents,
  folder: facturaFolderId
});
const pdfArchivoId = pdfArchivo.save();
log.debug('Archivo PDF guardado', 'ID: ${pdfArchivoId}');
```

```
return { success: true, message: 'Archivos almacenados correctamente.', xmlId: xmlArchivoId, pdfId: pdfArchivoId };
} catch (error) {
  log.error('Error al guardar archivos', error);
  return { success: false, message: 'Error al almacenar los archivos.' };
}
```

Ilustración 52: Función para almacenamiento de archivos XML y PDF. Fuente. Elaboración propia fragmento de código.

Estructura de carpetas final:



The screenshot shows the NetSuite File Cabinet interface. The top navigation bar includes the Oracle NetSuite logo, a search bar, and various menu items like Activities, Pagos, Transactions, Lists, Reports, Analytics, Documents, Setup, Customization, Commerce, SuiteApps, and Support. The main content area is titled 'File Cabinet' and displays a table of folders. The table has columns for EDIT, INTERNAL ID, NAME, SIZE, LAST MODIFIED, TYPE, and DOWNLOAD. The folders are organized by employee, with names like 'Empleado -15', 'Empleado -16', etc. The interface also includes a search bar on the left and a 'Búsqueda' button at the bottom.

EDIT	INTERNAL ID	NAME *	SIZE	LAST MODIFIED	TYPE	DOWNLOAD
Edit	-15	Empleado -15	415.66 KB	11/3/2025 1:42 PM	Folder	Download
Edit	-16	Empleado -16	807.39 MB	27/12/2024 8:01 AM	Folder	Download
Edit	-19	Empleado -19	1.7 MB	27/12/2024 8:01 AM	Folder	Download
Edit	-9	Empleado -9	2.09 MB	27/12/2024 8:01 AM	Folder	Download
Edit	-18	Empleado -18	0 B	27/12/2024 8:01 AM	Folder	Download
Edit	-4	Empleado -4	3.33 MB	11/2/2025 10:45 AM	Folder	Download
Edit	1625	Empleado 1625	26.0 KB	6/3/2025 12:09 PM	Folder	Download
Edit	-20	Empleado -20	0 B	27/12/2024 8:01 AM	Folder	Download
Edit	-10	Empleado -10	0 B	27/12/2024 8:01 AM	Folder	Download
Edit	-17	Empleado -17	0 B	27/12/2024 8:01 AM	Folder	Download
Edit	-100	Empleado -100	0 B	27/12/2024 8:01 AM	Folder	Download
Edit	-14	Empleado -14	1.05 MB	27/12/2024 8:01 AM	Folder	Download
Edit	376	Empleado 376	15.1 MB	17/1/2025 3:31 PM	Folder	Download

Ilustración 53. Gabinete de archivos con carpetas de empleados. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

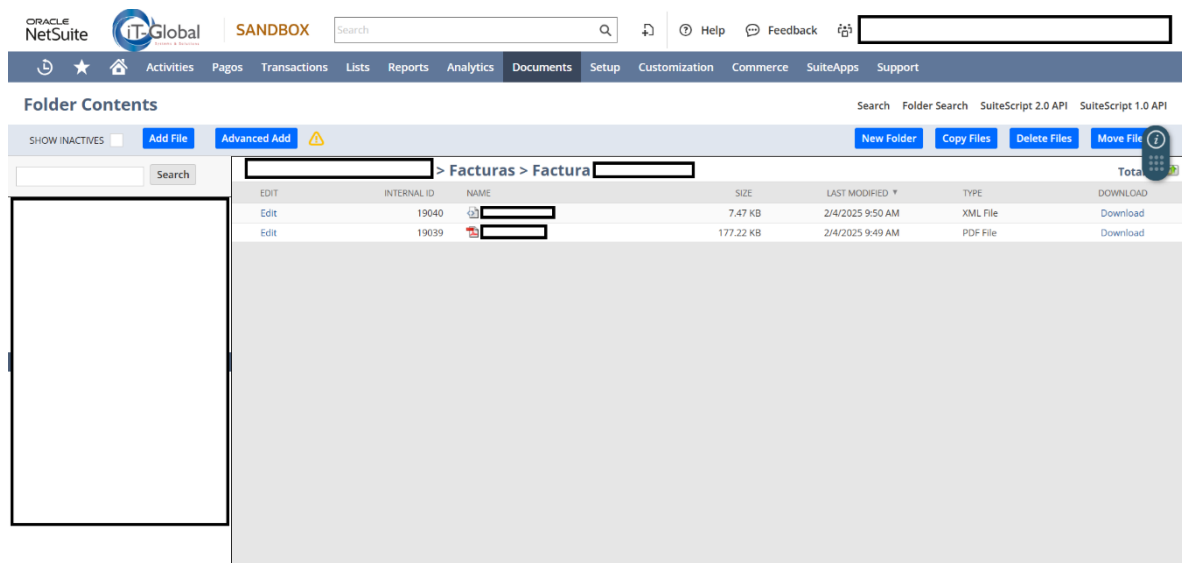


Ilustración 54. Carpeta de empleado con carpeta de factura. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Pantalla de confirmación

Una vez completado el registro exitoso del gasto, validación del CFDI con el SAT y almacenamiento de los archivos en la carpeta correspondiente del empleado, el sistema redirige al usuario a la ****Pantalla de Confirmación****. En esta pantalla, se presenta un resumen detallado de los resultados, incluyendo el ID del gasto generado, los identificadores de los archivos almacenados y el estado obtenido de la validación con el SAT. Esta etapa final asegura que el proceso haya concluido satisfactoriamente y brinda al usuario una confirmación clara y organizada del flujo completo.

Para implementar esta última pantalla, se emplea un código que crea un formulario específico. Este formulario permite mostrar los detalles del proceso al usuario, actuando como la pantalla principal del Suitelet.

```
// Crear formulario de confirmación
const form = serverWidget.createForm({
  title: 'Confirmación del Registro'
});
```

Ilustración 55: Creación del Formulario de Confirmación. Fuente. Elaboración propia fragmento de código.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Se obtienen los parámetros de éxito desde la solicitud del Suitelet.

```
// Obtener los parámetros de éxito
const gastoId = context.request.parameters.gastoId || 'No disponible';
const xmlArchivoId = context.request.parameters.xmlId || 'No disponible';
const pdfArchivoId = context.request.parameters.pdfId || 'No disponible';
const validacionSAT = context.request.parameters.validacionSAT || 'No disponible';
```

Ilustración 56: Recepción de Parámetros. Fuente. Elaboración propia fragmento de código.

Se agrega un campo al formulario para mostrar un mensaje que detalla el éxito del registro, los IDs del gasto y archivos, y el resultado de la validación con el SAT. El diseño utiliza etiquetas HTML para garantizar claridad y una presentación visual atractiva.

```
// Mensaje de éxito
const successMessage = form.addField({
  id: 'custpage_success_message',
  type: serverWidget.FieldType.INLINEHTML,
  label: ' '
});
successMessage.defaultValue = `
<h2 style="color: green;">¡Registro Completado Exitosamente!</h2>
<p>La factura se ha registrado correctamente y los archivos han sido almacenados exitosamente.</p>
<p><strong>Detalles:</strong></p>
<ul>
  <li><strong>ID del Gasto:</strong> ${gastoId}</li>
  <li><strong>ID del Archivo XML:</strong> ${xmlArchivoId}</li>
  <li><strong>ID del Archivo PDF:</strong> ${pdfArchivoId}</li>
  <li><strong>Validación SAT:</strong> ${validacionSAT}</li>
</ul>
<p>Gracias por utilizar el sistema. Puedes cerrar esta pantalla.</p>
`;
```

Ilustración 57: Campo de mensaje de éxito en pantalla. Fuente. Elaboración propia fragmento de código.

Pantalla de confirmación:

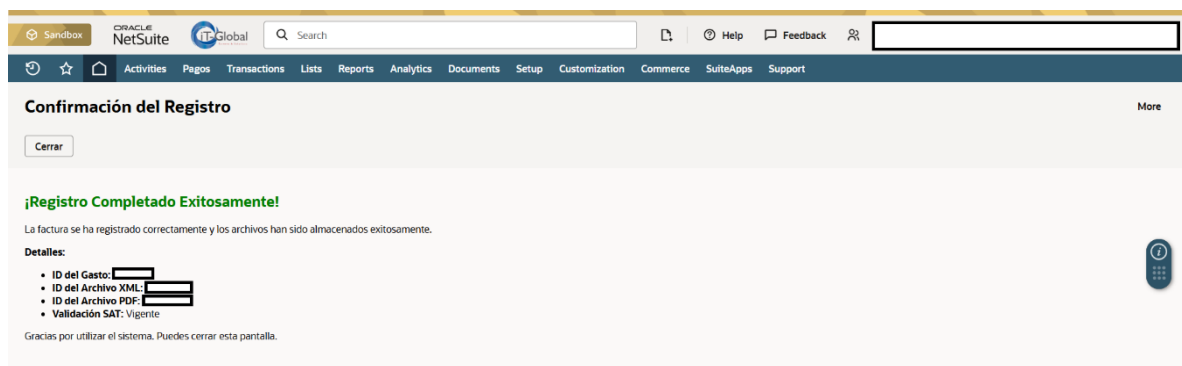


Ilustración 58: Pantalla de confirmación. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

En caso de validación incorrecta no se realizará el registro ni almacenamiento de los documentos. Se notificará al empleado del error y su causa terminando con el proceso.

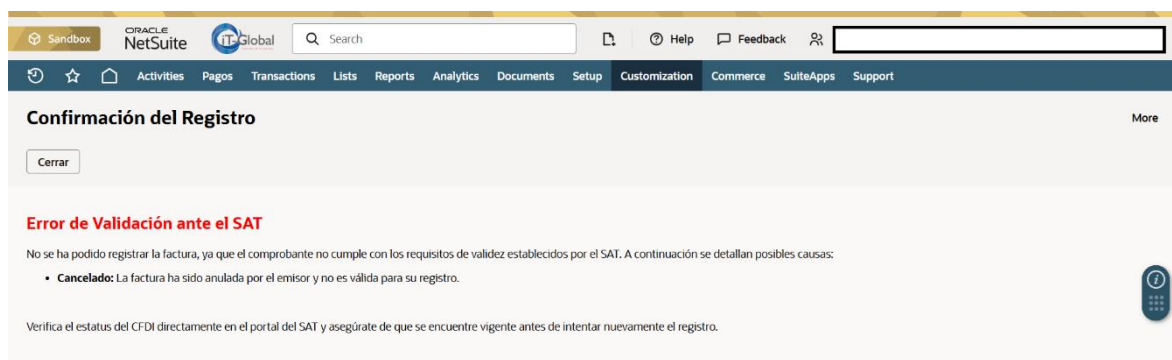


Ilustración 59: Pantalla de error. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Reporte general de egresos de empleados

Para mejorar la visualización y el análisis de los egresos de empleados dentro de NetSuite, cumpliendo con el requerimiento (Generación de Reportes) se implementó un Reporte General de Egresos de Empleados. Este reporte se basa en búsquedas guardadas, permitiendo filtrar los egresos por empleado y obtener una visión clara de los gastos asociados a cada uno. La personalización de estos filtros facilita el acceso a información relevante y agiliza la toma de decisiones.

El proceso para la creación de la búsqueda guardada se llevó a cabo siguiendo los siguientes pasos:

1. Seleccionar Tipo de Registro: Se eligió el tipo de registro “Transacción”, ya que los egresos se registran como transacciones dentro del sistema.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

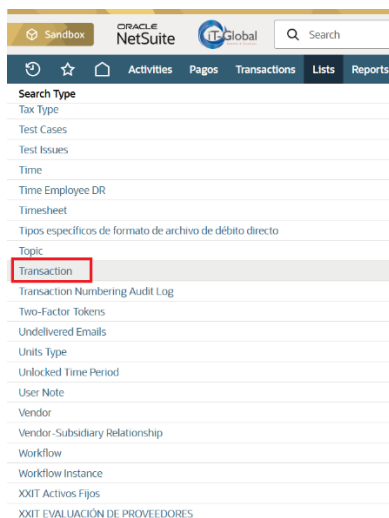


Ilustración 60: Selección de tipo de Registro. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

2. Definir Criterios de Búsqueda: Se establecieron los criterios que permiten identificar únicamente los egresos de empleados, como el tipo de reporte de gasto. Además de visualizar solo la línea principal de la transacción

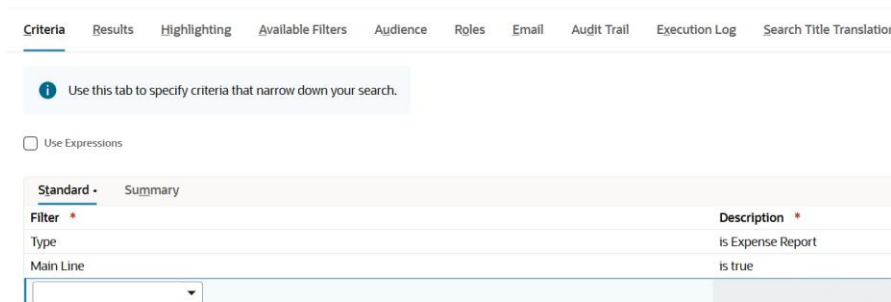


Ilustración 61: Criterios de búsqueda. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Los pasos 1 y 2 son comparables con la sentencia WHERE de una consulta SQL, ya que permiten establecer las condiciones que deben cumplirse para que los registros sean considerados dentro de la búsqueda. En este caso, al seleccionar el tipo de registro y definir los criterios de búsqueda, se delimitan las transacciones relevantes, filtrando únicamente aquellas que corresponden a egresos de empleados. Esto asegura que el conjunto de datos obtenido sea específico y útil para el análisis posterior.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

3. Definir Criterios de Filtro: Se configuraron filtros personalizables como nombre del empleado, supervisor y fecha permitiendo al usuario final refinar la información rápidamente según su necesidad y sin requerir conocimientos técnicos del funcionamiento de la búsqueda.

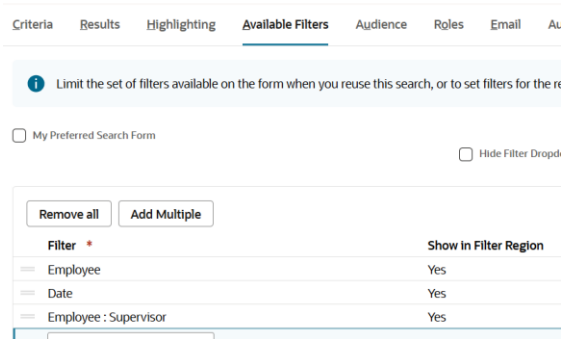


Ilustración 62: Criterios de filtro. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

4. Establecer Columnas de Resultados: Se seleccionaron las columnas que se mostrarán en el reporte, incluyendo campos como empleado, fecha, monto total, proveedor, entre otros datos clave. Este paso corresponde al SELECT de una consulta SQL, ya que define exactamente qué información será visualizada por el usuario final a partir de los registros filtrados previamente. La correcta elección de estas columnas permite construir un reporte claro, útil y alineado con los objetivos del análisis.

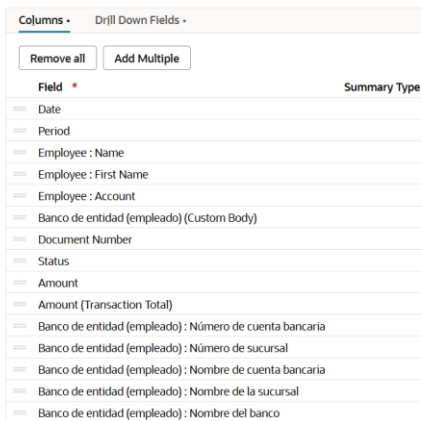


Ilustración 63: Columnas resultados. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

5. Configurar Permisos: Finalmente, se asignaron los permisos adecuados para asegurar que solo los usuarios autorizados, como los contadores, puedan acceder y utilizar esta búsqueda guardada.

CriteriaResultsHighlightingAvailable FiltersAudienceRolesEmailAudit TrailExecution LogSearch Title Translation

Use this tab to make this search the preferred list view, search form, dashboard view, or sublist view, either globally (with the top checkboxes) or by role (with the list view checkboxes).

Preferred Search Form

Preferred Search Results

Preferred List View

Preferred Dashboard View

Preferred Sublist View

List	Form	Results	Dashboard	Sublist	Role	Centre Type
☐	☐	☐	☐	☐	A/P Clerk	Accounting Centre
☐	☐	☐	☐	☐	A/R Clerk	Accounting Centre
☐	☐	☐	☐	☐	ABB Administrador del Sistema	System Administrator Centre
☐	☐	☐	☐	☐	AKRR Gestion de Inventario	Shipping Centre
☒	☒	☒	☒	☒	Accountant	Accounting Centre

Ilustración 64: Configuración de permisos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

SandboxORACLE NetSuiteGlobal

Search

HelpFeedback

ActivitiesPagosTransactionsListsReportsAnalyticsDocumentsSetupCustomizationCommerceSuiteAppsSupport

Informe de Gasto: Results

ListSearchAudit Trail

Edit this Search

Filters

Employee

- All -

Date

All

From

To

Supervisor

- All -

Style

Normal

Edit

Total: 8

Edit View	Internal ID	Date	Period	Name	First Name	Account	Banco de entidad (empleado)	Document Number	Status	Amount	Amount (Trans)
Edit View	7	7/2/2025	feb 2025						Approved by Accounting		
Edit View	8	7/2/2025	feb 2025						Approved by Accounting		
Edit View	9	10/2/2025	feb 2025						Pending Accounting Approval		
Edit View	10	10/2/2025	feb 2025						Approved by Accounting		
Edit View	17	18/2/2025	feb 2025						Approved by Accounting		

Ilustración 65: Búsqueda guardada de gastos de empleados. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Página 73 | 89

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Además, para ofrecer una experiencia más dinámica y visual, se implementó un Workbook que permite representar los datos en tablas interactivas y gráficos. Este Workbook se integra directamente en el dashboard de NetSuite mediante portlets, permitiendo a los usuarios acceder de manera inmediata a información clave sobre los egresos.

El propósito de estas personalizaciones es proporcionar herramientas eficientes para el análisis financiero, asegurando que la información de los egresos esté estructurada y fácilmente accesible. Esto no solo mejora la transparencia en la gestión de gastos, sino que también optimiza el proceso de toma de decisiones dentro de la organización.

El proceso para la creación del workbook se llevó a cabo siguiendo los siguientes pasos:

1. Seleccionar Dataset Base: Se eligió un dataset previamente generado a partir de búsquedas guardadas o registros específicos, que contiene la información relevante sobre egresos de empleados. Este dataset actúa como la fuente de datos estructurada sobre la cual se construirán las vistas del Workbook.

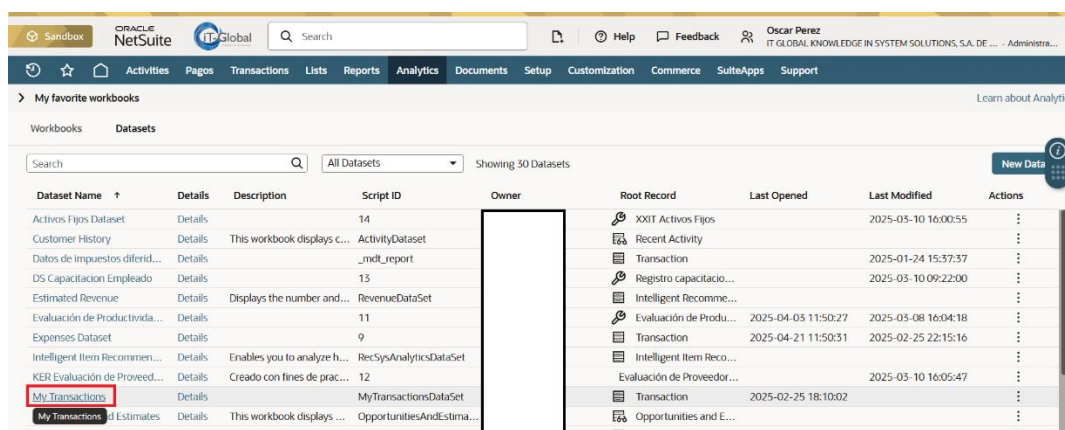


Ilustración 66: Selección dataset. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

2. Agregar Filtros Dinámicos: Se configuraron filtros dinámicos, como rango de fechas, permitiendo brindar datos del periodo contable actual de manera rápida al usuario final.

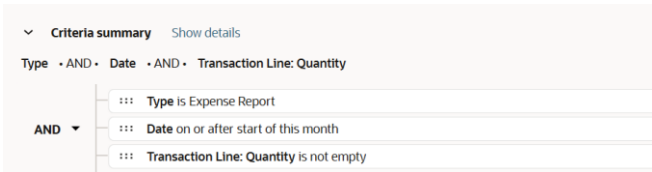


Ilustración 67: Filtros dinámicos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

3. Construcción de Vistas: A partir del dataset y los filtros definidos, se diseñaron dos tipos de vistas: tablas, gráficos de barras. Estas vistas permiten una comprensión más intuitiva del comportamiento de los egresos a lo largo del tiempo o por categorías específicas.

The screenshot shows the NetSuite interface with a table view of transaction data. The table has columns for ENTITY, DOCUMENT NUMBER/ID, DATE, MEMO, and AMOUNT (CREDIT). The data is filtered to show transactions from 7/2/2025 to 20/3/2025.

ENTITY :	DOCUMENT NUMBER/ID :	DATE :	MEMO :	AMOUNT (CREDIT) :
Ulises Paez	EXP-00000001	7/2/2025	Viaje a GDL	\$116.00
Daniel Carcaño	EXP-00000002	7/2/2025		\$116.00
Daniel Carcaño	EXP-00000003	10/2/2025	Expense report ITS	\$696.00
Daniel Carcaño	EXP-00000004	10/2/2025	Expense report Flujos ...	\$580.00
Daniel Carcaño	EXP-00000005	18/2/2025		\$120.00
Daniel Carcaño	EXP-00000006	27/2/2025		\$100.00
Daniel Carcaño	EXP-00000007	27/2/2025		\$150.00
Daniel Carcaño	EXP-00000008	20/3/2025		\$100.00

Ilustración 68: Construcción de tabla de datos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)



Ilustración 69: Construcción de gráfico de datos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

4. Publicar en Dashboard: Finalmente, las vistas construidas fueron publicadas en el dashboard de NetSuite a través de portlets, facilitando el acceso rápido y centralizado a la información desde la página principal del sistema, sin necesidad de ingresar a múltiples módulos.

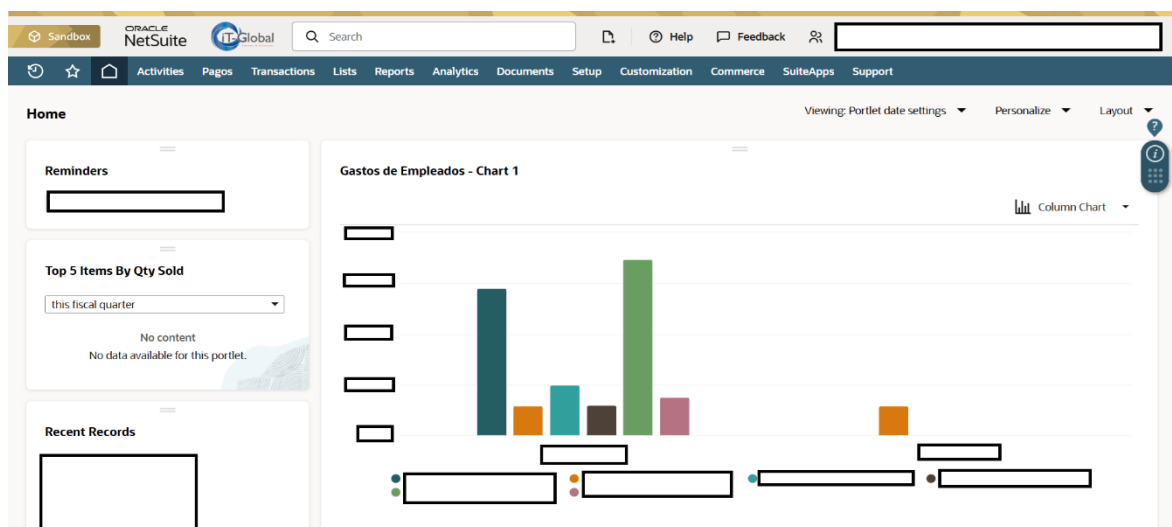


Ilustración 70: Portlet de gráfico de gastos de empleados en panel de inicio. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

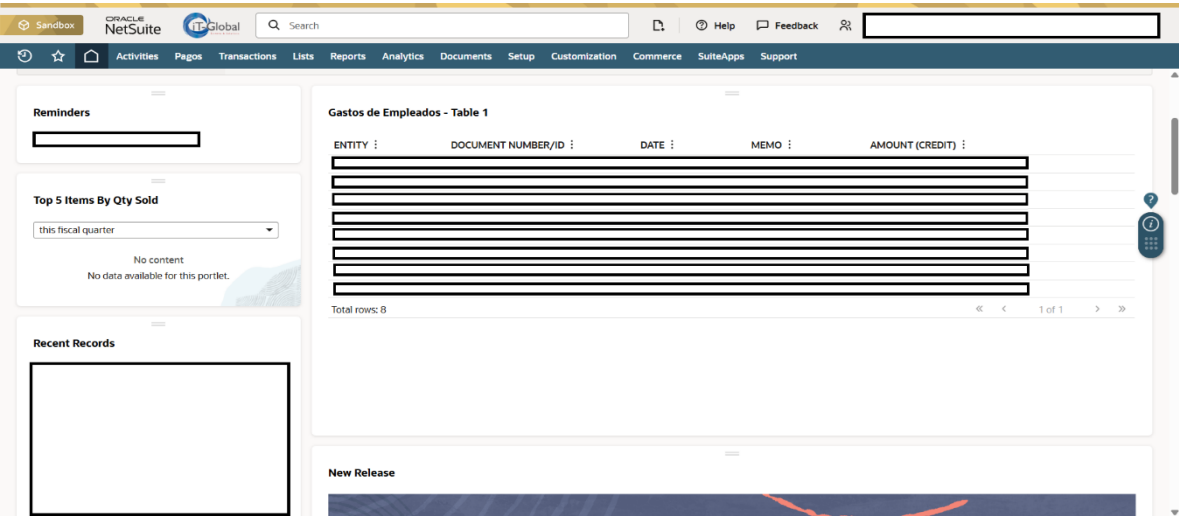


Ilustración 71: Portlet de tabla de gastos de empleados en panel de inicio. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Enviar Datos de Migración

Durante el desarrollo se utilizó la instancia de Sandbox, un entorno de pruebas que permite realizar cambios sin afectar la configuración ni los datos reales del entorno de producción. En este entorno, el usuario desarrollador cuenta con su propio espacio de trabajo, que le facilita la implementación, validación y depuración de funcionalidades antes de enviarlas al área de testing

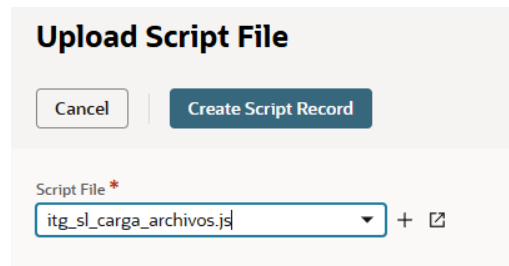
Para proceder con las pruebas, los archivos deben ser cargados en el File Cabinet de NetSuite y posteriormente publicados. En la ilustración 72 se muestran los archivos JS desarrollados, se encuentran almacenados en una carpeta creada por el líder del proyecto dedicada para cada desarrollador.

SuiteScripts > itg_scripts_jcc > Proyecto						Total: 5
Edit	Internal ID	Name	Size	Last Modified	Type	Download
Edit	19070	itg_sl_mostrar_datos.js	8.27 KB	21/4/2025 10:07 am	JavaScript File	Download
Edit	19181	pruebas_sl.js	1.75 KB	21/4/2025 9:39 am	JavaScript File	Download
Edit	19069	itg_sl_carga_archivos.js	2.58 KB	8/4/2025 5:31 pm	JavaScript File	Download
Edit	19074	itg_cs_validar_campos.js	1.09 KB	8/4/2025 5:30 pm	JavaScript File	Download
Edit	19073	itg_cs_redireccionar.js	651 B	8/4/2025 5:22 pm	JavaScript File	Download

Ilustración 72: Carpeta del proyecto con archivos JS desarrollados. Fuente. Captura de pantalla de file cabinet de NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Una vez cargados los archivos JS, debemos crear el registro de un script personalizado. Por cada archivo subido se debe de crear su respectivo registro en el sistema.



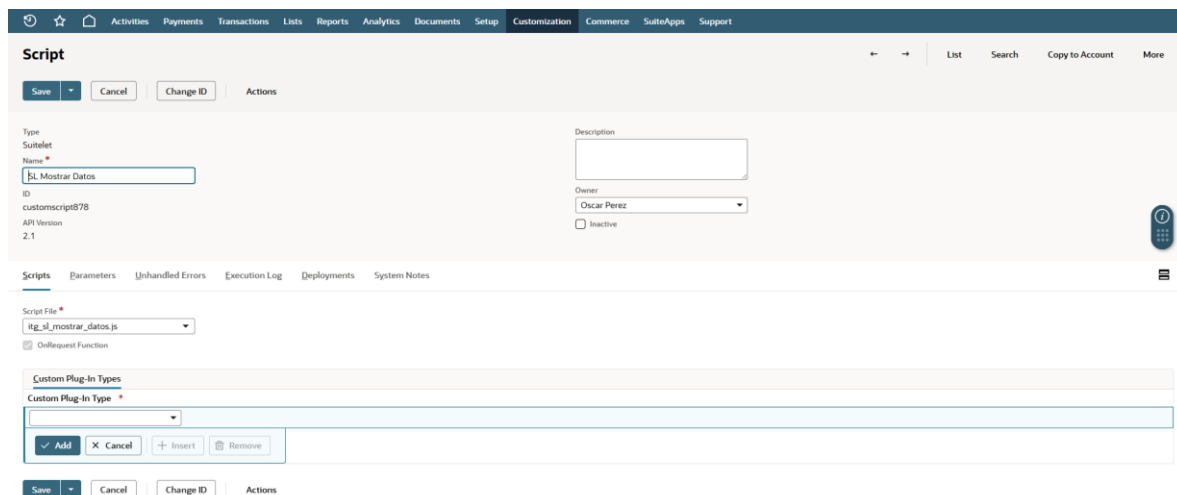
Upload Script File

Script File *

itg_sl_carga_archivos.js + 

Ilustración 73: Creación de registro de script. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

En el registro del script, se define el nombre y un id, por regla se ingresa tres letras de la carpeta de origen, el tipo de script y nombre, otros campos son descripción y propietario, dependiendo el tipo de script en la pestaña deployment se le debe de señalar a que registro debe de afiliarse con el objetivo que este sea su trigger a excepción de restlet esto no es necesario.



Script

Save + Cancel Change ID Actions

Type: Suitelet

Name: itg_sl_mostrar_datos

ID: customscript978

API Version: 2.1

Description:

Owner: Oscar Perez

☐ Inactive

Scripts Parameters Unhandled Errors Execution Log Deployments System Notes

Script File: itg_sl_mostrar_datos.js

☒ OnRequest Function

Custom Plug-In Types

Custom Plug-In Type:

Add Cancel Insert Remove

Save + Cancel Change ID Actions

Ilustración 74: Registro de script creado. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Cada usuario en Netsuite tiene por defecto un supervisor, que vigila su actividad dentro del sistema, cada acción es registrada y notificada mediante correos. Cuando se crea el registro del script se envía un correo al supervisor y a su vez notifica al área de testing. Además de que una vez creado el desarrollador tiene tres días hábiles para modificar lo necesario para su funcionamiento, si en ese lapso no logra hacerlo, simplemente puede eliminar el registro y se cancela el proceso.

Después de cargar todos los archivos y que el desarrollador corrobore que todo es correcto, solo le hace falta esperar la retroalimentación del área de testing.

Retroalimentación

Como parte del proceso de validación del desarrollo, se realizaron pruebas funcionales por parte del área de testing, quienes proporcionaron observaciones clave para mejorar la experiencia del usuario y asegurar el correcto funcionamiento del sistema. A continuación, se detallan los comentarios recibidos durante esta etapa, los cuales se enfocan principalmente en aspectos de presentación de datos extraídos del CFDI y usabilidad de la interfaz.

Comentarios del Área de Testing:

Durante la revisión del bundle correspondiente al flujo de recepción y validación de facturas electrónicas (CFDI), se detectó una mejora necesaria en la forma en que se muestran los datos extraídos del XML, particularmente en la sección de Conceptos.

Actualmente, los conceptos aparecen de manera continua y sin una estructura visual clara, lo cual dificulta la identificación de los datos correspondientes a cada línea y sus respectivos impuestos. Esta presentación puede resultar confusa para el usuario final al momento de revisar la información previa al envío para validación ante el SAT.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Para una mejor visualización, se sugiere ordenar los datos en una tabla para los conceptos, donde se muestre de forma estructurada toda la información de ese concepto (cantidad, unidad, descripción, importe, etc.), y en una subsección o tabla anidada, incluir los impuestos específicos aplicables a esa línea (tipo de impuesto, base, tasa/cuota, importe, tipo de factor).

Desarrollo de la corrección

Para el desarrollo de esta corrección se utilizaron sublistas (Sublist) dentro del Suitelet, permitiendo mostrar de forma dinámica y tabular los conceptos contenidos en el CFDI. Esta solución reemplaza los campos fijos previamente utilizados para representar los conceptos, los cuales no eran adecuados para manejar múltiples líneas.

```
// Sublista dinámica de Conceptos
const sublist = form.addSublist({
  id: 'custpage_sublist_conceptos',
  label: 'Conceptos',
  type: serverWidget.SublistType.LIST
});

const sublist2 = form.addSublist({
  id: 'custpage_sublist_impuestos',
  label: 'Impuestos',
  type: serverWidget.SublistType.LIST
});
```

Ilustración 75: Creación de sublistas. Fuente. Fragmento de código.

Conceptos		Impuestos						
Clave ProdServ	No Identificación	Cantidad	Clave Unidad	Unidad	Descripción	Valor Unitario	Importe	Descuento
No records to show.								

Ilustración 76: Sublista de conceptos. Fuente. Captura de pantalla de entorno de pruebas de NetSuite.

Conceptos		Impuestos		
Base	Impuesto	Tipo Factor	Tasa o Cuota	Importe Impuesto
No records to show.				

Ilustración 77: Sublista impuestos. Fuente. Captura de pantalla de entorno de pruebas de NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

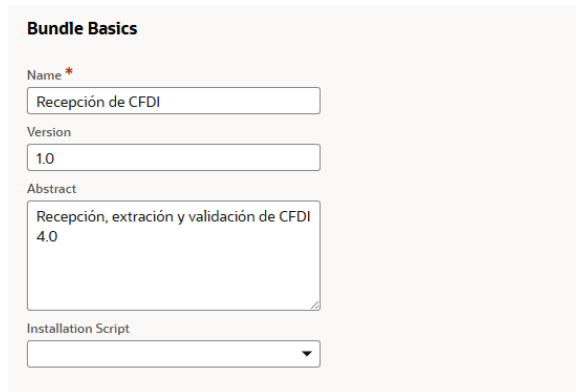
Migración a entorno de producción

Como parte del último paso en el desarrollo del bundle, se procede a la migración hacia el entorno de producción. Durante las fases previas, todas las pruebas y validaciones fueron realizadas en una cuenta sandbox, un entorno controlado destinado exclusivamente a pruebas, con el objetivo de evitar cualquier impacto en las operaciones reales de la empresa. Esta migración se lleva a cabo únicamente una vez que se ha verificado que los desarrollos funcionan correctamente, asegurando así una transición estable y sin contratiempos para el sistema en producción.

Creación del bundle

Para la creación del bundle que permitirá migrar a producción se siguieron los siguientes pasos:

1. **Llenar datos del bundle:** Se llenan datos básicos del bundle como nombre, versión, resumen.



Bundle Basics

Name *
Recepción de CFDI

Version
1.0

Abstract
Recepción, extracción y validación de CFDI 4.0

Installation Script
▼

Ilustración 78: Llenado de datos básicos. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

2. **Agregar componentes al bundle:** Se despliega una lista con todos los tipos de objetos personalizables en NetSuite que pueden ser incluidos en el bundle. Para incluir un componente, solo es necesario expandir el tipo correspondiente, localizar el elemento deseado y marcarlo para que forme parte del paquete.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

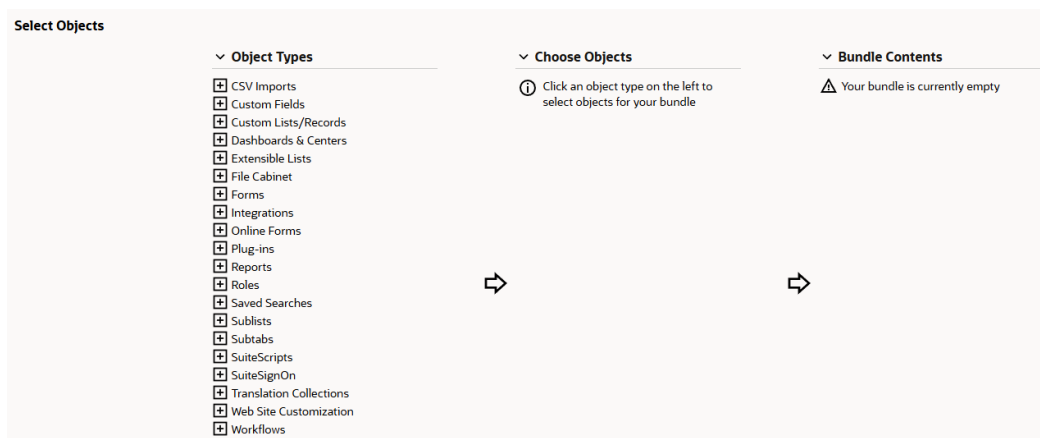


Ilustración 79: Selección de objetos del bundle. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

3. Comprobar el contenido del bundle y bloquear archivos para no permitir modificación (opcional pero recomendado).

Name	ID	Referenced By	Preference	Lock on Install
Saved Searches				
Electronic Payments Logs				
Payment File Format - EP Logs Search	customsearch_18023_file_format_ep_logs	Formato de archivo de pago		☒
Transaction				
Informe de Gasto	customsearch366			☐
Subtabs				
Entity				
Detalles de pago bancario	custtab_10_4533432_230	Detalles bancarios		
Detalles de pago bancario (crédito)	custtab_11_4533432_230	Detalles bancarios		
Detalles de pago bancario (débito)	custtab_12_4533432_230	Detalles bancarios		
Transaction				
TEF	custtab_9_4533432_230	Banco de entidad (empleado)		
SuiteScripts				
Client				
Redireccionar	customscript981		Do Not Update Deployments ▼	☐
Validar Campos	customscript982		Do Not Update Deployments ▼	☐

Ilustración 80: Objetos seleccionados para el bundle. Fuente. Captura de pantalla de entorno de pruebas NetSuite.

4. Guardar el bundle.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Instalación de bundle

Para instalar el bundle en la cuenta de producción, se deben seguir los siguientes pasos, partiendo del supuesto de que ya se ha iniciado sesión en dicha cuenta:

- 1. **Buscar el bundle:** Se necesitará una búsqueda avanzada indicando donde se encuentra ese bundle (cuenta sandbox) y el id de la cuenta.

Search & Install Bundles

Search

Basic | Advanced

Leave the keywords box blank and click Search to view the most popular SuiteApps

Keywords

Recepción de CFDI

Location

Sandbox Account

Availability

All

Publisher ID

Account ID *

Product

All

Vertical

All

Language

All

Ilustración 81: Búsqueda de bundle. Fuente. Captura de pantalla de entorno de producción NetSuite.

- 2. Seleccionar el resultado.

Name	Bundle ID	Version	Managed	Company Name	Publisher ID	Created on	Availability	No. Installs
Recepción de CFDI		1.0		IT GLOBAL KNOWLEDGE IN SYSTEM SOLUTIONS, S.A. DE C.V.		02-MAY-25	Private	0

Ilustración 82: Resultado de búsqueda. Fuente. Captura de pantalla de entorno de producción NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

3. Revisar el bundle y hacer clic en Install.

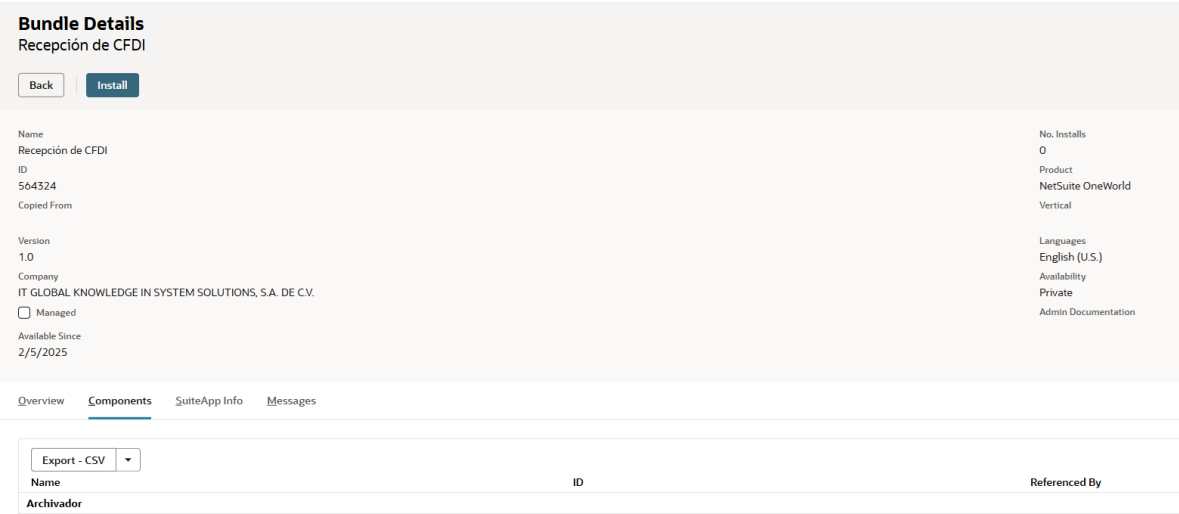


Ilustración 83: Bundle Recepción CFDI. Fuente. Captura de pantalla de entorno de producción NetSuite.

4. Esperar la instalación y comprobar en la lista de bundles instalados.

Bundle ID	Name	Version	Action	Start Date	End Date	Status	User	Source	Notes	Error Reference
564324	Recepción de CFDI	1.0	Install	2/5/2025 5:30 pm	2/5/2025 5:34 pm	Success				

Ilustración 84: Instalación de bundle. Fuente. Captura de pantalla de entorno de producción NetSuite.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Conclusiones

A lo largo del desarrollo del presente proyecto, se logró cumplir satisfactoriamente con todos los objetivos propuestos. Se implementaron soluciones personalizadas en NetSuite ERP orientadas a automatizar el ciclo de egresos, integrando la validación de facturas electrónicas (CFDI), generando registros de gasto de manera automática, y desarrollando reportes dinámicos que facilitan el análisis financiero en tiempo real. Estas soluciones no solo optimizaron los procesos internos de la empresa, sino que también permitieron mejorar el control, la trazabilidad y la transparencia de los egresos, aportando valor directo a la gestión financiera de la organización.

Entre las recomendaciones que se pueden ofrecer, se sugiere dar continuidad al mantenimiento y mejora de los bundles desarrollados, considerando retroalimentación constante por parte de los usuarios finales. Asimismo, se recomienda capacitar al personal contable y operativo en el uso de las herramientas implementadas para garantizar una adopción efectiva y maximizar los beneficios del sistema. Finalmente, sería conveniente evaluar la posibilidad de extender este tipo de automatizaciones a otros ciclos del ERP, como ingresos o inventarios, para obtener una integración aún más completa de los procesos empresariales.

Desde el punto de vista personal y profesional, esta experiencia fue sumamente enriquecedora. Permitió aplicar conocimientos técnicos adquiridos durante la formación académica, especialmente en el desarrollo con SuiteScript y el uso de plataformas ERP. Asimismo, favoreció el desarrollo de habilidades blandas como el trabajo colaborativo, la comunicación efectiva con equipos multidisciplinarios y la gestión de tiempos en entornos reales de trabajo. Esta residencia profesional no solo fortaleció mi perfil como desarrollador técnico, sino que también me brindó una visión más amplia de los retos y oportunidades dentro del ámbito de la consultoría en sistemas empresariales.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Competencias Desarrolladas y Aplicadas

Durante la realización de este proyecto, se desarrollaron y aplicaron múltiples competencias adquiridas a lo largo de la carrera de Tecnologías de la Información y Comunicaciones (TIC), tanto en el ámbito técnico como en el profesional.

1. Competencias técnicas:

- **Desarrollo de software:** Se aplicaron conocimientos avanzados en programación con JavaScript y el framework SuiteScript.
- **Diseño e implementación de sistemas:** Se diseñó e implementó un sistema de automatización del ciclo de egresos, que incluyó interfaces interactivas (Suitelet Assistant), procesamiento de archivos XML y generación automatizada de registros y reportes.
- **Manejo de datos estructurados:** Se adquirió experiencia en el análisis y manipulación de archivos XML y estructuras JSON para la extracción, validación y almacenamiento de información crítica.
- **Administración de plataformas empresariales:** Se fortalecieron conocimientos sobre la configuración y uso de un ERP en la nube (NetSuite), así como el uso de herramientas como búsquedas guardadas (Saved Searches), Workbooks y portlets para la visualización de datos financieros.
- **Validación e integración fiscal:** Se desarrollaron mecanismos para validar comprobantes fiscales digitales (CFDI), cumpliendo con los requisitos del SAT en su versión 4.0, asegurando la correcta gestión tributaria de los egresos.

2. Competencias profesionales:

- **Resolución de problemas:** Se abordaron desafíos reales del entorno empresarial mediante el análisis, diseño e implementación de soluciones tecnológicas adecuadas a los procesos de la organización.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

- **Trabajo colaborativo:** Se trabajó en estrecha colaboración con líderes técnicos y equipos multidisciplinarios, aplicando canales de comunicación efectiva.
- **Gestión de proyectos:** Se aplicaron principios de organización, planificación y seguimiento de actividades, cumpliendo con entregables y cronogramas establecidos por la empresa.
- **Autonomía y aprendizaje continuo:** Se fortaleció la capacidad de autoaprendizaje al tomar cursos oficiales de Oracle, investigar documentación técnica y adaptarse al entorno de desarrollo NetSuite, que representa un estándar internacional en soluciones empresariales.

Optimización de la Gestión Financiera ... (Ciclo de egresos)

Fuentes de Información

¿Qué es JavaScript (JS)? (2025). AWS. <https://aws.amazon.com/es/what-is/javascript/>

¿Qué es la ERP? (2025). Oracle. <https://www.oracle.com/mx/erp/what-is-erp/>

¿Qué son los egresos e ingresos y qué tipos existen? (2025). BBVA. <https://www.bbva.com/es/salud-financiera/que-son-los-egresos-e-ingresos-y-que-tipos-existen/>

Dynamics 365. (2021). Qué es Dynamics 365?: <https://www.microsoft.com/es-es/dynamics-365/what-is-dynamics-365>

Facturación electrónica en México. (2024). SOVOS. <https://sovos.com/es/iva/reglas-fiscales/mexico-facturacion-electronica/>

JavaScript. (2025). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/JavaScript>

La factura electrónica en México. (s.f). Edicom. <https://edicom.mx/factura-electronica/mexico>

NetSuite. (2021). NetSuite SuiteCloud Platform: <https://www.netsuite.com/portal/assets/public-pdf/ds-suitecloud-platform.pdf>

Oracle MyLearn. (2025). Componentes principales de SuiteCloud [imagen]. SuiteScript Foundations. <https://mylearn.oracle.com/netsuite/learning-path/suitescript-foundations/>

ORACLE Netsuite. (2021). Plataforma NetSuite SuiteCloud: <https://www.netsuite.com/portal/platform.shtml>

SAP. (2025). ¿Qué es ERP?: <https://www.sap.com/latinamerica/products/erp/what-is-erp.html>

Sistema de planificación de recursos empresariales (ERP) NetSuite. (2025). Oracle NetSuite. <https://www.netsuite.com/portal/products/erp.shtml>

SuiteScript Overview. (s.f). Oracle Help Center. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/article_163726005075.html

Optimización de la Gestión Financiera ... (Ciclo de egresos)

SuiteScript.

(2025).

NetSuite.

<https://www.netsuite.com.au/portal/au/developers/resources/suitescript.shtml>