



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

**OPTIMIZACIÓN DE LA GESTIÓN FINANCIERA Y OPERATIVA
MEDIANTE LA IMPLEMENTACIÓN DE UN SISTEMA BASADO EN
NETSUITE (FINANZAS)**

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

PRESENTA:

EFRÉN GARCÍA HERNÁNDEZ

ASESOR:

M.C. LUCÍA MUÑOZ DÁVILA



APIZACO, TLAX., SEPTIEMBRE 2025



Instituto Tecnológico de Apizaco
Sistemas y Computación

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

Oficio No. S.C./OPV/0254/25

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:	GARCÍA HERNÁNDEZ EFREN
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	20370954
NOMBRE DEL PROYECTO:	Optimización de la gestión financiera y operativa mediante la implementación de un sistema basado en NETSUITE (Finanzas)
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
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera
Expediente
MSP/mrol*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DEPTO. DE
SISTEMAS Y COMPUTACIÓN

LUCIA MUÑOZ DAVILA
ASESORA INTERNA



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**

GARCÍA HERNÁNDEZ EFRÉN
EGRESADO(A) DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES.
CON NÚMERO DE CONTROL 20370954.
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 (FINANZAS)**, 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
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

CECILIA YARELY ARREGUÍN 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

Agradezco a IT Global Knowledge in System Solutions S.A. de C.V., por brindarme la oportunidad de realizar mi residencia profesional en sus instalaciones y proporcionarme un entorno real de trabajo, permitiéndome adquirir experiencia práctica en el desarrollo e implementación de soluciones basadas en NetSuite ERP.

Agradezco también al Ingeniero Oscar Pérez Hernández por compartir sus conocimientos durante el desarrollo del proyecto. Su disposición fue de gran ayuda para alcanzar las metas esperadas dentro del proyecto.

A la licenciada Maricruz Vega Vásquez, al guiarme durante el desarrollo del proyecto y ayudarme a mi crecimiento profesional.

De igual forma agradezco a mis docentes y en especial a mi asesor académico M.C Lucía Muñoz Dávila del Tecnológico Nacional de México, Campus Apizaco, quienes me han brindado las herramientas teóricas y metodológicas necesarias a lo largo de mi formación académica.

A mi familia y amigos, por su apoyo incondicional, comprensión y motivación en cada etapa de este proceso.

A todos los que, de alguna forma, contribuyeron a la realización de este proyecto, les expreso mi más sincero agradecimiento.

Resumen

El presente proyecto tiene el objetivo de desarrollar e implementar soluciones personalizadas en NetSuite ERP, con el propósito de automatizar el proceso de dispersión de pago a proveedores. Estas soluciones buscan optimizar la gestión financiera y operativa dentro del sistema.

Netsuite ERP permite reestructurar los procesos internos para que se adapten a las necesidades de sus usuarios, convirtiéndolo en una herramienta flexible que empodera la transformación digital empresarial. Este proyecto se realizó con las herramientas de la plataforma SuiteCloud, como SuiteScript 2.0.

Índice de Contenidos

1. Generalidades del proyecto	11
1.1. Introducción.....	11
1.2. Descripción de la empresa.....	12
1.3. Descripción del cargo del residente	13
1.4. Problemas a resolver.....	13
1.5. Objetivos.....	14
1.5.1. General.....	14
1.5.2. Específicos.....	14
1.6. Justificación	15
2. Marco Teórico.....	16
2.1. Enterprise resource planning (ERP)	16
2.2. Tipos de ERP	17
2.3. Software como servicio (SaaS)	18
2.4. NetSuite Enterprise Resource Planning (ERP).....	18
2.5. Otros ERP populares.....	20
2.5.1 SAP	20
2.5.2 Microsoft Dynamics 365	21
2.5.3 Intelisis ERP	21
2.6. Sandbox de NetSuite	22
2.7. NetSuite SuiteCloud Platform	23
2.8. SuiteScript	24
2.8.1 Tipos de script de SuiteScript	25
2.9 SuiteBundler	29
2.10 Secure File Transfer Protocol	30
2.11 Búsquedas Guardadas	31

3. Procedimiento y resultados.....	32
3.1 Kick off.....	32
3.2 Personalización.....	34
3.2.1 De la adquisición al pago.....	34
3.2.2 Diagrama	38
3.2.3 Desarrollo del bundle.....	39
3.2.3.1 Pantalla de registro	39
3.2.3.2 Pago a proveedor	43
3.2.3.3 Secure File Transfer Protocol	47
3.2.3.4 Actualización de estado dentro de Netsuite	50
3.2.3.5 Historial de pagos a proveedores.....	54
3.3. Envío de datos de migración	57
3.4 Retroalimentación.....	61
3.5. Migración al entorno de producción.....	64
3.5.1. Creación de bundle	64
3.5.2 Instalación.....	66
3.5.3 ¿Porque hacer Bundles?.....	67
4. Conclusión	68
4.1 Experiencia personal profesional adquirida	68
5. Competencias Desarrolladas y Aplicadas.....	70
5.1. Competencias específicas:	70
5.2. Competencias genéricas:	71
6. Fuentes de información	72

Índice de Ilustraciones

Ilustración 2: Oracle Netsuite	19
Ilustración 3: SAP ERP	20
Ilustración 4: Microsoft Dynamics 365	21
Ilustración 5: SuiteCloud.....	23
Ilustración 6: SuiteCloud Platform Tools	25
Ilustración 7: Ejemplo de Client Script	26
Ilustración 8: Tipos de Script.....	27
Ilustración 9: Diagrama de ejecución de Suitelet	27
Ilustración 10: Ejemplo de RESTlet Script	28
Ilustración 11: De la adquisición al pago Diagrama.....	34
Ilustración 12: Orden de compra	34
Ilustración 13: Aprobación de Compra.....	35
Ilustración 14: Recibir artículo	35
Ilustración 15: Facturación del pedido	36
Ilustración 16: Registro de pago a proveedor.....	36
Ilustración 17: Arquitectura de solución.....	37
Ilustración 18. Diagrama casos de uso.	38
Ilustración 19: NetSuite Pago de factura	39
Ilustración 20. Fragmento de código: validación	40
Ilustración 21. Fragmento de código: Auto rellenó	41
Ilustración 22: NetSuite Pago de factura	41
Ilustración 23: Alerta de usuario.....	42
Ilustración 24: Fragmento de código: Envío de Datos	42
Ilustración 25: Suitelet módulos empleados.....	43
Ilustración 26: Fragmento de código: Consulta a base de datos.....	44
Ilustración 27: Fragmento de código: Campos bloqueados	44
Ilustración 28: Fragmentó de código: Creación de campos	45
Ilustración 29: Pantalla Pago a proveedor	45

Ilustración 30:Notificación de Envío exitoso	46
Ilustración 31: Notificación de error en el sistema.....	47
Ilustración 32: Notificación de pago agendado	47
Ilustración 33: Fragmento de código: Creación de archivo y guardado.....	48
Ilustración 34: Gabinete de archivos	48
Ilustración 35: Carpeta MT101.....	48
Ilustración 36:Modulos empleados en el Envío a SFTP.....	49
Ilustración 37: Fragmento de código conexión a SFTP	49
Ilustración 38: Detalle de pago	50
Ilustración 39: Fragmento de código: Recibo y consulta de datos	51
Ilustración 40: Fragmento de código: actualización de campos.....	51
Ilustración 41. Fragmento de código: suitelet detalle de pago	52
Ilustración 42: Fragmento de código Boton de detalle.....	52
Ilustración 43: Edición - visualización	52
Ilustración 44: Modo edición.....	53
Ilustración 45:Modo visualizar.....	53
Ilustración 46: Pantalla Detalle de pago	53
Ilustración 47: Saved search selección	54
Ilustración 48: Saved Search Nombre	55
Ilustración 49: Saved Search condición.....	55
Ilustración 50: Saved Search criteria	55
Ilustración 51:Vista Pago a proveedores	56
Ilustración 53: ejemplo de consulta SQL	56
Ilustración 53: Saved Search results.....	56
Ilustración 54:File Cabinet	57
Ilustración 55: archivos subidos	57
Ilustración 56: selección de script	58
Ilustración 57: Registro del script.....	58
Ilustración 58: Suitelet script registrado en el sistema	59
Ilustración 59: System notes.....	60
Ilustración 60: Fallo de aprobación	61

Ilustración 61: Error de aprobación	61
Ilustración 62: Cambios realizados en código de aprobación	62
Ilustración 63: Actualizar script.....	63
Ilustración 64: Notificación de correo a testing.....	63
Ilustración 65: Descripción básica de Bundle	64
Ilustración 66: Selección de objetos del bundle	65
Ilustración 67: Listado de objetos de instalación.....	65
Ilustración 68: Guardar el bundle	66
Ilustración 69: búsqueda de bundle	66

Índice de Tablas

Tabla 1: Módulos de SuiteScript	29
---------------------------------------	----

1. Generalidades del proyecto

1.1. Introducción

El proyecto “Optimización de la gestión financiera y operativa mediante la implementación de un sistema basado en NetSuite (Finanzas)” fue elaborado en la empresa IT GLOBAL KNOWLEDGE IN SYSTEM SOLUTIONS S.A. DE C.V. ubicada en la ciudad de Apizaco Tlaxcala, su giro principal es ofrecer servicios generales en consultoría de TI, principalmente sobre tecnologías Oracle.

El presente proyecto tiene el objetivo de desarrollar e implementar soluciones personalizadas en NetSuite ERP, con el propósito de automatizar el proceso de dispersión de pago a proveedores. Estas soluciones buscan optimizar la gestión financiera y operativa dentro del sistema.

Para el desarrollo se usaron las herramientas de la plataforma SuiteCloud en específico SuiteScript en sus versiones 2.0 y 2.1 que permitieron crear módulos adicionales enfocados en la solución. En el presente se detalla cómo se fue el desarrollo y resultado de estos módulos.

1.2. 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.

1.3. Descripción del cargo del residente

Netsuite es un modelo de software como servicio (SaaS), gestiona las funciones básicas de la empresa, como finanzas, contabilidad, inventarios, pedidos y compras.

Dado que se trata de una solución integral, Netsuite no siempre se adapta por completo a las necesidades del usuario, es ahí donde el ingeniero en Tecnologías de la información y comunicación, al cual me referiré como consultor técnico, entra en labor, su función es extender las funciones de NetSuite mediante el desarrollo de soluciones personalizadas (bundles).

El consultor técnico analiza los requerimientos del cliente, propone soluciones y finalmente desarrolla la solución que cumpla con las necesidades del cliente. Haciendo uso de SuiteScript, lenguaje basado en JavaScript, para la automatización de procesos, personalización de interfaces y gestión de los datos, aplicando sus conocimientos adquiridos en la carrera de Tecnologías de la Información y Comunicaciones, además de ética profesional.

1.4. Problemas a resolver

La empresa IT Global Knowledge in System Solutions S.A. de C.V. tiene el reto de adaptar NetSuite ERP a las necesidades específicas del área financiera de su cliente. Aunque NetSuite es una solución integral y robusta, su configuración estándar no contempla todos los procesos particulares de su cliente.

La problemática a resolver en específicos es:

1. Diseñar una solución que cumpla con los requerimientos del cliente.
2. Desarrollar personalizaciones (bundles) según la solución.

1.5. Objetivos

1.5.1. General

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

1.5.2. Específicos

1. Desarrollar soluciones personalizadas (bundles) según los requerimientos del cliente.
2. Enviar los bundles para ser probados y validados.
3. Recibir retroalimentación de pruebas unitarias y hacer correcciones en caso de ser necesario.
4. Migrar los bundles al entorno de producción.

1.6. 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 las necesidades particulares del cliente.

El desarrollo de bundles permitirá configurar NetSuite de acuerdo con los requerimientos 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 el cliente 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.

2. Marco Teórico

2.1. Enterprise resource planning (ERP)

Según (SAP, 2025), “La planificación de recursos empresariales (ERP) es un sistema de software que ayuda a las organizaciones a optimizar sus procesos de negocio centrales, incluyendo finanzas, recursos humanos, fabricación, cadena de suministro y ventas, con una visión unificada de la actividad y una única fuente de verdad”.

Un sistema de software de ERP actúa como el sistema nervioso central de una empresa, ayudando a gestionar eficientemente todos los procesos y operaciones cruciales. Están diseñados en torno a una estructura de datos común, garantizando que la información en toda la empresa esté normalizada y basada en definiciones y experiencias de usuario comunes. Estos sistemas interconectan procesos empresariales asegurando que cada departamento tenga acceso a datos precisos y actualizados.

Los ERP también garantizan que los datos se acumulen correctamente en el libro mayor general de la empresa, facilitando el seguimiento y representación adecuados de los costos. Esto elimina las inconsistencias que podrían surgir al utilizar diferentes nombres o formatos para los mismos elementos en varios sistemas.

Uno de los principios clave de los ERP es la centralización de datos. En lugar de múltiples bases de datos independientes, los sistemas de ERP organizan la información, permitiendo a todos los usuarios acceder a los mismos datos a partir de procesos comunes. Esto asegura integridad y precisión en toda la organización, desde informes financieros trimestrales hasta cuentas pendientes, eliminando la dependencia de hojas de cálculo propensas a errores.

Sus beneficios incluyen:

- Mejor visión del negocio con informes en tiempo real.
- Costos operativos más bajos mediante procesos optimizados.
- Aumento de la colaboración gracias a la compartición de datos.
- Eficiencia mejorada con experiencias de usuario comunes.
- Infraestructura homogénea para todas las actividades empresariales.

- Mayores tasas de adopción de usuarios debido a un diseño común.
- Menor riesgo con mayor integridad de datos y controles financieros
- Reducción de costos de gestión mediante sistemas integrados

2.2. Tipos de ERP

Un sistema de ERP tiene cuatro tipos de implementación: on-premise, en la nube, de dos niveles e híbrido. Cada tipo tiene sus beneficios e inconvenientes, las empresas deben identificar qué opción de implementación de ERP se adapta mejor a sus necesidades.

- ERP on-premise: es el modelo tradicional para implementar software de ERP, es el que permite el mayor nivel de control. El software normalmente se instala en el centro de datos de la empresa. Si bien tiene el control total como un beneficio también trae las responsabilidades de la instalación y el mantenimiento continuo.
- ERP en la nube: el software reside en un servidor remoto de terceros. Se accede a través de internet como un servicio por suscripción (SaaS), donde el proveedor de software gestiona mantenimiento, actualizaciones y seguridad en nombre de la empresa. El ERP en la nube es el método de implementación más popular, ofreciendo costos iniciales más bajos, mayor escalabilidad, innovación más rápida e integración más fácil.
- ERP de dos niveles: El ERP de nivel 1 puede ser un servidor on-premise en la sede central de la organización principal, y el ERP de nivel 2 puede ser un sistema subsidiario basado en la nube que gestiona funciones locales o especializadas. Una estrategia de dos niveles les permite a las empresas iniciar un paso controlado hacia la nube, migrando funciones o unidades de negocio específicas sin disrumpir las operaciones.
- ERP híbrido: Es un tipo de sistema ERP de dos niveles que combina la implementación on-premise y en la nube para darles a las empresas la flexibilidad de elegir la mejor implementación para cada aplicación. El ERP híbrido puede ser un paso hacia un modelo de dos niveles exclusivamente en la nube, mientras se

mantienen aplicaciones que deben estar on-premise para cumplir con regulaciones de la industria o requisitos de seguridad.

2.3. Software como servicio (SaaS)

El software como servicio (SaaS) es un modelo de entrega de software basado en la nube en el que el proveedor de la nube desarrolla y mantiene el software, proporciona actualizaciones automáticas del mismo y lo pone a disposición de sus clientes a través de Internet con un sistema de pago por uso. El proveedor de la nube pública administra todo el hardware y el software tradicional, incluidos middleware, software de aplicaciones y seguridad. De ese modo, los clientes de SaaS pueden reducir drásticamente los costos, implementar, ampliar y actualizar las soluciones empresariales con mayor rapidez de la que proporciona el mantenimiento de sistemas y software locales y predecir el costo total de propiedad con mayor precisión.

2.4. NetSuite Enterprise Resource Planning (ERP)

NetSuite ERP es una solución de gestión empresarial en la nube "todo en uno" basada en IA que ayuda a las organizaciones a operar de manera más eficaz mediante la automatización de los procesos principales y la visibilidad en tiempo real del rendimiento operativo y financiero. Con un conjunto único e integrado de aplicativos para gestionar la contabilidad, el procesamiento de pedidos, la gestión de inventarios, la producción, la cadena de suministro y las operaciones de almacén, NetSuite ERP ofrece a las empresas una clara visibilidad de sus datos y un mayor control de sus negocios.

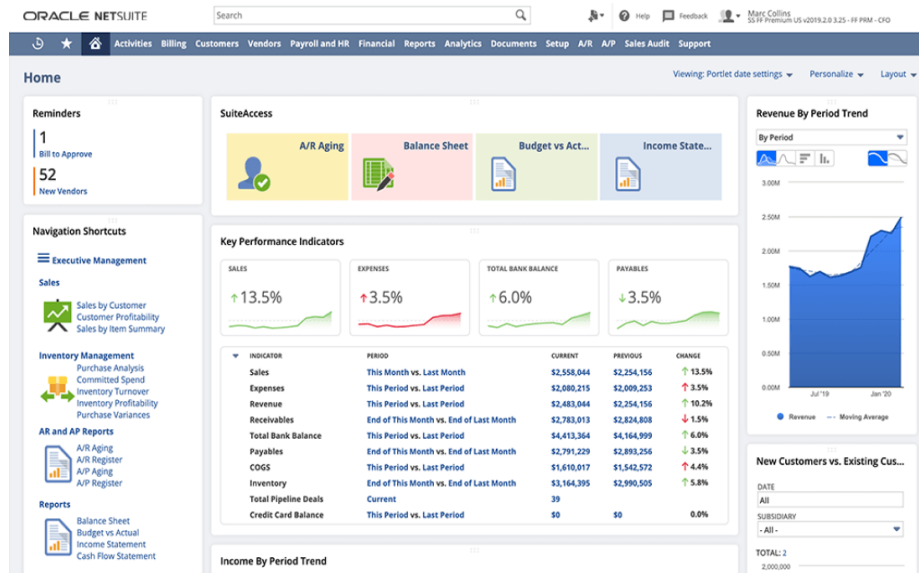


Ilustración 2: Oracle Netsuite

Nota. Netsuite ERP [Imagen], Oracle Netsuite ,2025,

Beneficios de NetSuite ERP

- Automatiza los procesos financieros. Mejora las operaciones financieras, la eficiencia y la productividad.
- Obtén visibilidad del inventario. Controla los niveles de inventario, minimiza los costos de transporte y entrega los pedidos a tiempo.
- Optimización de la cadena de suministro. Controla el flujo de mercaderías en toda la cadena de valor, desde los proveedores hasta el cliente.
- Excelente gestión de pedidos. Gestión de pedidos y aprovisionamiento a prueba de errores.
- Aumenta la eficiencia del almacén. Optimiza el proceso de entradas en almacén y reduce los errores de picking.

2.5. Otros ERP populares

2.5.1 SAP

SAP ERP es un sistema de software integral que optimiza los procesos, mejora la productividad y brinda información estratégica en tiempo real a toda su organización.

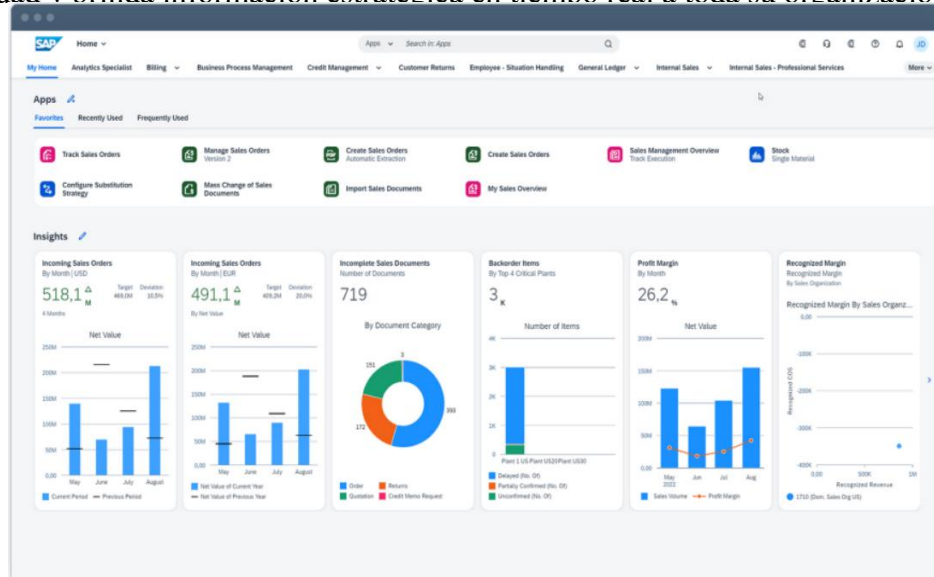


Ilustración 5: SAP ERP

Nota. SAP [Imagen], SAP, 2025,

Beneficios del sistema SAP

- **Eficiencia operativa:** optimiza los procesos de negocio centrales, reduciendo la necesidad de tareas manuales y manteniendo los errores al mínimo.
- **Acceso a datos en tiempo real:** brinda acceso instantáneo a la información más reciente.
- **Informes automatizados:** automatiza la creación de informes detallados y precisos, reduciendo el tiempo dedicado a la recopilación de datos y liberando recursos para actividades más estratégicas.
- **Proyecciones para obtener una ventaja competitiva:** con analíticas avanzadas y capacidades predictivas, SAP ERP empodera a las organizaciones para que proyecten tendencias futuras a fin de lograr una toma de decisiones más proactiva.
- **Ciberseguridad optimizada:** SAP ERP incorpora funciones avanzadas de ciberseguridad y herramientas de compliance para proteger la información de negocios confidencial y mantener el cumplimiento de las regulaciones de la industria.

- Toma de decisiones basada en datos: SAP ERP brinda una única fuente de verdad consolidando datos entre toda la organización. Este enfoque integrado les permite a los líderes basar las iniciativas estratégicas en información estratégica integral en tiempo real.

2.5.2 Microsoft Dynamics 365

Dynamics 365 es una cartera de aplicaciones empresariales inteligentes que ofrece una eficiencia operativa superior y experiencias de cliente innovadoras que permiten a las empresas ser más ágiles y reducir la complejidad sin aumentar los costes. Permite integrar todos tus procesos de negocio en una misma plataforma, desde las finanzas hasta el marketing, pasando por customer service o ventas.

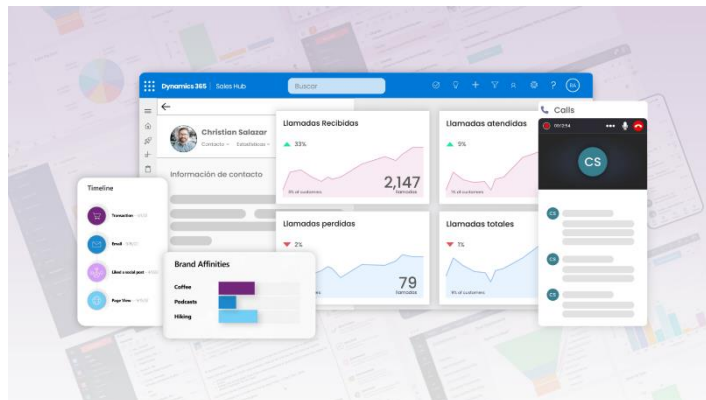


Ilustración 8: Microsoft Dynamics 365

Nota. Microsoft Dynamics [Imagen], Dynamics 365, 2021

Ilustración 9: Microsoft Dynamics 365

2.5.3 Intelisis ERP

Intelisis ERP es la solución de gestión empresarial que integra toda la administración y operación de manera inteligente. Accede y controla la información que se genera a diario en las diferentes áreas clave de la empresa, permite que todos trabajen bajo los mismos

lineamientos, mediante el registro de cada movimiento en línea y tiempo real, mismo que se va desarrollando de forma lógica e intuitiva dentro de un flujo operacional determinado.

Beneficios:

- Sistema responsivo. Accede a tu información desde cualquier dispositivo, en tiempo real.
- Intelisis ERP crece a la misma velocidad que tu negocio sin interrumpir tus operaciones. Integración de las mejores prácticas en seguridad de datos independientemente de la arquitectura del sistema
- Control de procesos de trabajo para el incremento de la productividad y gestión de errores.
- Servicio de soporte técnico directo, implementación, migración de datos y actualizaciones.
- Distintos tipos de arquitectura dependiendo de las necesidades de tu empresa: Híbrido, en sitio o en la nube.
- Personaliza el sistema a multi-moneda, multi-nivel, multi-usuario y multi-idioma.

2.6. Sandbox de NetSuite

Un sandbox de NetSuite es una cuenta de prueba segura y aislada donde puede desarrollar y probar nuevas aplicaciones y personalizaciones sin preocuparse por afectar la cuenta de producción. Puede tener uno o más sandboxes aprovisionados para cada cuenta de producción. Cada cuenta de sandbox tiene la misma configuración, datos y personalizaciones que la cuenta de producción de su empresa.

Las cuentas Sandbox ofrecen lugares ideales para:

- Prueba de elementos de personalización antes de la implementación.
- Solución de problemas con personalizaciones existentes.
- Probar funciones de integración de terceros.

- Capacitar a los empleados en prácticas comerciales existentes y nuevas en una cuenta idéntica a su cuenta de producción.

Las acciones que realice en cuentas de sandbox, como entradas de transacciones, eliminación de registros y personalizaciones de formularios, no afectan a su cuenta de producción de NetSuite. Puede realizar cambios en una cuenta de sandbox sin modificar sus datos de producción.

2.7. NetSuite 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 tecnologías de la información, 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.

SuiteCloud ofrece una amplia gama de opciones, desde personalización sin código hasta desarrollo de aplicaciones comerciales, que minimizan el tiempo y el esfuerzo necesarios para agregar funcionalidad personalizada y brindar experiencias de usuario adaptadas a sus necesidades comerciales específicas.

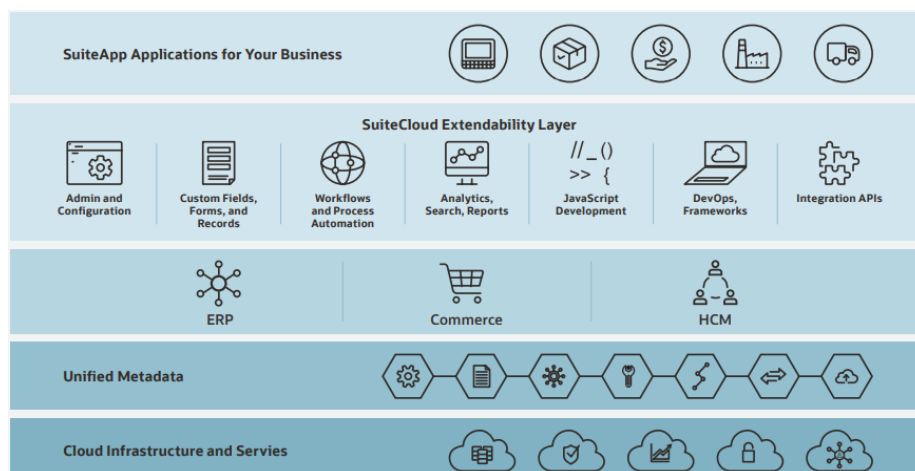


Ilustración 11: SuiteCloud

Nota. Suitecloud application task [Imagen], Netsuite, 2025. Página 23 | 73

2.8. SuiteScript

Dentro del catálogo de herramientas provistas por SuiteCloud Platform se encuentra SuiteScript. El lenguaje de programación basado en JavaScript, ofrece funciones completas a nivel de aplicación que admiten lógica procedimental tanto en el lado del cliente como del servidor. SuiteScript se usa para ampliar NetSuite y personalizar, buscar y procesar sus datos. También dispone de un depurador de SuiteScript para depurar scripts de servidor.

Las características destacadas de SuiteScript son:

- Modularidad: admite la encapsulación, proporciona una organización de código intuitiva y garantiza que no haya conflictos de nombres de métodos o variables globales.
- Gestión automática de dependencias: le permite concentrarse en la lógica en lugar de en las dependencias y el orden de carga.
- JavaScript: diseñado para soportar la mayoría de los estándares de JavaScript, brindándole acceso programático a la funcionalidad de NetSuite.
- Bibliotecas de JavaScript de terceros: utilice módulos personalizados para cargar sus bibliotecas de JavaScript de terceros preferidas.
- Funcionalidad mejorada: compatibilidad con todos los tipos de contenido HTTP, transferencia de archivos SFTP, almacenamiento en caché de datos, paginación de búsqueda, transmisión de archivos planos, cifrado, descifrado y hash mejorados.
- Procesamiento asíncrono de scripts de cliente: proporciona promesas y scripts de mapeo/reducción; un marco estructurado para scripts de servidor que procesan una gran cantidad de registros

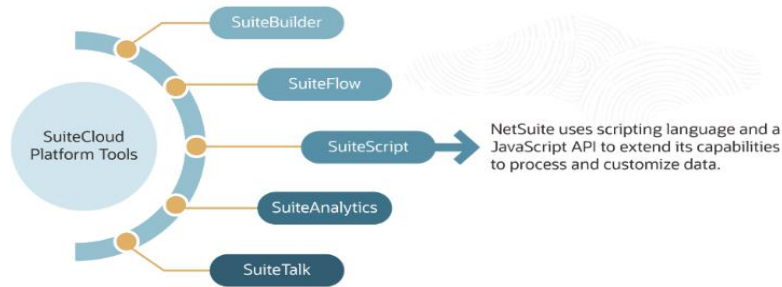


Ilustración 14: SuiteCloud Platform Tools

Nota. SuiteScript Foundations [Curso], ORACLE Mylearn, 2025,

2.8.1 Tipos de script de SuiteScript

SuiteScript ofrece varios tipos de scripts, Cada tipo tiene un propósito específico y se ejecuta en diferentes momentos del ciclo de vida de los registros o del sistema. Es fundamental conocer todos los scripts con el objetivo de usar los que mayor se acoplen a las necesidades del desarrollo,

Client Script

Son scripts que se ejecutan mediante activadores de eventos predefinidos en el navegador del cliente. Permiten validar los datos introducidos por el usuario y rellenar automáticamente campos o sublistas en diversos eventos de formulario. Se pueden ejecutar en la mayoría de los registros estándar, tipos de registros y páginas personalizados de NetSuite, como Suitelets.

Los activadores (triggers) que pueden ejecutar el script son:

- Cargar un formulario para editarlo
- Ingresar o cambiar un valor en un campo (antes y después de ingresarlo)
- Ingresar o cambiar un valor en un campo que genera otro campo
- Seleccionar un elemento de línea en una sublista
- Agregar una línea de artículo (antes y después de ingresarla)
- Guardar un formulario

Los scripts de cliente a nivel de registro se ejecutan después de que se ejecuta cualquier cliente basado en formulario existente y antes de que se ejecute cualquier script de evento de usuario.

```
/**
 *@NApiVersion 2.x
 *@NScriptType ClientScript
 */
define(['N/error'], function(error) {
    function pageInit(context) {
        if (context.mode !== 'create')
            return;
        var currentRecord = context.currentRecord;
        currentRecord.setValue({
            fieldId: 'entity',
            value: 107
        });
    }
    function saveRecord(context) {
        var currentRecord = context.currentRecord;
        if (!currentRecord.getValue({fieldId: 'entity'})
            throw error.create({
                name: 'MISSING_REQ_ARG',
                message: 'Please enter all the necessar
            });
        return true;
    }
});
```

Ilustración 17: Ejemplo de Client Script

Nota. *Client Script* [Imagen], Netsuite help center,

User Event Script

Los scripts de eventos de usuario se ejecutan en el servidor de NetSuite. Se ejecutan cuando los usuarios realizan ciertas acciones en los registros, como crear, cargar, actualizar, copiar, eliminar o enviar. La mayoría de los registros estándar de NetSuite y los tipos de registros personalizados admiten scripts de eventos de usuario.

Los scripts de eventos de usuario se pueden utilizar para realizar las siguientes tareas:

- Implementar validación personalizada en registros
- Hacer cumplir la integridad de los datos definidos por el usuario y las reglas comerciales

- Realizar comprobaciones de permisos definidos por el usuario y registrar restricciones
- Implementar la sincronización de datos en tiempo real
- Definir flujos de trabajo personalizados (acciones de redirección y seguimiento)
- Personalizar formularios.

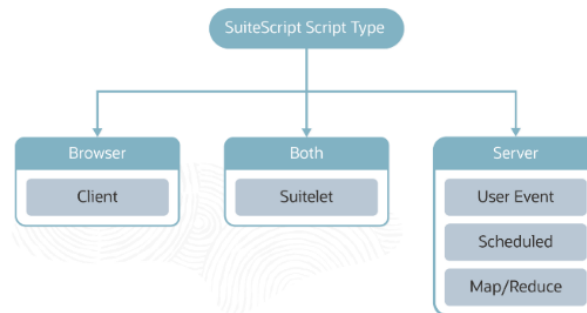


Ilustración 20: Tipos de Script

Nota. SuiteScript Script Type [Imagen], Oracle Mylearn, 2025,

Suitelet Script

Los Suitelets son extensiones de la API SuiteScript que permiten crear páginas NetSuite personalizadas y lógica de backend. Los Suitelets son scripts de servidor que operan con un modelo de solicitud-respuesta y se invocan mediante solicitudes HTTP GET o POST a las URL generadas por el sistema.

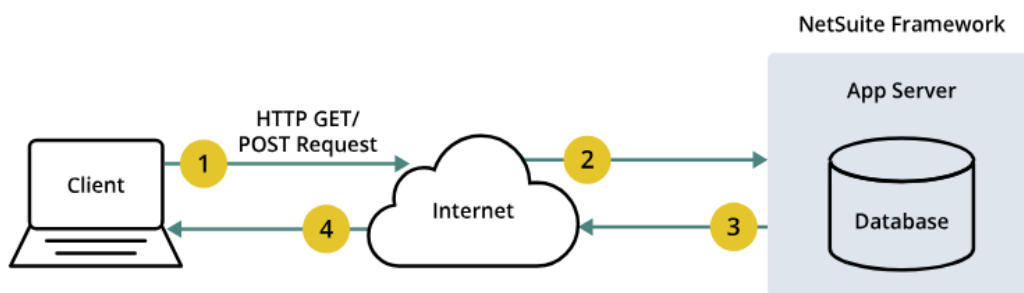


Ilustración 23: Diagrama de ejecución de Suitelet

Nota. Suitelet execution [Imagen], Netsuite help center, 2025,

RESTlet Script

Un RESTlet es un SuiteScript que se pone a disposición de otras aplicaciones para que lo invoquen. Puede invocarse desde una aplicación externa o desde otro script dentro de NetSuite. Un RESTlet se ejecuta solo cuando se invoca y, en algunos casos, devuelve un valor a la aplicación que lo invoca. Los RESTlets pueden ser útiles al importar datos a NetSuite desde otro sistema o al extraerlos de NetSuite. También pueden usarse en combinación con otros scripts para personalizar el comportamiento de una página en NetSuite.

```
/**
 * @NApiVersion 2.x
 * @NScriptType Restlet
 */
define([ 'N/record' ], function(record) {
    return {
        post: function(restletBody) {
            var restletData = restletBody.data;
            for (var contact in restletData) {
                var objRecord = record.create({
                    type : record.Type.CONTACT,
                    isDynamic : true
                });
                var contactData = restletData[contact];
                for (var key in contactData) {
                    if (contactData.hasOwnProperty(key)) {
                        objRecord.setValue({
                            fieldId : key,
                            value : contactData[key]
                        });
                    }
                }
            }
            var recordId = objRecord.save({
                enableSourcing : false
            });
        }
    };
});
```

Ilustración 25: Ejemplo de RESTlet Script

Nota. *RESTlet Script* [Imagen], Netsuite help center, 2025,

Módulos suiteScript

AMD (Asynchronous Module Definition) es un patrón de codificación que permite seleccionar qué módulos cargar en una función de script, en lugar de tener un script pesado que cargue todos los módulos posibles o bibliotecas completas de funcionalidad, incluso cuando no todos son necesarios.

Con AMD, los scripts son mucho más livianos, rápidos y eficientes, ya que utilizan solo los módulos necesarios para un contexto de ejecución específico.

Tabla 1: Módulos de SuiteScript

Módulo	Descripción
N/record	Se utiliza para crear, leer, actualizar y eliminar registros estándar o personalizados en NetSuite.
N/file	Permite interactuar con archivos en el File Cabinet: crear, leer, actualizar, copiar y eliminar archivos.
N/search	Se utiliza para ejecutar búsquedas guardadas o personalizadas para consultar datos dentro de NetSuite. Es similar a usar SQL
N/redirect	Redirige al usuario a registros, páginas o scripts específicos dentro de NetSuite.
N/ui/serverWidget	Permite crear interfaces personalizadas del lado del servidor como formularios, campos y botones.

Nota: Elaboración propia

2.9 SuiteBundler

SuiteBundler permite a los usuarios de NetSuite agrupar grupos de objetos para distribuirlos a otras cuentas. Estos paquetes se denominan bundles o SuiteApps. Los creadores de bundles pueden ser desarrolladores internos que crean personalizaciones para sus empresas, proveedores independientes de software (ISV) que distribuyen soluciones a sus clientes o administradores que publican sus bundles a través de una cuenta de implementación.

Se pueden crear dos tipos de paquetes y ponerlos a disposición para su instalación:

- Un paquete de personalización es un grupo de objetos personalizados que implementan un comportamiento personalizado en la aplicación NetSuite.
- Un paquete de configuración es un grupo de entradas de configuración y configuraciones de preferencias de NetSuite.

2.10 Secure File Transfer Protocol

File Transfer Protocol, o Protocolo de Transferencia de Archivos (FTP) es un protocolo de Internet para la transferencia de archivos. FTP resulta especialmente importante porque es el mecanismo que se usa para subir los ficheros de los sitios web, generalmente desarrollados en un ordenador local, al servidor remoto.

Este es el mecanismo habitual en los alojamientos tradicionales, o también llamado hosting compartido. Dado que el servidor web se encuentra en un centro de datos al que no tenemos acceso físico, es necesario conectarnos con él por otros medios para enviarle los archivos html, CSS, Javascript, imágenes, txt, etc.

Para conectarnos por FTP al servidor usamos clientes de FTP, que permiten enviar y recibir datos entre el servidor y nuestro ordenador local.

SFTP es lo mismo que el protocolo FTP pero implementado por un canal seguro. Son las siglas de SSH File Transfer Protocol y consiste en una extensión de Secure Shell Protocol (SSH) creada para poder hacer transmisiones de archivos.

La seguridad que nos aporta SFTP es importante para la transferencia de archivos porque, si no disponemos de ella, los archivos viajarán tal cual, por la red, sin ningún tipo de encriptación. Así pues, usando FTP tradicional, si algún agente consigue escuchar las transferencias, podría ocurrir que la información quedase al descubierto. Esto sería especialmente importante si los archivos que subimos contienen información confidencial o datos personales.

2.11 Búsquedas Guardadas

Una búsqueda guardada es una definición de búsqueda reutilizable que puede incluir numerosos filtros de búsqueda avanzados y opciones de visualización de resultados. Los resultados de búsqueda guardados proporcionan informes y seguimiento, y pueden servir de base para el análisis empresarial y la toma de decisiones estratégicas. Es muy similar a utilizar SQL sin embargo la búsqueda guardada tiene la peculiaridad de hacerlo por interfaz gráfica, diseñada para facilitar la toma de decisiones y optimizar el tiempo.

Las búsquedas guardadas o Saved Searches son útiles para:

- Generar reportes personalizados sin necesidad de crear consultas manuales cada vez.
- Automatizar alertas por correo electrónico cuando ciertos datos cambian.
- Filtrar información de transacciones, clientes, proveedores y otros registros.
- Integrar datos con otras herramientas dentro de NetSuite.

3. Procedimiento y resultados

3.1 Kick off

El proyecto dio inicio con la asignación formal del equipo de trabajo por parte del líder de proyecto. Posteriormente, se llevó a cabo una reunión grupal, donde se presentó el equipo y se discutieron los pormenores del proyecto.

En una segunda reunión, se presentó la lista de requerimientos del cliente y definieron las actividades específicas además de la distribución de las tareas dentro del equipo. También se acordaron los medios de comunicación y las metodologías de trabajo que se utilizarían a lo largo del proyecto.

Como parte del proceso de formación, se organizó la inscripción a un curso en línea proporcionado por Oracle, enfocado en el uso y desarrollo de SuiteScript, con el objetivo de familiarizar con la herramienta y aplicar los conocimientos adquiridos al desarrollo del proyecto.

Los cursos mencionados son:

- SuiteScript Foundations: proporciona una introducción a SuiteScript engloba desde temas básicos hasta un nivel intermedio, está diseñado para desarrolladores principiantes en SuiteScript que deseen aprender a personalizar NetSuite.
- SuiteScript Development: curso avanzado ofrecido por NetSuite que está diseñado para desarrolladores que ya tienen experiencia básica con SuiteScript y desean profundizar en técnicas de desarrollo más complejas y en la personalización avanzada de NetSuite

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

Requerimientos del sistema:

1. Automatizar la dispersión de pago a proveedores en Netsuite.

2. Validar información del proveedor previas al pago:
 - a. Confirmar que la cuenta del proveedor es válida.
 - b. Verificar disponibilidad de fondos en la cuenta de la empresa.
3. El sistema debe enviar la información de pago al banco una vez hecho el registro.
4. Mantener un registro de los pagos realizados y su estado (pendiente, aprobado, rechazado).
5. Solo usuarios con los permisos adecuados pueden registrar y autorizar pagos.
6. Todos los intentos de pago deben ser registrados para auditoría y diagnóstico en caso de errores.

3.2 Personalización

En esta etapa del desarrollo del bundle, se enfocó en la creación de una solución adaptada a los requerimientos del cliente. Esta solución fue diseñada específicamente para la dispersión de pagos a proveedores, como una extensión de las funciones de Netsuite. Para dar inicio al proceso, es fundamental comprender el flujo de trabajo conocido como “De la adquisición al pago”. A continuación, se describe el proceso.

3.2.1 De la adquisición al pago

El proceso De la adquisición al pago de NetSuite permite iniciar y realizar el seguimiento de los bienes y servicios suministrados por los proveedores, que son fundamentales para el funcionamiento de la empresa.



Ilustración 28: De la adquisición al pago Diagrama

De la adquisición al pago [imagen]. Oracle sites, 2016

1. El proceso de compra inicia mediante la creación de un Pedido de compra hecho por el empleado. (ilustración 12)

La imagen muestra la interfaz de usuario de NetSuite para la creación de un "Purchase Order". La barra superior incluye el logo de Oracle NetSuite, el nombre de usuario "Global", el entorno "SANDBOX" y una barra de búsqueda. El menú principal contiene opciones como Activities, Vendors, Financial, Reports, Analytics, Documents, Setup, Pagos, SuiteApps y Support. El formulario principal está dividido en secciones: "Primary Information" (donde se selecciona el "CUSTOM FORM" como "Standard Purchase Order", se ingresa el "VENDOR #", se elige el "VENDOR" como "Vendor Performance" y se ingresa el "EMPLOYEE"), "Classification" (donde se selecciona el "SUBSIDIARY" como "SALTECH", el "DEPARTMENT", la "CURRENCY" como "MEX" y el "TIPO DE OPERACIÓN" como "Others"), y un "Summary" que muestra el "SUBTOTAL" y el "TAX TOTAL" como "0.00".

Ilustración 29: Orden de compra

2. Posteriormente pasa a ser una Solicitud de compra, que luego se deriva para su aprobación por parte del supervisor. (Ilustracion 13)

Purchase Order PO-00000003 7 PENDING SUPERVISOR APPROVAL

Información principal

VENDOR #	RECEIVE BY	21/02/2025
VENDOR	DATE	20/02/2025
EMPLOYEE	PO #	PO-00000003
SUPERVISOR APPROVAL	MEMO	Pedido de neumático

SUBTOTAL	40,000.00
TAX TOTAL	6,400.00
IVA 16%	6,400.00
TOTAL	46,400.00

Clasificación

SUBSIDIARY	CLASS	LOCATION
IT SOFTWARE IMPLEMENTATION	CONSULTORIA FUNCIONAL	CIUDAD DE MEXICO
DEPARTMENT		
DIRECCION GENERAL : DIRECCION CONSULTORIA FUNCIONAL : GERENCIA		
CONSULTORIA FUNCIONAL : LIDER CONSULTORIA FUNCIONAL :		
CONSULTORIA FUNCIONAL		

Ilustración 32: Aprobación de Compra

3. Una vez aprobado se convierte en un Pedido de compra, el cual se envia al proveedor para ser surtido.
4. El proveedor entrega los artículos comprados, al mismo tiempo se hace la recepción del producto.

Item Receipt RCV-00000008 6

Primary Information

REFERENCE #	DATE
RCV-00000008	20/02/2025
VENDOR	POSTING PERIOD
6	feb 2025
CREATED FROM	MEMO
Purchase Order #PO-00000012	Total

Classification

SUBSIDIARY	CLASS
IT SOFTWARE IMPLEMENTATION	CONSULTORIA FUNCIONAL
DEPARTMENT	
6	
CURRENCY	
MEX	

Items & Expenses

ITEM	VENDOR NAME	DESCRIPTION	TO LOCATION	ON HAND	QUANTITY	UNITS	INVENTORY DETAIL	OPTIONS	RATE	CURRENCY	LANDED COST
Paquete palo de paleta	MP 33	Paquete con palos para paleta	TLAXCALA	5	5	Pq			150.00	MEX	

Ilustración 33: Recibir articulo

5. Una vez que el proveedor envía la factura correspondiente a la compra, el contador crea dentro del sistema una Factura del proveedor a partir del Pedido de compra(3) o del Recibo del artículo(4) y liga la factura recibida y la creada en el sistema para llevar un control.

EXP-00000004		Summary
EXP. REPT. A EXP-00000004	POSTING PERIOD feb 2025	EXPENSES TOTAL
EMPLOYEE	DATE DUE	TOTAL IN BASE CURRENCY
CURRENCY MEX	APPROVAL STATUS Approved	REIMBURSABLE EXPENSES
EXCHANGE RATE 1.00	NEXT APPROVAL	CORPORATE CARD
ADVANCE TO APPLY		TAX TOTAL
PURPOSE F		IVA 16%
DATE 10/02/2024		ADVANCE TO APPLY
		TOTAL REIMBURSABLE AMOUNT

Ilustración 36: Facturación del pedido

6. Se genera un Registro de pago a partir de la Factura del proveedor cuando se paga la factura.

Ilustración 39: Registro de pago a proveedor

Una vez registrado el pago, el proceso dentro del sistema concluye. Sin embargo, en este punto, el pago al proveedor aún no ha sido efectuado. Para completar la dispersión, es necesario enviar la información del pago al banco para su procesamiento.

Para dar una idea más a detalle de lo que se requiere implementar, en la ilustración 17, se presenta el diagrama: arquitectura de solución.

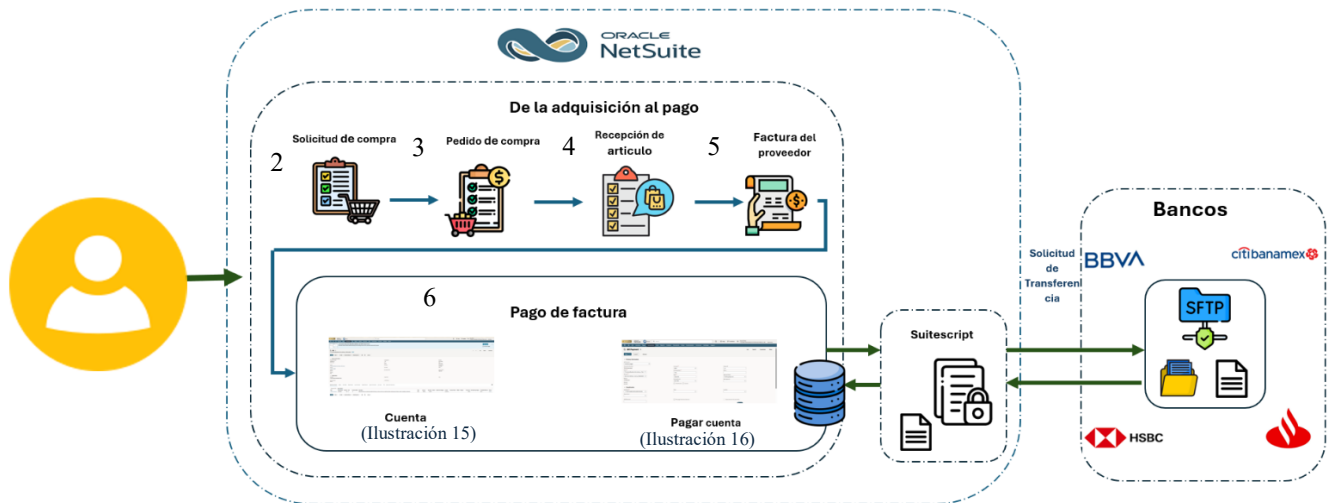


Ilustración 41: Arquitectura de solución

Nota. Elaboración propia

Retomando el proceso de la Adquisición al pago.

- Pago factura: Registra el pago en el sistema.
- SuiteScript: Accede a la base de datos, extrae los datos de transferencia, protege la información y la envía a bancos a través de SFTP (Protocolo de Transferencia Segura de Archivos).
- Bancos: Reciben los datos, ejecuta la transacción y notifican una respuesta.
- SuiteScript: Lee la respuesta del banco y actualiza el estado del pago en el sistema.

Este flujo mejorado no solo garantiza el registro del pago, sino también la dispersión efectiva de este, y asegura el cumplimiento de las necesidades del cliente.

3.2.2 Diagrama

En este apartado se mostrará como es el proceso de ejecución del sistema a desarrollar desde la perspectiva del usuario. Este proceso se encuentra en el paso Pago de factura del proceso de la adquisición al pago.

Diagrama de Casos de Uso

Describe las principales funcionalidades del sistema desde la perspectiva del usuario. En otras palabras, muestran el comportamiento esperado del sistema.

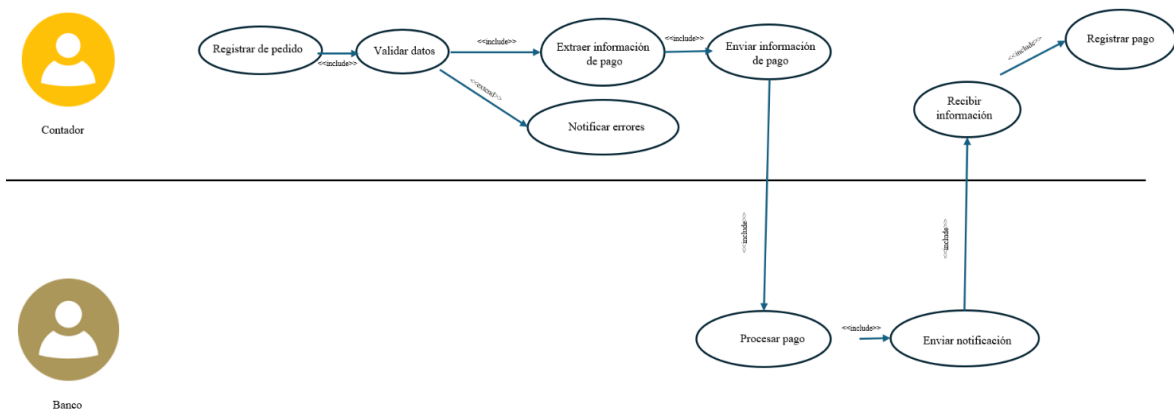


Ilustración 43. Diagrama casos de uso.

Nota: Elaboración propia

- Contador: es el responsable de realizar los pagos a proveedor.
- Registrar información de pedido: el contador registra la información del movimiento, por ejemplo, a que cuenta del catálogo de cuentas será registrado el movimiento y que banco depositará a la cuenta del proveedor, etc.
- Validar datos: el sistema revisa datos internamente para asegurar que los datos son correctos.
- Extraer información de pago: el sistema extrae la información de pago de la base de datos.

- Enviar información de pago al banco: el sistema crea un archivo que posteriormente envía al banco a través de un servicio SFTP (Protocolo de Transferencia Segura de Archivos).
- Registrar pago: una vez realizado el pago, el sistema registra la información en la base de datos

3.2.3 Desarrollo del bundle

Una vez explicada la solución y como es el proceso que ejecuta el sistema, se da paso con la demostración del desarrollo.

3.2.3.1 Pantalla de registro

El primer paso en el desarrollo es la pantalla de registro, esta pantalla es propia de NetSuite (Ilustración 19), desde esta pantalla se configuran los parámetros fundamentales que darán inicio al flujo automatizado dentro del sistema.

Bill Payment

Save Cancel Actions

Primary Information

A/P ACCOUNT: Cuentas a pagar 1

TRANSACTION NUMBER To Be Generated: 2

PAYEE: 3

ACCOUNT: 4

BALANCE: 579090.00

AMOUNT: 626.40 5

CURRENCY: MEX

EXCHANGE RATE: 1.00

DATE: 02/04/2025 6

POSTING PERIOD: abr 2025

CHECK #: 22

MEMO: 7

APPROVAL STATUS: Pending Approval 8

NEXT APPROVER:

Classification

Apply Payee Address Relationships Communication IEF Payment Method

SELECT ITEM:

Mark All Unmark All Customise

APPLY	DATE DUE	TYPE	REF NO.	INSTALLMENT REF NO.	ORIG. AMT.	AMT. DUE	CURRENCY	DISC. DATE	DISC. AVAIL.	DISC. TAKEN	PAYMENT
9	25/03/2025	Bill			626.40	626.40	MEX			0.00	626.40

Save Cancel Actions

Ilustración 46: NetSuite Pago de factura

En la pantalla Bill Payment (Pago de factura) se registra la información de pago al proveedor:

1. A/P ACCOUNT (Cuenta de AP): Registra a que cuenta del catálogo de cuentas será abonado este movimiento
2. TRANSACTION NUMBER (Numero de transacción): número de referencia del movimiento
3. PAYEE (Beneficiario): Selecciona al proveedor a pagar
4. ACCOUNT (Cuenta): Selecciona la cuenta bancaria beneficiaria
5. AMOUNT (Importe): Describe la cantidad a pagar.
6. DATE (Fecha): Fecha en la que se realiza el pago
7. MEMO (Nota): Descripción del pago que realizas.
8. APPROVAL STATUS (Estado de aprobación): Tiene tres estados, rechazado, aprobación pendiente, aprobado.
9. APPLY: Muestra el detalle de la orden

Se modifico la pantalla para hacer validaciones antes de registrar el pago, se presenta un fragmento del código desarrollado y el resultado (Ilustración 20):

```

115
116     var currentRecord = context.currentRecord;
117     var approvalStatus = currentRecord.getValue({ fieldId: 'approvalstatus' });
118
119     if (approvalStatus !== 'Aprobado') {
120         dialog.alert({
121             title: 'Validación Requerida',
122             message: 'El campo "Estado de Aprobación" debe ser aprobado antes de guardar.'
123         })
124         .then(() => {
125             log.debug({ title: 'Validación fallida', details: 'Estado de aprobación no válido.' });
126         })
127         .catch((error) => {
128             log.error({ title: 'Error mostrando la alerta', details: error });
129         });
130
131         return false;
132     }
133
134
135

```

Ilustración 48. Fragmento de código: validación

Se obtiene el valor del campo "Approval Status" y se establece una alerta si el valor no es "Aprobado"

Anteriormente se podía registrar el pago sin necesidad de cambiar el estado, porque el contador era su propio supervisor, además de que era una forma de saber que pagos se habían postergado, sin embargo, en los reportes había incongruencia con los datos, esta validación permite notificar al usuario a cambiar el estado y como complemento este campo se tornó obligatorio.

Otro elemento necesario es el relleno del campo “memo”, aquí se registra la factura que se liquida, y se usa como referencia. Se accede al registro y sustraemos la información requerida (Ilustración 21).

```

69
70 var form = context.form;
71 var approvalStatusField = form.getField({ id: 'approvalstatus' });
72
73 if (approvalStatusField) {
74     approvalStatusField.isMandatory = true;
75 }
76
77 var vendorPayment = context.newRecord;
78 var lineCount = vendorPayment.getLineCount({ sublistId: 'apply' });
79
80 for (var i = 0; i < lineCount; i++) {
81     var refNo = vendorPayment.getSublistValue({ sublistId: 'apply', fieldId: 'refnum', line: i });
82     var amountDue = vendorPayment.getSublistValue({ sublistId: 'apply', fieldId: 'due', line: i });
83     var paymentAmount = vendorPayment.getSublistValue({ sublistId: 'apply', fieldId: 'amount', line: i });
84
85     log.debug({
86         title: 'Registro Aplicado en Inicio',
87         details: {
88             refNo: refNo,
89             amountDue: amountDue,
90             paymentAmount: paymentAmount
91         }
92     });
93     memoText = 'Pago Factura: ${refNo}';
94     vendorPayment.setValue({ fieldId: 'memo', value: memoText });
95 }
96
97
98
99

```

Ilustración 49. Fragmento de código: Auto relleno

Se muestran las modificaciones realizadas (Ilustración 22)

The screenshot displays the 'Pago de factura' (Invoice Payment) form in NetSuite. The form is organized into several sections:

- Información principal:** Includes fields for 'Cuenta de AP' (Accounts Payable), 'Número de transacción' (Transaction Number), 'A generar' (To be generated), 'Beneficiario' (Beneficiary), 'Cuenta' (Account), 'Saldo' (Balance), 'Importe' (Amount), and 'Moneda' (Currency).
- Clasificación:** Includes fields for 'Subcategoría' (Subcategory), 'Departamento' (Department), 'Clase' (Class), and 'Ubicación' (Location).
- Approval and Other Fields:** Includes 'Cheque N°' (Check Number), 'Nota' (Note), 'Estado de aprobación' (Approval Status), 'Próximo aprobador' (Next Approver), 'Tipo de cambio' (Exchange Rate), 'Fecha' (Date), 'Período contable' (Accounting Period), and checkboxes for 'Para imprimir' (To print) and 'Comprobante' (Receipt).

The form is currently showing a balance of 579090.00 and an amount of 626.40. The approval status is set to 'Aprobación pendiente' (Pending approval).

Ilustración 52: NetSuite Pago de factura

1. Se auto rellena con el número de factura y la leyenda “Pago Factura”.
2. El campo ahora es obligatorio y debe de cumplir el estado “Aprobado” para continuar, en caso contrario se muestra una alerta. (Ilustración 23)
3. Si la cuenta seleccionada no tiene fondos, se alerta al usuario antes de continuar.

Ilustración 53: Alerta de usuario

Después de pasar las primeras validaciones, y de dar clic en guardar, el sistema nos redirige a la pantalla Pago a proveedor, desarrollada completamente con SuiteScript 2.1.

```

112
113 | var newRecord = context.newRecord;
114 | var vendorId = newRecord.getValue({ fieldId: 'entity' });
115 | var descripcion = newRecord.getValue({ fieldId: 'memo' });
116 | var divisa = newRecord.getValue({ fieldId: 'currency' });
117 | var titular = newRecord.getText({ fieldId: 'subsidiary' });
118 | var total = newRecord.getValue({ fieldId: 'total' });
119
120 | log.debug({
121 |   title: "Redirección - Datos del Registro",
122 |   details: {
123 |     vendorId: vendorId,
124 |     descripcion: descripcion,
125 |     divisa: divisa,
126 |     titular: titular,
127 |     total: total
128 |   }
129 | });
130
131 | redirect.toSuitelet({
132 |   scriptId: 'c...',
133 |   deploymentId: 'c...',
134 |   parameters: {
135 |     sdr_vendor_id: vendorId,
136 |     sdr_descripcion: descripcion,
137 |     sdr_divisa: divisa,
138 |     sdr_titular: titular,
139 |     sdr_total: total
140 |   }
141 | });

```

Ilustración 56: Fragmento de código: Envío de Datos

Antes de ser redirigido, el sistema obtiene los datos y los reenvía a la pantalla Pago proveedor, usando el Método Get

3.2.3.2 Pago a proveedor

Esta pantalla muestra un resumen de la información bancaria de la empresa y del proveedor, es una forma de verificar que el movimiento sea correcto. Para el desarrollo se usó un Suitscript del tipo suitelet y el uso de los módulos:

- N/record: permite crear, cargar, editar y guardar registros en NetSuite.
- N/redirect: se utiliza para redirigir la interfaz del usuario a una página específica en NetSuite
- N/ui/serverWidget: para la creación de formularios personalizados en NetSuite, como formularios de entrada de datos, botones personalizados o interfaces gráficas.
- N/search: crear búsquedas personalizadas programáticamente, similar a hacer consultas en SQL.

```

1  /**
2   * @NApiVersion 2.1
3   * @NScriptType Suitelet
4   */
5  define(['N/email', 'N/record', 'N/redirect', 'N/ui/serverWidget', 'N/search'],
6
7      /**
8       * @param{email} email
9       * @param{record} record
10      * @param{redirect} redirect
11      * @param{serverWidget} serverWidget
12      * @param{search} search
13      */
14      (email, record, redirect, serverWidget, search) => {

```

Ilustración 57: Suitelet módulos empleados

La pantalla recibe la información de la actividad anterior “Pago de factura”, y consulta la base de datos para obtener la información bancaria del proveedor, por ejemplo: número de cuenta, banco, titular entre otros datos relevantes para hacer el pago. (Ilustración 26)

```

var bankDetailsSearch = search.create({
  type: 'CUSTOMRECORD_2663_ENTITY_BANK_DETAILS',
  filters: [['custrecord_2663_parent_vendor', 'anyof', vendorId]],
  columns: [
    'name',
    'custrecord_2663_entity_bank_type'
  ]
});

var resultSet = bankDetailsSearch.run().getRange({ start: 0, end: 1 });

var vendorName = ''; var bankType = ''; var accountNo = ''; var accountName = ''; var bankNo =

if (resultSet.length > 0) {
  var bankDetailId = resultSet[0].id;
  var bankRecord = record.load({
    type: 'CUSTOMRECORD_2663_ENTITY_BANK_DETAILS',
    id: bankDetailId
  });

  vendorName = bankRecord.getValue({ fieldId: 'name' });
  bankType = bankRecord.getValue({ fieldId: 'custrecord_2663_entity_bank_type' });
}

```

Ilustración 60: Fragmento de código: Consulta a base de datos

Posteriormente, una vez obtenidos los datos y validarlos, se crean y asigna la información en los campos que se mostraran en la pantalla, es fundamental que esta información no pueda ser modificada por el usuario por obvios motivos de seguridad, por lo cual también se han bloqueado los campos para solo visualizar, si se requiere hacer una modificación el usuario debe retornar a la pantalla “Pago de factura”.

```

205 descripcionFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
206 cuentaRetiroFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
207 divisaFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
208 titularFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
209 bancoBeneficiarioFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
210 fechaFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
211
212 importeFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
213 cuentaDepositotFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
214 divisaProvFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
215 titularProvFld.updateDisplayType({ displayType: serverWidget.FieldDisplayType.INLINE });
216

```

Ilustración 63: Fragmento de código: Campos bloqueados

```
100
101
102     var form = serverWidget.createForm({
103         title: 'Pago a Proveedor',
104     });
105
106     var noteFld = form.addField({
107         id: 'custpage_impt_note',
108         type: serverWidget.FieldType.HELP,
109         label: 'Note: A continuación se muestran los detalles de pago.',
110     });
111
112     // Agrupación de campos
113     form.addFieldGroup({
114         id: 'custpage_grp_general',
115         label: 'Informacion de tranferencia'
116     });
117
118     form.addFieldGroup({
119         id: 'custpage_grp_vendorInfo',
120         label: 'Informacion del proveedor'
121     });
122
123     // Campos de información
124     var descripcionFld = form.addField({
125         id: 'custpage_nfo_descripcion',
126         type: serverWidget.FieldType.TEXT,
127         label: 'Descripcion',
128         container: 'custpage_grp_general'
129     });
130
131     var cuentaRetiroFld = form.addField({
132         id: 'custpage_nfo_cuentaRetiro',
133         type: serverWidget.FieldType.TEXT,
134         label: 'Cuenta de Retiro',
135         container: 'custpage_grp_general'
136     });
137
138     var divisaFld = form.addField({
139         id: 'custpage_nfo_divisa',
140         type: serverWidget.FieldType.TEXT,
141         label: 'Divisa de la cuenta',
142         container: 'custpage_grp_general'
143     });
```

Ilustración 66: Fragmentó de código: Creación de campos

La pantalla pago a proveedor se visualiza de la siguiente manera (Ilustración 29):

ActivitiesPagosTransactionsListsReportsAnalyticsDocumentsSetupCustomizationCommerceSuiteAppsSupport

Pago a Proveedor

Continuar13

Note: A continuación se muestran los detalles de pago.

Información de transferencia

Descripción

Pago Factura: N° 2025-0041

Cuenta de Retiro

2

Divisa de la cuenta

MXN3

Referencia

4

Titular de la cuenta

5

Banco Beneficiario

Bancomer6

Fecha de creación

08/04/20257

Información del proveedor

Importe

626.408

Cuenta de depósito

9

Clabe

10

Divisa de la cuenta

MXN11

Titular de la cuenta

12

Ilustración 69: Pantalla Pago a proveedor

Página 45 | 73

1. Descripción: Breve detalle del propósito del pago.
2. Cuenta de retiro: Número de la cuenta desde la que se realiza el retiro.
3. Divisa de la cuenta: Moneda de la cuenta de retiro.
4. Referencia: Información adicional opcional para identificar el pago.
5. Titular de la cuenta: Nombre del titular de la cuenta de retiro.
6. Banco beneficiario: Banco del proveedor o beneficiario del pago.
7. Fecha de creación: Fecha en la que se creó el movimiento de pago.
8. Importe: Monto total del pago.
9. Cuenta de depósito: Número de cuenta del proveedor o beneficiario.
10. Clabe: CLABE interbancaria de la cuenta de depósito.
11. Divisa de la cuenta (depósito): Moneda de la cuenta de depósito.
12. Titular de la cuenta (depósito): Nombre del titular de la cuenta de depósito.
13. Continuar: Botón de continuar

Si en la revisión del usuario todo es correcto al dar clic en “Continuar” se inicia el proceso de dispersión de pago, y se notifica al usuario el estatus. (Ilustración 30, 31, 32)

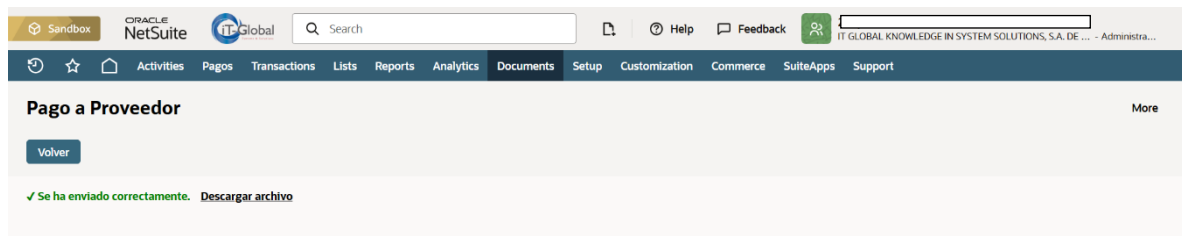


Ilustración 70: Notificación de Envío exitoso

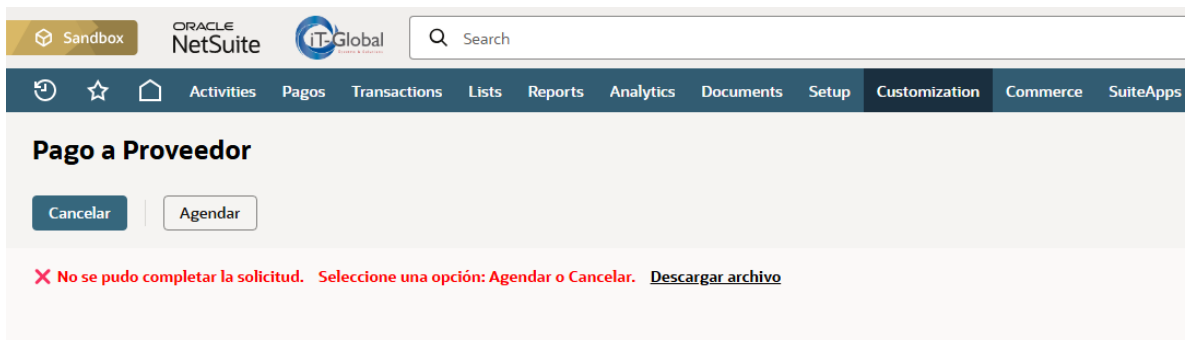


Ilustración 72: Notificación de error en el sistema

En caso de surgir un error interno el usuario tiene la opción de cancelar o agendar el pago para después, el sistema reintentara posteriormente él envío.

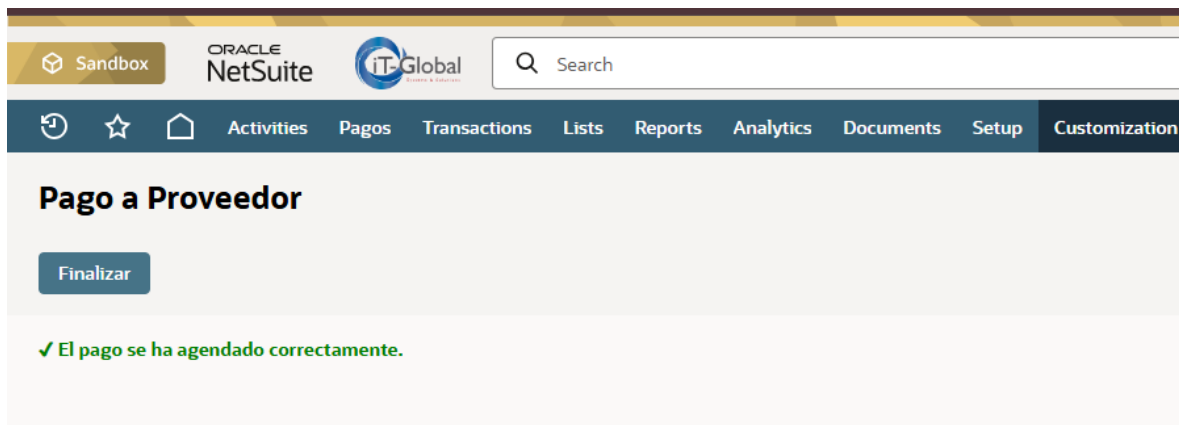


Ilustración 71: Notificación de pago agendado

3.2.3.3 Secure File Transfer Protocol

NetSuite cobra por el espacio de almacenamiento utilizado, y estos costos pueden ser elevados. Por ello, el cliente dispone de un servidor SFTP donde almacena sus documentos.

Internamente se genera el archivo de solicitud de transferencia donde se guarda internamente una copia temporal y la otra es puesta a disposición servidor SFTP para efectuar el pago. (Ilustración 33)

```
const txtFile = file.create({
  name: `${referencia}.txt`,
  fileType: file.Type.PLAINTEXT,
  contents: contenidoMT101,
  encoding: file.Encoding.UTF_8,
  folder: 1684
});

const savedFileId = txtFile.save();
const fileUrl = file.load({ id: savedFileId }).url;

const form = serverWidget.createForm({
  title: 'Pago a Proveedor',
});

form.addField({
  id: 'custpage_mensaje_envio',
  type: serverWidget.FieldType.INLINEHTML,
  label: ' ',
  defaultValue: '<div style="color: gray; font-weight: bold; margin-bottom: 10px;">
```

El archivo es creado y se almacena en la carpeta 1684

Ilustración 73: Fragmento de código: Creación de archivo y guardado

Netsuite asigna IDs internos a las carpetas creadas y documentos, de esta manera tiene un mayor control de la gestión, la carpeta antes mencionada 1684 tiene el nombre de MT101. (Ilustración 34)

SuiteScripts > Suite Dreams EGH Total: 14

Edit	Internal ID	Name	Size	Last Modified	Type	Download
Edit	1680	Bill Payment	11.61 KB	08/04/2025 1:22 PM	Folder	Download
Edit	1664	Eval	7.76 KB	22/03/2025 3:39 PM	Folder	Download
Edit	1684	MT101	274 B	09/04/2025 12:59 PM	Folder	Download
Edit	1661	restlet	8.7 KB	21/03/2025 10:17 AM	Folder	Download
Edit	1669	Suitelets Course	11.88 KB	31/03/2025 11:26 AM	Folder	Download

Ilustración 76: Gabinete de archivos

Dentro de la carpeta se encuentra nuestro archivo generado. (Ilustración 35)

Folder Contents

Search Folder Search SuiteScript 2.0 API SuiteScript 1.0 API

Show Inactives Add File Advanced Add

New Folder Copy Files Delete Files Move Files

Search

SuiteScripts > Suite Dreams EGH > MT101

Total: 1

Edit	Internal ID	Name	Size	Last Modified	Type	Download
Edit	19079	Nº 2025-004.txt	274 B	08/04/2025 12:59 PM	Plain Text File	Download

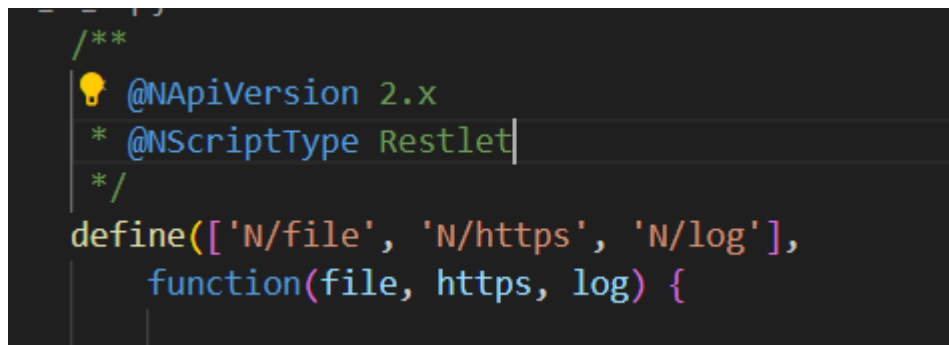
Ilustración 79: Carpeta MT101

El archivo contiene la información a detalle para el pago a proveedor, por motivos de confidencialidad no se puede mostrar el contenido.

Para el envío del archivo por SFTP es necesario incorporar los módulos N/file, N/https, N/log

1. N/file : Manejo archivos en el File Cabinet (crear, leer, actualizar, eliminar).
2. N/https : Realizar solicitudes HTTPS a servicios o APIS externas.
3. N/log : Registrar información y depurar scripts.

En conjunto estos módulos se encargan de cargar y enviar el archivo al servidor también de registrar la ejecución dentro del sistema. (Ilustración 36)



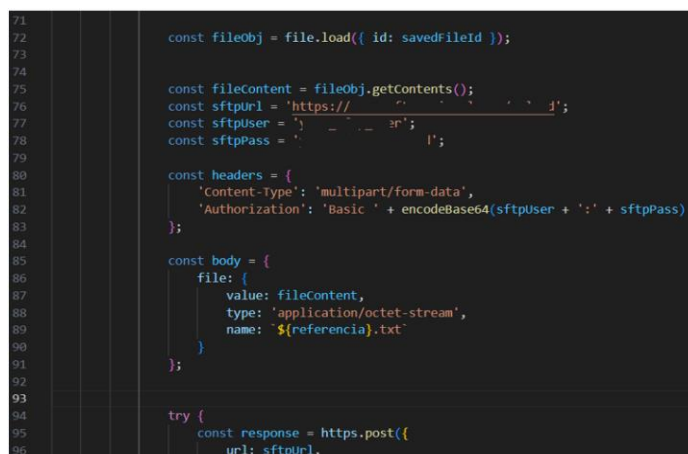
```

/**
 * @NApiVersion 2.x
 * @NScriptType Restlet
 */
define(['N/file', 'N/https', 'N/log'],
  function(file, https, log) {

```

Ilustración 80: Módulos empleados en el Envío a SFTP

En la ilustración 37, se presenta un fragmento de código de la conexión a SFTP, posterior a ser recibido el archivo es procesado y enviado a la respectiva API rest bancaria.



```

71
72
73     const fileObj = file.load({ id: savedFileId });
74
75     const fileContent = fileObj.getContents();
76     const sftpUrl = 'https://';
77     const sftpUser = ' ';
78     const sftpPass = ' ';
79
80     const headers = {
81       'Content-Type': 'multipart/form-data',
82       'Authorization': 'Basic ' + encodeBase64(sftpUser + ':' + sftpPass)
83     };
84
85     const body = {
86       file: {
87         value: fileContent,
88         type: 'application/octet-stream',
89         name: `${referencia}.txt`
90       }
91     };
92
93
94     try {
95       const response = https.post({
96         url: sftpUrl,

```

Ilustración 81: Fragmento de código conexión a SFTP

Como resultado de todo el proceso el archivo es entregado al servidor SFTP, donde se almacena y espera para ser procesado, Después de haber sido ejecutado el pago, se recibe un detalle completo del movimiento en el servidor y el retorna una notificación que se registra en el sistema

BBVA
BBVA Net Cash

Fecha y hora de consulta: 10/11/2012 10:11:52 Contador: 422
Nombre del Cliente:

BBVA Net Cash: Pago Interbancario

Operación autorizada

1 Transferecia en proceso de validación y aplicación

Datos del firmante

Usuario: USUARIO2 Poder: 0%

Datos de la operación

Tipo de operación: Grupo Pago Interbancario

Descripción: Importe: 0.00

Cuenta de retiro: Cuenta de depósito:

Divisa de la cuenta: MXN Divisa de la cuenta: MXN

Titular de la cuenta: Titular de la cuenta:

Banco beneficiario: SANTANDER Disponibilidad de pago: Mismo día

Fecha de creación: Fecha de aplicación:

Concepto de pago: Referencia:

Hora de captura en el canal: 10/11/2012

Datos de confirmación de la transferencia

Folio interbancario: Clave de rastreo:

Folio de firma: Folio único:

Estado operación

Porcentaje firmado: 100% Estado: Operado

1 PARA CONSULTAR EL ESTATUS DEL PAGO O TRANSFERENCIA, O SI DESEA OBTENER UN COMPROBANTE ELECTRONICO DE PAGO CON VALIDEZ OFICIAL VISITE: <http://www.bancomex.com/mex/pago>
Este vinculo se activará a más tardar dentro de los cinco minutos siguientes a la aceptación de la operación.

BBVA MEXICO, S.A. INSTITUCION DE BANCA MULTIPLE GRUPO FINANCIERO BBVA MEXICO

Internet
www.bbvanetcash.mx

Ilustración 84: Detalle de pago

3.2.3.4 Actualización de estado dentro de Netsuite

En las secciones anteriores, se ha guardado el registro de pago y enviado al servidor para su procesamiento. Una vez completado el pago, se recibe el detalle correspondiente, de esta forma se puede actualizar el estado del pago dentro del sistema de NetSuite.

La ilustración 39 muestra cómo se reciben los datos a través de un restlet. Además de consultar la base de datos para extraer el pago correspondiente al detalle y actualizar el estado de pago.

```

17 const post = (requestBody) => {
18   try {
19     function call() {
20     }
21     log.debug('Datos recibidos', requestBody);
22
23     const paymentRecord = search.create({
24       type: 'customrecord_pago',
25       filters: [
26         ['custrecord_referencia_pago', 'is', requestBody.referencia_pago]
27       ],
28       columns: ['internalid']
29     });
30     const result = paymentRecord.run().getRange({ start: 0, end: 1 });
31
32     if (result.length === 0) {
33       return {
34         success: false,
35         message: 'No se encontró un registro de pago con la referencia indicada.'
36       };
37     }
38     const paymentId = result[0].getValue('internalid');
39     log.debug('Registro encontrado, ID:', paymentId);
40
41     const paymentRecordObj = record.load({
42       type: 'customrecord_pago',
43       id: paymentId
44     });

```

Ilustración 85: Fragmento de código: Recibo y consulta de datos

Si el registro es encontrado en la búsqueda, se carga ese registro para poder modificarlo y actualizarlo. Ilustración (40)

```

    paymentRecordObj.setValue({
      fieldId: 'custrecord_codigo_autorizacion',
      value: requestBody.codigo_autorizacion
    });

    paymentRecordObj.setValue({
      fieldId: 'custrecord_banco',
      value: requestBody.banco
    });

    paymentRecordObj.setValue({
      fieldId: 'custrecord_num_rastreo',
      value: requestBody.num_rastreo
    });

    const updatedRecordId = paymentRecordObj.save();
    log.debug('Registro actualizado, ID:', updatedRecordId);

    return {
      success: true,
      message: 'El registro de pago fue actualizado correctamente.'
    };

```

Ilustración 86: Fragmento de código: actualización de campos

También es necesario mostrar el detalle de pago recibido en caso de que el usuario quiera consultarlo, para hacerlo se usó nuevamente un suitlet, ilustración(41)

```

const ref = form.addField({ id: 'custpage_referencia_pago', label: 'Referencia de Pago', type: ui.FieldType.TEXT });
ref.defaultValue = data.referencia_pago;
ref.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const monto = form.addField({ id: 'custpage_monto', label: 'Monto', type: ui.FieldType.CURRENCY });
monto.defaultValue = data.monto.toString();
monto.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const estatus = form.addField({ id: 'custpage_estatus_pago', label: 'Estatus', type: ui.FieldType.TEXT });
estatus.defaultValue = data.estatus_pago;
estatus.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const fecha = form.addField({ id: 'custpage_fecha_pago', label: 'Fecha de Pago', type: ui.FieldType.TEXT });
fecha.defaultValue = data.fecha_pago;
fecha.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const codigo = form.addField({ id: 'custpage_codigo_autorizacion', label: 'Código de Autorización', type: ui.FieldType.TEXT });
codigo.defaultValue = data.codigo_autorizacion;
codigo.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const banco = form.addField({ id: 'custpage_banco', label: 'Banco', type: ui.FieldType.TEXT });
banco.defaultValue = data.banco;
banco.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

const rastreo = form.addField({ id: 'custpage_num_rastreo', label: 'Número de Rastreo', type: ui.FieldType.TEXT });
rastreo.defaultValue = data.num_rastreo;
rastreo.updateDisplayType({ displaytype: ui.FieldDisplayType.INLINE });

form.addSubmitButton({ label: 'Back' });

```

Ilustración 87. Fragmento de código: suitlet detalle de pago

La forma de acceder a este registro es a través de la pantalla pago de factura (Bill payment), mostrada en un comienzo, la diferencia radica en que este acceso solo se permite si el registro es accedido solo como una visualización(view), de otro modo el enlace no se muestra, para lograrlo se desarrolló un User event para agregar el botón de enlace con la condición de que solo en modo de view.

```

const beforeLoad = (context) => {
  if (context.type !== context.UserEventType.VIEW) return;

  const form = context.form;
  const record = context.newRecord;

  form.addButton({
    id: 'custpage_boton_pago',
    label: 'Detalle',
    functionName: 'detalle_pago'
  });

  form.clientScriptModulePath = 'SuiteScripts/mis_scripts/cs_enviar_banco.js';
};

return { beforeLoad };
};

```

Ilustración 91: Fragmento de código Boton de detalle

Edit View	Internal ID	Date	Document Number ↑	Transaction Number
Edit View	387	27/02/2025	1	VENDPYMT8

Ilustración 90: Edición - visualización

Se puede acceder al registro en modo edición o solo visualizar, si se elige edición el botón creado no se mostrará. Ilustración (44)

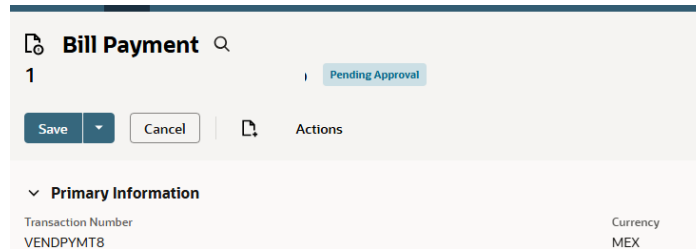


Ilustración 92: Modo edición

Ilustración 93: Modo visualizar Ilustración 94: Modo edición

Y en modo visualizar ilustración (45) se muestra el botón detalle:

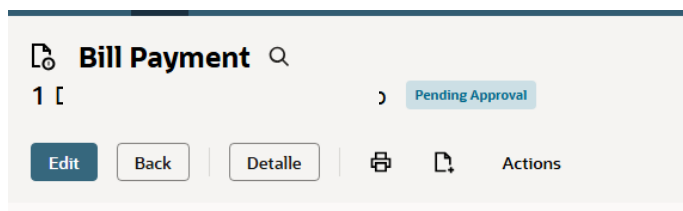


Ilustración 95: Modo visualizar

Si damos clic a Detalle, nos redirige a la pantalla de detalle donde se muestra la información de la transacción realizada ilustración (46)

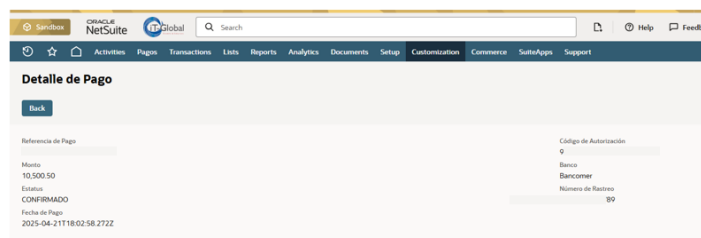


Ilustración 96: Pantalla Detalle de pago

3.2.3.5 Historial de pagos a proveedores

Para visualizar el historial de pagos a proveedores, se utiliza Saved Searches, una herramienta de NetSuite que permite acceder a los registros de la base de datos de manera eficiente. Su funcionamiento es similar al de las consultas SQL, pero con una interfaz más intuitiva y accesible, lo que facilita el desarrollo y análisis de información sin necesidad de implementar codificación, aunque si requiere conocimientos avanzados en bases de datos para la relación de tablas.

El proceso para hacer la Saved search fue el siguiente:

1. Seleccionar el tipo de búsqueda guardada, Netsuite lista algunos de los procesos de negocio algunos son estándar otros propios de la empresa, dependiendo el tipo de información que se desee consultar se hace una selección, Los pagos a proveedores entran en clasificación transacciones. (ilustración 47)

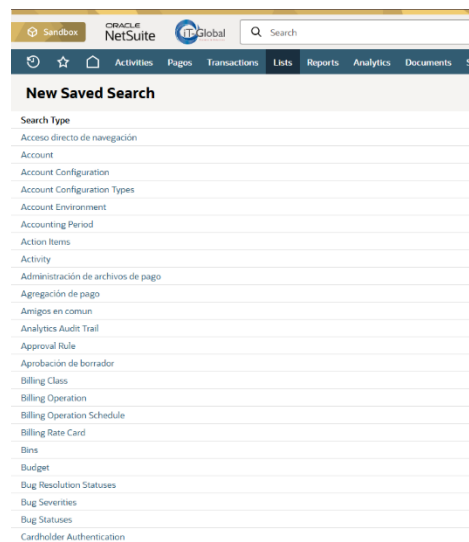


Ilustración 97: Saved search selección

2. Posterior a la selección se define el nombre de la búsqueda y el identificador que es necesario para el sistema. (ilustración 48)

Search Title *

Pago a Proveedor

ID

_sdr_pago_a_proveedor_sch_egh

☐ Public

☐ Available as List View

☐ Available as Dashboard View

☐ Available as Sublist View

☐ Available for Reminders

☐ Show In Menu

Ilustración 98: Saved Search Nombre

3. En la pestaña criteria, se ingresan todas las condiciones de búsqueda que se deben de cumplir, la selección es extensa es preciso conocer con exactitud lo que se quiere conocer, algunas veces la condición se encuentra dentro de otra condición y tiende a ser tardado encontrarla. (ilustración 49, 50)

Saved Transaction Search

SetCancel

Type

Transfer Bill

Bill

Bill Credit

Bill Payment

Bin Putaway Worksheet

Bin Transfer

CriteriaResultsHighlightingAvailable FiltersAudienceRolesEmailAudit TrailExecution Log

Transfer Order Quantity Packed

Transfer Order Quantity Picked

Transfer Order Quantity Received

Type

Type

OKCancelInsertRemove

+ Add Row

Ilustración 99: Saved Search condición

Ilustración 102: Saved Serach criteria

6. En la pestaña results, se seleccionan todos los campos que se mostraran, se define el orden y si se mostraran en grupo, se contara, máximos mínimos etc. Se muestra una comparativa de cómo sería una consulta SQL y su equivalencia en saved searches. (ilustración 52).

```

clienteordenes_ooo EXPIRATION:
clientequotas_ooo EXPIRATION:

--consulta para cotizaciones
cursor c_quotas (v_idcustomer IN NUMBER) is
select
    cu.name as nombre,
    c.modelo as modelokuto,
    c.versao as versaoikuto,
    c.prixe as preciokuto,
    q.deposit as deposito,
    q.quantity as cantidad,
    q.tot as impuesto
from KX001M_K0M_firrenciaia_quotas q
left join KX001M_K0M_firrenciaia_Cus c on c.id_cus = q.id_cus
left join KX001M_K0M_firrenciaia_customer cus on cus.id_customer = q.id_customer
where c.id_customer = v_idcustomer;

--consulta para ordenes de compra
cursor c_order (v_idcustomer IN NUMBER) is
select
    cu.name as nombre,
    c.modelo as modelokuto,
    c.versao as versaoikuto,
    c.prixe as preciokuto,
    q.deposit as deposito,
    q.quantity as cantidad,
    q.tot as impuesto,
    pur.line_number as subordenes,
    ship.delivery_address as delivery,
    ship.delivery_type as ship_type
from KX001M_K0M_firrenciaia_purchase_order pur
left join KX001M_K0M_firrenciaia_quotas q on q.id_customer = pur.id_customer
left join KX001M_K0M_firrenciaia_Cus c on c.id_cus = q.id_cus
left join KX001M_K0M_firrenciaia_customer cus on cus.id_customer = pur.id_customer
where pur.id_customer = v_idcustomer;

```

Ilustración 109: ejemplo de consulta SQL

Criteria
Results
Highlighting
Available Filters
Agencies
Rules
Email
Audit Trail
Execution Log
Search Title Translation

Use this tab to indicate columns to be included in the search results as well as sort order.

Sort By
Order Type
Then By
Then By

☐ Descending
☐ Descending
☐ Descending

Output Type
Normal
Consolidated Exchange Rate
Per-Account
Show Totals
Max Results
Run Unrestricted
My Preferred Search Results

Columns -
Drill Down Fields -

Remove all
Add Multiple

Field *	Summary Type	Function	Formula	When Ordered By Field	Custom Label
Date					
Document Number					
Transaction Number					
Name					
Account					
Status					

Ilustración 109: Saved Search results

Pago a Proveedor: Results

Return to Criteria

Edit this Search

Filters

Status

- All -

Name

Style

Normal

Edit

Total: 40

Edit View	Internal ID	Date	Transaction Number	Name	Account	Status	Memo	Currency	Amount	Banco de entidad de preferencia
Edit View	888	31/03/2025	VENDPYMT18	ó B...	100-1	Approved		MEX	-13.92	
Edit View	888	31/03/2025	VENDPYMT18	ó Bi de C...	Cue	Approved		MEX	-13.92	
Edit View	889	01/04/2025	VENDPYMT19	ó B de ...	Tl M 000000001	Approved		MEX	-870.00	
Edit View	889	01/04/2025	VENDPYMT19	ó Ba de C.V.	Cia	Approved		MEX	-870.00	
Edit View	2213	08/04/2025	VENDPYMT20			Approved	Pago Factura: N° 2025-004	MEX	-626.40	

Ilustración 105: Vista Pago a proveedores

7. Finalmente se visualiza la saved search, esta vista permite visualizar los pagos realizados y el estado en el que se encuentran, de esta manera se da seguimiento al proceso de dispersión de pago a proveedores. También se pueden filtrar los resultados, por proveedor y estado (pendiente: el pago se agendo, aprobado: pago realizado con éxito, rechazado: el pago fue rechazado por la entidad bancaria).

3.3. Envío de 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. A continuación, se detalla el proceso de envío:

- 1. Cargar archivos al File Cabinet, siguiendo la ruta Documentos > archivos > File Cabinet > SuiteScript, en esta carpeta el líder del proyecto crea una carpeta dedicada para cada desarrollador, donde se almacenarán los documentos JavaScript desarrollados. Ilustración (54).

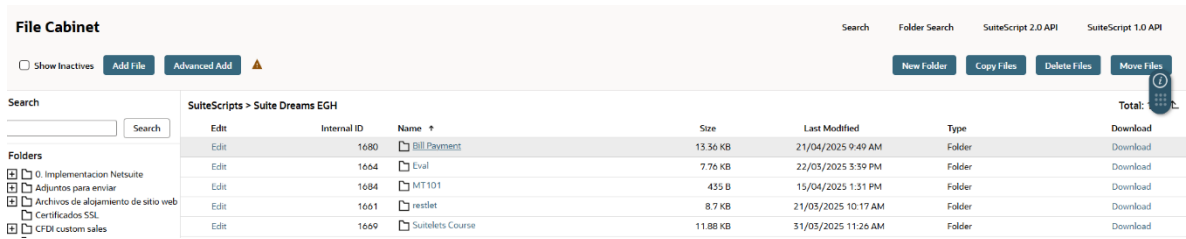


Ilustración 112:File Cabinet

- 2. clic en add file, se abre ventana emergente del explorador de archivos, seleccionas los archivos que se deseen subir.

SuiteScripts > Suite Dreams EGH > Bill Payment			
Edit	Internal ID	Name ↑	Size
Edit	19064	sdr_cs_validaciones.js	921 B
Edit	19183	sdr_lectura.js	4.02 KB
Edit	19042	sdr_sl_bill_info.js	7.94 KB
Edit	19066	sdr_us_envio.js	520 B

Ilustración 113: archivos subidos

3. Es importante recordar el nombre de cada archivo, ahora en la ruta: personalización > scripting > nuevo script, por cada archivo subido se debe de crear su respectivo registro en el sistema. Ilustración (56), selecciona el script y después clic en crear script.

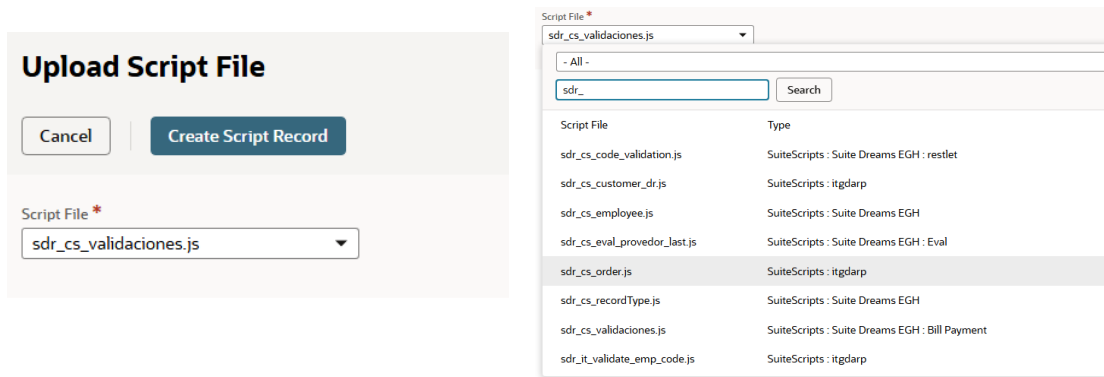


Ilustración 114: selección de script

4. Ahora muestra 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.

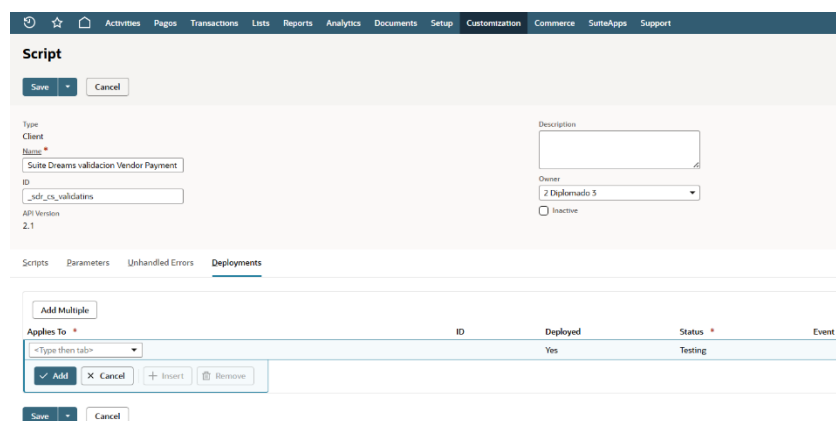


Ilustración 115: Registro del script

5. Cada usuario en Netsuite tiene por defecto un supervisor, que vigila su actividad dentro del sistema, cada acción es registrada y notificada mediante correos por Netsuite. 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 script, el desarrollador tiene tres días hábiles para modificar lo necesario para su funcionamiento, si en ese lapso no logro hacerlo, simplemente puede eliminar el registro y se cancela el proceso.
6. 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.

The screenshot displays the 'Script Deployment' interface in Netsuite. The top section shows script details: Script Name 'SuiteDreams detalle de pago egh', Title 'SuiteDreams detalle de pago egh', ID 'customdeploy_sdr_detalle_pago_2', and a 'Deployed' status. The right side shows configuration options like 'Status' (Testing), 'Event Type', 'Log Level' (Debug), 'Execute as Role' (Current Role), and 'Available Without Login' (unchecked). The URL is '/app/site/hosting/scriptlet.nl?script=9806&deploy=1'.

The bottom section, 'Execution Log', shows a table of script executions. The table has columns for #, View, Type, Title, Date, Time, User, and Details. One execution is listed with a 'TypeError' on 21/04/2025 at 9:52 AM, performed by user '2 Diplomado 3'. The error details are 'TypeError: form.getFields is not a function [at Object.onRequest (/SuiteScripts/Suite Dreams EGH/Bill Payment/sdr_lectura.js:36:14)]'.

Ilustración 116: Suitelet script registrado en el sistema

Independientemente de que tipo de script sea, el sistema guarda las ejecuciones y notas de sistema, durante el desarrollo se usaron manejadores de errores try catch, para prevenir que el sistema se detenga inesperadamente, además de que se pueda conocer los errores y solucionarlos, estos manejadores notifican al sistema y se registra en la pestaña execution log. En notas del sistema, se registra a detalle cada cambio realizado en el registro, esto es perfecto para la auditoria.

Audience - Links Execution Log - System Notes						
View Field						
Default - All						
Customise View						
Date	Set by	Context	Type	Field	Old Value	New Value
21/04/2025 9:52 AM		UI	Set	Deployed		T
21/04/2025 9:52 AM		UI	Set	Available Without Login		F
21/04/2025 9:52 AM		UI	Set	Mobile Menu		F
21/04/2025 9:52 AM		UI	Set	ID		CUSTOMDEPLOY_SDR_DETALLE_PAGO_2
21/04/2025 9:52 AM		UI	Set	Log Level		Debug
21/04/2025 9:52 AM		UI	Set	Title		SuizaDreams detalle de pago igh
21/04/2025 9:52 AM	by: longines@nabors.es	UI	Set	Status		Testing

Ilustración 117: System notes

3.4 Retroalimentación

Después de concluir el proceso de pruebas realizado por el área de testeo. Se recibieron como resultado, observaciones orientadas a asegurar la correcta operación del sistema. El error detectado fue un problema relacionado con el proceso de aprobación, el cual falla cuando el idioma de la cuenta o del usuario es diferente al idioma configurado inicialmente.

El mensaje recibido fue el siguiente.

Cuando el idioma de la cuenta o del usuario en NetSuite se cambia de Español a Inglés (u otro idioma diferente al configurado inicialmente), la funcionalidad de aprobación de pagos deja de funcionar correctamente. El botón o la acción de "Aprobar" no se activa o genera un error, impidiendo continuar con el flujo de aprobación. (Área de testing)

Al comprobarlo efectivamente no permite continuar el proceso de pago, el mensaje mostrado se muestra en la ilustración 60.

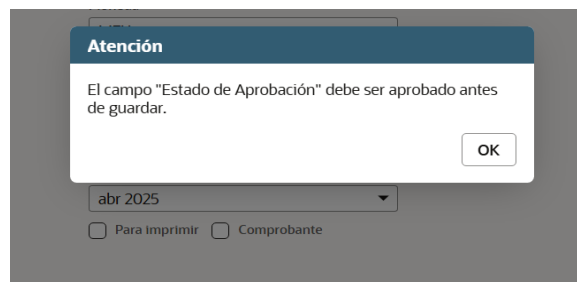


Ilustración 120: Fallo de aprobación

Después de analizar el problema se encontró que efectivamente había un problema en la codificación, la sentencia if evalúa que el parámetro recibido sea “Aprobado” por lo cual al cambiar de idioma nunca va a permitir continuar.

```

115
116     var currentRecord = context.currentRecord;
117     var approvalStatus = currentRecord.getValue({ fieldId: 'approvalstatus' });
118
119     if (approvalStatus !== 'Aprobado') {
120         dialog.alert({
121             title: 'Validación Requerida',
122             message: 'El campo "Estado de Aprobación" debe ser aprobado antes de guardar.'
123         });
124         .then(() => {

```

Ilustración 123: Error de aprobación

Para poder solucionarlo es necesario cambiar la codificación para corregir el error, así fue como el código cambio al mostrado en la ilustración 62. Los cambios realizados incluyeron agregar el módulo N/runtime para determinar el idioma de la sesión y así cambiar el idioma de la advertencia, sobre la validación se consultó la documentación de Netsuite para conocer el id del campo dentro de la lista usada “approvalstatus”, ese valor nunca cambia. Otra consideración fue usar la función `getValue`, esta función cuando se trata de listas extrae el id y en campos extrae la cadena de texto.

```
define(['N/runtime', 'N/log', 'N/ui/dialog'], function(runtime, log, dialog) {
```

```
var currentRecord = context.currentRecord;
var approvalStatus = currentRecord.getValue({ fieldId: 'approvalstatus' });

var STATUS_APPROVED = '2';

if (approvalStatus !== STATUS_APPROVED) {
  var userLanguage = runtime.getCurrentUser().getPreference({ name: 'language' });

  var messages = {
    'es_ES': {
      title: 'Validación Requerida',
      message: 'El campo "Estado de Aprobación" debe ser aprobado antes de guardar.'
    },
    'en_US': {
      title: 'Validation Required',
      message: 'The "Approval Status" field must be approved before saving.'
    }
  };
  var selectedMessage = messages[userLanguage] || messages['en_US'];
```

Ilustración 124: Cambios realizados en código de aprobación

Posterior a la corrección, nuevamente se envía a el área de testing. Para hacerlo basta con solo actualizar el documento dentro del registro de script.



Ilustración 125: Actualizar script

Ilustración 126: Notificación de correo a testing Ilustración 127: Actualizar script

Finalmente, se envía el correo al área para que lo validen

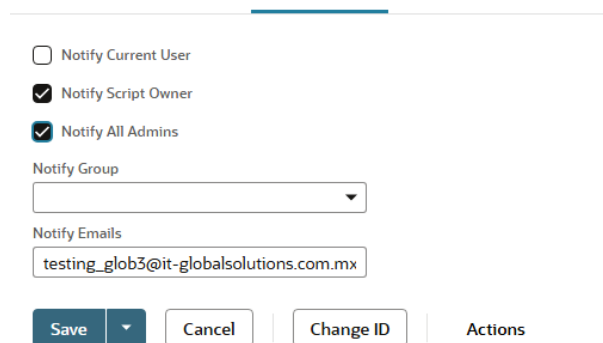


Ilustración 128: Notificación de correo a testing

Ilustración 129: Descripción básica de Bundle Ilustración 130: Notificación de correo a testing

3.5. Migración al entorno de producción

Una vez aprobada la funcionalidad del desarrollo, el área de testing permite dar continuidad a la siguiente y última etapa, migrar la personalización al entorno de producción, recordemos que todas las pruebas y validaciones fueron realizadas en una cuenta sandbox, espacio asilado con el objetivo de no afectar la producción de la empresa. A continuación, se detalla como es el proceso de migración.

3.5.1. Creación de bundle

Un bundle es una forma de empaquetar y distribuir personalizaciones, scripts, y configuraciones para ser instalados en una cuenta de NetSuite. Permite añadir o modificar funcionalidades de NetSuite, y pueden ser creados por los usuarios o distribuidos por Oracle NetSuite.

Para la creación del bundle se siguieron los siguientes pasos:

Ilustración 131: Descripción básica de Bundle

Ilustración 132: Selección de objetos del bundle
Ilustración 133: Descripción básica de Bundle

1. Se define un nombre de bundle, e información básica como lo es la descripción, versión. Ilustración 65

2. Selección de archivos que forman parte del empaquetado (bundle), para incluir un componente, solo es necesario expandir el tipo correspondiente, localizar el elemento deseado y marcarlo para que forme parte del paquete.

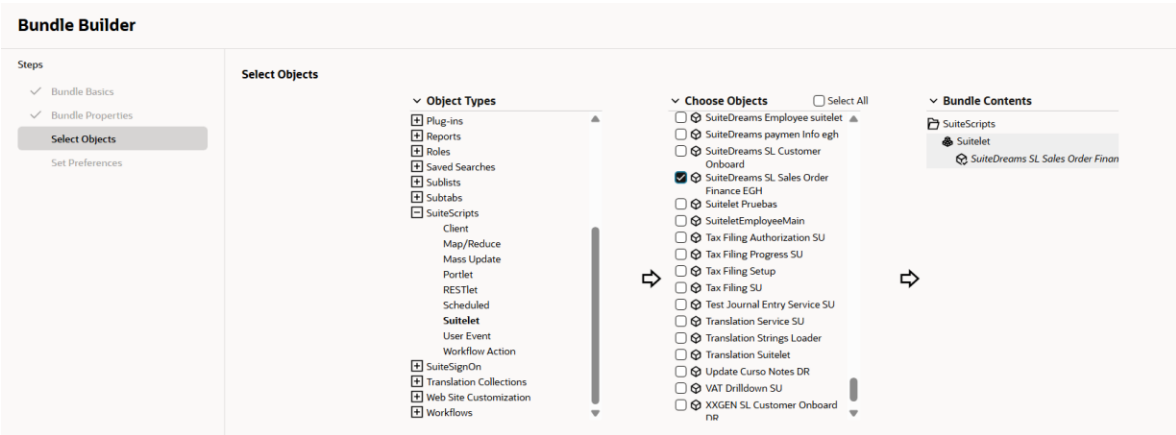


Ilustración 134: Selección de objetos del bundle

Ilustración 135: Listado de objetos de instalaciónIlustración 136: Selección de objetos del bundle

3. Finalmente se revisa el contenido del bundle y también, se bloquea para que solo el

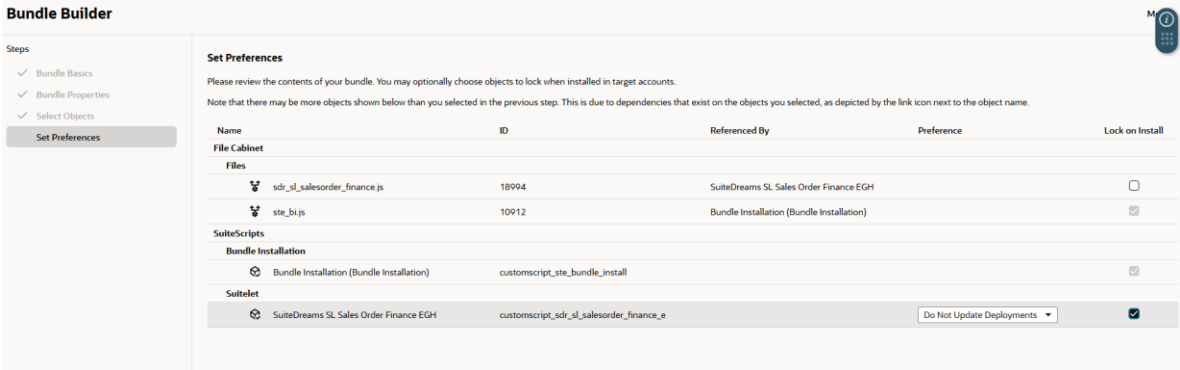


Ilustración 137: Listado de objetos de instalación

Ilustración 138: Guardar el bundleIlustración 139: Listado de objetos de instalación

propietario pueda modificar, es una medida de seguridad que garantiza el correcto funcionamiento

4. Guardar el bundle.

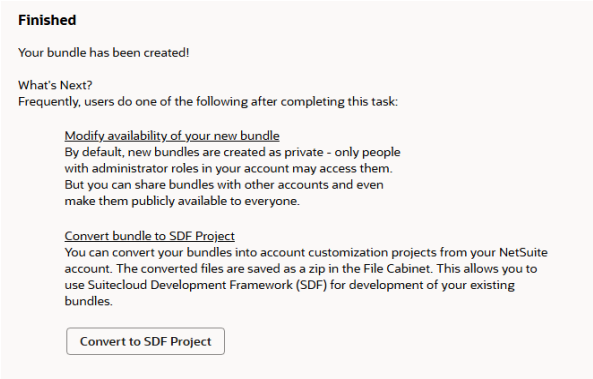


Ilustración 140: Guardar el bundle

3.5.2 Instalación

Para instalar un bundle, solo se requiere estar logeado a una cuenta Netsuite del tipo productiva

En la sección de Bundles & install, se busca el bundle requerido, la ventana es la siguiente:

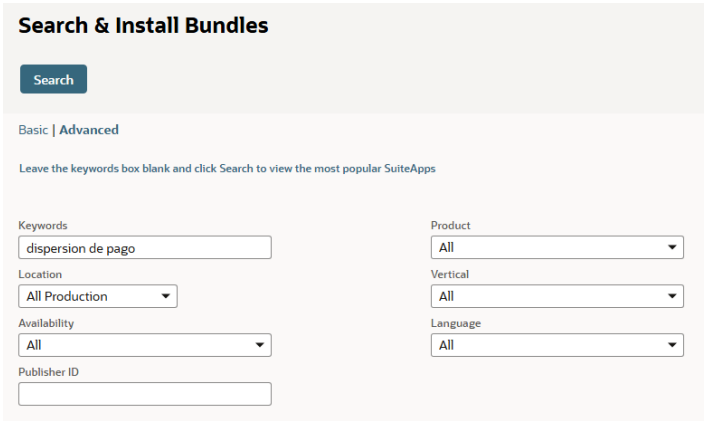
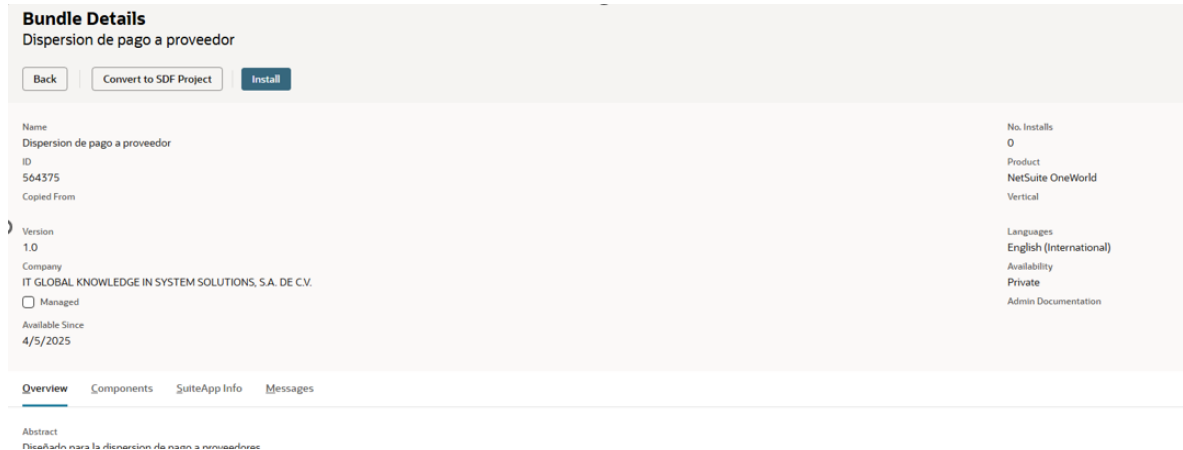


Ilustración 143: busqueda de bundle

El resultado de la búsqueda muestra el bundle creado

Action	Name ↑	Bundle ID	Version	App ID	Abstract	Admin Document
	Dispersion de pago a proveedor	564375	1.0		Diseñado para la dispersion de pago a proveedores	

Abrimos, para revisar los detalles, y encontrar el botón de instalar, basta con dar clic y el proceso se ejecutará en automático



3.5.3 ¿Porque hacer Bundles?

Aunque si es posible solo migrar los scripts desarrollados al entorno productivo, no es la mejor opción, debido a que dentro del sistema cada registro tiene un id en específico y al migrarlos, dichos ids pueden ser duplicados, esto afectaría a todo el sistema, por lo consiguiente la mejor practica es crear bundles, esta solución renombra los ids al ser instalados en una nueva cuenta, además de que la instalación no se limita a una sola cuenta.

4. Conclusión

La personalización del proceso de dispersión de pagos en NetSuite ha permitido ajustar la plataforma a los requerimientos del cliente, asegurando que el sistema contemple sus flujos de trabajo y cumpla con las exigencias dentro del ciclo financiero. Este desarrollo facilita la operación diaria del cliente, agiliza tareas y ayuda a tener un mejor control sobre la información financiera, haciendo que cada etapa del proceso sea más clara y ordenada.

Automatizar la dispersión de pago representa un avance importante para la empresa ya que permite el envío de múltiples pagos a proveedores mediante la generación de órdenes de pago en archivos de formato TXT compatibles con los sistemas bancarios. La implementación de esta solución con SuiteScript simplifica el manejo de pagos, eliminando tareas manuales repetitivas y reduciendo errores humanos.

Además, integrar esta funcionalidad dentro de NetSuite hace que las operaciones financieras se lleven a cabo de manera más organizada y alineada con los requerimientos del negocio. Con este desarrollo, la empresa logra mayor precisión en sus procesos, minimiza riesgos y mejora la administración de sus recursos.

Por último, es importante esclarecer que los objetivos planteados en un principio se cumplieron:

1. Desarrollar soluciones personalizadas (bundles) según los requerimientos del cliente.
2. Enviar los bundles para ser probados y validados.
3. Recibir retroalimentación de pruebas unitarias y hacer correcciones en caso de ser necesario.
4. Migrar los bundles al entorno de producción.

4.1 Experiencia personal profesional adquirida

Cada etapa del proceso significó un reto personal y profesional, entender el flujo de trabajo de la plataforma Netsuite y como se relaciona con los procesos externos de la empresa fue

primordial para así poder conocer las áreas de mejora del ERP y que están permitidas por el entorno. Aprendí a utilizar nuevas herramientas tecnológicas proporcionadas por la plataforma SuiteCloud, especialmente enfocadas en el desarrollo con SuiteScript.

Desde el punto de vista profesional, este proyecto representó una experiencia de gran valor en mi carrera como futuro ingeniero en Tecnologías de la Información y Comunicaciones. El desarrollo de la funcionalidad de dispersión de pagos fue un desafío técnico que requirió un arduo esfuerzo al aplicar conocimientos adquiridos en la formación académica y los adquiridos en la marcha, responder a los tiempos de entrega y asegurar el correcto funcionamiento y comprender la terminología técnica utilizada dentro de la empresa y fortalecer el trabajo en equipo, en general considero que la experiencia laboral adquirida fue muy enriquecedora.

5. Competencias Desarrolladas y Aplicadas

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

5.1. Competencias específicas:

- **Programación II:** Aplica tecnologías y herramientas actuales y emergentes de programación para desarrollar sistemas de información que ofrezcan soluciones a problemas del entorno.
- **Programación WEB:** Desarrolla aplicaciones web dinámicas del lado cliente y del servidor, considerando la conectividad a orígenes de datos, la interconectividad entre aplicaciones y cómputo en la nube.
- **Fundamentos de Bases de Datos:** Analiza requerimientos y diseña bases de datos para generar soluciones al tratamiento de información basándose en modelos y estándares.
- **Interacción Humano Computadora:** Crea interfaces humano computadora que faciliten el acceso y la interacción con las aplicaciones de cómputo.
- **Taller de Ingeniería de Software:** Aplica métodos y herramientas de la ingeniería del software en el desarrollo de software aplicando estándares de calidad y productividad.
- **Herramientas Emergentes de Desarrollo de Software:** Identifica, evalúa, y aplica tecnologías y herramientas actuales y emergentes en el diseño y desarrollo de software para atender necesidades del entorno

5.2. Competencias genéricas:

- Capacidad de abstracción, análisis y síntesis
- Capacidad para identificar, plantear y resolver problemas
- Capacidad de aplicar los conocimientos en la práctica
- Capacidad de generar nuevas ideas.
- Trabajo en equipo
- Capacidad de comunicación oral y escrita.

6. Fuentes de información

arsys. (2021). SFTP: Qué es y cómo se utiliza: <https://www.arsys.es/blog/sftp>

BBVA. (2022). SWIFT: <https://www.bbva.com/es/salud-financiera/swift-el-sistema-que-facilita-el-movimiento-de-capitales-entre-paises/>

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

Help Center. (2025). SuiteScript: https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/set_1502135122.html

it-globalsolutions. (2025). <https://www.it-globalsolutions.com/inicio>

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

NetSuite Applications Suite. (2025). SuiteBundler Overview:
https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/chapter_N3363483.html

Oracle. (2023). ¿Qué es la ERP?: <https://www.oracle.com/mx/erp/what-is-erp/>

Oracle. (2023). <https://doi.org/https://www.oracle.com/mx/applications/what-is-saas/#link6>

Oracle Help Center. (2025). Búsquedas guardadas:
https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/chapter_N675442.html

Oracle Help Center. (2025). Sandbox de NetSuite:
https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/chapter_N333400.html

Oracle Mylearn. (2025). SuiteScript Foundations:
<https://mylearn.oracle.com/netsuite/learning-path/suitescript-foundations/138080>

Oracle Mylearn. (2025). SuiteScript Development:

<https://mylearn.oracle.com/netsuite/learning-path/suitescript-development/125743>

ORACLE Netsuite. (2021). Plataforma NetSuite SuiteCloud:

<https://www.netsuite.com/portal/platform.shtml>

Oracle NetSuite. (2023). NetSuite Enterprise Resource Planning (ERP):

<https://www.netsuite.com.mx/portal/mx/products/erp.shtml>

Oracle sites. (2016). De la adquisición al pago: <https://sites.oracle.com/site/NES-FF-es/procure-to-pay.html>

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