



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Sistemas**

**Implementación de un módulo personalizado en el  
ERP Oracle NetSuite para automatizar la gestión de  
contratos de alquiler en una empresa española de  
bienes raíces**

**TRABAJO DE SUFICIENCIA PROFESIONAL**

**Para optar el Título Profesional de Ingeniero de Sistemas**

**AUTOR**

**Vicente Omar MORI ROJAS**

**ASESOR**

**Mg. Joel Fernando MACHADO VICENTE**

**Lima, Perú**

**2026**



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Mori, V. (2026). *Implementación de un módulo personalizado en el ERP Oracle NetSuite para automatizar la gestión de contratos de alquiler en una empresa española de bienes raíces*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

---

### Metadatos complementarios

|                                          |                                                                                           |
|------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>Datos de autor</b>                    |                                                                                           |
| Nombres y apellidos                      | Vicente Omar Mori Rojas                                                                   |
| Tipo de documento de identidad           | DNI                                                                                       |
| Número de documento de identidad         | 71238397                                                                                  |
| URL de ORCID                             | <a href="https://orcid.org/0000-0003-1302-5592">https://orcid.org/0000-0003-1302-5592</a> |
| <b>Datos de asesor</b>                   |                                                                                           |
| Nombres y apellidos                      | Joel Fernando Machado Vicente                                                             |
| Tipo de documento de identidad           | DNI                                                                                       |
| Número de documento de identidad         | 40476778                                                                                  |
| URL de ORCID                             | <a href="https://orcid.org/0000-0002-4056-0438">https://orcid.org/0000-0002-4056-0438</a> |
| <b>Datos del jurado</b>                  |                                                                                           |
| <b>Presidente del jurado</b>             |                                                                                           |
| Nombres y apellidos                      | Arturo Alejandro Bartra More                                                              |
| Tipo de documento                        | DNI                                                                                       |
| Número de documento de identidad         | 40233946                                                                                  |
| <b>Miembro del jurado 1</b>              |                                                                                           |
| Nombres y apellidos                      | William Martin Enríquez Maguiña                                                           |
| Tipo de documento                        | DNI                                                                                       |
| Número de documento de identidad         | 06179457                                                                                  |
| <b>Datos de investigación</b>            |                                                                                           |
| Línea de investigación                   | No aplica                                                                                 |
| Grupo de investigación                   | No aplica                                                                                 |
| Agencia de financiamiento                | Sin financiamiento                                                                        |
| Ubicación geográfica de la investigación | País: Perú<br>Departamento: Lima<br>Provincia: Lima                                       |

|                                                        |                                                                                                                                                           |
|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                        | Distrito: Cercado de Lima<br>Jr. Carlos Amezaga No. 375<br>Universidad Nacional Mayor de San Marcos<br>Latitud: -12.0564232<br>Longitud: -77.0843327      |
| Año o rango de años en que se realizó la investigación | 2025                                                                                                                                                      |
| URL de disciplinas OCDE                                | 2.02.04 -- Ingeniería de sistemas y comunicaciones<br><a href="https://purl.org/pe-repo/ocde/ford#2.02.04">https://purl.org/pe-repo/ocde/ford#2.02.04</a> |



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**Escuela Profesional de Ingeniería de Sistemas**

**Acta Virtual de Sustentación**  
**del Trabajo de Suficiencia Profesional**

Siendo las 14:00 pm. horas del día 01 de abril del año 2026, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Bartra More Arturo Alejandro (Presidente), Mg. Enríquez Maguiña William Martin (Miembro) y el Mg. Machado Vicente Joel Fernando (Miembro Asesor), usando la plataforma Meet (<http://meet.google.com/hot-tmmg-afm>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE UN MÓDULO PERSONALIZADO EN EL ERP ORACLE NETSUITE PARA AUTOMATIZAR LA GESTIÓN DE CONTRATOS DE ALQUILER EN UNA EMPRESA ESPAÑOLA DE BIENES RAÍCES”**, por el Bachiller **Mori Rojas Vicente Omar**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 (DIECIOCHO)**.

A continuación el Presidente de Jurados el Mg. Bartra More Arturo Alejandro, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 14:55 horas, se levantó la sesión.

**Presidente**

Mg. Bartra More Arturo Alejandro

**Miembro**

Mg. Enríquez Maguiña William Martin

**Miembro Asesor**

Mg. Machado Vicente Joel Fernando



UNIVERSIDAD NACIONAL MAYOR DE  
**SAN MARCOS**  
Universidad del Perú, Decana de América

## CERTIFICADO DE SIMILITUD

Yo, Joel Fernando Machado Vicente en mi condición de asesor acreditado mediante Resolución Decanal N° 001268-2025-D-FISI/UNMSM y OFICIO N° 0022-PTTSP-FISI-2025 del Trabajo de Suficiencia Profesional, cuyo título es Implementación de un módulo personalizado en el ERP Oracle NetSuite para automatizar la gestión de contratos de alquiler en una empresa española de bienes raíces, presentado por el bachiller Mori Rojas Vicente Omar para optar el Título Profesional de Ingeniero de Sistemas, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 7% de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del **TITULO PROFESIONAL DE INGENIERO DE SISTEMAS**.

Lima, 4 de junio de 2026

Firma del Asesor:

Nombres y Apellidos del asesor: Joel Fernando Machado Vicente

DNI: 40476778

## **FICHA CATALOGRÁFICA**

IMPLEMENTACIÓN DE UN MÓDULO PERSONALIZADO EN EL ERP  
ORACLE NETSUITE PARA AUTOMATIZAR LA GESTIÓN DE  
CONTRATOS DE ALQUILER EN UNA EMPRESA ESPAÑOLA DE  
BIENES RAÍCES

AUTOR: MORI ROJAS, VICENTE OMAR

ASESOR: MACHADO VICENTE, JOEL FERNANDO

LIMA – PERÚ, 2026

TÍTULO PROFESIONAL: Título Profesional de Ingeniero de Sistemas

ÁREA / PROGRAMA / LÍNEA DE INVESTIGACIÓN:

Ingenierías / Tecnología de Información y Comunicación / Ingeniería de  
Software

PREGRADO: Universidad Nacional Mayor de San Marcos - Facultad de  
Ingeniería de Sistemas e Informática - Escuela Profesional de Ingeniería de  
Sistemas

Formato 21 x 29.7 cm

Páginas: xi, 81

## **DEDICATORIA**

Dedico este trabajo principalmente a mis padres, quienes se esforzaron y nunca dudaron por brindarnos, a mí y a mi hermano, la oportunidad de acceder a una educación que no tuvieron. Gracias a su confianza y amor hoy cumplimos el logro de titularme en mi carrera profesional.

A mi hermano, por acompañarme en todos mis logros, impulsar y recordarme el buen ejemplo que debo ser, espero que este logro te ayude y motive para que te conviertas en un mejor profesional.

A mi novia por su incondicional apoyo y amor, el cual me impulsa a ser un mejor profesional y persona.

Se lo dedico también a toda mi familia por enseñarme lo importante que es la unión familiar, lo cual me es de gran ayuda ahora que nos encontramos lejos.

## **AGRADECIMIENTOS**

A mis padres y hermano por su constante apoyo y motivación.

A mi asesor por su orientación y apoyo en todo el desarrollo de este trabajo de suficiencia profesional.

A mi querida Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática (FISI) por brindarme la educación y experiencias que me convirtieron en un profesional de éxito.

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**  
**Implementación de un módulo personalizado en el ERP Oracle NetSuite**  
**para automatizar la gestión de contratos de alquiler en una empresa**  
**española de bienes raíces**

**Autor:** Mori Rojas, Vicente Omar  
**Asesor:** Machado Vicente, Joel Fernando  
**Título:** Trabajo de Suficiencia Profesional para optar el título de Ingeniero de Sistemas  
**Fecha:** Enero 2026

---

**RESUMEN**

La implementación de los procesos de una empresa en el ERP Oracle NetSuite no siempre es posible en su totalidad con los módulos estándar debido a las particularidades de cada empresa y de la región en que operan, por lo cual se debe recurrir a desarrollar personalizaciones dentro del ERP para poder satisfacer las necesidades de los clientes. Los contratos de alquiler en España se conforman por una serie de cobros mensuales como la fianza, renta, garantías y gastos, que adicionalmente pueden cambiar por actualizaciones del IPC (Índice de Precios de Consumo) o renovaciones en el contrato. Lo que obliga a generar facturas con estas particularidades. El reto principal consiste en que el módulo estándar “Advance Billing” no ofrece la flexibilidad para incluir estos escenarios y frecuencias, esto obliga a que se evalúen la adquisición de planes adicionales, que no solo implica un mayor costo para el cliente, sino también un mayor trabajo para la configuración del entorno. El presente trabajo tiene como objetivo la implementación de un módulo personalizado para automatizar la facturación de los contratos de alquiler, permitiendo la visualización del pronóstico de facturación y actualización del contrato en NetSuite representado como una Orden de Venta para generar correctamente las facturas.

Palabras Claves: ERP, NetSuite, Personalización, Pronóstico de facturación, Contrato de alquiler

**MAJOR NATIONAL UNIVERSITY OF SAN MARCOS**  
**FACULTY OF SYSTEMS AND INFORMATIC ENGINEERING**  
**PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING**

**Implementation of a customized module in Oracle NetSuite ERP to automate the management of rental contracts in a Spanish real estate company**

**Author:** Mori Rojas, Vicente Omar

**Advisor:** Machado Vicente, Joel Fernando

**Title:** Professional Sufficiency Work to opt for the Professional Title of Systems Engineer

**Date:** January 2026

---

**ABSTRACT**

Implementation of company processes in Oracle NetSuite ERP is not always possible by just using standard modules due to the particularities of each company and the region in which they operate, this demands to develop customizations within the ERP to meet customer needs. Rental contracts in Spain consist of a series of monthly payments such as the deposit, rent, guarantees, and expenses, which may also change due to CPI (Consumer Price Index) and updates or contract renewals. This requires invoices to be generated with these specific scenarios. The main challenge here is that the standard Advance Billing module does not offer the flexibility to include these scenarios and frequencies. This forces to evaluate the acquisition of additional plans, which not only implies a higher cost for the customer, but also more work for the configuration of the environment. This professional sufficiency work focuses on the implementation of a customized module to automate the billing of rental contracts. This will allow the billing forecast visibility and contract updates, represented in Sales Order in NetSuite, in order to generate invoices correctly.

Key words: ERP, NetSuite, ERP customization, Billing Forecast, Rental contract

## TABLA DE CONTENIDO

|                                                                   |     |
|-------------------------------------------------------------------|-----|
| DEDICATORIA .....                                                 | iii |
| AGRADECIMIENTOS .....                                             | iv  |
| ÍNDICE DE TABLAS .....                                            | ix  |
| ÍNDICE DE FIGURAS.....                                            | x   |
| INTRODUCCIÓN .....                                                | 1   |
| CAPÍTULO I - TRAYECTORIA PROFESIONAL .....                        | 3   |
| 1.1    Presentación profesional .....                             | 3   |
| 1.2    Educación y certificaciones.....                           | 3   |
| CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA..   | 7   |
| 2.1    Empresa - Actividad que realiza .....                      | 7   |
| 2.2    Misión de la empresa .....                                 | 7   |
| 2.3    Visión de la empresa .....                                 | 8   |
| 2.4    Organización de la empresa .....                           | 8   |
| 2.5    Área, cargo y funciones desempeñadas.....                  | 10  |
| 2.6    Experiencia profesional realizada en la organización ..... | 10  |
| CAPÍTULO III – ACTIVIDADES DESARROLLADAS .....                    | 13  |
| 3.1    Situación problemática.....                                | 13  |
| 3.1.1    Definición del problema.....                             | 14  |
| 3.2    Solución .....                                             | 14  |
| 3.2.1    Objetivos .....                                          | 14  |
| 3.2.1.1    Objetivo General. ....                                 | 14  |
| 3.2.1.2    Objetivos específicos. ....                            | 15  |
| 3.2.2    Alcance .....                                            | 15  |
| 3.2.3    Etapas y metodología .....                               | 15  |
| 3.2.4    Fundamentos utilizados.....                              | 18  |
| 3.2.4.1    Contrato de alquiler.....                              | 18  |

|                                                         |                                                                             |    |
|---------------------------------------------------------|-----------------------------------------------------------------------------|----|
| 3.2.4.2                                                 | Fianza.....                                                                 | 18 |
| 3.2.4.3                                                 | Renta .....                                                                 | 18 |
| 3.2.4.4                                                 | ERP (Enterprise Resource Planning). ....                                    | 19 |
| 3.2.4.5                                                 | Oracle NetSuite.....                                                        | 21 |
| 3.2.4.6                                                 | NetSuite SuiteCloud Platform.....                                           | 22 |
| 3.2.4.7                                                 | SuiteScript.....                                                            | 24 |
| 3.2.4.8                                                 | User Event Script. ....                                                     | 27 |
| 3.2.4.9                                                 | Map Reduce Script.....                                                      | 29 |
| 3.2.4.10                                                | Jira.....                                                                   | 31 |
| 3.2.4.11                                                | Confluence .....                                                            | 31 |
| 3.2.4.12                                                | Git .....                                                                   | 31 |
| 3.2.4.13                                                | BitBucket .....                                                             | 31 |
| 3.2.4.14                                                | Visual Studio Code .....                                                    | 31 |
| 3.2.5                                                   | Implementación de las áreas, procesos, sistemas y sus buenas prácticas..... | 32 |
| 3.3                                                     | Evaluación.....                                                             | 73 |
| CAPITULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA ..... |                                                                             | 76 |
| CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES .....       |                                                                             | 78 |
| 5.1                                                     | Conclusiones .....                                                          | 78 |
| 5.2                                                     | Recomendaciones.....                                                        | 79 |
| 5.3                                                     | Fuentes de información.....                                                 | 80 |

## ÍNDICE DE TABLAS

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Tabla 1. <i>Experiencia laboral del autor</i> .....                                           | 3  |
| Tabla 2. <i>Formación académica del autor del trabajo</i> .....                               | 5  |
| Tabla 3. <i>Cursos realizados</i> .....                                                       | 5  |
| Tabla 4. <i>Certificaciones obtenidas</i> .....                                               | 6  |
| Tabla 5. <i>Formación lingüística</i> .....                                                   | 6  |
| Tabla 6. <i>Puntos de entrada por tipo de SuiteScript</i> . ....                              | 25 |
| Tabla 7. <i>Tareas del desarrollo de la gestión automática de contratos de alquiler</i> ..... | 32 |
| Tabla 8. <i>Documento de especificación para la tarea 1</i> .....                             | 38 |
| Tabla 9. <i>Documento de especificación para la tarea 2</i> .....                             | 51 |
| Tabla 10. <i>Documento de especificación para la tarea 3</i> .....                            | 56 |
| Tabla 11. <i>Documento de especificación para la tarea 4</i> .....                            | 59 |
| Tabla 12. <i>Documento de especificación para la tarea 5</i> .....                            | 63 |
| Tabla 13. <i>Cálculo del IPC</i> .....                                                        | 66 |
| Tabla 14. <i>Documento de especificación para la tarea 6</i> .....                            | 66 |
| Tabla 15. <i>Costos mensuales del equipo del proyecto</i> .....                               | 73 |
| Tabla 16. <i>Costos de Licencias de software</i> .....                                        | 73 |
| Tabla 17. <i>Ingresos por venta</i> .....                                                     | 74 |
| Tabla 18. <i>Proyección de ingresos y gastos</i> .....                                        | 75 |

## ÍNDICE DE FIGURAS

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| Figura 1. <i>Organigrama principal de la empresa</i> .....                                       | 9  |
| Figura 2. <i>Organigrama centrado en subárea “Software Factory”</i> .....                        | 9  |
| Figura 3. <i>Modelo cascada del Ciclo de Vida del Desarrollo de Software</i> .....               | 16 |
| Figura 4. <i>Módulos de un ERP</i> .....                                                         | 19 |
| Figura 5. <i>Principales beneficios del ERP</i> .....                                            | 21 |
| Figura 6. <i>Capa SuiteCloud en la arquitectura tecnológica de NetSuite</i> .....                | 24 |
| Figura 7. <i>Operación en beforeLoad de un User Event</i> .....                                  | 28 |
| Figura 8. <i>Operación beforeSubmit y afterSubmit de un User Event</i> .....                     | 29 |
| Figura 9. <i>Etapas del Script Map Reduce</i> .....                                              | 30 |
| <b>Figura 10. Diagrama de arquitectura de solución</b> .....                                     | 37 |
| Figura 11. <i>Flujo del proceso para la tarea 1</i> .....                                        | 39 |
| Figura 12. <i>Custom Record Billing Schedule</i> .....                                           | 40 |
| Figura 13. <i>Custom Record Billing Schedule Detail</i> .....                                    | 41 |
| Figura 14. <i>Custom Record Billing Frequency</i> .....                                          | 42 |
| Figura 15. <i>Lista de frecuencias</i> .....                                                     | 42 |
| Figura 16. <i>Configuración de campo Fulfillment Start Date</i> .....                            | 43 |
| Figura 17. <i>Configuración de campo Fulfillment Schedule</i> .....                              | 43 |
| Figura 18. <i>Código JavaScript de librería lib_billing.js</i> .....                             | 44 |
| Figura 19. <i>Código JavaScript de UserEvent UE Sales Order</i> .....                            | 45 |
| Figura 20. <i>Código JavaScript de Map/Reduce MR Billing Schedule</i> .....                      | 46 |
| Figura 21. <i>Script Map/Reduce en NetSuite</i> .....                                            | 47 |
| Figura 22. <i>Parámetros del Script Map/Reduce</i> .....                                         | 47 |
| Figura 23. <i>Implementaciones del Script Map/Reduce en NetSuite</i> .....                       | 48 |
| Figura 24. <i>Implementación que genera Item Fulfillments</i> .....                              | 48 |
| Figura 25. <i>Implementación que elimina los registros personalizados Billing Schedule</i> ..... | 49 |
| Figura 26. <i>Líneas de un contrato con frecuencias personalizadas</i> .....                     | 49 |
| Figura 27. <i>Pronóstico de facturación generado para el contrato</i> .....                      | 50 |
| Figura 28. <i>Flujo del proceso de la solución para la tarea 2</i> .....                         | 52 |
| Figura 29. <i>Código JavaScript de librería lib_fam.js</i> .....                                 | 53 |
| Figura 30. <i>Código JavaScript del RESTlet</i> .....                                            | 54 |
| Figura 31. <i>Registro FAM Process creado por el RESTlet</i> .....                               | 55 |
| Figura 32. <i>Activo fijo con el estado actualizado a Disposal</i> .....                         | 55 |

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| Figura 33. Código JavaScript de librería lib_fianza.js .....                 | 57 |
| Figura 34. Código JavaScript de User Event UE Fianza .....                   | 58 |
| Figura 35. Flujo del proceso de la solución para la tarea 4.....             | 60 |
| Figura 36. Código JavaScript de UserEvent UE Sales Order actualizado. ....   | 61 |
| Figura 37. Código JavaScript de la librería lib_renting.....                 | 62 |
| Figura 38. Segmento Personalizado Activo. ....                               | 64 |
| Figura 39. Registro Personalizado Activo .....                               | 64 |
| <b>Figura 40.</b> Código javascript de UE Asset.....                         | 65 |
| <b>Figura 41.</b> Código javascript de lib_asset.js.....                     | 65 |
| Figura 42. Flujo del proceso de la solución para la tarea 6.....             | 68 |
| <b>Figura 43.</b> Registro personalizado Índice Contratos.....               | 69 |
| <b>Figura 44.</b> Campo Fecha Inicio Contrato.....                           | 69 |
| <b>Figura 45.</b> Campo Periodo de ajuste.....                               | 70 |
| <b>Figura 46.</b> Código javascript de Map Reduce Update IPC Automation..... | 71 |
| <b>Figura 47.</b> Código javascript de librería lib_ipc .....                | 72 |

## INTRODUCCIÓN

Las soluciones de planificación de recursos empresariales (ERP, por sus siglas en inglés) deben ser una de las principales herramientas tecnológicas utilizadas por las organizaciones para gestionar eficientemente sus procesos de negocio. El uso de estos sistemas mejora la competitividad de las empresas al permitir la integración de procesos y reducción de costos a lo largo del tiempo, por lo que su implementación es un factor clave para las empresas (Alshalfi, 2018).

En las últimas dos décadas estos sistemas han evolucionado a soluciones en la nube, este avance ha permitido que los ERP extiendan su uso más allá de los sectores tradicionales, incrementando su importancia y adopción en el sector inmobiliario por su capacidad de ofrecer mayor transparencia financiera y un mayor control operativo (Nasirzada, N., 2025).

En este ámbito, el ERP Oracle NetSuite es considerado como una de las soluciones implementada totalmente en la nube líder a nivel global. Gracias a su enfoque modular y a la posibilidad de personalización, las organizaciones pueden adaptar la herramienta a sus necesidades particulares, optimizando así la eficiencia operativa y facilitando la toma de decisiones estratégicas basadas en datos en tiempo real. Esto permite a las empresas a centralizar sus operaciones de forma segura, escalable y accesible desde cualquier lugar (Oracle NetSuite, 2025).

La empresa donde se elaboró este proyecto es uno de los principales socios de Oracle para la implementación del ERP Oracle NetSuite y recientemente tuvo la oportunidad de realizar una implementación en una empresa española de bienes raíces. La gestión eficiente de contratos de alquiler implica la correcta determinación de la renta, actualización periódica mediante índices como el IPC y administración de garantías como la fianza. A pesar de que existen módulos estándar que permiten la gestión general de estos procesos, no llegan a ser una solución completa debido a las particularidades de la frecuencia de facturación y actualizaciones periódicas del IPC.

Este trabajo detalla el proceso del desarrollo e implementación de un módulo que automatiza la gestión de los contratos de alquiler específicamente para el ERP Oracle NetSuite y adaptado a las necesidades particulares de una empresa de bienes raíces. Es importante mencionar que el módulo no solo incluyó las funcionalidades esenciales en la gestión de los contratos de alquiler, sino que también generan registros personalizados,

los cuales son necesarios para un mayor entendimiento de la facturación periódica y generación de reportes para un análisis más extenso. Finalmente, esta solución no solo permitió completar una exitosa implementación, sino que también sirve como base de un producto que puede aplicarse a otros clientes del sector.

A continuación, se detalla la estructura del presente trabajo de suficiencia profesional:

- CAPÍTULO I: Se describe mi trayectoria profesional, seguido de los aprendizajes y certificaciones obtenidas que complementan mi formación profesional.
- CAPÍTULO II: Se presenta el contexto en el que se desarrolló el proyecto, además de mencionar las funciones y responsabilidades desempeñadas, destacando los proyectos más relevantes lo que muestra las habilidades obtenidas y aplicadas dentro de la organización.
- CAPÍTULO III: Se analiza la situación problemática relacionada con el proyecto, definiendo el problema y presentando la solución propuesta. Se detallan los objetivos, el alcance, metodología e implementación, terminando con la evaluación económica que respalda la viabilidad de la solución.
- CAPÍTULO IV: Se presenta una reflexión crítica sobre la experiencia adquirida durante el desarrollo del proyecto. Se evalúan los desafíos enfrentados y las lecciones aprendidas, destacando su impacto en el crecimiento profesional del autor.
- CAPÍTULO V: Se describen las conclusiones del trabajo y se brindan recomendaciones en base a la experiencia y los resultados obtenidos a lo largo del desarrollo del proyecto.

## CAPÍTULO I - TRAYECTORIA PROFESIONAL

### 1.1 Presentación profesional

El autor tiene el grado de Bachiller en Ingeniería de Sistemas y cuenta con más de 5 años de experiencia laboral en el diseño y desarrollo de soluciones y personalizaciones en torno al ERP Oracle NetSuite.

Actualmente se encuentra laborando bajo el rol “Software Engineer” en una consultora socia de Oracle NetSuite con presencia en las regiones de Norteamérica, Latinoamérica y EMEA (Europe, Middle East, and Africa) encargándose de gestión y desarrollo de proyectos con clientes de la compañía.

### 1.2 Educación y certificaciones

A lo largo del trayecto, el autor trabajó para empresas del sector de Tecnología de Información de diferentes tamaños e industrias mayormente en empresas socias de Oracle NetSuite. Así como también se formó de manera complementaria con cursos y certificaciones que se detallan en las siguientes tablas.

**Tabla 1.** *Experiencia laboral del autor*

| Empresa/Periodo                                               | Educación Académica                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Empresa Socia de Oracle NetSuite I (2023 - actualidad)</b> | <b>Cargo:</b> Software Engineer, NetSuite<br><b>Actividades:</b> <ul style="list-style-type: none"><li>- Diseñar y desarrollar personalizaciones de procesos según necesidades de los clientes en ERP NetSuite mediante SuiteScripts en sus versiones 1.0, 2.0 y 2.1, SavedSearch, SuiteAnalytics, entre otros.</li><li>- Desarrollo de generador de reportes para módulo de libros electrónicos usando el framework Angular para la generación de vistas y SuiteScripts para procesar y almacenar información.</li><li>- Implementar y personalizar la solución “Field Service Management” de NetSuite en clientes de la región de Norteamérica.</li></ul> |

---

|                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                               | <ul style="list-style-type: none"> <li>- Integración con Operador de Servicios Electrónicos para emisión de documentos electrónicos.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>INFOSYS</b><br><b>(2022 - 2023)</b>                                        | <b>Cargo:</b> Technology Analyst<br><b>Actividades:</b> <ul style="list-style-type: none"> <li>- Desarrollo y actualización para migrar el módulo de flujo de tickets de un proyecto legacy usando la librería JBPM.io para la interfaz de usuario y Spring para la creación de microservicios.</li> <li>- Diseño y ejecución de pruebas usando Junit y Mockito.</li> </ul>                                                                                                                                                                                                                                                                                     |
| <b>Empresa Socia de</b><br><b>Oracle NetSuite II</b><br><b>(2021 - 2022)</b>  | <b>Cargo:</b> Senior Developer<br><b>Actividades:</b> <ul style="list-style-type: none"> <li>- Implementación de módulo personalizado de facturación eletrónica por medio de integración con proveedores de servicios electrónicos para la generación, envío y consulta de documentos electrónicos.</li> <li>- Experiencia técnica en: SuiteScript 1.0, SuiteScript 2.0, SuiteScript 2.1, SavedSearch, NetSuite Scripting: Suitelet, User Event, Schedule, Map/Reduce and Client Script, SuiteGL Plugins.</li> <li>- Ejecución de pruebas unitarias.</li> <li>- Revisión de incidencias reportados por clientes en ambientes de prueba y producción.</li> </ul> |
| <b>ENTELGY</b><br><b>(2020 - 2021)</b>                                        | <b>Cargo:</b> Java Programmer Analyst<br><b>Actividades:</b> <ul style="list-style-type: none"> <li>- Desarrollo y modificación de servicios REST para banco BBVA usando metodología Scrum.</li> <li>- Diseño y ejecución de pruebas usando Junit y Mockito.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Empresa Socia de</b><br><b>Oracle NetSuite III</b><br><b>(2019 - 2020)</b> | <b>Cargo:</b> SuiteCloud Developer<br><b>Actividades:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

---

- Adquirir conocimientos de flujos de compra y venta, gestión de inventarios, facturación electrónica, entre otros, propios del ERP Oracle NetSuite.
- Refactorizar y actualización de scripts de versiones 1.0 a 2.0.
- Implementación de facturación electrónica propia de la localización para países de Latinoamérica.
- Estandarización de scripts en la generación y envío de documentos electrónicos para reducir la cantidad de scripts de 1 por documento y país a solo 1 que incluye todos los documentos por país.

*Nota.* Fuente: Elaboración propia

**Tabla 2. Formación académica del autor del trabajo**

| Periodo     | Educación Académica                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------|
| 2015 - 2021 | Bachiller de la Escuela Profesional de Ingeniería de Sistemas<br>Universidad Nacional Mayor de San Marcos |

*Nota.* Fuente: Elaboración propia

Los cursos realizados por el autor se encuentran en la tabla 3

**Tabla 3. Cursos realizados**

| Institución     | Curso                                                                        |
|-----------------|------------------------------------------------------------------------------|
| Oracle NetSuite | Partner Introduction to NetSuite                                             |
| Oracle NetSuite | Partner SuiteSuccess Methodology for Delivery (PARSSMETDEL)                  |
| Oracle NetSuite | Performance Strategies in SuiteScript                                        |
| Oracle NetSuite | SuiteCloud Development Framework: Building and Deploying Customizations CD.2 |
| Oracle NetSuite | Advanced PDF/HTML Templates: Customizing Printouts and Emails                |
| Oracle NetSuite | SuiteScript 2.1: Working with the Query API and SuiteQL                      |
| Next Technik    | Next Service Technical Consultant track                                      |

|                             |                                           |
|-----------------------------|-------------------------------------------|
| Mitocode Network            | Microservices using Spring Cloud          |
| Mitocode Network            | Docker Kubernetes Cloud                   |
| Oracle Cloud Infrastructure | Oracle Cloud Infrastructure Foundations   |
| Mitocode Network            | Java Fullstack Spring Boot 2 - Angular 10 |

*Nota.* Fuente: Elaboración propia

Las certificaciones obtenidas por el autor se listan en la Tabla 4:

**Tabla 4. Certificaciones obtenidas**

| Periodo | Certificado                                                  | Institución                 |
|---------|--------------------------------------------------------------|-----------------------------|
| 2025    | Oracle NetSuite Certified AI Foundations Associate           | Oracle NetSuite             |
| 2025    | Oracle NetSuite Certified Application Developer Professional | Oracle NetSuite             |
| 2024    | NetSuite Certified SuiteFoundation                           | Oracle NetSuite             |
| 2024    | NextService Technical Consultant                             | Next Technik                |
| 2021    | Oracle Cloud Infrastructure Foundations                      | Oracle Cloud Infrastructure |
| 2020    | Java Fullstack Spring Boot 2 - Angular 10                    | Mitocode Network            |
| 2017    | International Certificate in English (FCE)                   | Cambridge University        |

*Nota.* Fuente: Elaboración propia

Los estudios de idiomas realizados por el autor se listan en la Tabla 5:

**Tabla 5. Formación lingüística**

| Periodo     | Idioma | Institución |
|-------------|--------|-------------|
| 2014 - 2016 | Inglés | Británico   |

*Nota.* Fuente: Elaboración propia

## **CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA**

### **2.1 Empresa - Actividad que realiza**

La empresa donde se realizó el proyecto es una compañía dedicada a la transformación digital de negocios a través de la implementación y consultoría del ERP Oracle NetSuite. Destacada por su excelencia en desempeño y compromiso en el servicio al cliente ha sido reconocida internacionalmente como socio estratégico de NetSuite, lo que demuestra su compromiso con entregar soluciones personalizadas, adaptaciones profundas y experiencia técnica para ayudar a los clientes a lograr un crecimiento rápido, sostenible, visibilidad operativa y escalabilidad.

Inicialmente la empresa orientó en ofrecer sus servicios en la región de Norteamérica donde logró posicionarse como un referente en consultoría e implementación de Oracle NetSuite, haciéndose acreedor de reconocimientos por su rápido crecimiento durante tres años consecutivos, lo que reflejó su expansión, crecimiento comercial y consolidación como compañía importante en el mercado de transformación digital. Gracias a su crecimiento y consolidación en la región, le permitió expandir su presencia en países de Asia, América Latina y Europa utilizando su metodología para ayudar a distintas organizaciones a optimizar sus operaciones y adaptándose a las particularidades de cada mercado.

Actualmente cuenta con un equipo de más de 400 empleados, incluyendo más de 300 consultores certificados en NetSuite que operan en más de 10 países y servido a más de 800 clientes a nivel global.

### **2.2 Misión de la empresa**

La misión de la empresa consiste en impulsar la digitalización y automatización de industrias sostenibles para ampliar su capacidad de servicio a un mayor número de personas, reduciendo el coste y optimizando los tiempos de atención.

## 2.3 Visión de la empresa

La visión de la empresa es impulsar iniciativas de transformación tecnológica en los negocios de nuestros clientes para generar valor constantemente y promover un impacto positivo en la industria.

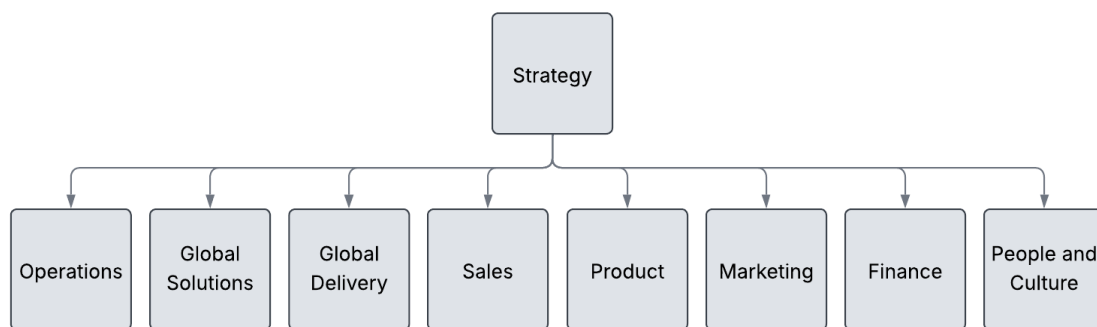
## 2.4 Organización de la empresa

La empresa principalmente se conforma por las áreas presentadas en la Figura 1 y detalladas a continuación:

- **Strategy:** Área responsable de definir y ejecutar la visión y misión de la empresa, así como de establecer los objetivos estratégicos que guíen la dirección general de la organización.
- **Operations:** Área encargada del diseño y desarrollo de soluciones que satisfacen las necesidades del cliente, asegurando la calidad y eficiencia en cada etapa del proceso. Además, se ocupa de gestionar la infraestructura tecnológica de la empresa, garantizando un rendimiento óptimo de las personalizaciones de cada cliente.
- **Global Solutions:** Área encargada del diseño y desarrollo de soluciones relacionadas a integraciones de terceros y análisis de datos. Garantiza consistencia de datos entre los sistemas de los clientes con NetSuite, así como también la visualización de información y seguimiento de indicadores de rendimiento de los empleados en la empresa.
- **Global Delivery:** Área que asegura que los servicios ofrecidos por la empresa sean entregados de forma eficiente, con calidad y en los plazos establecidos. Gestiona la continuidad y soporte post-implementación.
- **Sales:** Área responsable de la gestión comercial y de la búsqueda de oportunidades de colaboración. Construir y mantener relaciones de confianza con diversas organizaciones que se benefician de las soluciones que ofrece la empresa.
- **Product:** Área destinada a la investigación y desarrollo de herramientas y aplicaciones propias de la empresa para complementar sus servicios y para mejorar el funcionamiento de la empresa.
- **Marketing:** Área dedicada a implementar campañas y estrategias para promover los servicios y/o productos de la empresa, con el propósito de posicionar la presencia de la empresa y maximizar la captación y fidelización de clientes.

- **Finance:** Área responsable de gestionar los recursos económicos de la empresa, asegurando su sostenibilidad financiera y proporcionar la información necesaria para la toma de decisiones estratégicas.
- **People and Culture:** Área encargada de la gestión de talento humano y fomentar la cultura organizacional de la empresa, promoviendo iniciativas de desarrollo profesional y personal para lograr un entorno de bienestar y desarrollo dentro de la organización.

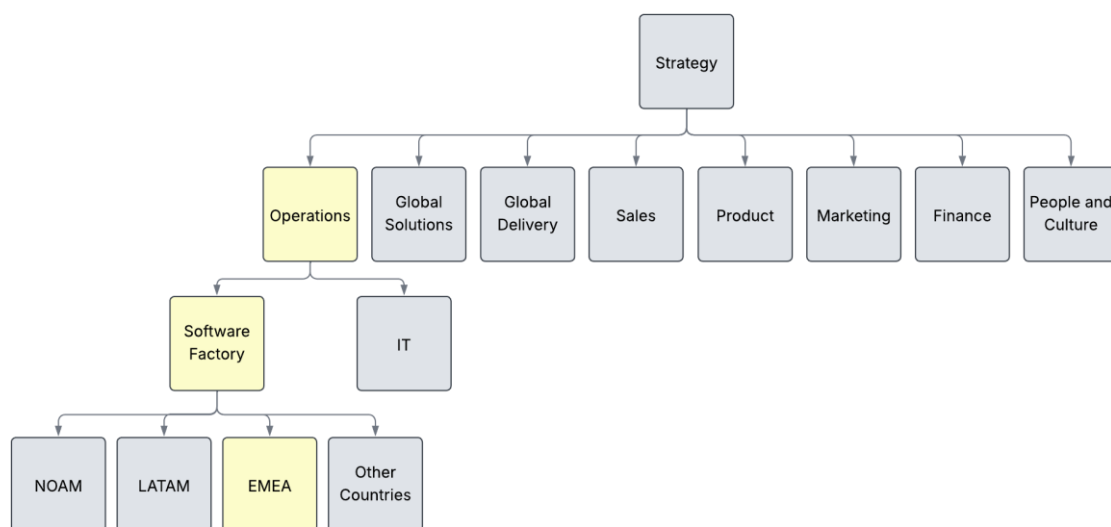
**Figura 1.** Organigrama principal de la empresa



*Nota.* Fuente: Elaboración propia

Lás áreas principales se subdividen en departamentos y en la región que operan. El presente trabajo toma lugar en el área de “Operations” (operaciones) en la subárea de “Software Factory” que opera en la región de EMEA que se muestra en la Figura 2.

**Figura 2.** Organigrama centrado en subárea “Software Factory”



*Nota.* Fuente: Elaboración propia

## **2.5 Área, cargo y funciones desempeñadas**

Para el presente trabajo el autor asumió el rol *Software Engineer, NetSuite* prestando servicios en la subárea *Software Factory* para la región EMEA donde desempeñó las funciones listadas a continuación:

- Desarrollo de personalizaciones en el ERP Oracle NetSuite acorde a las necesidades del cliente. Principalmente en la región EMEA.
- Diseño e implementación de soluciones técnicas utilizando SuiteScripts 1.0, 2.0 y 2.1, flujos de trabajo, entre otros.
- Dominio de SuiteScript: Client Script, User Event, Suitelet, Schedule, Map/Reduce, RESTlet, Portlet y Custom Plugin Implementations.
- Entendimiento y ejecución de consultas avanzadas en SavedSearch.
- Dominio en FreeMarker y generación de plantillas PDF avanzadas.
- Manejo de objetos usando SuiteBundler y SuiteCloud Development Framework.
- Análisis de requerimientos y crear soluciones en colaboración con consultores funcionales y arquitectos.
- Manejo de Git y Bitbucket para el control de versiones de los desarrollos.
- Generar documentación técnica de las soluciones desarrolladas en Confluence.
- Atención de requerimientos por medio de tareas utilizando la herramienta Jira.
- Participar en reuniones con equipo funcional y arquitectos para diseñar nuevas soluciones y dar seguimiento a incidencias.
- Revisar y mantener soluciones existentes, proponer mejoras y optimizar scripts para mejorar el producto.
- Entrenamiento y apoyo técnico a los integrantes que se incorporan al equipo.

## **2.6 Experiencia profesional realizada en la organización**

El autor del presente trabajo ha sido encargado de la implementación y desarrollo de distintos proyectos en la organización con la oportunidad de colaborar con los equipos de las distintas regiones. Entre ellos se destacan los siguientes proyectos:

- Implementación de módulo de facturación electrónica para la generación, envío y consulta de documentos electrónicos como facturas, notas de crédito y guías de remisión para una empresa de Perú. Este proyecto destaca en el uso de los servicios web del Operador de Servicios Electrónicos, el uso de SuiteScript y Custom Records para montar una solución escalable a incluir más documentos

electrónicos o actualizaciones futuras sin necesidad de grandes modificaciones. Lo que permitió que el cliente cuente con sus procesos de facturación electrónica automática, además de tener los datos enviados y recibidos de la SUNAT incluyendo los documentos en formato XML y PDF.

- Desarrollo de un módulo en las transacciones que genera distintos tipos de etiquetas en formato PDF con la información de los artículos serializados en una empresa de Estados Unidos que ofrece servicios de instalación de suelos para interiores. La solución permite al cliente seleccionar el tipo y cantidad de etiquetas que necesita generar, así como facilita la asignación de lotes en el proceso de recepción de los artículos. Este proyecto destaca en el uso de la librería SweetAlert2 para mostrar el formulario del módulo en un pop-up ofreciendo una forma más intuitiva, personalizable y responsiva al usuario, además de que el módulo sirvió de base para la implementación en otro cliente con cambios menores en la plantilla de la etiqueta y lógica para la generación de estas.
- Automatización en la generación de facturas con la información del sistema de un cliente que provee sistemas y dispositivos de juegos para loterías. Este proyecto destaca en el desarrollo de una integración consumiendo la API del sistema del cliente para la recepción de la información de alquiler de sus dispositivos de manera mensual, almacenamiento y transformación de datos en NetSuite, culminando con la generación de facturas permitiendo al cliente tener la facturación actualizada en NetSuite.
- Creación de flujo de aprobación en órdenes de compra usando SuiteFlow que permite a los usuarios aprobar, rechazar y reiniciar el flujo en su proceso de compra. Además, permite a los consultores o usuarios con un rol específico, modificar las condiciones y aprobadores en el flujo de trabajo.
- Implementación de un módulo para la visualización del estado y conciliación de Órdenes de trabajo en una empresa proveedora de insumos de café y elaboración de bebidas. Este proyecto destaca en el uso del framework VUE y SuiteScripts para montar un panel de control donde se muestra la información actualizada constantemente de las órdenes de trabajo, sobre todo su producción actual, lo que permite al usuario conocer si puede realizar o no la conciliación de la orden de trabajo.

En adición a los proyectos mencionados anteriormente, se realizó el desarrollo del presente proyecto, el cual se centra en la creación de un módulo que permite la gestión de los contratos de alquiler mediante el cálculo correcto de la renta, modificación de contratos por la actualización del IPC y generación de la proyección de facturación.

## **CAPÍTULO III – ACTIVIDADES DESARROLLADAS**

### **3.1 Situación problemática**

De acuerdo con Vera (2006), las empresas deciden implementar el ERP por la necesidad de tener acceso a información confiable, optimizar sus procesos y disponer de la información en todas las áreas de la organización.

Para toda implementación de ERP en una empresa, conocer y entender los procesos de la empresa es de vital importancia para una implementación exitosa ya que el sistema debe permitir, mediante sus diversos módulos, el correcto funcionamiento de las operaciones (Gupta et al., 2018). Esto incluye los procesos particulares de cada organización, lo cual es un desafío en cada implementación debido a que los módulos estándar no siempre cumplen con la incorporación total de los procesos de la organización.

Factores como el presupuesto, conocimientos y una cultura de innovación también son fundamentales para el éxito de la adopción del ERP y la colaboración de socios de NetSuite ayuda en este sentido para reducir costos y acelerar los ciclos de implementación o desarrollo de productos (Hong et al., 2024).

En el escenario específico de empresas del sector inmobiliario en España y de acuerdo con la Ley 29/1994, de 24 de noviembre, de Arrendamientos Urbanos, el cálculo y cobro de la fianza se debe depositar en los organismos de cada comunidad autónoma, según la obligatoriedad en cada una de ellas. Actualmente existe dos indicadores para la actualización de alquileres que son el IPC (Índice de Precios al Consumo) y el IRAV (Índice de Referencia de Arrendamientos de Vivienda) los cuales son definidos por el INE (Instituto Nacional de Estadística) de acuerdo con la Ley 12/2023, de 24 de mayo, por el derecho a la vivienda. Adicionalmente, suelen contar o usar diversas plataformas que se encargan de la reserva de los inmuebles, firmas de contrato, cálculo de mensualidades y división de pagos. Lo que implica que el sistema debe asegurar la sincronización de estos sistemas con el ERP mediante integraciones.

En este contexto donde las necesidades de la organización se vuelven más complejas, NetSuite ofrece una capa de personalización que, por medio del uso de lenguajes de programación, se construyan soluciones que permitan incorporar estos procesos adaptando los servicios del ERP a las diferentes lógicas de negocio de cada empresa. El principal lenguaje de programación de este ERP es SuiteScript, el cual está

basado en JavaScript, permite personalizar y automatizar procesos dentro del ERP tanto del lado del cliente como del servidor (Oracle NetSuite, 2024).

### **3.1.1 Definición del problema**

La empresa donde se intenta implementar el ERP Oracle NetSuite enfrenta desafíos significativos debido a que las integraciones estándar de la plataforma no permiten la incorporación total de los procesos principales para el alquiler de vivienda y automatización de contratos de alquiler. El módulo estándar *Advanced Billing* permite la generación automática de facturas en base a la configuración que dictan la frecuencia en la cual deben ser generadas, pero no existe una configuración o combinación de reglas que incluya las frecuencias y tiempo de contrato usados por la empresa inmobiliaria. Esto pone en riesgo la implementación y salida a producción del proyecto.

## **3.2 Solución**

La solución presentada en el presente trabajo de suficiencia profesional es el desarrollo e implementación de un módulo personalizado que permita automatizar la gestión de contratos de alquiler dentro del ERP Oracle NetSuite cumpliendo los requisitos establecidos por la empresa inmobiliaria. La solución busca visualizar la previsión de la facturación de cada contrato, actualizar los montos a facturar en base a la actualización del IPC y complementar las funcionalidades nativas del sistema. Con la automatización, se espera que la empresa inmobiliaria cuente con sus procesos internos integrados en el ERP, así como la sincronización de sus sistemas con el ERP. Además de una exitosa implementación de NetSuite en la empresa.

### **3.2.1 Objetivos**

#### **3.2.1.1 Objetivo General.**

Implementar un módulo personalizado para la gestión de contratos de alquiler en el ERP Oracle NetSuite, que integre los procesos internos para una empresa inmobiliaria, asegurando la conformidad del servicio.

### 3.2.1.2 Objetivos específicos.

- Analizar los requerimientos del cliente que requieren de personalizaciones en el ERP Oracle NetSuite.
- Diseñar las funcionalidades que se implementarán en el ERP Oracle NetSuite siguiendo los lineamientos de la empresa bajo el modelo de cascada del marco que sigue el Ciclo de Vida del Desarrollo de Software.
- Desarrollar un módulo automatizado de gestión de contratos mediante el uso de herramientas proporcionadas por la plataforma SuiteCloud, que asegure el cumplimiento de los requerimientos funcionales definidos.
- Validar el funcionamiento de las personalizaciones desarrolladas, mediante pruebas internas y con el usuario final antes de su despliegue a producción.
- Documentar los procesos técnicos y pruebas realizadas para garantizar la trazabilidad de los requerimientos.

### 3.2.2 Alcance

La ejecución del proyecto fue iniciada para que la implementación del ERP Oracle NetSuite en una empresa inmobiliaria cumpla con los requisitos exigidos por la misma con relación a sus procesos internos para gestión de alquiler.

#### **Limitaciones del alcance:**

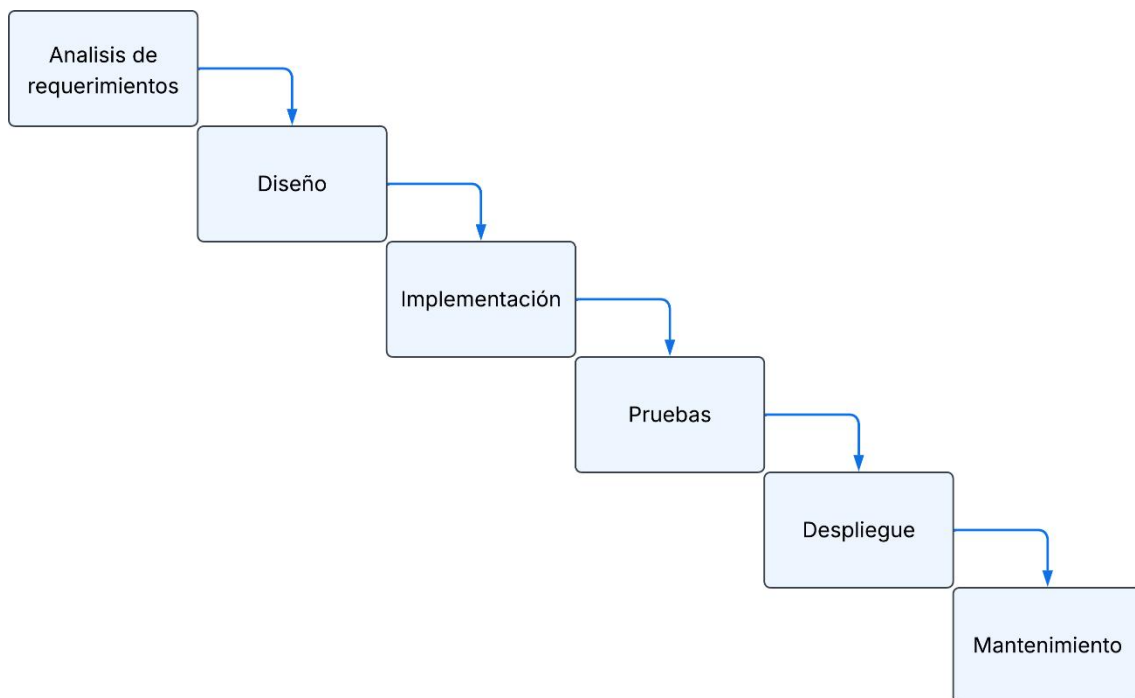
- El módulo personalizado no genera facturas automáticamente debido a que sí es posible crearlas usando la funcionalidad estándar *Billing Operations* creando las facturas desde los *Item Fulfillments* generados por el módulo personalizado.
- El módulo no incluye procesos fiscales como el envío de la factura electrónica a Hacienda, debido a que se usa la facturación electrónica de la localización española.

### 3.2.3 Etapas y metodología

Este proyecto se ha desarrollado bajo el modelo de cascada del marco que sigue el Ciclo de Vida del Desarrollo de Software, SDLC por sus siglas en inglés, el cual se utiliza para diseñar, desarrollar y producir productos o soluciones de software

de alta calidad, fiables, rentables y en los plazos previstos en la industria de software. De acuerdo con Shravan Pargaonkar (2023). Si bien las metodologías ágiles son adecuadas para proyectos que requieren flexibilidad y mejora continua, se recomienda el método de cascada para proyectos donde los requerimientos ya se encuentran definidos o son predecibles y donde se requiera documentación detallada que cubra los límites del proyecto (*scope*). Este modelo consiste en el desarrollo secuencial por etapas que no se superponen como se muestra en la Figura 3.

**Figura 3.** *Modelo cascada del Ciclo de Vida del Desarrollo de Software*



*Nota.* Fuente: Elaboración propia

Esta metodología y modelo es la designada para el desarrollo de soluciones para este tipo de proyectos en el área de la empresa donde se realizó la solución presentada en el trabajo de suficiencia profesional.

Adicionalmente, es fundamental para la empresa tener el control de todos los componentes creados y usados en cada solución desarrollada. Esto permite revisar los cambios, estados y dónde han sido desplegados los componentes. Lo que ayuda a evitar pérdida de componentes y errores al momento de desplegar las soluciones, así como también permite reusar componentes entre otras cuentas que requieran de la misma solución o con ligeros cambios. Para lo cual se usan las siguientes herramientas:

- Jira: Para crear, asignar y dar seguimiento a las tareas. Esta herramienta permite también implementar el flujo de la metodología con ligeros cambios que se ajustan a lo que la empresa implementa.
- Confluence: Para la gestión de toda la documentación generada tanto para tareas como para elaboración de guías y plantillas propios de la empresa.
- BitBucket: Es la plataforma de colaboración para el control de versiones de código basado en Git, usada por la empresa para alojar, gestionar y compartir código.

Para el desarrollo del proyecto, el equipo se conformó por los siguientes roles en los términos de la empresa:

- Project Manager: Tiene la responsabilidad de coordinar y dar seguimiento a las tareas de desarrollo. Mantener comunicación con el cliente con respecto al progreso del proyecto, así como también manejar la comunicación interna entre los equipos involucrados.
- NetSuite Implementation Consultant: Es responsable del trabajo con el equipo del cliente para entender los requerimientos y determinar la mejor solución funcional que cubra los requerimientos. En caso de que el requerimiento requiera de una personalización, debe generar un documento de especificación para el equipo de desarrollo, y posterior al desarrollo, realizar las pruebas con el cliente para su aceptación.
- Software Engineer, NetSuite: Se encarga del desarrollo de la solución para cubrir los requerimientos solicitados por el equipo funcional. Debe revisar el documento de especificación y entender los requerimientos, realizar el desarrollo de la solución, incluyendo pruebas unitarias que cubran los casos de prueba y realizar el respaldo de los componentes y código generado en el repositorio. Por último, generar los documentos técnicos y de pruebas unitarias.
- Software Engineer, Deployment: Se encarga de realizar el despliegue de la solución, siguiendo el documento técnico realizado por el desarrollador, así como también gestionar los cambios en el repositorio y, por último, generar un documento post implementación.

### **3.2.4 Fundamentos utilizados**

#### **3.2.4.1 Contrato de alquiler**

Galán (2021) define el alquiler como “*el proceso mediante el cual dos partes efectúan la cesión temporal de un bien o servicio a cambio de una contraprestación que generalmente es de tipo económico.*” El alquiler de una vivienda debe tener sustento legal por lo que un contrato de alquiler es un acuerdo legal entre un propietario y un inquilino donde el primero cede el uso temporal de un inmueble a cambio de una renta periódica. Cuenta con cláusulas donde se especifican elementos como la duración, fianza, renta, gastos y finalización del contrato (Junta de Andalucía, s. f.).

#### **3.2.4.2 Fianza**

Es la suma de dinero que se guarda como garantía en caso de que el inquilino incumpla con alguna cláusula del contrato o si no entrega la vivienda en buen estado. De no ser este el caso, la fianza es devuelta al inquilino cuando finaliza el contrato. De acuerdo con el artículo 36.1 de la Ley 29/1994, de 24 de noviembre, se calcula como “*cantidad equivalente a una mensualidad de renta en el arrendamiento de viviendas y de dos en el arrendamiento para uso distinto del de vivienda*”.

#### **3.2.4.3 Renta**

De acuerdo con el artículo 17 de la Ley 29/1994, de 24 de noviembre. Es un monto generalmente mensual que debe efectuarse en los primeros siete días del mes y es un monto libremente pactado entre el inquilino y el propietario.

#### 3.2.4.4 ERP (Enterprise Resource Planning)

Un ERP o sistema de planificación de recursos empresariales, por sus siglas en español, es un sistema completo que administra los procesos principales y áreas de una empresa como contabilidad, gestión de personal, compras, ventas, facturación y gestión de inventario, integrados en una base de datos. Tiene como principal objetivo optimizar las operaciones, mejorar la rentabilidad, eliminar redundancia de información y permite facilitar la visibilidad en tiempo real para facilitar la toma de decisiones estratégicas. (Oracle, 2024)

Los ERP se componen de diversos módulos, funciones y capas que garantizan el funcionamiento correcto de las operaciones de la empresa. Como se muestra en la Figura 4, Según (Oracle NetSuite, 2024), los principales módulos que incluye un ERP son Gestión financiera, Gestión de recursos humanos, Gestión de la cadena de suministro y Gestión de las relaciones con el cliente (CRM). Y los módulos adicionales comprenden el de Fabricación, Gestión de inventario, Gestión de proyectos, Comercio Electrónico, Automatización de marketing, Compras, Gestión de pedidos, Gestión de almacén y Gestión de fuerza laboral.

**Figura 4.** Módulos de un ERP



*Nota.* Módulos de un ERP Fuente: (Oracle, 2025)

Existe diferentes modelos de soluciones ERP como servidores locales, en la nube o híbridas, y la elección entre los modelos se centra en las necesidades particulares de cada empresa, así como también su presupuesto e infraestructura.

Con el fin de que la empresa optimice sus operaciones y obtenga el mayor beneficio de la solución, se debe realizar un análisis detallado de las características y funcionalidades que ofrece un ERP. De acuerdo con Oracle (2024), un ERP en la nube resalta por los siguientes beneficios mostrados en la Figura 5:

- **Rápida integración:** Los ERP en la nube se componen de módulos, lo que facilita su implementación frente a implementaciones locales, lo que permite que las empresas empiecen a utilizar la solución más rápidamente. En cuanto al tiempo de implementación, un ERP en la nube suele tardar entre unas semanas o pocos meses, mientras que implementaciones locales tradicionales llegan a tardar de 1 año a más.
- **Seguridad:** Cuentan con equipos de seguridad para resguardar la información en el sistema y control de accesos basados en roles.
- **Productividad:** Automatiza tareas repetitivas y permite incrementar la productividad de los empleados.
- **Movilidad:** Permite a los usuarios acceder desde cualquier dispositivo y lugar.
- **Soporte:** Proporciona asistencia las 24 horas en la gestión de incidencias o aplicaciones.
- **Datos centralizados:** Cuenta con diversas fuentes de datos integrados y paneles de control que reflejan los datos a los usuarios tan pronto estén disponibles.
- **Visibilidad:** Permiten a los usuarios generar informes, reportes y analizar datos, además del acceso de dispositivos móviles.
- **Menor coste:** Las soluciones en la nube permiten a las empresas pagar por los usuarios que utilizan el sistema, ahorrando costos de instalación, mantenimiento y hardware.
- **Cumplimiento:** Provee informes para ayudar a las empresas a cumplir con las regulaciones locales e internacionales.
- **Escalabilidad:** El sistema se adapta a medida que la empresa va creciendo.
- **Reportería:** Permite la generación de reportes detallados para su análisis y ayuda en toma de decisiones.
- **Flexibilidad:** Se adapta a las necesidades de la organización y permite una implementación gradual de sus procesos.

**Figura 5. Principales beneficios del ERP**



*Nota.* Principales beneficios del ERP. Fuente: (Oracle, 2025)

### 3.2.4.5 Oracle NetSuite

Es uno de los pioneros en el modelo de software empresarial en la nube. Fue fundado en 1998 con el nombre de NetLedger y fue de las primeras plataformas en ofrecer servicios de ERP en línea, en un momento donde la mayoría de los sistemas se instalaban de manera local. Años más tarde fue renombrada a NetSuite e inicialmente ofrecía aplicaciones financieras como Libro Mayor, Cuentas por cobrar, Cuentas por pagar, recursos humanos, gestión de pedidos, inventario, envío y facturación. En 2016, es adquirida por Oracle por aproximadamente 9.3 billones de dólares y toma el nombre de Oracle NetSuite, manteniendo su identidad como ERP en la nube con el respaldo de la infraestructura de Oracle y consolidándose como una de las principales plataformas ERP implementadas totalmente en la nube. (Oracle NetSuite, 2019)

De acuerdo con Kelly, Braud, & Huffman, M (2010), NetSuite es una solución que combina las necesidades de una empresa en una sola aplicación. Incorpora las operaciones fundamentales del negocio en una base de datos y cuenta con una interfaz personalizable que permite a los usuarios acceder de cualquier dispositivo con conexión a internet.

### 3.2.4.6 NetSuite SuiteCloud Platform

Es un entorno tecnológico de herramientas de desarrollo en la nube, aplicaciones e infraestructura que permite a los usuarios y desarrolladores aprovechar al máximo las ventajas de la nube. Brinda tecnologías de infraestructura como servicio (IaaS por sus siglas en inglés), plataforma como servicio (PaaS por sus siglas en inglés) y software como servicio (SaaS por sus siglas en inglés). Está diseñado para profesionales de tecnologías de información, analistas de procesos y desarrolladores con las herramientas necesarias para la configuración, personalización e integración programática de las funcionalidades nativas del sistema. La capacidad de adaptación que provee lo convierte en un componente estratégico dentro de la arquitectura tecnológica de las organizaciones que buscan optimizar sus operaciones mediante soluciones ERP en la nube. (Oracle NetSuite, 2024)

Las funcionalidades más importantes que ofrece SuiteCloud Platform se detallan a continuación:

- **Personalizaciones:** Cuenta con herramientas visuales y configurables que permiten a los usuarios disponer de sus datos y manejarlos a su criterio creando registros personalizados, formularios y otros componentes que se ajustan a los objetivos de la organización.
- **Automatización de procesos:** Brinda la capacidad de definir visualmente los flujos de trabajo usando herramientas sin código permitiendo a usuarios sin conocimiento técnico a automatizar procesos de negocio comunes o tareas repetitivas como flujos de aprobaciones.
- **Internacionalización:** Permite configurar el sistema para usuarios finales o administradores independientemente del idioma, la moneda, formatos de fecha y hora y otras características específicas de cada región.
- **Monitoreo de rendimiento:** Cuenta con paneles de control en tiempo real que brindan visibilidad del rendimiento y la estabilidad de NetSuite, adicionalmente, incluye herramientas para identificar rápidamente los problemas.
- **Integración:** El sistema brinda conectores y API que diseñados para facilitar la transferencia de datos entre sistemas externos con NetSuite lo que asegura la interoperabilidad y sincronización de datos.

- **Desarrollo de la plataforma:** El sistema ofrece un entorno completo que permite a desarrolladores crear y gestionar el ciclo de vida de aplicaciones y componentes personalizados. Esto mejora y amplía las funcionalidades del ERP en base a las necesidades del negocio.

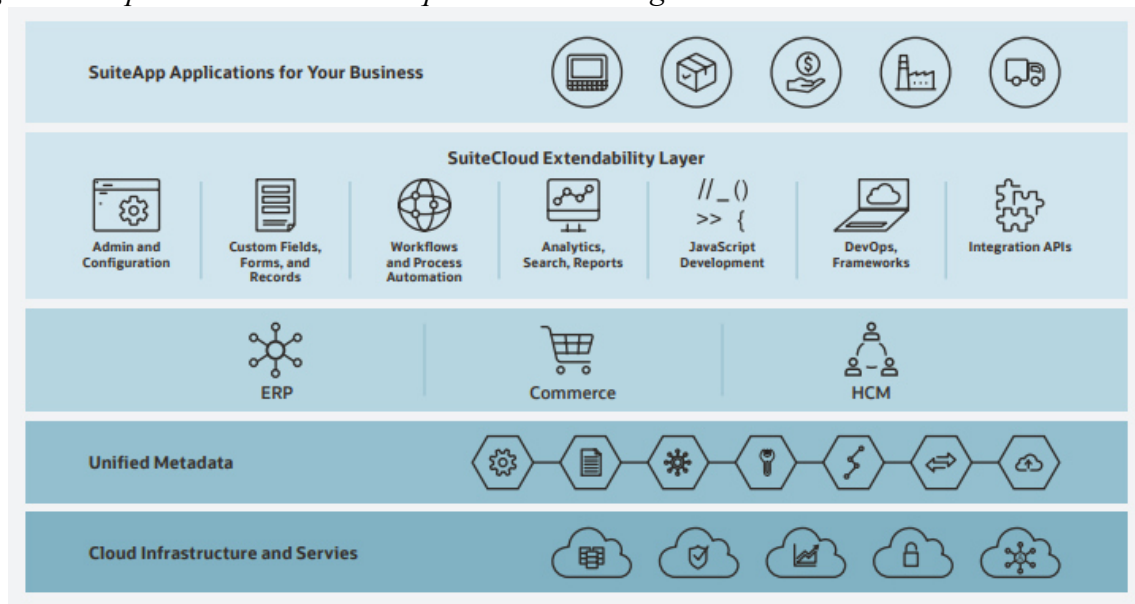
La plataforma cuenta con herramientas para desarrolladores que permiten tener un entorno completo y personalizado en la nube para adaptar NetSuite a las necesidades empresariales como para desarrollar aplicaciones totalmente nuevas. Estas herramientas que se detallan a continuación incluyen la gestión de proceso, análisis de datos, scripting e integración y en la Figura 6 se muestra la capa de SuiteCloud en el conjunto de tecnologías de NetSuite

- SuiteBuilder: Engloba todas las opciones de personalización disponibles a través de la interfaz de usuario sin la necesidad de escribir código. Entre sus principales funcionalidades se encuentran la personalización de dashboards, creación de formularios, campos, listas y registros personalizados que se orientan a tablas de bases de datos llamados *Custom Records*.
- SuiteFlow Workflow Engine: Es una herramienta tipo “arrastrar y hacer clic” fácil de usar para modificar y automatizar los procesos de trabajo en NetSuite.
- SuiteAnalytics Business Intelligence: Es la herramienta de inteligencia de negocio que proporciona la creación de conjunto de datos y la generación de informes independientes que facilitan la toma de decisiones. Cuenta con un conjunto de herramientas gráficas intuitivas llamada SuiteAnalytics, permite a los usuarios finales realizar análisis en tiempo real en todas las áreas de pedidos, clientes, finanzas y también con los componentes personalizados para obtener la información y respuestas que necesitan.
- SuiteTalk Web Services: Esta herramienta facilita a los desarrolladores a integrar NetSuite con aplicaciones en la nube de terceros, utiliza estándares como SOAP y REST, lo que permite una comunicación entre plataformas para compartir datos y crear componentes en NetSuite.
- SuiteBundler: Es la herramienta de empaquetado y distribución de aplicaciones. Facilita a los socios y clientes finales la implementación de

aplicaciones personalizadas que han sido creadas en la plataforma SuiteCloud. Permite el traslado de estas aplicaciones de entornos de pruebas a un entorno de producción. Además, cuenta con la supervisión de versiones y la organización de personalizaciones que componen cada *Bundle*.

- SuiteScript: Permite a los desarrolladores crear soluciones para implementar las lógicas de negocio dentro de NetSuite, desde procesos específicos hasta aplicaciones completamente nuevas.

**Figura 6.** *Capa SuiteCloud en la arquitectura tecnológica de NetSuite*



*Nota.* Capa SuiteCloud en la arquitectura tecnológica de NetSuite. Fuente: (Oracle, 2024)

### 3.2.4.7 SuiteScript

Es el lenguaje de programación principal de NetSuite, está basado en JavaScript y permite la personalización y automatización completa de los procesos de negocio. Cuenta adicionalmente con un API que facilita el acceso a registros del sistema, manejar información de usuarios, gestión de archivos, realizar búsquedas, entre otros. La versión más actual y recomendada para usar es la 2.1 pero sigue siendo compatible con sus versiones anteriores 2.0 y 1.0.

Los scripts de SuiteScript se clasifican principalmente en dos categorías según su ejecución, script de tipo Cliente y Servidor, ya que se ejecutan en el navegador del usuario o en el servidor de NetSuite. A continuación, se describen cada una de las categorías y los tipos de script en cada categoría:

- Cliente: Existe solo 1 tipo de script cliente, Client Script, el cual se ejecuta en el navegador en base a eventos realizados por el usuario, como cuando se carga la página, ingresa o actualiza valores en campos, agregar líneas o al grabar la página. Es el tipo de script que se usa cuando se necesita automatizar o validar algún proceso mientras el usuario está interactuando con el sistema en algún registro o componente en donde esté implementado el script, como puede ser una transacción o algún registro personalizado.
- Servidor: Estos tipos de script son utilizados para el procesamiento de registros, crear pantallas personalizadas, exponer *endpoints* para realizar algún proceso en específico, procesar gran cantidad de datos mediante ejecuciones periódicamente. En esta categoría encontramos a los User Event, Suitelet, Workflow Action, Schedule, Map/Reduce, RESTlet, Portlet y Mass Update.

Cada tipo de SuiteScript tiene puntos de entrada que definen a las funciones principales propias de cada tipo de SuiteScript. Para crear un SuiteScript, se necesita de un archivo JavaScript que tenga un encabezado al inicio donde se indique la versión y tipo de SuiteScript que se quiera crear, esto sirve como identificador para que la plataforma reconozca el tipo de SuiteScript que se quiera crear y muestre el registro correspondiente. Por último, este archivo JavaScript debe exponer los puntos de entrada. Los SuiteScript también brindan la opción de crear parámetros para configurar el script, el uso de parámetros permite personalizar el comportamiento del script sin la necesidad de modificar el código. En la Tabla 6 se detallan los puntos de entrada por cada tipo de SuiteScript.

**Tabla 6.** *Puntos de entrada por tipo de SuiteScript.*

| Tipo de SuiteScript | Puntos de entrada                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Client              | <ul style="list-style-type: none"> <li>- pageInit: Define la función que se ejecuta cuando la página termina de cargarse o cuando se restablece el formulario.</li> <li>- fieldChanged: Define la función cuando el usuario modifica un campo.</li> <li>- postSourcing: Define la función que se ejecuta cuando se modifica un campo que obtiene información de otro campo.</li> <li>- lineInit: Define la función que se ejecuta cuando se inicializa una línea o se añade una línea nueva.</li> </ul> |

---

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <ul style="list-style-type: none"> <li>- <code>sublistChanged</code>: Define la función que se ejecuta después de que una sublista haya sido insertada, eliminada o actualizada.</li> <li>- <code>validateField</code>: Define la función que se ejecuta cuando el valor de un campo luego de que haya sido modificado por el usuario.</li> <li>- <code>validateInsert</code>: Define la función que se ejecuta cuando se inserta una línea de una sublista.</li> <li>- <code>validateLine</code>: Define la función que se ejecuta antes de añadir una línea de una sublista.</li> <li>- <code>validateDelete</code>: Define la función que se ejecuta cuando se elimina una línea de una sublista.</li> <li>- <code>saveRecord</code>: Define la función que se ejecuta cuando se guarda el registro.</li> <li>- <code>localizationContextEnter</code>: Define la función que se ejecuta cuando el registro entra en el contexto de la localización especificado en la implementación del script.</li> <li>- <code>localizationContextExit</code>: Define la función que se ejecuta cuando el registro sale del contexto de localización</li> </ul> |
| User Event | <ul style="list-style-type: none"> <li>- <code>beforeLoad</code>: Define la función que se ejecuta cada vez que se produce una operación de lectura en un registro y antes de retornar el registro o la página.</li> <li>- <code>beforeSubmit</code>: Define la función que se ejecuta antes que el registro sea actualizado en la base de datos.</li> <li>- <code>afterSubmit</code>: Define la función que se ejecuta luego que el registro haya sido actualizado en la base de datos de NetSuite.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Schedule   | <ul style="list-style-type: none"> <li>- <code>execute</code>: Define la función que se ejecuta cuando se activa el script programado.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Map/Reduce | <ul style="list-style-type: none"> <li>- <code>getInputData</code>: Define la función de la primera etapa donde se genera los datos de entrada.</li> <li>- <code>map</code>: Define la función que invoca la etapa de mapeo, donde los datos de entrada se analizan en pares de clave-valor.</li> <li>- <code>reduce</code>: Define la función que invoca la etapa de reducción donde se puede realizar el procesamiento por cada par clave-valor.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <ul style="list-style-type: none"> <li>- summarized: Define la función que invoca la etapa donde se resumen los datos procesados.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Suitelet        | <ul style="list-style-type: none"> <li>- onRequest: Define la función que se ejecuta cuando un Suitelet es llamado por medio de una solicitud HTTPS de tipo GET o POST.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                       |
| RESTlet         | <ul style="list-style-type: none"> <li>- get: Define la función que se activa cuando se envía una solicitud HTTPS de tipo GET. Retorna una respuesta HTTPS.</li> <li>- post: Define la función que se activa cuando se envía una solicitud HTTPS de tipo POST. Retorna una respuesta HTTPS.</li> <li>- put: Define la función que se activa cuando se envía una solicitud HTTPS de tipo PUT. Retorna una respuesta HTTPS.</li> <li>- delete: Define la función que se activa cuando se envía una solicitud HTTPS de tipo DELETE. Retorna una respuesta HTTPS.</li> </ul> |
| Workflow Action | <ul style="list-style-type: none"> <li>- onAction: Define la función que activa una acción del flujo de trabajo.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                              |

---

*Nota.* Fuente: Elaboración propia

Cada tipo de script tiene una capacidad máxima de utilización preestablecida llamada gobernanza (governance), estas unidades se van consumiendo a medida que se hace uso de las API que tiene SuiteScript lo que implica que al desarrollar estos scripts se debe tener en cuenta estas limitaciones para evitar errores durante su ejecución.

Para que estos scripts sean ejecutados, no basta con subir el archivo al sistema y crear el registro del script, deben también ser implementados. Esta implementación puede ser que sea sobre un registro, como lo son los Client Script, User Event y Workflow Action, o en el servidor y se ejecuten cuando sean llamados por otra tarea.

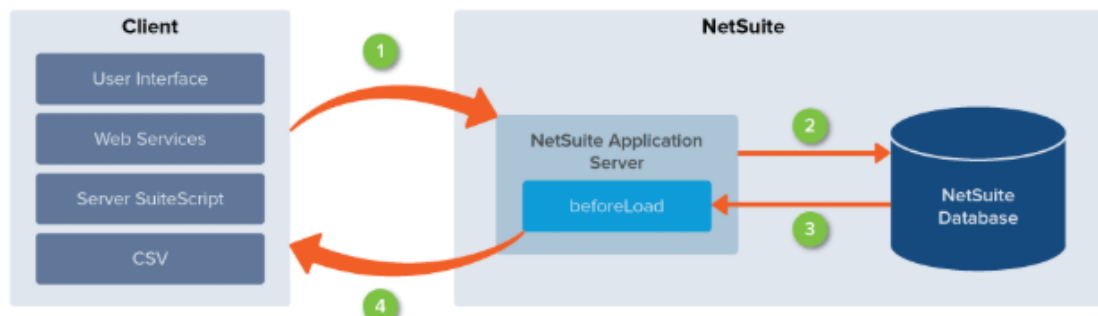
#### **3.2.4.8 User Event Script**

Es un tipo de SuiteScript que se ejecuta en el servidor de NetSuite, se utiliza para aplicar validación de datos y reglas de negocio. Este tipo de SuiteScript se activa cuando el usuario trabaja con registros y cambios de NetSuite al abrir, editar o guardar los registros. Adicionalmente, permite personalizar la interfaz de usuario del formulario del registro actual como agregar botones, crear o inhabilitar campos, entre otras funciones antes que cargue la página; así como también, sincronizar datos con fuentes externas en tiempo real.

La ejecución de este script puede activarse también desde otros SuiteScripts, importando archivos tipo CSV, entre otros. Tiene un límite de 1000 unidades de uso en lo que respecta a su *governance*.

Como se menciona anteriormente, este tipo de script tiene los puntos de entrada *beforeLoad*, *beforeSubmit* y *afterSubmit*. En la Figura 7 y Figura 8 se muestra una descripción general de lo que ocurre en cada operación.

**Figura 7.** Operación en *beforeLoad* de un User Event

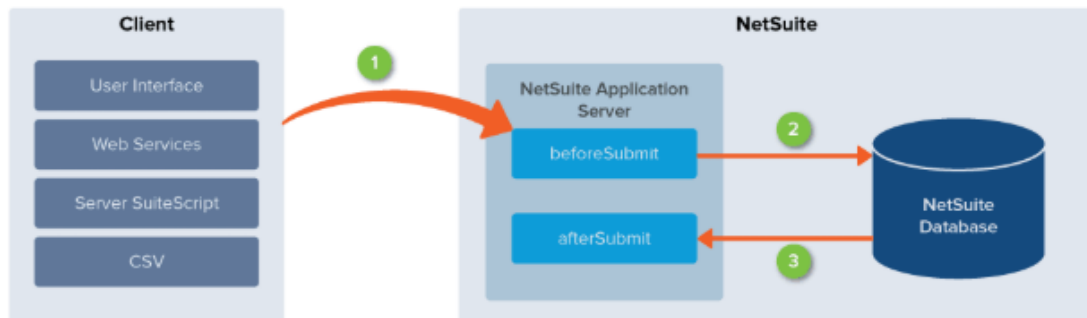


*Nota.* Operación en *beforeLoad* de un User Event. Fuente: (Oracle, 2024)

1. El cliente envía una solicitud para leer los datos del registro. Esta solicitud puede venir de la interfaz de usuario, servicios web SOAP y REST, importación CSV u otro SuiteScript de tipo servidor.
2. El servidor comprueba los permisos.
3. La base de datos carga la información en el servidor de aplicaciones para su procesamiento. Aquí es donde se realiza la operación *beforeLoad*, antes de que los datos vuelvan al cliente.
4. El cliente obtiene los datos del *beforeLoad* validados y procesados.
5. El cliente envía una solicitud de escritura al servidor que puede venir de la interfaz de usuario, servicios web SOAP o REST, ejecución de otro SuiteScript o importación CSV. El servidor comprueba los permisos y procesa los datos comprobando validaciones durante el *beforeSubmit*. Es importante resaltar que, en este punto, los datos aún no se han guardado en la base de datos.
6. Luego de la validación, los datos se guardan en la base de datos, es decir, el registro ya está creado o actualizado en NetSuite.

7. Cuando se ejecuta un afterSubmit, se toma los nuevos datos de la base de datos y los envía al servidor de aplicaciones para más procesamiento. Estas acciones pueden ser el envío de correos electrónicos sobre los nuevos datos, crear registros secundarios en base a los nuevos datos, entre otros.

**Figura 8.** Operación beforeSubmit y afterSubmit de un User Event



*Nota.* Operación en beforeLoad de un User Event. Fuente: (Oracle, 2024)

### 3.2.4.9 Map Reduce Script

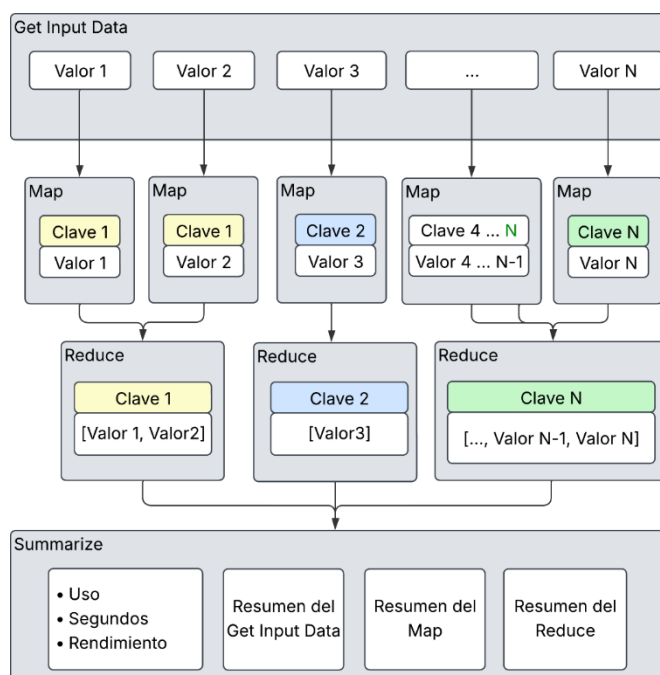
Es un tipo de SuiteScript que está diseñado para manejar grandes cantidades de datos ya que permite dividir la información que se procesa en partes pequeñas e independientes. Tiene la opción de ejecutar en paralelo y se puede elegir el nivel de paralelismo en la implementación del script. Como ha sido mencionado anteriormente, tiene cuatro puntos de entrada que son las etapas que se ejecutan secuencialmente. A continuación, se describe con mayor detalle cada etapa y en la Figura 9 se muestra su ejecución secuencial.

- Get Input Data: Es la primera etapa y es donde se arma todo el conjunto de datos a procesar, usualmente en esta etapa se cargan búsquedas donde los resultados son los que se necesitan agrupar y procesar respectivamente, también se puede realizar una serie de operaciones donde el resultado final sea el conjunto de datos que se quiere procesar. Esta etapa tiene un límite de 10000 unidades de uso de API (*governance*) y un máximo tiempo de ejecución de 60 minutos.
- Map: Esta etapa procesa individualmente cada par clave-valor provisto de la etapa Get Input Data, aquí es donde se pueden realizar ciertos ajustes correspondientes al mapeo de cada elemento. Esta etapa es opcional, por lo que, si no se incluye, será necesario implementar la etapa Reduce; de incluirse, y que el script use la etapa Reduce, puede generar nuevos pares clave-valor que son los datos de entrada de la

etapa Shuffle y luego al Reduce, sino, va directo al Summarize para terminar su ejecución. Por último, esta etapa tiene un máximo de 1000 unidades de uso de API (governance) y un máximo tiempo de ejecución de 5 minutos por cada ejecución individual.

- Shuffle: No es un punto de entrada del script, por lo que no puede ser modificada por código, pero sí es una etapa controlada por el sistema que se ejecuta inmediatamente después del Map y se encarga de organizar los pares clave-valor para la etapa Reduce, finalmente, genera un nuevo conjunto de pares clave-valor donde cada clave es única y cada valor es un arreglo (array).
- Reduce: Esta etapa procesa los pares clave-valor agrupados provistos de la etapa Map, si se implementó, o del Get Input Data. Brinda una mayor capacidad de procesamiento ya que tiene un máximo de 5000 unidades de uso de API (governance) y un máximo de tiempo de ejecución de 15 minutos por cada ejecución individual.
- Summarize: Es la etapa final del script que consolida los resultados de las etapas previas y permite obtener y registrar estadísticas sobre el funcionamiento del script. Además, permite procesar y registrar datos finales como notificar que la ejecución total ha terminado o ejecutar otras tareas. Tiene un máximo de 10000 unidades de uso de API (governance) y un máximo de tiempo de ejecución de 60 minutos.

**Figura 9.** Etapas del Script Map Reduce



*Nota.* Representación del funcionamiento del Script Map Reduce. Fuente: Elaboración propia

#### **3.2.4.10 Jira**

Es una herramienta para la gestión de proyectos que permite planificar, supervisar y realizar entregas de forma segura para diversos tipos de proyectos. Funciona como un repositorio centralizado de información relacionado a la gestión de proyectos y tareas, lo que permite a los equipos desarrollar sus actividades de manera ágil y alineada con la estrategia de la organización (Atlassian, s. f.).

#### **3.2.4.11 Confluence**

Es una herramienta que brinda a los equipos de proyectos, un espacio de trabajo para la gestión de documentos. Permite a los miembros de un equipo crear y compartir información para sus actividades y de esa forma, generar una base de conocimiento accesible a todo el equipo (Atlassian, s. f.).

#### **3.2.4.12 Git**

Es un sistema de control de versiones distribuido que permite a desarrolladores gestionar cambios de código. Facilita el trabajo colaborativo al permitir que cada miembro trabaje en copias independientes (ramas) de un proyecto, para más adelante integrar los cambios y conservar un historial completo de versiones (Git, s. f.).

#### **3.2.4.13 BitBucket**

Es una plataforma para la gestión de repositorios de código y control de versiones basada en Git que permite a los equipos de desarrollo de software revisar código y dar seguimiento de actividades o problemas. Gracias a su integración con herramientas de gestión de proyectos como Jira, facilita la trazabilidad entre el código fuente y las actividades de proyectos (BitBucket, s. f.).

#### **3.2.4.14 Visual Studio Code**

Es un editor de código gratuito desarrollado por Microsoft, el cual cuenta con un entorno ligero y funciones avanzadas de desarrollo para diversos lenguajes de programación. Brinda capacidades para la edición de código, depuración, control de versiones con Git y muchas más gracias a las extensiones que se pueden instalar y facilitan aún más la forma de trabajar (Microsoft, s. f.).

### 3.2.5 Implementación de las áreas, procesos, sistemas y sus buenas prácticas

La implementación de personalizaciones se dividió en 6 tareas, cada tarea enfocada en los procesos particulares y fundamentales que tiene la empresa inmobiliaria que no pudieron ser solventadas en su totalidad con los módulos estándar de NetSuite o porque requerían la automatización de sus procesos. Siguiendo la metodología, cada una de estas tareas fue creada y gestionada en la herramienta Jira, documentada en Confluence y todo lo relacionado al código fue alojado y versionado en un repositorio de BitBucket. En la Tabla 7 se muestra el detalle general de cada tarea.

**Tabla 7.** *Tareas del desarrollo de la gestión automática de contratos de alquiler*

| Tarea en Jira                                                    | Detalles                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tarea 1: Desarrollo del calendario de facturación personalizado. | <p><b>Problema:</b> El módulo <i>Advanced Billing</i> carece de la flexibilidad para generar reglas de facturación que se adaptan al negocio de la empresa inmobiliaria.</p> <p><b>Objetivo:</b> Mostrar el pronóstico de facturación en base a la frecuencia, las fechas de inicio y término del cumplimiento de la facturación.</p> <p><b>Solución:</b></p> <ul style="list-style-type: none"><li>- Crear registros personalizados para las reglas de facturación que brinden la flexibilidad en el pronóstico de la facturación que se debe realizar.</li><li>- Crear un Script de tipo User Event para crear y actualizar un calendario de facturación de cada contrato.</li><li>- Crear un Script Map/Reduce que se ejecute periódicamente en base a las fechas del calendario y realice la creación de la transacción que da origen a la factura.</li></ul> |

|                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Tarea 2: Ejecutar el proceso <i>Assets Sales Disposal</i> usando una solicitud HTTPS (integración).</p> | <p><b>Problema:</b> El proceso del módulo nativo de Gestión de Activos Fijos, FAM por sus siglas en inglés, solo puede ser ejecutado por medio de una interfaz y el cliente necesita poder ejecutar por integración mediante una solicitud HTTP.</p> <p><b>Objetivo:</b> Ejecutar el proceso <i>Assets Sales Disposal</i> mediante una solicitud HTTPS.</p> <p><b>Solución:</b> Crear un Script RESTlet para exponer un <i>endpoint</i> que ejecuta el proceso.</p>                              |
| <p>Tarea 3: Actualizar datos de Fianzas en contrato</p>                                                    | <p><b>Problema:</b> Se solicita una automatización para actualizar datos de la fianza en el contrato.</p> <p><b>Objetivo:</b> Tener control de las fianzas que recibe el inquilino en el contrato.</p> <p><b>Solución:</b> Crear un SuiteScript de tipo User Event con la lógica para actualizar los datos de la Fianza cuando esta se crea en el sistema.</p>                                                                                                                                   |
| <p>Tarea 4: Cálculo de mensualidad en la finalización del contrato</p>                                     | <p><b>Problema:</b> Se solicita la automatización en el manejo de las transacciones al finalizar el contrato.</p> <p><b>Objetivo:</b> Gestionar la última mensualidad correctamente cuando el Portal del alquiler de la empresa inmobiliaria envíe la fecha de la entrega de llaves.</p> <p><b>Solución:</b> Crear Script User Event que se active cuando se actualice el contrato con la fecha de entrega de llaves y se ejecute la creación de las transacciones de la última mensualidad.</p> |
| <p>Tarea 5: Sincronización de Segmentos contables personalizados</p>                                       | <p><b>Problema:</b> Se solicita una automatización en los activos fijos de la empresa.</p> <p><b>Objetivo:</b> Tener actualizado los segmentos personalizados con los activos creados en el módulo nativo de Gestión de Activos Fijos, Fixed Asset Management (FAM) por sus siglas en inglés.</p>                                                                                                                                                                                                |

**Solución:** Crear un SuiteScript de tipo User Event con la lógica para actualizar los segmentos personalizados cada vez que el activo cambie de estado.

Tarea 6: Actualización automática de IPC o IRAV

**Problema:** Se debe realizar una revisión anual en función del IPC establecido por el gobierno.

**Objetivo:** Revisar todos los contratos periódicamente, identificar las rentas y recalcular el importe.

**Solución:** Crear Script Map/Reduce para evaluar todos los contratos, identificar y actualizar la renta de cada contrato con el IPC o IRAV correspondiente.

---

*Nota.* Fuente: Elaboración propia.

Todas las tareas fueron generadas durante la implementación de NetSuite en una empresa inmobiliaria, debido a las carencias detectadas del ERP en ciertos procesos y por las solicitudes de automatización requeridas por el cliente. El equipo de proyecto para las tareas se conformó por los siguientes integrantes, donde el autor del proyecto desempeñó el rol de Software Engineer, NetSuite, encargándose de todas las actividades respondientes a la consultoría técnica para el diseño de las soluciones y al desarrollo de la solución de cada tarea.

- 1 Project Manager
- 1 NetSuite Implementation Consultant
- 1 Software Engineer, NetSuite
- 1 Software Engineer, Deployments

Es importante mencionar ciertas características del cliente y de los elementos de NetSuite que se utilizaron en la solución para tener un mejor entendimiento del contexto de la solución.

El cliente es una empresa del sector inmobiliario que cuenta con sistemas propios encargados de la venta, reserva y firma de contratos de alquiler, por lo que la mayor parte de sus procesos requieren que esté integrado con NetSuite mediante su API. Las tareas presentadas en el trabajo de suficiencia profesional son para los procesos que no pueden

ser solventados por el uso de SuiteTalk que es la herramienta de integración nativa de NetSuite o para complementar los procesos con automatizaciones solicitadas por el cliente.

La representación de los contratos de alquiler en NetSuite son por medio del tipo de transacción *Sales Order* (orden de venta). En NetSuite, las órdenes de venta se ingresan para capturar los artículos y/o servicios que los clientes han comprado y se planea entregar esa compra. Estas órdenes permiten rastrear el estado del pedido del cliente y planear operativamente las actividades relacionadas a la venta que se generó. A partir de estas órdenes se puede generar la transacción Invoice (Factura de venta).

En NetSuite, se cuenta con los *Custom Segments* (Segmentos personalizados), que permiten crear campos de clasificación y definir valores posibles para cada segmento y relacionarlo con un tipo de registro personalizado. Los segmentos personalizados que se crearán en el proyecto son los de Delegación, Promoción, Comunidad Autónoma, Provincia y Activo.

- Delegación: Representa las oficinas distribuidas por las diferentes regiones de España.
- Promoción: Representa una ubicación que contiene varios activos.
- Comunidad Autónoma: Representa las comunidades autónomas de España que contiene activos (pisos, garajes, locales, entre otros).
- Provincia: Representa las regiones provinciales de España y será independiente de la segmentación Delegación.

Otro elemento nativo de NetSuite que se usa en el proyecto son los *Assets* (activos), un activo es cualquier equipo o recurso que necesita amortizarse, o que requiere de un seguro, un arrendamiento, un mantenimiento o detalles personalizados que deben registrarse. Las empresas necesitan un control y seguimiento de los activos. La SuiteApp *Fixed Asset Management* (FAM), o Gestión de Activos fijos por sus siglas en español, brinda una única fuente de información a todo lo que está relacionado con los activos.

Para el proyecto, el cliente maneja los siguientes tipos de activos

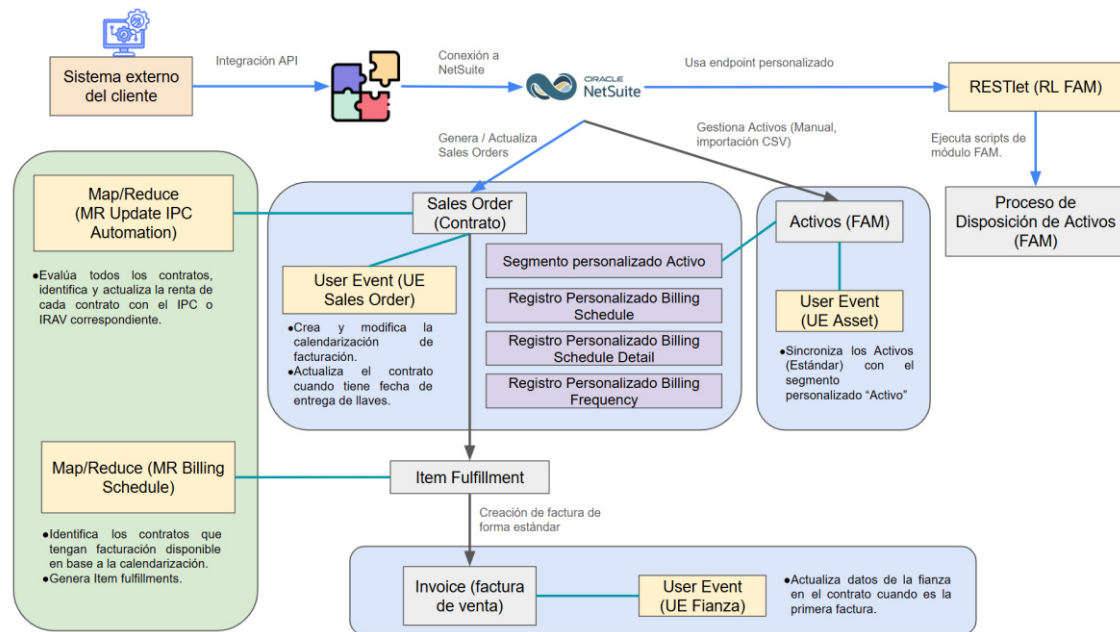
- Suelo
- Nave Industrial
- Garaje Moto
- Chalet Adosado
- Dúplex

- Zona Común
- Chalet Individual
- Garaje
- Local
- Oficina
- Trastero
- Vivienda
- Edificio
- Estación de Servicio
- Finca rústica
- Ático
- Hotel
- Teléf. Antena
- Biciletero

Para la implementación se creará una lista conectada al módulo de Activos Fijos (Fixed Asset Management, FAM) usando el nombre del activo para poder ser seleccionado en el segmento personalizado.

Para tener una visión general, la Figura 10 muestra el diagrama de arquitectura de la solución la cual incluye los componentes nativos del ERP, registros personalizados, scripts creados y cómo se integran en la solución. El sistema del cliente y su conexión por medio de integraciones API con el ERP permite la gestión de contratos (creación, modificación y término de contrato) y ejecutar procesos del módulo de activos fijos. Los scripts User Event son implementados en cada componente de NetSuite y son ejecutados cuando se realiza algún evento como creación o modificación. El RESTlet script expone un endpoint que es usado en la integración para ejecutar el proceso de disposición de activos y los Map/Reduce scripts se ejecutan periódicamente usando la información de las transacciones o registros personalizados que se mantienen actualizados por los User Event.

**Figura 10. Diagrama de arquitectura de solución**



*Nota.* Fuente: Elaboración propia

## Tarea 1: Desarrollo de calendario de facturación personalizado.

### Análisis de requerimientos

Debido a que un contrato puede incluir la gestión de diversos activos, con distinta periodicidad y duración en el cobro de la mensualidad, el consultor de implementaciones detecta que el módulo nativo de NetSuite *Advanced Billing* no cuenta con una configuración de periodicidad y duración del contrato que maneja el cliente y eso imposibilita la generación automática de facturas con los datos esperados y necesita de alguna forma generar las facturas correctamente.

Luego de coordinar y proponer soluciones alternas, junto al autor y otros consultores o arquitectos, se llegó a determinar la viabilidad de la siguiente solución. Otra forma de crear un *Invoice* es mediante una transacción intermedia entre la *Sales Order* y el *Invoice* que es el *Item Fulfillment* (Ejecución de orden de artículo) que se usa para determinar que la totalidad o parcialidad de una orden de venta está lista para facturarse. Esto permitirá representar la mensualidad correcta que se necesita facturar. Luego de determinar la solución, el consultor genera el documento de especificación con todos los requerimientos que se detallan a continuación en la Tabla 8.

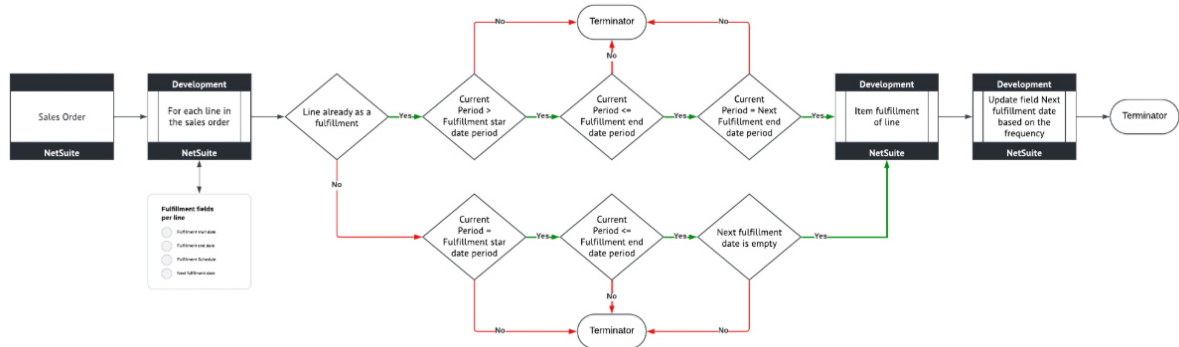
**Tabla 8.** Documento de especificación para la tarea 1

| Tarea 1 – Documento de especificación |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                               | Se requiere de una personalización para crear <i>Item Fulfillments</i> basándose en una fecha de inicio de periodo para facturar mensualidades y el tipo de frecuencia. Además, se necesita tener un pronóstico de la facturación de mensualidades y así validar que ya fue procesado.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Supuestos                             | El cliente estima tener unas 7000 <i>Sales Orders</i> que pertenecen a este proceso                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Requerimientos técnicos               | <ul style="list-style-type: none"><li>- Crear campos personalizados a nivel línea de transacción para la fecha de inicio de la mensualidad y tipo de frecuencia.</li><li>- Crear campo personalizado a nivel de línea de transacción para seleccionar la periodicidad de la mensualidad.</li><li>- Crear registros personalizados que permitan la configuración flexible de la periodicidad de las mensualidades.</li><li>- Cada vez que el contrato sea creado o actualizado, se debe generar o actualizar el pronóstico de facturación.</li><li>- Crear un Script que diariamente detecte si alguna mensualidad debe generarse, y de ser así, crear el <i>Item Fulfillment</i> respectivo para que esté listo a facturarse.</li></ul> |

*Nota.* Fuente: Elaboración propia.

A continuación, en la Figura 11 se muestra el flujo del proceso de la solución.

**Figura 11.** Flujo del proceso para la tarea 1



*Nota.* Fuente: Elaboración propia.

## Desarrollo

Para esta tarea se crearon los siguientes componentes en base a los requerimientos técnicos y a lo que se determinó que fue necesario para llevar a cabo la solución de la mejor manera y usando buenas prácticas.

### Registro Personalizado Billing Schedule

Este registro representa a cada elemento que se verá en el pronóstico de facturación que incluye todas las líneas en la Sales Order que deben facturarse en determinada fecha. Está compuesto por 4 campos que se detallan a continuación y en la Figura 12 se muestra el registro creado en NetSuite.

#### Start Date

Campo de tipo fecha donde se almacena la fecha de facturación.

#### Amount

Campo tipo moneda donde se almacena el monto total de la mensualidad a facturar. Este monto es la suma de las mensualidades de todas las líneas que se facturan en una misma fecha.

#### Sales Order:

Campo de tipo selección de Transacción donde se almacena la Sales Order (contrato) al cual pertenece esta facturación. Este campo tiene la configuración *Record is Parent* que enlaza al registro personalizado con la transacción (el registro es un “hijo” de la transacción) lo que permite ver todos los registros relacionados en la transacción. Esto hará que se muestre el listado de todas las facturaciones (pronóstico de facturación) en el formulario de la Sales Order, es decir, del contrato.

## Item Fulfillment

Campo de tipo lista de Transacción donde se almacena el *Item Fulfillment* que es generado por el Script que detecta cuando se debe generar la mensualidad.

**Figura 12.** *Custom Record Billing Schedule*

The screenshot shows the NetSuite 'Custom Record Type' configuration page for 'Billing Schedule'. The page includes a search bar at the top, navigation tabs, and a 'Save' button. The configuration details are as follows:

- NAME:** Billing Schedule
- ID:** customrecord\_ \_billing\_schedule
- INTERNAL ID:** 1170
- ORIGINATING CUSTOM SEGMENT:**
- OWNER:** (dropdown menu)
- DESCRIPTION:** (text area)
- ACCESS TYPE:** Require Custom Reco... Entries Permission
- ALLOW UI ACCESS:** ☒
- ALLOW MOBILE ACCESS:** ☐
- ALLOW ATTACHMENTS:** ☒
- SHOW NOTES:** ☒
- ENABLE MAIL MERGE:** ☐
- RECORDS ARE ORDERED:** ☐
- SHOW REMOVE LINK:** ☒
- ALLOW CHILD RECORD EDITING:** ☒
- ALLOW DELETE:** ☒
- ALLOW QUICK SEARCH:** ☐
- ALLOW QUICK ADD:** ☒
- ENABLE SYSTEM NOTES:** ☒
- INCLUDE IN GLOBAL SEARCH:** ☒
- INCLUDE IN SEARCH MENU:** ☒
- ENABLE OPTIMISTIC LOCKING:** ☒
- ENABLE INLINE EDITING:** ☒
- ENABLE NAME TRANSLATION:** ☐
- HIERARCHY:** ☐
- INACTIVE:** ☐

The 'Fields' section shows a table with the following data:

| DESCRIPTION      | ID                            | TYPE        | LIST RECORD | TAB | SHOW IN LIST |
|------------------|-------------------------------|-------------|-------------|-----|--------------|
| Start Date       | customrecord_ _btl_start_date | Date        |             |     | Yes          |
| Amount           | customrecord_ _btl_amount     | Currency    |             |     | Yes          |
| Sales Order      | customrecord_ _btl_so         | List/Record | Transaction |     | Yes          |
| Item Fulfillment | customrecord_ _btl_if         | List/Record | Transaction |     | Yes          |

*Nota.* Fuente: Elaboración propia.

## Registro Personalizado Billing Schedule Detail

Este registro representa el detalle de cada línea del contrato que forma parte de la mensualidad total. Está compuesto por los campos que se detallan a continuación y en la Figura 13 se muestra el registro creado en NetSuite.

### Item

Campo de tipo lista de Item donde se almacena el artículo de la línea de un contrato

### Amount

Campo de tipo moneda donde se almacena el monto de la línea de un contrato.

### Line

Campo de tipo número entero donde se almacena el número de la línea de un contrato.

### Line Unique Key

Campo de tipo texto libre donde se almacena el identificador único de cada línea de un contrato.

### Billing Schedule

Campo de tipo lista de Registro Personalizado Billing Schedule. Este almacena el elemento del cual es parte este detalle. Este campo tiene la configuración *Record is*

Parent que lo enlaza al Registro Personalizado Billing Schedule como registro “padre”.

Figura 13. Custom Record Billing Schedule Detail

Oracle NetSuite

Custom Record Type  
Billing Schedule Detail

Save Cancel Change ID Actions

NAME \*  
Billing Schedule Detail

ID  
customrecord\_billing\_schedule\_detail

INTERVAL ID  
1169

ORIGINATING CUSTOM SEGMENT

OWNER

DESCRIPTION

ACCESS TYPE  
Require Custom Reco..Errors Permission

ALLOW UI ACCESS  
ALLOW MOBILE ACCESS  
ALLOW ATTACHMENTS  
SHOW NOTES  
ENABLE MAIL MERGE  
RECORDS ARE ORDERED  
SHOW REMOVE LINK  
ALLOW CHILD RECORD EDITING  
ALLOW DELETE  
ALLOW QUICK SEARCH

ALLOW QUICK ADD  
ENABLE SYSTEM NOTES  
INCLUDE IN GLOBAL SEARCH  
INCLUDE IN SEARCH MENU  
ENABLE OPTIMISTIC LOCKING  
ENABLE INLINE EDITING  
ENABLE NAME TRANSLATION  
HIERARCHY  
INACTIVE

INCLUDE NAME FIELD  
SHOW ID  
SHOW CREATION DATE ON RECORD ON LIST  
SHOW LAST MODIFIED ON RECORD ON LIST  
SHOW OWNERS ON RECORD ON LIST ALLOW CHANGE

Fields Sublists Icon Numbering Forms Online Forms Permissions Links Managers Translation Child Records Parent Records History System Notes

SHOW INACTIVES

| DESCRIPTION      | ID                                   | TYPE           | LIST RECORD        | TAB | SHOW IN LIST |
|------------------|--------------------------------------|----------------|--------------------|-----|--------------|
| Item             | customrecord_billing_item            | List/Record    | Item               |     | Yes          |
| Amount           | customrecord_billing_amount          | Currency       |                    |     | Yes          |
| Line             | customrecord_billing_line            | Integer Number |                    |     | Yes          |
| Line Unique Key  | customrecord_billing_line_unique_key | Free Form Text |                    |     | Yes          |
| Billing Schedule | customrecord_billing_schedule        | List/Record    | - Billing Schedule |     | No           |

Nota. Fuente: Elaboración propia.

### Registro Personalizado Billing Frequency

Este registro representa la frecuencia en meses que usa el cliente. Está compuesto por un campo. En la Figura 14 se muestra el registro en NetSuite y en la Figura 15 la lista de frecuencias que usa el cliente.

#### Frequency (in months)

Campo de tipo número entero donde se almacena la cantidad de meses que tiene la frecuencia definida por el cliente.

**Figura 14.** *Custom Record Billing Frequency*

**Custom Record Type**

- | Billing Frequency

**Saves** Cancel Change ID Actions

NAME \*  
Billing Frequency

ID  
customerrecord\_\_billing\_frequency

INTERNAL ID  
1168

ORIGINATING CUSTOM SEGMENT

OWNER

DESCRIPTION

☒ INCLUDE NAME FIELD  
☒ SHOW ID

SHOW CREATION DATE ☐ ON RECORD ☐ ON LIST

SHOW LAST MODIFIED ☐ ON RECORD ☐ ON LIST

SHOW OWNER ☐ ON RECORD ☐ ON LIST ☐ ALLOW CHANGE

ACCESS TYPE  
Require Custom Rec.Entries Permission

- ☒ ALLOW UI ACCESS
- ☐ ALLOW MOBILE ACCESS
- ☒ ALLOW ATTACHMENTS
- ☒ SHOW NOTES
- ☐ ENABLE MAIL MERGE
- ☐ RECORDS ARE ORDERED
- ☒ SHOW REMOVE LINK ☐ ALLOW CHILD RECORD EDITING ☐ ALLOW DELETE
- ☐ ALLOW QUICK SEARCH

- ☒ ALLOW QUICK ADD
- ☒ ENABLE SYSTEM NOTES
- ☒ INCLUDE IN GLOBAL SEARCH
- ☒ INCLUDE IN SEARCH MENU
- ☒ ENABLE OPTIMISTIC LOCKING
- ☒ ENABLE INLINE EDITING
- ☐ ENABLE NAME TRANSLATION
- ☐ HIERARCHY
- ☐ INACTIVE

**Fields** Subtabs Sublists Icon Numbering Firms Online Forms Permissions Links Managers Translation Child Records Parent Records History System Notes

☐ SHOW INACTIVES

| New Field | Move To Top | Move To Bottom | ID                    | TYPE                   | LIST/RECORD    | TAB | SHOW IN LIST |
|-----------|-------------|----------------|-----------------------|------------------------|----------------|-----|--------------|
|           |             |                | Frequency (in months) | custorecord__frequency | Integer Number |     | Yes          |

*Nota.* Fuente: Elaboración propia.

**Figura 15. Lista de frecuencias.**

The screenshot displays the NetSuite Billing Frequency List interface. At the top, there is a navigation bar with various menu items like Activities, Localization, Payments, Transactions, Lists, Reports, Analytics, Documents, Setup, Customization, Commerce, Administration and Controls, Configuration Management, Fixed Assets, SR, SuiteApps, and Support. Below the navigation bar, the page title is "Billing Frequency List". There are tabs for "Default" and "Customize View", and a "New" button. A "FILTERS" section shows "STYLE" set to "Normal". Below the filters, there is a table with columns "ID", "Interval", and "ID". The "Interval" column is highlighted with a red box and a red arrow pointing to it. The "ID" column is also highlighted with a red box and a red arrow pointing to it. The table contains 5 rows of data. The "QUICK SORT" dropdown is set to "FREQUENCY (BY MONTHS)".

| ID | Interval       | ID |
|----|----------------|----|
| 1  | Annual (JR)    | 1  |
| 2  | Bimonthly (JR) | 2  |
| 6  | Quarterly (JR) | 6  |
| 3  | Monthly (JR)   | 3  |
| 4  | Semiannual     | 4  |
| 5  | Triannual      | 5  |

*Nota.* Fuente: Elaboración propia.

**Fulfillment Start Date**

Es un campo de línea de transacción de tipo fecha donde se selecciona la fecha de inicio de la mensualidad. En la Figura 16 se muestra la configuración del campo en NetSuite.

**Figura 16.** Configuración de campo *Fulfillment Start Date*

The screenshot shows the NetSuite interface for configuring a Transaction Line Field. The field is named 'Fulfillment Start Date' with ID 'custcol\_...fulfillment\_start\_date' and INTERNAL ID '4443'. The TYPE is set to 'Date'. The 'Applies To' section includes checkboxes for EXPENSE, PURCHASE ITEM, SALE ITEM (checked), OPPORTUNITY ITEM, JOURNAL, EXPENSE REPORT, and ITEM RECEIPT. The 'Display' section includes checkboxes for ITEM FULFILLMENT, VIEW FROM ORDER ONLY, PRINT ON STANDARD FORMS, PRINT ON PICKING TICKET, PRINT ON PACKING SLIP, PRINT ON RETURN FORM, and STORE WITH ITEM GROUPS. The 'Validation & Defaulting' section includes checkboxes for GROUP ON INVOICES and APPLY TO KIT/ASSEMBLY COMPONENTS. The 'Sourcing & Filtering' section includes checkboxes for CUSTOM TRANSACTIONS, FAM Special Depreciation Entry, Fixed Asset Depreciation Entry, Fixed Asset Disposal Entry, Fixed Asset Revaluation Entry, and Fixed Asset Transfer Entry. The 'Access' section includes checkboxes for INACTIVE and USE ENCRYPTED FORMAT.

*Nota.* Fuente: Elaboración propia.

## Fulfillment Schedule

Es un campo de línea de transacción de tipo lista de Registro Personalizado Billing Frequency donde se indica el tipo de frecuencia. En la Figura 17 se muestra la configuración del campo en NetSuite.

**Figura 17.** Configuración de campo *Fulfillment Schedule*

The screenshot shows the NetSuite interface for configuring a Transaction Line Field. The field is named 'Fulfillment Schedule' with ID 'custcol\_...fulfillment\_schedule' and INTERNAL ID '4442'. The TYPE is set to 'List/Record'. The 'Applies To' section includes checkboxes for EXPENSE, PURCHASE ITEM, SALE ITEM (checked), OPPORTUNITY ITEM, JOURNAL, EXPENSE REPORT, and ITEM RECEIPT. The 'Display' section includes checkboxes for ITEM FULFILLMENT, VIEW FROM ORDER ONLY, PRINT ON STANDARD FORMS, PRINT ON PICKING TICKET, PRINT ON PACKING SLIP, PRINT ON RETURN FORM, and STORE WITH ITEM GROUPS. The 'Validation & Defaulting' section includes checkboxes for GROUP ON INVOICES and APPLY TO KIT/ASSEMBLY COMPONENTS. The 'Sourcing & Filtering' section includes checkboxes for CUSTOM TRANSACTIONS, FAM Special Depreciation Entry, Fixed Asset Depreciation Entry, Fixed Asset Disposal Entry, Fixed Asset Revaluation Entry, and Fixed Asset Transfer Entry. The 'Access' section includes checkboxes for INACTIVE and USE ENCRYPTED FORMAT.

*Nota.* Fuente: Elaboración propia.

Los componentes que se detallan a continuación corresponden a todo lo relacionado a SuiteScript y librerías. Para no sobrecargar cada SuiteScript con toda la lógica de programación, se hace el uso de librerías que tienen las funciones que se necesitan y el SuiteScript solo las invoque y haga uso de ellas, adicionalmente, esto permite desacoplar el código y que diferentes SuiteScripts hagan uso de las funciones de las librerías.

## Librería lib\_billing.js

Esta librería fue creada para toda la lógica relacionada al proceso personalizado de facturación que se realizó en la tarea. Expone las funciones que son usadas por los SuiteScripts de User Event y Map Reduce que se crearon para la tarea. En la figura 18 se muestra parte del código JavaScript donde se indica el tipo de versión de API que se usa y las funciones expuestas por la librería.

**Figura 18.** Código JavaScript de librería lib\_billing.js

```
1  /**
2   * Task ID          Date          Author          Remarks
3   *                  28 Nov 2024    Vicente Mori <vicente.mori>
4   *                  >              - Custom Billing schedule
5   * @APIVersion 2.1
6   * @NModuleScope public
7   */
8
9  define(['N/record', 'N/search', 'N/format'], (record, search, format) => {
10
175 >   const createBillingSchedule = (params) => { ...
176
177 >   const getCurrentBillingSchedules = (soId) => { ...
248
249   };
250
251   };
252
253   return { createBillingSchedule, loadSearchMr, processFulfillment, removeBillingSchedules };
254 });
```

Se indica la versión de SuiteScript a usar

Se definen las API de la plataforma a usar

Funciones que se exponen para ser usadas por los SuiteScripts

*Nota.* Fuente: Elaboración propia.

## UE Sales Order

Este es el Script User Event que fue desarrollado para que incluya todas las personalizaciones que sean necesarias en los contratos. Debido a que los contratos son creados por integración, debemos usar el punto de entrada *AfterSubmit* para que nuestra personalización se ejecute una vez que la integración haya creado o actualizado el contrato en NetSuite.

Al crear el archivo JavaScript se debe indicar el tipo de SuiteScript que se quiere crear en el JSDoc tag `@NScriptType` y retornar las funciones relacionadas a los puntos de entrada, en este caso solo el *AfterSubmit* fue necesario. En la Figura 19 se puede visualizar el contenido del archivo JavaScript correspondiente al User Event.

**Figura 19.** Código JavaScript de UserEvent UE Sales Order

```
1  /**
2   * @APIVersion 2.1
3   * @NScriptType UserEventScript
4   *
5   * Task Date Author Remarks
6   * 28 Nov 2024 Vicente Mori <vicente.mori> - Custom Billing schedule
7
8  */
9  define(['N/runtime', './_lib_billing'], (runtime, LibBilling) => {
10     const afterSubmit = (context) => {
11         const FN = 'afterSubmit';
12         try {
13             let { newRecord } = { ...context };
14             const scriptObj = runtime.getCurrentScript();
15             const parameters = getParameters(scriptObj);
16             if (['salesorder'].includes(newRecord.type)) {
17                 LibBilling.createBillingSchedule({ context, parameters });
18             }
19         } catch (e) { ... }
20     };
21     return {
22         afterSubmit,
23     };
24 });
```

*Nota.* Fuente: Elaboración propia.

### MR Billing Schedule

Este es el SuiteScript Map/Reduce que fue desarrollado para periódicamente identificar las mensualidades de los contratos y crear los *Item Fulfillments* correspondientes. Fue diseñado para que cargue los resultados de una *Saved Search* (búsqueda guardada) que retorna las *Sales Orders*(contratos) que cumplen con las condiciones establecidas para ser validadas y procesadas. Luego, por cada contrato retornado, validar si en su pronóstico de facturación se encuentra algún registro que tenga la fecha correspondiente al día actual y de ser así, generar el *Item Fulfillment*.

Para dar un poco más de flexibilidad a la solución, se creó unos parámetros en el Script para elegir la fuente de información que quiera ser procesada, la fecha de procesamiento (en caso se requiera procesar alguna fecha distinta a la actual), y parámetros que contienen el nombre de la función a ejecutar para cada punto de entrada del Script. Se diseñó de esta manera para permitir crear *Item Fulfillments* y también eliminar los *registros personalizados de Billing Schedule* en caso de ser necesario, simplemente creando diferentes implementaciones, pero usando el mismo Script Map/Reduce.

Debido a que este requerimiento implica la carga de contratos para luego validarlos y generar una transacción por cada resultado, solo fue necesario definir los puntos de entrada *GetInputData*, *Map* y *Summarize*. A continuación, En la Figura 20 se muestra el código JavaScript del Map/Reduce.

**Figura 20.** Código JavaScript de Map/Reduce MR Billing Schedule



*Nota.* Fuente: Elaboración propia.

A modo de resumen, las actividades que se realizan en cada etapa del Script son las siguientes:

- Get Input Data: En esta etapa se cargan los resultados de la búsqueda especificada en el parámetro Search.
- Map: En esta etapa se valida cada resultado, dependiendo de las funciones colocadas en los parámetros, se puede generar el *Item Fulfillment* o eliminar los *registros personalizados de Billing Schedule* de cada contrato.
- Summarize: En esta no se realiza alguna tarea adicional a la de mostrar el rendimiento de las etapas ejecutadas.

En la Figura 21 se muestra el registro del Script Map/Reduce en NetSuite.

Figura 21. Script Map/Reduce en NetSuite

ORACLE NetSuite

Search

Activities Localización Española Payments Transactions Lists Reports Analytics Documents Setup Customization Commerce Administration and Controls

Script

Save Cancel Change ID Actions

TYPE: Map/Reduce El sistema identifica el tipo de Script del archivo JavaScript.

NAME: MR Billing Schedule

ID: customscript\_ \_mr\_billing\_schedule

API VERSION: 2.1 El sistema identifica la versión de API del archivo JavaScript.

DESCRIPTION:

OWNER: customscript\_ \_mr\_billing\_schedule

☐ INACTIVE

Scripts Parameters Unhandled Errors Execution Log Deployments System Notes

SCRIPT FILE: \_mr\_billing\_schedule.js

☒ GET INPUT DATA

☒ MAP El sistema identifica los puntos de entrada del archivo JavaScript.

☐ REDUCE

☒ SUMMARIZE

Nota. Fuente: Elaboración propia.

En la Figura 22 se muestra los parámetros del Script Map/Reduce.

Figura 22. Parámetros del Script Map/Reduce

ORACLE NetSuite

Search

Activities Localización Española Payments Transactions Lists Reports Analytics Documents Setup Customization Commerce Administration and Controls Configuratio

Script

Save Cancel Change ID Actions

TYPE: Map/Reduce

NAME: MR Billing Schedule

ID: customscript\_ \_mr\_billing\_schedule

API VERSION: 2.1

DESCRIPTION:

OWNER: customscript\_ \_mr\_billing\_schedule

☐ INACTIVE

Scripts Parameters Unhandled Errors Execution Log Deployments System Notes

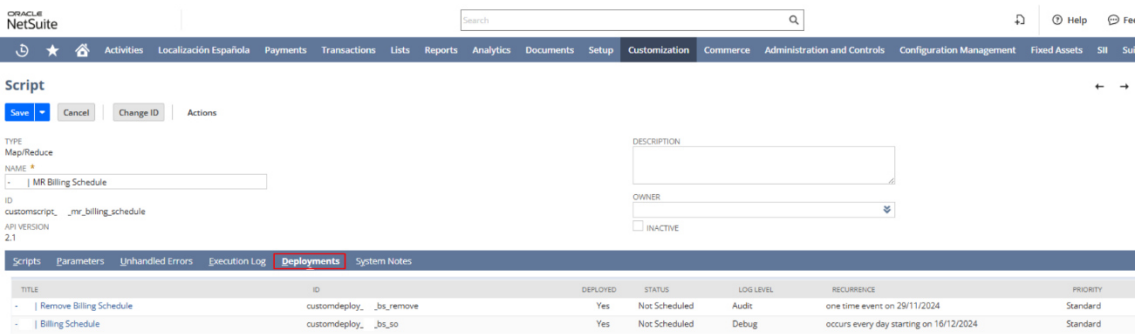
New Parameter

| DESCRIPTION             | ID                           | TYPE           | LIST/RECORD  |
|-------------------------|------------------------------|----------------|--------------|
| SEARCH                  | custscript_ _bs_search       | List/Record    | Saved Search |
| GET INPUT DATA FUNCTION | custscript_ _bs_get_fn       | Free-Form Text |              |
| MAP FUNCTION            | custscript_ _bs_map_fn       | Free-Form Text |              |
| FULFILLMENT DATE        | custscript_ _bs_fulfill_date | Date           |              |

Nota. Fuente: Elaboración propia.

En la Figura 23 se muestra las implementaciones del Script Map/Reduce en NetSuite.

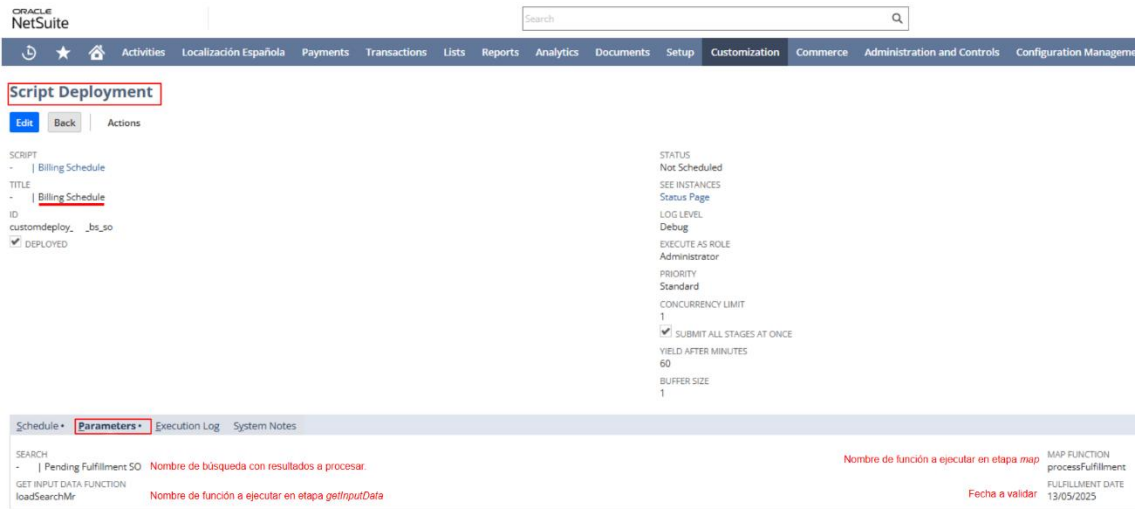
**Figura 23.** *Implementaciones del Script Map/Reduce en NetSuite*



*Nota.* Fuente: Elaboración propia.

En la Figura 24 se muestra el detalle de la implementación que genera *Item Fulfillments*

**Figura 24.** *Implementación que genera Item Fulfillments*



*Nota.* Fuente: Elaboración propia.

En la Figura 25 se muestra el detalle de la implementación que elimina los *registros personalizados de Billing Schedule*.

**Figura 25.** Implementación que elimina los registros personalizados Billing Schedule

Oracle NetSuite

Script Deployment

Script: Billing Schedule

Title: Remove Billing Schedule

ID: customdeploy...\_bs\_remove

DEPLOYED

Parameters

SEARCH: loadSearchMr

DELETE: Nombre de búsqueda con resultados a procesar.

MAP FUNCTION: removeBillingSchedules

EXECUTE AS ROLE: Administrator

PRIORITY: Standard

CONCURRENCY LIMIT: 1

SUBMIT ALL STAGES AT ONCE: ☒

YIELD AFTER MINUTES: 60

BUFFER SIZE: 1

*Nota.* Fuente: Elaboración propia.

Con los scripts desarrollados e implementados, los contratos que cuenten con las frecuencias de facturación personalizadas como se muestra en la Figura 26, cuentan con el pronóstico de facturación actualizado de todas sus líneas, y es un reporte que se encuentra adjunto en la transacción como se muestra en la Figura 27.

**Figura 26.** Líneas de un contrato con frecuencias personalizadas.

| Shipping                                 | Billing      | Accounting                   | Relationships | Communication | Related Records | System Information | Custom    | Zip Reporting | Spain SI Reporting | Tag Reporting | EET       | Impuestos España | Fiscal | Printing History |                            |                      |                     |          |                    |                  |
|------------------------------------------|--------------|------------------------------|---------------|---------------|-----------------|--------------------|-----------|---------------|--------------------|---------------|-----------|------------------|--------|------------------|----------------------------|----------------------|---------------------|----------|--------------------|------------------|
| Items 38,750.00 • Withholding Tax 0.00 • |              |                              |               |               |                 |                    |           |               |                    |               |           |                  |        |                  |                            |                      |                     |          |                    |                  |
| ITEM                                     | DISPLAY NAME | DESCRIPTION                  | QUANTITY      | FULFILLED     | INVOICED        | RATE               | AMOUNT    | OPTIONS       | TAX CODE           | TAX RATE      | GROSS AMT | TAX AMT          | ACTIVO | DEPARTMENT       | DELEGACIÓN                 | DELEGACIÓN/PROMOCIÓN | CLASS               | LOCATION | BILLING START DATE | BILLING SCHEDULE |
| Test Renta IF                            |              |                              | 1             | 0             | 0               | 1,000.00           | 1,000.00  |               | IVA EX             | 0.0%          | 1,000.00  | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2024         | Trimestral (3)   |
| Test Renta IF                            |              | Gastos Trimestrales          | 4             | 0             | 0               | 100.00             | 400.00    |               | IVA EX             | 0.0%          | 400.00    | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2024         | Trimestral (3)   |
| Test Promesa Mensualidad IF              |              |                              | 1             | 0             | 0               | 750.00             | 750.00    |               | IVA EX             | 0.0%          | 750.00    | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2024         | Mensual (3)      |
| Test Renta IF                            |              | Año 1                        | 12            | 0             | 0               | 1,000.00           | 12,000.00 |               | IVA EX             | 0.0%          | 12,000.00 | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2024         | Mensual (3)      |
| Test Renta IF                            |              | Año 2                        | 12            | 0             | 0               | 1,000.00           | 12,000.00 |               | IVA EX             | 0.0%          | 12,000.00 | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2025         | Mensual (3)      |
| Test Renta IF                            |              | Año 3                        | 12            | 0             | 0               | 1,000.00           | 12,000.00 |               | IVA EX             | 0.0%          | 12,000.00 | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/12/2026         | Mensual (3)      |
| Test Renta IF                            |              | Gastos Anuales               | 2             | 0             | 0               | 100.00             | 200.00    |               | IVA EX             | 0.0%          | 200.00    | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/06/2025         | Annual (3)       |
| Test Renta IF                            |              | Gastos Semestrales           | 3             | 0             | 0               | 100.00             | 300.00    |               | IVA EX             | 0.0%          | 300.00    | 0.00             |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/06/2025         | Semestral (3)    |
| Test Renta IF                            |              | Gastos Semestrales to remove | 1             | 0             | 0               | 100.00             | 100.00    | Yes           | IVA21 ES           | 21.0%         | 121.00    | 21.00            |        | Finanzas         | Alicante - Torres Gran Via | INGRESOS ALQUILERES  | INGRESOS ALQUILERES |          | 01/06/2025         | Semestral (3)    |

*Nota.* Fuente: Elaboración propia.

Figura 27. Pronóstico de facturación generado para el contrato.

★

📄

Activities

Localización Española

Payments

Transactions

Reports

Analytics

Documents

Setup

Customization

Commerce

Administration

Controls

Configuration Management

Fixed Assets

SR

Subscriptions

Support

28/11/2024

LAST MODIFIED DATE

28/11/2024

SALES REP

SALES EFFECTIVE DATE

28/11/2024

PAUSED INTERCOMPANY TRANSACTION

INTERCOMPANY STATUS

OPPORTUNITY

RATE

PARTNER

☐ EXCLUDE FROM ELECTRONIC BANK PAYMENTS PROCESSING

NUMBER IN FACTURA INDOE

NUMBER IN FACTURA FINAL

END DATE

DELIVERY DATE

☐ KEYS DELIVERED

Items

Shipment

Billing

Accounting

Relationships

Communication

Related Records

System Information

Custom

Tax Reporting

Spain SR Reporting

Tax Reporting

ET

Exposición Capital

Fiscalia

Printing History

Billing Information

BILLING SCHEDULE

PAYMENT METHOD

BILL TO

\*BSC Text

Spin Map

2

| Billing Schedule

| Payment Events

| Schedule

VIEW

Default View

▼

▼

BILLING SCHEDULE

New

| Billing Schedule

| Attach

| Customer View

| EDIT | ID  | START DATE | AMOUNT   | ITEM IN CURRENCY | REMOVE |
|------|-----|------------|----------|------------------|--------|
| EDIT | 256 | 27/11/2024 | 1,750.00 |                  | Remove |
| EDIT | 257 | 01/12/2024 | 1,100.00 |                  | Remove |
| EDIT | 261 | 01/01/2025 | 1,000.00 |                  | Remove |
| EDIT | 262 | 01/02/2025 | 1,000.00 |                  | Remove |
| EDIT | 263 | 01/03/2025 | 1,100.00 |                  | Remove |
| EDIT | 263 | 01/04/2025 | 1,000.00 |                  | Remove |
| EDIT | 264 | 01/05/2025 | 1,000.00 |                  | Remove |
| EDIT | 265 | 01/06/2025 | 1,200.00 |                  | Remove |
| EDIT | 265 | 06/06/2025 | 0.00     |                  | Remove |
| EDIT | 265 | 01/07/2025 | 1,000.00 |                  | Remove |
| EDIT | 266 | 01/08/2025 | 1,000.00 |                  | Remove |
| EDIT | 266 | 01/09/2025 | 1,100.00 |                  | Remove |
| EDIT | 267 | 01/10/2025 | 1,000.00 |                  | Remove |
| EDIT | 268 | 01/11/2025 | 1,000.00 |                  | Remove |
| EDIT | 269 | 01/12/2025 | 1,100.00 |                  | Remove |
| EDIT | 270 | 01/01/2026 | 1,000.00 |                  | Remove |
| EDIT | 271 | 01/02/2026 | 1,000.00 |                  | Remove |
| EDIT | 272 | 01/03/2026 | 1,000.00 |                  | Remove |
| EDIT | 273 | 01/04/2026 | 1,000.00 |                  | Remove |
| EDIT | 274 | 01/05/2026 | 1,000.00 |                  | Remove |
| EDIT | 275 | 01/06/2026 | 1,200.00 |                  | Remove |

1 to 25 of 30

Nota. Fuente: Elaboración propia.

## **Tarea 2: Ejecutar el proceso Assets Sales Disposal usando una solicitud HTTPS (integración).**

### **Análisis de requerimientos**

El consultor funcional junto al equipo de integraciones detectó que no es posible realizar una conexión directa para la activación del proceso del módulo de activos fijos de NetSuite.

Luego de coordinar y realizar una investigación por parte del autor. Se encontró que es posible ejecutar el proceso requerido por script, pero es necesario crear un registro personalizado del módulo FAM llamado *FAM Process*, el cual debe tener una información en formato JSON por lo que se llegó a determinar la viabilidad de la siguiente solución. Crear un SuiteScript tipo RESTlet que exponga un *endpoint*, y reciba un formato tipo JSON que contiene los datos que pueden ser usados por el módulo de activos fijos y ejecutar el proceso del módulo de activos fijos.

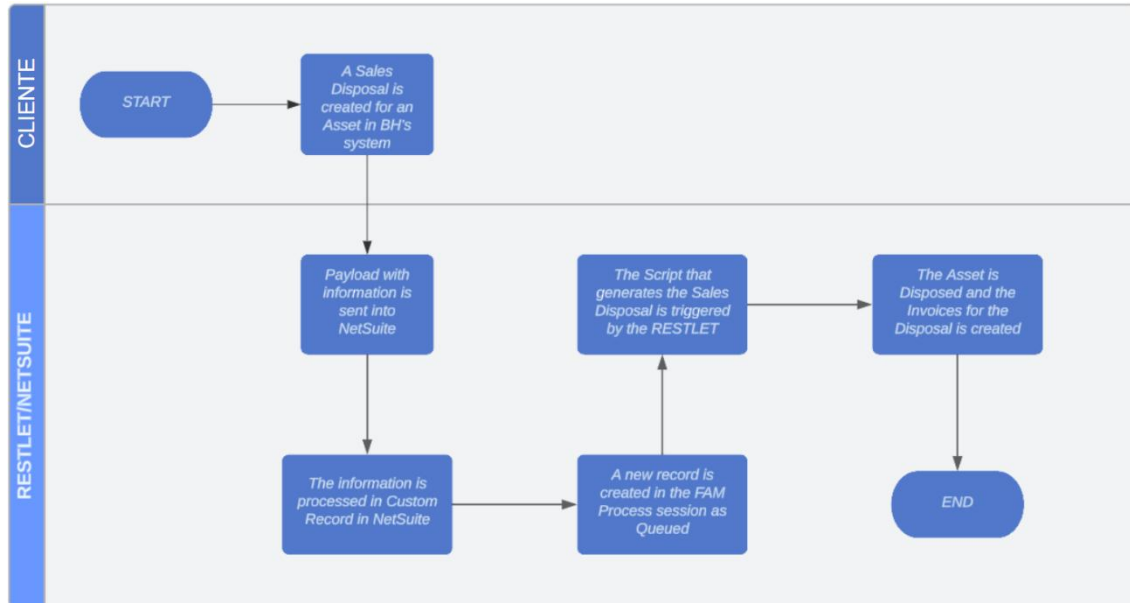
Luego de determinar la solución, el consultor genera el documento de especificación con todos los requerimientos que se detallan a continuación en la Tabla 9.

**Tabla 9.** Documento de especificación para la tarea 2

| <b>Tarea 2– Documento de especificación</b> |                                                                                                                                                                                                                                                                                       |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                                     | Se requiere de una personalización para crear una conexión directa con NetSuite para ejecutar el proceso Assets Sales Disposal del módulo nativo FAM de NetSuite                                                                                                                      |
| Supuestos                                   | <p>El formato JSON es personalizado por el RESTlet y será provisto al cliente.</p> <p>El proceso Assets Sales Disposal se ejecutará mediante un proceso alternativo en lugar de la solución manual que existe en el módulo FAM de NetSuite.</p>                                       |
| Requerimientos técnicos                     | <ul style="list-style-type: none"><li>- Crear un Script RESTlet que use la información en formato JSON recibida, realice la creación de un registro personalizado de tipo <i>FAM Process</i> y active la ejecución del SuiteScript tipo Schedule del módulo FAM de NetSuite</li></ul> |

A continuación, en la Figura 28 se muestra el flujo del proceso de la solución.

**Figura 28.** Flujo del proceso de la solución para la tarea 2



*Nota.* Fuente: Elaboración propia.

### Desarrollo

Para esta tarea se crearon los siguientes componentes en base a los requerimientos técnicos y a lo que se determinó que fue necesario para llevar a cabo la solución de la mejor manera usando buenas prácticas.

#### Librería `lib_fam.js`

Esta librería fue creada para toda la lógica necesaria para la creación del registro Personalizado *FAM Process* y la ejecución del Script del módulo FAM de NetSuite responsable del proceso *Assets Sales Disposal* requerido. Expone la función principal que es usada por el SuiteScript tipo RESTlet que se creó para la tarea. En la Figura 29 se muestra parte del código JavaScript donde se indica el tipo de versión de API que se usa y la función expuesta por la librería.

Figura 29. Código JavaScript de librería lib\_fam.js

```
JS lib_fam.js X
JS lib_fam.js > ...
1  /**
2  * @APIVersion 2.1 Versión de API a usar
3  * Task Date Author Remarks
4  * 23 Jan 2025 Vicente Mori <vicente.mori> - Trigger Asset Sales Disposal process
5  */
6
7  define(['N/search', 'N/record', 'N/task', 'N/runtime'], (search, record, task, runtime) => {
8      /**
9       * Processes disposal by creating a record and triggering a scheduled script.
10     */
11     @param {Object} dataIn - The input data for the process.
12     @param {Object} dataIn.parameters - Parameters for the process.
13     @param {Object} dataIn.recToProc - The records to process.
14     @throws {Object} - Throws an error with a message property if any issue occurs during the operation.
15     */
16
17     Windsurf: Refactor | Explain | Generate JSDoc | X
18     > const disposeProcess = ({ dataIn }) => { ...
19     63     };
20     64
21     65     /** ...
22     Windsurf: Refactor | Explain | X
23     76     > const createRecord = (params) => { ...
24     116     };
25     117
26     118     /** ...
27     Windsurf: Refactor | Explain | X
28     128     > const triggerScript = (params) => { ...
29     139     };
30     140
31     141     /** ...
32     149
33     Windsurf: Refactor | Explain | Generate JSDoc | X
34     150     > const rollback = (params) => { ...
35     164     };
36     165
37     166     return {
38     167         disposeProcess, Función que se expone para ser usado por el RESTlet.
39     168     };
40     169 });
41     170
```

Nota. Fuente: Elaboración propia.

## RL FAM

Es el SuiteScript tipo RESTlet que fue desarrollado, el cual expone un *endpoint* que se activa cuando se realice una petición HTTPS tipo POST. La Figura 30 muestra la porción de código que compone el script.

**Figura 30.** Código JavaScript del RESTlet.

```
1  /**
2   * @APIVersion 2.1
3   * @NScriptType Restlet Se indica el tipo de SuiteScript RESTlet
4   * @author
5   * @date 23 Jan 2025
6   * @author Vicente Mori <vicente.mori>
7   * @remarks - Trigger Asset Sales Disposal process
8   */
9  define(['/N/runtime', './_lib_fam'], (runtime, LibFam) => {
10     const doPost = (dataIn) => {
11         try {
12             log.debug('dataIn', dataIn);
13             if (!dataIn.recsToProc) throw { message: 'Field recsToProc is mandatory' };
14             if (!Object.keys(dataIn.recsToProc).length) throw { message: 'recsToProc is empty' };
15             const scriptObj = runtime.getCurrentScript();
16             const parameters = getParameters(scriptObj);
17             dataIn.parameters = parameters;
18             let response = LibFam.disposeProcess({ dataIn: dataIn }); Se invoca la función de la librería.
19             return {
20                 isSuccess: response.isSuccess,
21                 data: response.data,
22             };
23         } catch (e) {
24             return {
25                 isSuccess: false,
26                 message: e.message || 'Unexpected error, contact your administrator',
27             };
28         }
29     };
30     /**
31     * @param {string} scriptObj
32     * @return {Array} data
33     */
34     function getParameters(scriptObj) {
35         // ...
36     }
37     return {
38         post: doPost, Punto de entrada post propio del RESTlet
39     };
40 });
```

*Nota.* Fuente: Elaboración propia.

Con estos componentes, se permite al cliente que haga uso de este *endpoint* y pueda realizar el *Asset Sales Disposal* en los activos que debe procesar. Cuando el cliente haga la petición HTTPS POST, el RESTlet se ejecutará y creará un registro *FAM Process* como se muestra en la Figura 31 y posteriormente activará el SuiteScript Process Manager del módulo FAM de NetSuite que ejecutará el *Disposal* del activo enviado. El resultado final se puede observar en la Figura 32.

Figura 31. Registro FAM Process creado por el RESTlet.

Search

HelpFeedback

ActivitiesLocalización EspañolaPaymentsTransactionsListsReportsAnalyticsDocumentsSetupCustomizationCommerceAdministration and Controls

FAM Process

EditBackActions

DATE CREATED  
22/01/2025 19:13

LAST MODIFIED  
22/01/2025 19:13

BY

INACTIVE

PROCESS ID  
disposal

TRIGGERED BY

STATUS  
Queued

PARAMETERS  
{ "prmt":4,"recoToProc":{"836":  
{"date":1737360000000,"type":1,"qty":1,"item":62,"cust":138113,"amt":10000.00,"tax":6,"loc":  
"2"}}}

CURRENT STAGE

TOTAL STAGES

CURRENT STAGE STATUS

POST PROCESS DATA

STATE VALUES

Nota. Fuente: Elaboración propia.

Figura 32. Activo fijo con el estado actualizado a Disposal

Search

ActivitiesLocalización EspañolaPaymentsTransactionsListsReportsAnalyticsDocumentsSetupCustomizationCommerce

FAM Asset

FAM000836

EditBackActions

ID  
FAM000836

NAME  
Auto Generated Asset

INACTIVE

ASSET DESCRIPTION  
Auto Generated Asset

ASSET SERIAL NUMBER

ALTERNATE ASSET NUMBER

PARENT ASSET

PARENT PROMOTION

PROJECT

ASSET TYPE  
Training Test

ASSET ORIGINAL COST  
10.000,00

ASSET CURRENT COST  
10.000,00

RESIDUAL VALUE PERCENTAGE  
0,0%

RESIDUAL VALUE  
0,00

DEPRECIATION METHOD  
Straight Line

ASSET LIFETIME  
120

DEPRECIATION PERIOD  
Monthly

ASSET LIFETIME USAGE

CURRENT NET BOOK VALUE  
0,00

CUMULATIVE DEPRECIATION  
10.000,00

ASSET STATUS  
Disposed

CUSTOMER LOCATION

CÓDIGO POSTAL

TIENE GIMNASIO

ORIGEN DEL ACTIVO

ID DEL ACTIVO

EXPEDIENTE CLIENTE

FECHA DE CONSTRUCCION

FECHA DE ADQUISICION DEL ACTIVO

FECHA DE INCORPORACION DEL ACTIVO

FECHA DE ACTUALIZACION EN CATASTRO

NATURALEZA

TIPOLOGIA

SUB TIPOLOGIA

CODIGO IDUFIR

CODIGO REGISTRAL UNICO

OCUPACION

METROS CUADRADOS CONSTRUIDOS

METROS CUADRADOS CONSTRUIDOS CON COMUNES

METROS CUADRADOS UTILES

METROS CUADRADOS EN CATASTRO

METROS CUADRADOS PARCELA

METROS CUADRADOS TERRAZA

METROS CUADRADOS JARDIN

Nota. Fuente: Elaboración propia.

### Tarea 3: Actualizar datos de Fianzas en contrato

#### Análisis de requerimientos

El cliente requiere tener un control de las fianzas que reciben del inquilino en el contrato por lo que el consultor funcional detecta que es necesario crear una sección con los datos de la Fianza en el contrato y que se actualicen automáticamente cuando se emite la fianza.

Cuando se genera la factura de la primera mensualidad desde el contrato (orden de venta), esta debe incluir la fianza.

**Tabla 10.** Documento de especificación para la tarea 3

| Tarea 3 – Documento de especificación |                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                               | Se requiere de una personalización para actualizar los datos de la fianza en el contrato cuando es emitida la fianza.                                                                                                                                                                                                                                                                                                              |
| Supuestos                             | Sin supuestos.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Requerimientos técnicos               | <ul style="list-style-type: none"><li>- Crear pestaña personalizada en el contrato para informar campos de la Fianza.</li><li>- Crear campos personalizados a nivel de cuerpo de transacción para informar los datos de la Fianza en la pestaña correspondiente.</li><li>- Crear un Script de tipo User Event que obtenga y actualice los datos de la fianza en el contrato cuando la primera mensualidad es generada. .</li></ul> |

*Nota.* Fuente: Elaboración propia.

#### Desarrollo

Para esta tarea se crearon los siguientes componentes en base a los requerimientos técnicos y a lo que se determinó que fue necesario para llevar a cabo la solución de la mejor manera y usando buenas prácticas.

#### Registros personalizados para el tipo de depósito y tipo de organización

Se crearon estos dos registros para poder administrar la información que maneja el cliente para sus tipos de depósito y organización que se referencian en la Fianza.

## Listas personalizadas

Se crearon listas para administrar los datos que maneja el cliente que se referencian en la Fianza, las listas creadas son:

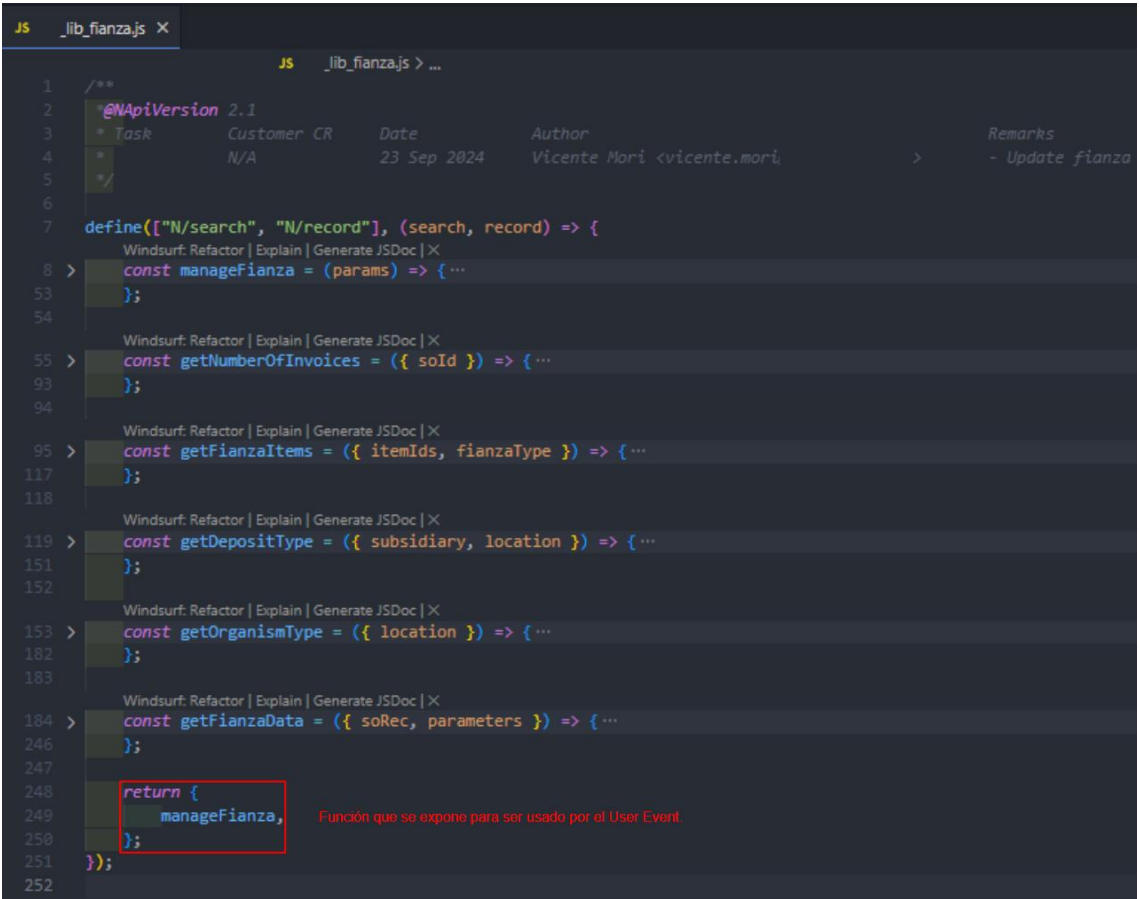
- Estado de la Fianza
- Estado Inquilino
- Estado Organismos
- Depósito
- Categoría del artículo

## Librería lib\_fianza.js

Esta librería contiene toda la lógica necesaria para actualizar los datos de la fianza en el contrato. Expone la función principal que es usada por el Script User Event y además contiene las funciones para obtener la información necesaria de los registros y listas personalizadas.

En la Figura 33 se muestra parte del código JavaScript donde se indica el tipo de versión de API que se usa y la función expuesta por la librería.

**Figura 33.** Código JavaScript de librería lib\_fianza.js



```
1  /**
2   * @APIVersion 2.1
3   * Task Customer CR Date Author Remarks
4   * N/A 23 Sep 2024 Vicente Mori <vicente.mori> - Update fianza
5   */
6
7  define(["N/search", "N/record"], (search, record) => {
8    Windsurf: Refactor | Explain | Generate JSDoc | X
9    > const manageFianza = (params) => { ...
53  };
54
55  Windsurf: Refactor | Explain | Generate JSDoc | X
56  > const getNumberOfInvoices = ({ soId }) => { ...
93  };
94
95  Windsurf: Refactor | Explain | Generate JSDoc | X
96  > const getFianzaItems = ({ itemIds, fianzaType }) => { ...
117  };
118
119  Windsurf: Refactor | Explain | Generate JSDoc | X
120  > const getDepositType = ({ subsidiary, location }) => { ...
151  };
152
153  Windsurf: Refactor | Explain | Generate JSDoc | X
154  > const getOrganismType = ({ location }) => { ...
182  };
183
184  Windsurf: Refactor | Explain | Generate JSDoc | X
185  > const getFianzaData = ({ soRec, parameters }) => { ...
246  };
247
248  return {
249    manageFianza,
250  };
251  });
252
```

*Nota.* Fuente: Elaboración propia.

## UE Fianza

Es el script tipo User Event creado para actualizar los datos de la fianza en el contrato (orden de venta). Está desplegado en la transacción factura de venta porque es la transacción donde se registra la fianza y se valida que corresponda a la primera factura de venta relacionada a la Orden de venta (contrato). En la Figura 34 se muestra el archivo JavaScript correspondiente al User Event.

**Figura 34.** Código JavaScript de User Event UE Fianza

```
1  /**
2   * @APIVersion 2.1
3   * @NScriptType UserEventScript Se indica que es un SuiteScript tipo User Event
4   *
5   * Task Customer CR Date Author Remarks
6   * N/A N/A 23 Sep 2024 Vicente Mori <vicente.mori> - Update fianza
7   */
8  define(["N/runtime", "../_lib_fianza"], (runtime, LibFianza) => {
9      Windsurf: Refactor | Explain | Generate JSDoc | X
10     const afterSubmit = (context) => {
11         const FN = "afterSubmit";
12         try {
13             let { newRecord } = { ...context };
14             const scriptObj = runtime.getCurrentScript();
15             const parameters = getParameters(scriptObj);
16             if ([["invoice"].includes(newRecord.type)] {
17                 LibFianza.manageFianza({ context, parameters });
18             }
19         } catch (e) {
20             log.error({
21                 title: `${FN} error`,
22                 details: {
23                     message: `${FN} - ${e.message || "Unexpected error"}`,
24                 },
25             });
26             throw new Error({
27                 message: `${FN} - ${e.message || "Unexpected error"}`,
28             });
29         }
30     };
31     const getParameters = (scriptObj) => {
32         const parameters = [
33             "custscript_ _ue_f_estado_finanza",
34             "custscript_ _ue_f_estado_org",
35             "custscript_ _ue_f_estado_inquilino",
36             "custscript_ _ue_f_fianza_type",
37         ];
38         const data = {};
39         for (let i = 0; i < parameters.length; i += 1) {
40             if (scriptObj.getParameter(parameters[i])) {
41                 data[parameters[i]] = scriptObj.getParameter(parameters[i]);
42             }
43         }
44         return data;
45     };
46     return {
47         afterSubmit,
48     };
49 });
```

Punto de entrada *afterSubmit* propio del User Event.

*Nota.* Fuente: Elaboración propia.

#### **Tarea 4: Cálculo de mensualidad en la finalización del contrato**

##### **Análisis de requerimientos**

El cliente requiere tener un cálculo automático de la última mensualidad en la finalización del contrato. Esto se da a cabo cuando se informa una fecha fin de contrato y entrega de llaves.

El equipo funcional determinó 2 (dos) escenarios posibles en base a si se facturó el mes de la entrega de llaves o no. De haberse facturado, se debe calcular los días no consumidos y generar una Nota de crédito por ese monto. De no haberse facturado, se genera la Factura de venta correspondiente a la última mensualidad por el monto de los días consumidos.

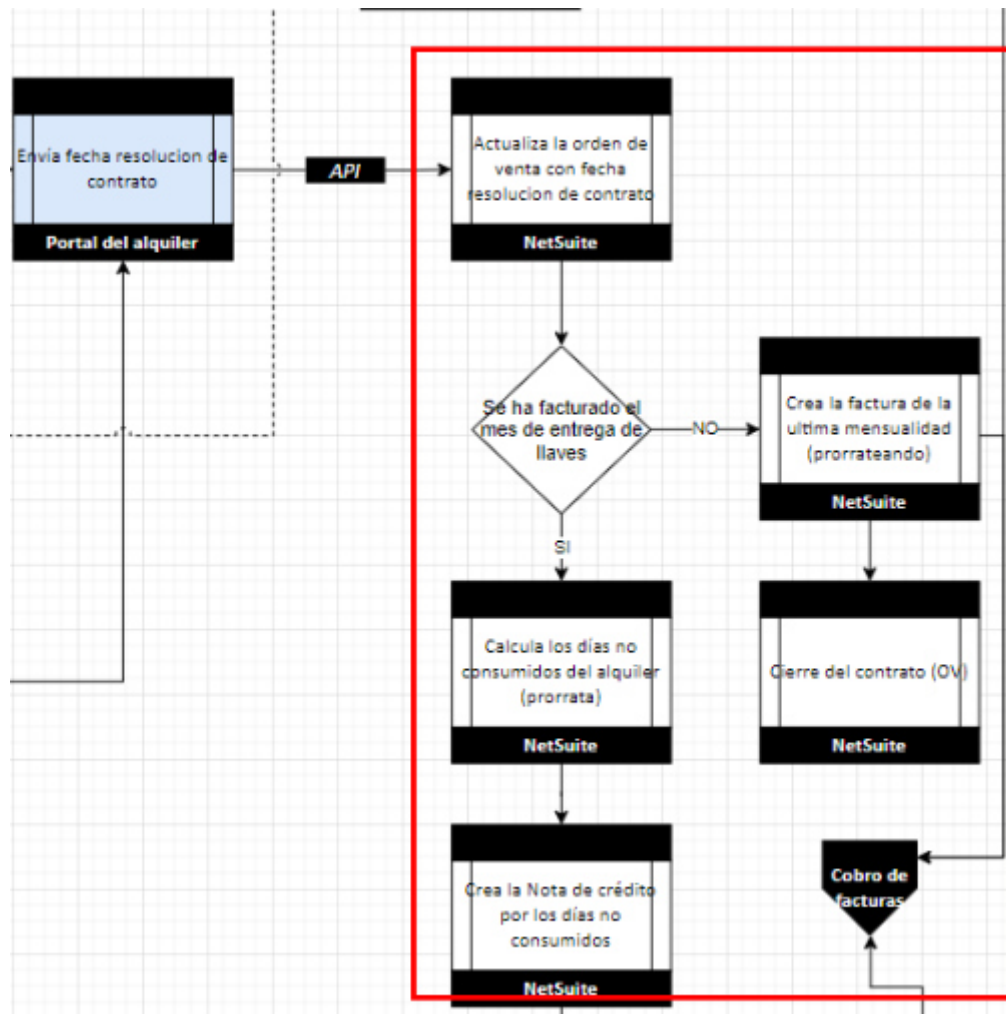
**Tabla 11.** *Documento de especificación para la tarea 4*

| <b>Tarea 4 – Documento de especificación</b> |                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                                      | Se requiere de una personalización para calcular y generar la última mensualidad.                                                                                                                                                                                                                                                     |
| Supuestos                                    | Sin supuestos.                                                                                                                                                                                                                                                                                                                        |
| Requerimientos técnicos                      | <ul style="list-style-type: none"><li>- Crear campos personalizados a nivel de cuerpo de transacción para informar los datos de la entrega de llaves.</li><li>- Crear un Script de tipo User Event que detecte la finalización de contrato y genere la Factura o Nota de crédito por los días consumidos según corresponda.</li></ul> |

*Nota.* Fuente: Elaboración propia.

A continuación, en la Figura 35 se muestra el flujo que sigue la solución.

**Figura 35.** Flujo del proceso de la solución para la tarea 4



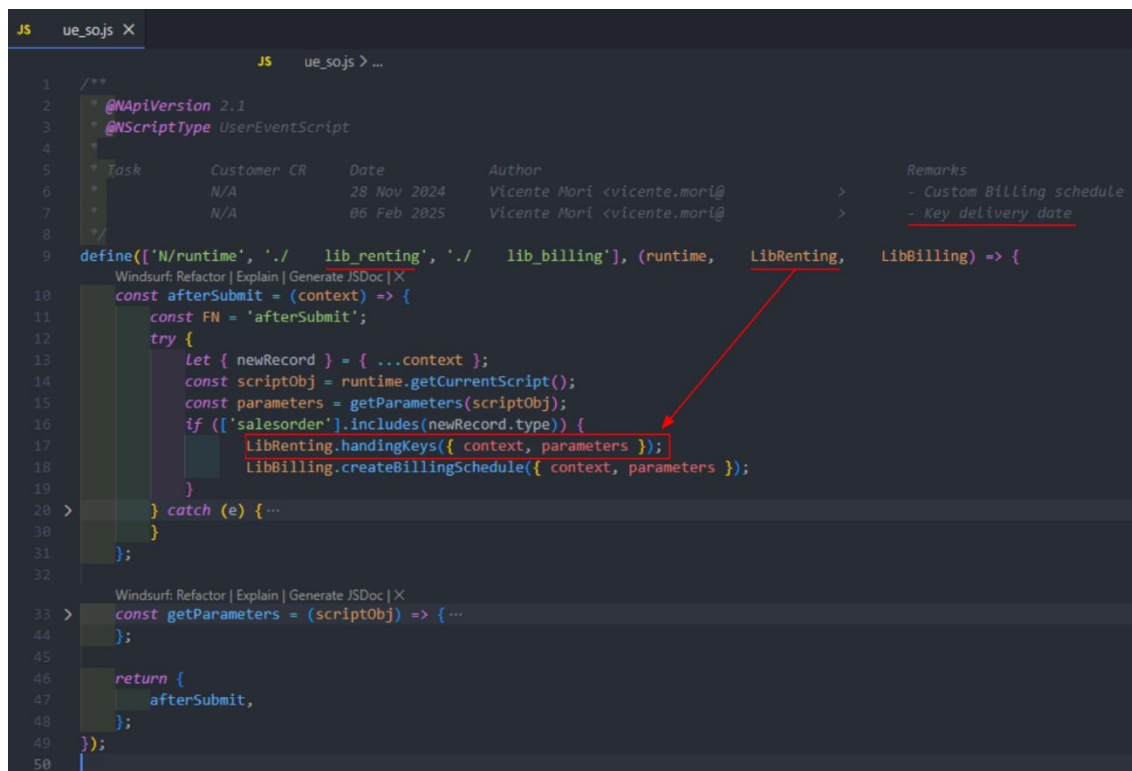
*Nota.* Fuente: Elaboración propia.

## Desarrollo

Para realizar esta personalización se creó un campo personalizado donde se informa la fecha de entrega de llaves, identificando que se da por terminado el contrato y es la fecha con la que se calcula los días consumidos.

Debido a que esta actualización se realiza en la transacción Orden de venta (contrato), se actualizó el Script User Event **UE Sales Order** (creado en la Tarea 1), agregando la lógica para este proceso en particular en la librería destinada a este proceso. De esta manera, no necesitamos crear un nuevo Script User Event, solo la librería destinada a este proceso, lo que permite administrar la lógica de los procesos más fácilmente. En la Figura 36 se muestra el Script actualizado con este proceso.

**Figura 36.** Código JavaScript de UserEvent UE Sales Order actualizado.



```
1  /**
2   * @NapiVersion 2.1
3   * @NScriptType UserEventScript
4   *
5   * Task      Customer CR      Date      Author      Remarks
6   *      N/A      28 Nov 2024      Vicente Mori <vicente.mori@
7   *      N/A      06 Feb 2025      Vicente Mori <vicente.mori@
8   */
9  define(['N/runtime', './ lib_renting', './ lib_billing'], (runtime, LibRenting, LibBilling) => {
10     Windsurf: Refactor | Explain | Generate JSDoc | X
11     const afterSubmit = (context) => {
12         const FN = 'afterSubmit';
13         try {
14             let { newRecord } = { ...context };
15             const scriptObj = runtime.getCurrentScript();
16             const parameters = getParameters(scriptObj);
17             if (!['salesorder'].includes(newRecord.type)) {
18                 LibRenting.handlingKeys({ context, parameters });
19                 LibBilling.createBillingSchedule({ context, parameters });
20             }
21         } catch (e) { ...
22     };
23 }
24
25 Windsurf: Refactor | Explain | Generate JSDoc | X
26 const getParameters = (scriptObj) => { ...
27 };
28
29 return {
30     afterSubmit,
31 };
32 }));
```

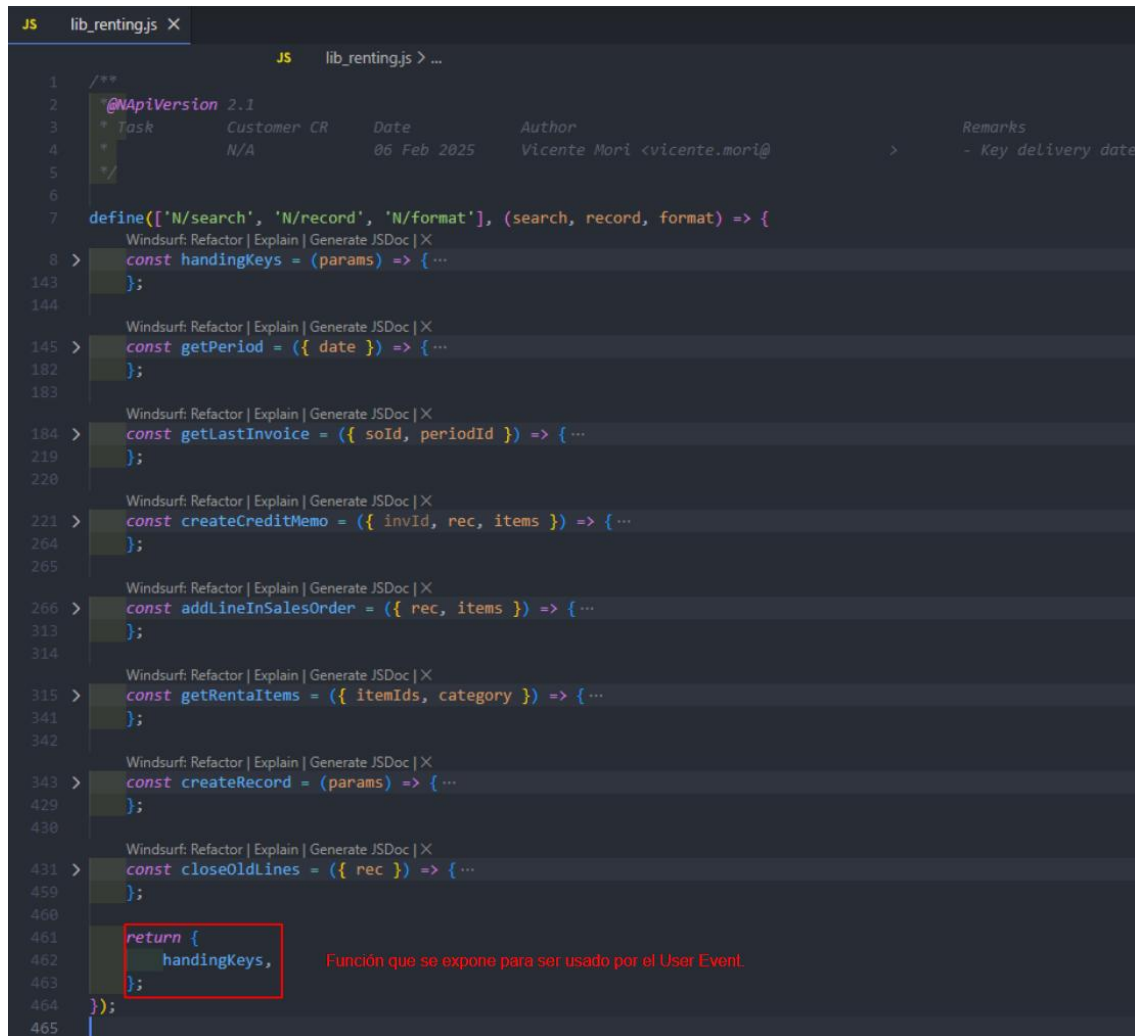
*Nota.* Fuente: Elaboración propia.

### Librería lib\_renting.js

Esta librería contiene toda la lógica necesaria para el cálculo de los días no consumidos y la generación de las transacciones correspondientes. Para el cálculo se requiere usar el precio unitario de los artículos de tipo Renta, multiplicarlo por 12 (meses del año) y dividirlo entre 365 (días del año). El resultado vendría a ser la prorrata, el cual se multiplica por los días consumidos dando como resultado el cálculo final.

En la Figura 37 se muestra parte del código JavaScript donde se muestra la función expuesta por la librería.

**Figura 37.** Código JavaScript de la librería *lib\_renting*.



```
1  /**
2   * @APIVersion 2.1
3   * Task Customer CR Date Author Remarks
4   * N/A 06 Feb 2025 Vicente Mori <vicente.mori@ > - Key delivery date
5   */
6
7  define(['N/search', 'N/record', 'N/format'], (search, record, format) => {
8    Windsurf: Refactor | Explain | Generate JSDoc | X
9    const handingKeys = (params) => { ...
143  };
144
145  Windsurf: Refactor | Explain | Generate JSDoc | X
146  const getPeriod = ({ date }) => { ...
182  };
183
184  Windsurf: Refactor | Explain | Generate JSDoc | X
185  const getLastInvoice = ({ soId, periodId }) => { ...
219  };
220
221  Windsurf: Refactor | Explain | Generate JSDoc | X
222  const createCreditMemo = ({ invId, rec, items }) => { ...
264  };
265
266  Windsurf: Refactor | Explain | Generate JSDoc | X
267  const addLineInSalesOrder = ({ rec, items }) => { ...
313  };
314
315  Windsurf: Refactor | Explain | Generate JSDoc | X
316  const getRentalItems = ({ itemIds, category }) => { ...
341  };
342
343  Windsurf: Refactor | Explain | Generate JSDoc | X
344  const createRecord = (params) => { ...
429  };
430
431  Windsurf: Refactor | Explain | Generate JSDoc | X
432  const closeOldLines = ({ rec }) => { ...
459  };
460
461  return {
462    handingKeys,
463  };
464  });
465
```

Función que se expone para ser usado por el User Event.

*Nota.* Fuente: Elaboración propia.

## **Tarea 5: Sincronización de Segmentos contables personalizados**

### **Análisis de requerimientos**

Se necesita sincronizar automáticamente el segmento contable personalizado Activo con el módulo estándar de Activos Fijos. El segmento personalizado debe tener todos los activos del módulo de Activos Fijos de NetSuite.

**Tabla 12.** *Documento de especificación para la tarea 5*

| <b>Tarea 5 – Documento de especificación</b> |                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                                      | Debido a que no es posible realizar una conexión directa con el módulo de Activos Fijos, se requiere de una personalización para tener actualizado la lista de activos fijos con el segmento personalizado asociado.                                                                                                                                                                            |
| Supuestos                                    | El cliente provee la lista inicial de activos fijos importado por CSV en el módulo FAM.                                                                                                                                                                                                                                                                                                         |
| Requerimientos técnicos                      | <ul style="list-style-type: none"><li>- Crear segmento personalizado “Activo” para replicar los activos del módulo FAM.</li><li>- Cada vez que un activo fijo sea creado o actualizado, se debe sincronizar en el segmento personalizado.</li><li>- Crear un Script que detecte cuando un activo fijo sea creado o actualizado y sincronice los cambios en el segmento personalizado.</li></ul> |

*Nota.* Fuente: Elaboración propia.

### **Desarrollo**

Para realizar esta tarea se creó un segmento personalizado que hará referencia a un activo del módulo de Activos Fijos, este registro se tendrá sincronizado mediante un Script User Event desplegado en los Activos del módulo de Activos Fijos que se ejecutará cuando el registro sea creado o actualizado.

En la Figura 38 se muestra la configuración del Segmento Personalizado Activo y en la Figura 39, el Registro Personalizado asociado al Segmento Personalizado.

Figura 38. Segmento Personalizado Activo.

The screenshot shows the 'Custom Segment' configuration page in Oracle NetSuite. The page has a top navigation bar with 'básico' and 'SANDBOX' tabs. Below the navigation bar, there are tabs for 'Edit', 'Back', 'Manage Values', and 'Actions'. The 'Primary Information' section includes fields for 'LABEL' (set to 'Activo'), 'ID' (set to 'cseg\_activo\_bh\_'), 'RECORD ID' (set to 'customrecord\_cseg\_activo\_bh\_'), and 'CUSTOM RECORD TYPE' (set to 'Activo'). The 'Accounting' section has a checked 'GL IMPACT' checkbox. The 'Details' section has a 'HELP' link and a 'DESCRIPTION' field. At the bottom, there is a 'Values' section with a table showing 'Application & Sourcing', 'Validation & Defaulting', 'Permissions', 'Dependent Segments', 'Display Order', and 'Translation'. A note at the bottom states: 'NO MORE THAN 1,000 VALUES CAN BE DISPLAYED. TO UPDATE VALUES, CLICK MANAGE VALUES AT THE TOP OF THE PAGE.'

Nota. Fuente: Elaboración propia.

Figura 39. Registro Personalizado Activo

The screenshot shows the 'Custom Record Type' configuration page in Oracle NetSuite. The page has a top navigation bar with 'básico' and 'SANDBOX' tabs. Below the navigation bar, there are tabs for 'Edit', 'Back', and 'Actions'. The 'NAME' section includes fields for 'NAME' (set to 'Activo'), 'ID' (set to 'customrecord\_cseg\_activo\_bh\_'), 'ORIGINATING CUSTOM SEGMENT' (set to 'Activo'), and 'OWNER'. The 'DESCRIPTION' section has a checked 'INCLUDE NAME FIELD' checkbox. The 'ACCESS TYPE' section includes a checked 'ALLOW UI ACCESS' checkbox and a 'SHOW REMOVE LINK' checkbox. The 'ALLOW QUICK ADD' section includes a checked 'ENABLE SYSTEM NOTES' checkbox and a 'HIERARCHY' checkbox. At the bottom, there is a table with columns 'DESCRIPTION', 'ID', 'TYPE', 'LIST/RECORD', 'TAB', and 'SHOW IN LIST'. The table contains two rows: 'Promotion' with ID 'custrecord\_bsc\_activo\_promocion' and 'Asset' with ID 'custrecord\_ref\_asset'.

| DESCRIPTION | ID                              | TYPE        | LIST/RECORD | TAB | SHOW IN LIST |
|-------------|---------------------------------|-------------|-------------|-----|--------------|
| Promotion   | custrecord_bsc_activo_promocion | List/Record | Promoción   |     | No           |
| Asset       | custrecord_ref_asset            | List/Record | FAM Asset   |     | Yes          |

Nota. Fuente: Elaboración propia.

A continuación, en las Figuras 40 y 41 se muestra el código javascript del User Event UE Asset y librería lib\_asset.js.

Figura 40. Código javascript de UE Asset.

```
JS ue_asset.js X
1  /**
2   * @NApiVersion 2.1
3   * @NScriptType UserEventScript
4   *
5   * Task      Customer CR      Date      Author      Remarks
6   *          N/A      23 Sep 2024      Vicente Mori <vicente.mori@...com>      - Sync Asset creation/disposal with custom segment
7   */
8  define(["N/runtime", "N/ lib_asset"], (runtime, LibAsset) => {
9      Windsurf: Refactor | Explain | Generate JSDoc | X
10     const afterSubmit = (context) => {
11         const FN = "afterSubmit";
12         try {
13             let { newRecord, oldRecord, type } = { ...context };
14             const scriptObj = runtime.getCurrentScript();
15             const parameters = getParameters(scriptObj);
16             if (["create", "copy", "edit", "xedit"].includes(type)) {
17                 if (["customrecord_ncfar_asset"].includes(newRecord.type)) {
18                     LibAsset.syncWithSegment({ newRecord, oldRecord, type, parameters });
19                 }
20             } catch (e) { ...
21             }
22         };
23     }
24
25     Windsurf: Refactor | Explain | Generate JSDoc | X
26     const getParameters = (scriptObj) => { ...
27     }
28
29     return {
30         afterSubmit,
31     };
32 });
```

Nota. Fuente: Elaboración propia.

Figura 41. Código javascript de lib\_asset.js.

```
JS lib_asset.js X
1  /**
2   * @NApiVersion 2.1
3   * Task      Customer CR      Date      Author      Remarks
4   *          N/A      23 Sep 2024      Vicente Mori <vicente.mori@...com>      - Sync Asset creation/disposal with custom segment
5   */
6
7  define(["N/search", "N/record"], (search, record) => {
8      /** ...
9      Windsurf: Refactor | Explain | X
10     const syncWithSegment = (params) => { ...
11     };
12
13     /** ...
14     Windsurf: Refactor | Explain | X
15     const get Activo = ({ assetId }) => { ...
16     };
17
18     /** ...
19     Windsurf: Refactor | Explain | X
20     const handleRecord = (params) => { ...
21     };
22
23     return {
24         syncWithSegment,
25     };
26 });
```

Nota. Fuente: Elaboración propia.

## Tarea 6: Actualización automática de IPC o IRAV

### Análisis de requerimientos

Dentro de las operaciones de la gestión de alquiler, existe la necesidad de hacer una revisión de la renta de los contratos de manera anual en función al IPC (Índice de Precios al Consumo) o IRAV (Índice de Referencia de Arrendamientos de Vivienda) que son publicados mensualmente en la página del INE (Instituto Nacional de Estadística).

El sistema deberá calcular la parte proporcional de la mensualidad, siempre que la anualidad no corresponda al primer día del mes. A partir de ahí se debe realizar la prorrata y solo aplicar a los conceptos renta/RTA. Ejemplo: Suponemos un contrato con fecha de inicio 21/05/2019 y renta 151,93€, su actualización por tanto será el 21 de mayo de cada año (2020, 2021, 2022, 2023, etc).

El cálculo de IPC se realiza en base a los conceptos detallados en la tabla 13.

**Tabla 13.** *Cálculo del IPC.*

|      | %                                           | ¿A qué aplica?                                                               |
|------|---------------------------------------------|------------------------------------------------------------------------------|
| IPC  | Publicado mensualmente en la página de INE. | Contratos de vivienda cuya fecha de firma es anterior al 26 de mayo de 2023. |
| IRAV | Publicado mensualmente en la página de INE. | Contratos de vivienda a partir del 26 de mayo de 2023.                       |

*Nota.* Fuente: Elaboración propia

**Tabla 14.** *Documento de especificación para la tarea 6.*

| Tarea 13 – Documento de especificación |                                                                                                                                                                                                                |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resumen                                | Se requiere de una personalización para revisar los contratos anualmente en función de su fecha de inicio de contrato y se recalcula el importe del IPC que debe aplicarse cada año en función de dicho valor. |
| Supuestos                              | Los valores del IPC e IRAV serán ingresados manualmente al sistema en base a lo publicado en la página del INE de forma mensual.                                                                               |

---

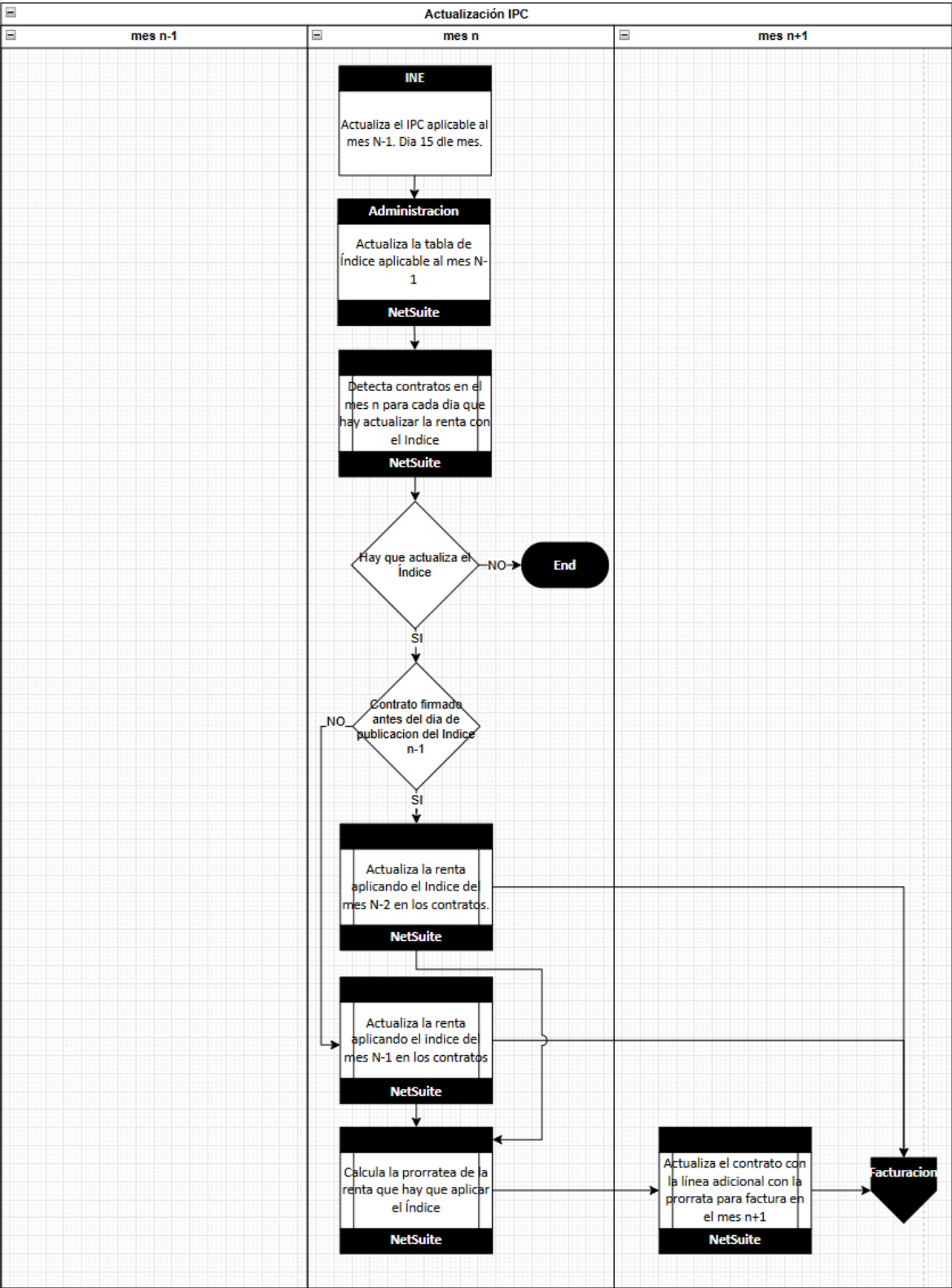
|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Requerimientos<br>técnicos | <ul style="list-style-type: none"> <li>- Crear un registro personalizado para el registro del IPC e IRAY.</li> <li>- Crear campo personalizado a nivel de cuerpo de transacción para informar la fecha de inicio de contrato.</li> <li>- Crear campo personalizado a nivel de línea de transacción para indicar el periodo de ajuste.</li> <li>- Crear un Script de tipo Map Reduce que detecte los contratos que cumplen las condiciones para actualizar la renta.</li> </ul> |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

*Nota.* Fuente: Elaboración propia

A continuación, en la Figura 42 se muestra el flujo que sigue la solución.

Figura 42. Flujo del proceso de la solución para la tarea 6.

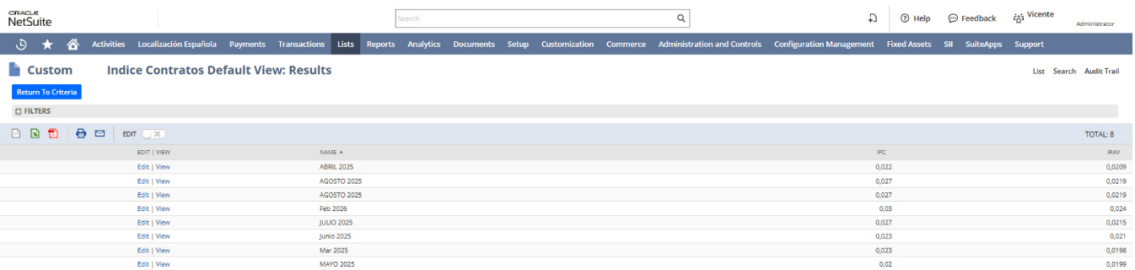


Nota. Fuente: Elaboración propia.

Desarrollo

Para realizar esta tarea se creó un registro personalizado para registrar los valores de IPC e IRAV, campos personalizados para la fecha de inicio de contrato y periodo de ajuste. En las siguientes figuras se muestran el Registro personalizado Índice Contratos y los campos personalizados a utilizar.

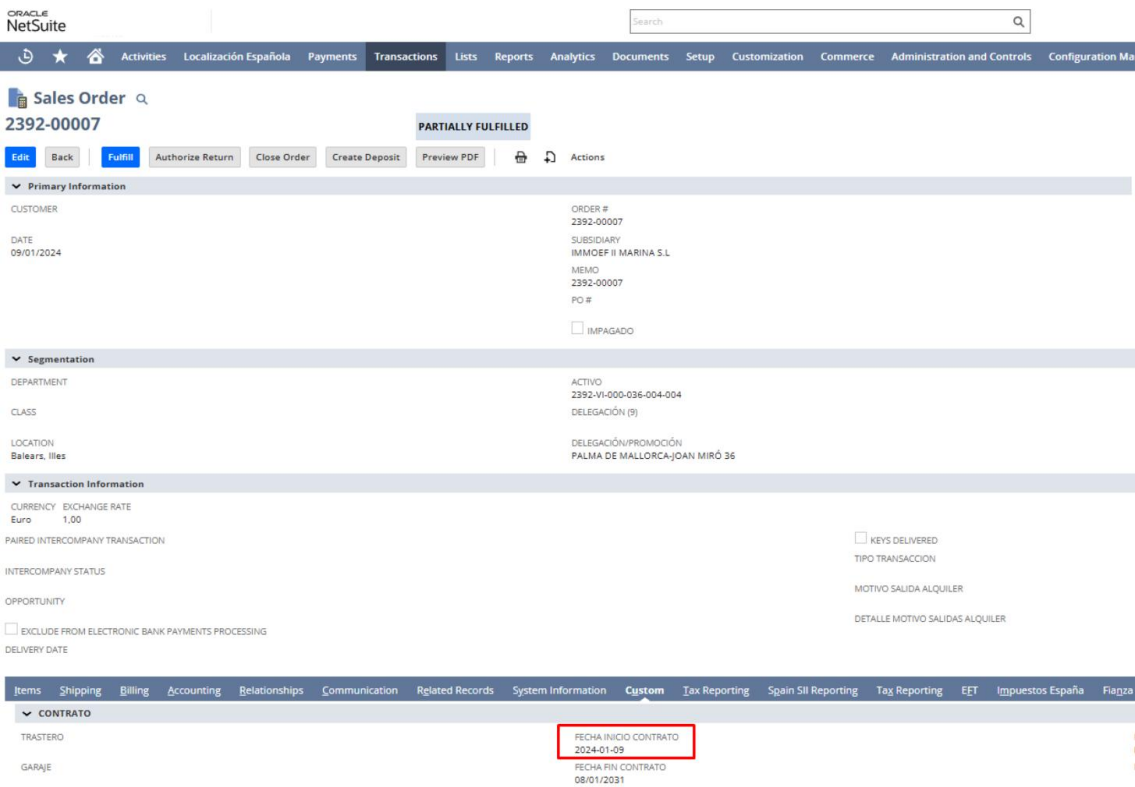
Figura 43. Registro personalizado Índice Contratos.



| EDIT   VIEW | NAME        | IPC   | IRAY   |
|-------------|-------------|-------|--------|
| EDIT   VIEW | ABRIL 2025  | 0,022 | 0,0209 |
| EDIT   VIEW | AGOSTO 2025 | 0,027 | 0,0219 |
| EDIT   VIEW | AGOSTO 2025 | 0,027 | 0,0219 |
| EDIT   VIEW | Feb 2025    | 0,03  | 0,024  |
| EDIT   VIEW | JULIO 2025  | 0,027 | 0,0219 |
| EDIT   VIEW | Junio 2025  | 0,023 | 0,021  |
| EDIT   VIEW | Mar 2025    | 0,023 | 0,0198 |
| EDIT   VIEW | MAYO 2025   | 0,02  | 0,0199 |

Nota. Fuente: Elaboración propia.

Figura 44. Campo Fecha Inicio Contrato



**Primary Information**

CUSTOMER: ORDER # 2392-00007  
DATE: 09/01/2024  
SUBSIDIARY: IMMOEF II MARINA S.L.  
MEMO: 2392-00007  
PO #: ☐ IMPAGADO

**Segmentation**

DEPARTMENT: ACTIVO  
CLASS: 2392-VI-000-036-004-004  
LOCATION: Balears, illes  
DELEGACIÓN (9)  
DELEGACIÓN/PROMOCIÓN: PALMA DE MALLORCA-JOAN MIRÓ 36

**Transaction Information**

CURRENCY: Euro  
EXCHANGE RATE: 1,00  
PAIRED INTERCOMPANY TRANSACTION: ☐ KEYS DELIVERED  
INTERCOMPANY STATUS: TIPO TRANSACCION  
OPPORTUNITY: MOTIVO SALIDA ALQUILER  
☐ EXCLUDE FROM ELECTRONIC BANK PAYMENTS PROCESSING  
DELIVERY DATE: DETALLE MOTIVO SALIDAS ALQUILER

**Custom**

CONTRATO  
TRASTERO: FECHA INICIO CONTRATO 2024-01-09  
GARAJE: FECHA FIN CONTRATO 08/01/2031

Nota. Fuente: Elaboración propia.

Figura 45. Campo Periodo de ajuste.

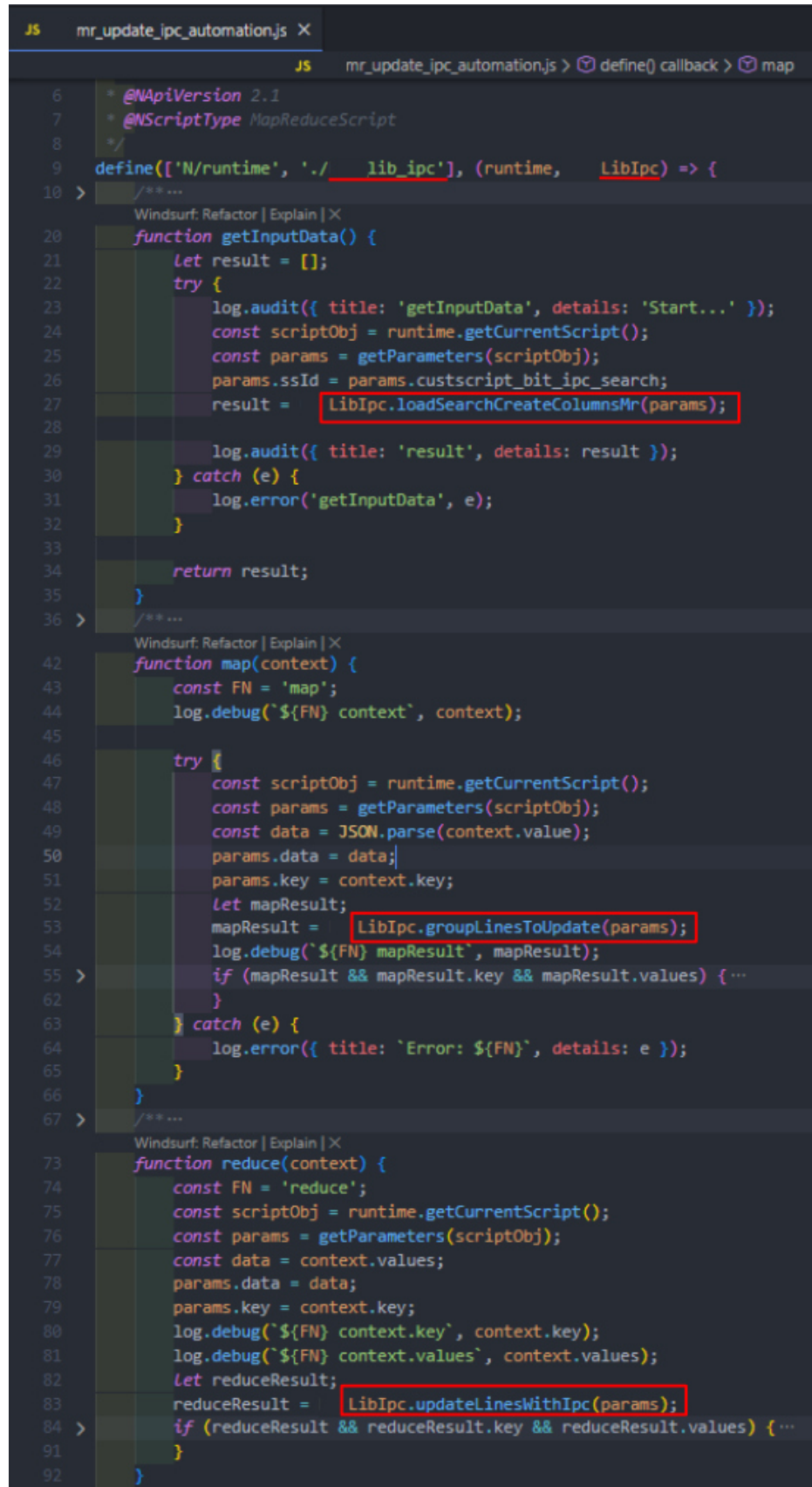
| Items Shipping Billing Accounting Relationships Communication Related Records System Information Custom Tax Reporting Spain SE Reporting Tax Reporting EET Impuestos España Reglas Pricing History |                     |                     |              |           |           |          |           |        |         |        |          |          |           |         |                         |        |                                |                            |               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|---------------------|--------------|-----------|-----------|----------|-----------|--------|---------|--------|----------|----------|-----------|---------|-------------------------|--------|--------------------------------|----------------------------|---------------|
| Items 60,792,96+ Including Tax 0.00+                                                                                                                                                               |                     |                     |              |           |           |          |           |        |         |        |          |          |           |         |                         |        |                                |                            |               |
| DATE                                                                                                                                                                                               | DEPARTMENT          | CLASS               | LOCATION     | COMMITTED | FULFILLED | INVOICED | QUANTITY  | AMOUNT | OPTIONS | CLOSED | TAX CODE | TAX RATE | GROSS AMT | TAX AMT | APPLY NET TAX           | ACTIVO | DELEGACION/PROVINCIA           | DELEGACION                 | EMPLACAMIENTO |
| 1.083.85                                                                                                                                                                                           | INGRESOS ALQUILERES | INGRESOS ALQUILERES | Balears, Res | 0         | 0         | 6        | 6.503.10  |        |         |        | IVAEX    | 0.0%     | 6.503.10  | 0.00    | 2380-VI-000-036-004-004 |        | PALMA DE MALLORCA-JOAN MARO 36 | LEVANTE, TERRITORIAL, ESTE | Manual        |
| 1.083.85                                                                                                                                                                                           | INGRESOS ALQUILERES | INGRESOS ALQUILERES | Balears, Res | 0         | 0         | 12       | 13.006.20 |        |         |        | IVAEX    | 0.0%     | 13.006.20 | 0.00    | 2380-VI-000-036-004-004 |        | PALMA DE MALLORCA-JOAN MARO 36 | LEVANTE, TERRITORIAL, ESTE | Manual        |
| 1.083.85                                                                                                                                                                                           | INGRESOS ALQUILERES | INGRESOS ALQUILERES | Balears, Res | 0         | 0         | 12       | 13.006.20 |        |         |        | IVAEX    | 0.0%     | 13.006.20 | 0.00    | 2380-VI-000-036-004-004 |        | PALMA DE MALLORCA-JOAN MARO 36 | LEVANTE, TERRITORIAL, ESTE | Manual        |
| 1.083.85                                                                                                                                                                                           | INGRESOS ALQUILERES | INGRESOS ALQUILERES | Balears, Res | 0         | 0         | 12       | 13.006.20 |        |         |        | IVAEX    | 0.0%     | 13.006.20 | 0.00    | 2380-VI-000-036-004-004 |        | PALMA DE MALLORCA-JOAN MARO 36 | LEVANTE, TERRITORIAL, ESTE | Manual        |

Nota. Fuente: Elaboración propia.

MR Update IPC Automation

Debido a que para esta tarea se debe detectar contratos a actualizar cada cierto tiempo, significa la posibilidad de procesar grandes cantidades de datos por lo que decidió crear un script Map Reduce para este procesamiento, diseñado para cargar los resultados de una *Saved Search* (búsqueda guardada) para luego agrupar todas las líneas (que pueden ser de diferentes contratos), y finalmente, realizar el recálculo usando los índices de IPC o IRAY. En la Figura 46 se muestra el código javascript del Map Reduce y en la Figura 47 se muestra el código javascript de la librería que contiene las funciones necesarias para esta tarea.

Figura 46. Código javascript de Map Reduce Update IPC Automation.

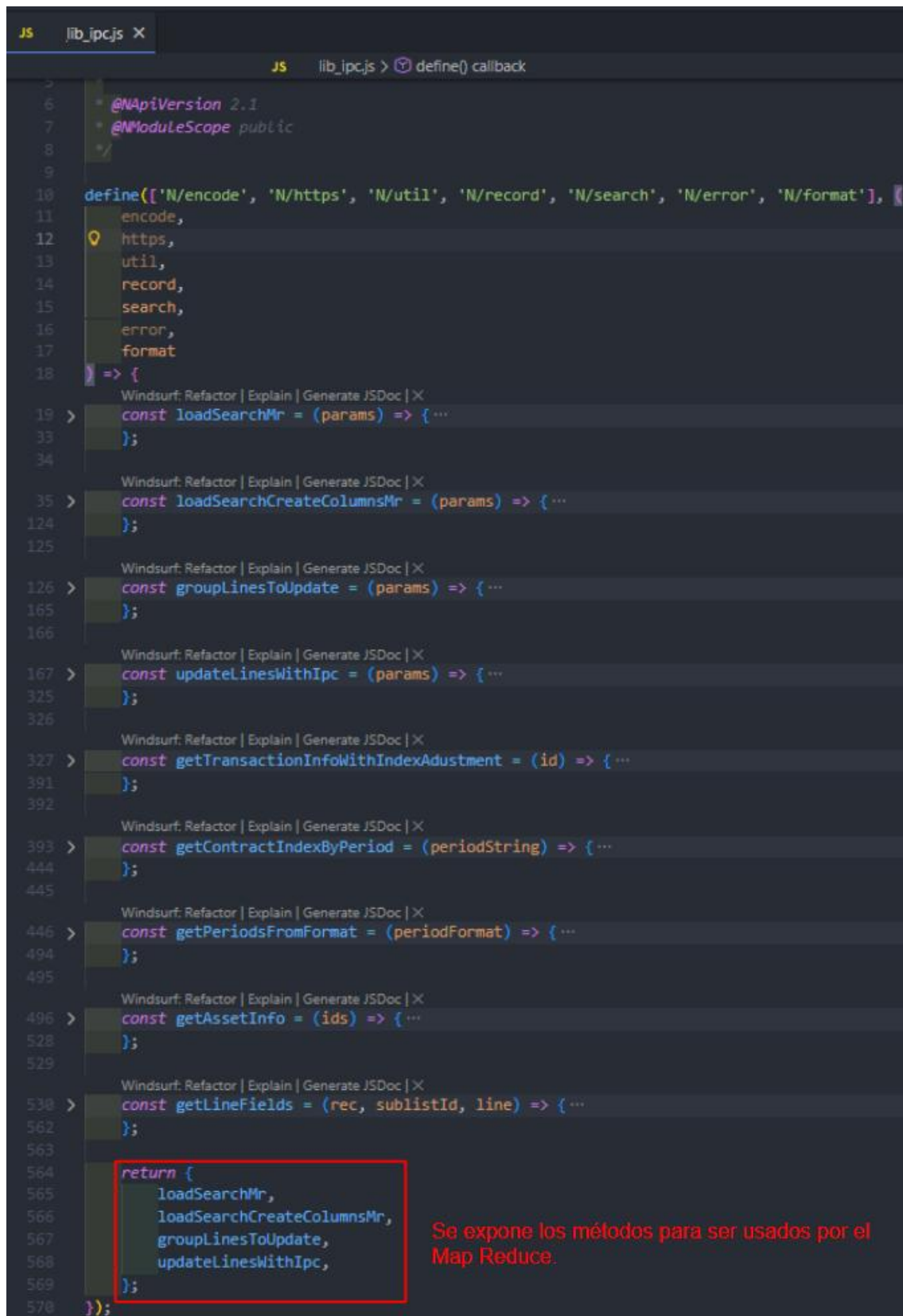


```
JS mr_update_ipc_automation.js X
JS mr_update_ipc_automation.js > define() callback > map

6  * @NApiVersion 2.1
7  * @NScriptType MapReduceScript
8  */
9  define(['N/runtime', './ lib_ipc'], (runtime, LibIpc) => {
10 >  /** ...
11  Windsurf: Refactor | Explain | X
12  function getInputData() {
13    let result = [];
14    try {
15      log.audit({ title: 'getInputData', details: 'Start...' });
16      const scriptObj = runtime.getCurrentScript();
17      const params = getParameters(scriptObj);
18      params.ssId = params.custscript_bit_ipc_search;
19      result = LibIpc.loadSearchCreateColumnsMr(params);
20    } catch (e) {
21      log.error('getInputData', e);
22    }
23    return result;
24  }
25  /** ...
26  Windsurf: Refactor | Explain | X
27  function map(context) {
28    const FN = 'map';
29    log.debug(`${FN} context`, context);
30    try {
31      const scriptObj = runtime.getCurrentScript();
32      const params = getParameters(scriptObj);
33      const data = JSON.parse(context.value);
34      params.data = data;
35      params.key = context.key;
36      let mapResult;
37      mapResult = LibIpc.grouplinesToUpdate(params);
38      log.debug(`${FN} mapResult`, mapResult);
39      if (mapResult && mapResult.key && mapResult.values) { ...
40    } catch (e) {
41      log.error({ title: 'Error: ${FN}', details: e });
42    }
43  }
44  /** ...
45  Windsurf: Refactor | Explain | X
46  function reduce(context) {
47    const FN = 'reduce';
48    const scriptObj = runtime.getCurrentScript();
49    const params = getParameters(scriptObj);
50    const data = context.values;
51    params.data = data;
52    params.key = context.key;
53    log.debug(`${FN} context.key`, context.key);
54    log.debug(`${FN} context.values`, context.values);
55    let reduceResult;
56    reduceResult = LibIpc.updateLinesWithIpc(params);
57    if (reduceResult && reduceResult.key && reduceResult.values) { ...
58  }
59 }
```

Nota. Fuente: Elaboración propia.

Figura 47. Código javascript de librería lib\_ipc



```
JS lib_ipc.js x
JS lib_ipc.js > define() callback
6  * @NApiVersion 2.1
7  * @ModuleScope public
8  */
9
10 define(['N/encode', 'N/https', 'N/util', 'N/record', 'N/search', 'N/error', 'N/format'], [
11   encode,
12   https,
13   util,
14   record,
15   search,
16   error,
17   format
18 ] => {
19   Windsurf: Refactor | Explain | Generate JSDoc | X
20   const loadSearchMr = (params) => { ...
33   };
34
35   Windsurf: Refactor | Explain | Generate JSDoc | X
36   const loadSearchCreateColumnsMr = (params) => { ...
124   };
125
126   Windsurf: Refactor | Explain | Generate JSDoc | X
127   const grouplinesToUpdate = (params) => { ...
165   };
166
167   Windsurf: Refactor | Explain | Generate JSDoc | X
168   const updatelinesWithIpc = (params) => { ...
325   };
326
327   Windsurf: Refactor | Explain | Generate JSDoc | X
328   const getTransactionInfoWithIndexAdustment = (id) => { ...
391   };
392
393   Windsurf: Refactor | Explain | Generate JSDoc | X
394   const getContractIndexByPeriod = (periodString) => { ...
444   };
445
446   Windsurf: Refactor | Explain | Generate JSDoc | X
447   const getPeriodsFromFormat = (periodFormat) => { ...
494   };
495
496   Windsurf: Refactor | Explain | Generate JSDoc | X
497   const getAssetInfo = (ids) => { ...
528   };
529
530   Windsurf: Refactor | Explain | Generate JSDoc | X
531   const getLineFields = (rec, sublistId, line) => { ...
562   };
563
564   return {
565     loadSearchMr,
566     loadSearchCreateColumnsMr,
567     grouplinesToUpdate,
568     updatelinesWithIpc,
569   };
570 });
```

Se expone los métodos para ser usados por el Map Reduce.

Nota. Fuente: Elaboración propia.

### 3.3 Evaluación

#### 3.3.1 Evaluación económica

Esta sección se enfoca en el análisis de los costos desde el punto de vista de la empresa consultora que realiza la implementación en el desarrollo de las personalizaciones, el cual tuvo una duración de 4 meses, donde se han tenido en cuenta los costos del personal involucrado en el proyecto y las licencias de software necesarias. En la tabla 15, se presenta el detalle de los costos mensuales, expresados en euros, asociado al equipo que desarrolló el proyecto.

**Tabla 15.** *Costos mensuales del equipo del proyecto*

| <b>Rol</b>                         | <b>Costo mensual (€)</b> | <b>Costo Total (€)</b> |
|------------------------------------|--------------------------|------------------------|
| Project Manager                    | € 1 900                  | € 7 600                |
| NetSuite Implementation Consultant | € 1 750                  | € 7 000                |
| Software Engineer, NetSuite        | € 1 750                  | € 7 000                |
| Software Engineer, Deployments     | € 1 000                  | € 4 000                |
| Renta de oficina                   | € 2500                   | € 10 000               |
| Mantenimiento equipos físicos      | € 100                    | € 400                  |
| <b>Total</b>                       |                          | <b>€ 36 000</b>        |

*Nota.* Costos mensuales del equipo durante los 4 meses de desarrollo. Fuente: Elaboración propia.

La Tabla 16 detalla el costo anual de las licencias de software utilizadas durante la duración del proyecto.

**Tabla 16.** *Costos de Licencias de software*

| <b>Descripción</b>    | <b>Cantidad</b> | <b>Costo anual (€)</b> | <b>Costo Total (€)</b> |
|-----------------------|-----------------|------------------------|------------------------|
| Licencias de NetSuite | 4               | € 500                  | € 2 000                |
| Licencias Atlassian   | 4               | € 180                  | € 720                  |
| <b>Total</b>          |                 |                        | <b>€ 2 720</b>         |

*Nota.* Costos anuales de las licencias de software utilizadas en el proyecto. Fuente: Elaboración propia.

Con la información proporcionada relacionada a los costos asociados al proyecto, se concluye que los costos totales realizados por la consultora durante los 4 meses de proyecto fueron de 38,720 euros.

### 3.3.2 Proyección económica de la implementación del proyecto

El análisis financiero del proyecto evalúa la rentabilidad del proyecto y su viabilidad económica a lo largo del tiempo. Las tablas siguientes proporcionan una visión clara si los gastos asumidos con la implementación generarán beneficios económicos suficientes para justificar los costos incurridos.

El proyecto fue encargado por un cliente con subsidiarias en España. Para la estimación de flujo de efectivo, se considera únicamente este cliente. Los ingresos generados por el desarrollo de las personalizaciones son asociados a una bolsa de 500 horas y un precio de 120 euros por hora al término del proyecto. La Tabla 17 detalla los ingresos por venta.

**Tabla 17.** *Ingresos por venta*

| Descripción                       | Cantidad | Costo por unidad (€) | Costo Total (€) |
|-----------------------------------|----------|----------------------|-----------------|
| Bolsa de horas de personalización | 500      | € 120                | € 60 000        |
| Suscripción anual                 | 1        | € 5 000              | € 5 000         |
| Bolsa de horas de soporte         | 200      | € 120                | € 24 000        |

*Nota.* Ingresos por venta. Fuente: Elaboración propia.

Al realizar el análisis de los ingresos del proyecto, es importante considerar los gastos relacionados al soporte luego de la implementación a partir del segundo año que se detalla en la tabla 18.

**Tabla 18.** *Proyección de ingresos y gastos*

| Año | Ingresos | Gastos   | Monto Neto |
|-----|----------|----------|------------|
| 1   | € 0      | € 38 720 | - € 38 720 |
| 2   | € 65 000 | € 17 000 | € 48 000   |
| 3   | € 29 000 | € 17 000 | € 12 000   |

*Nota.* Proyección de ingresos y gastos. Fuente: Elaboración propia.

Estos datos permiten medir el beneficio generado en relación con el costo al calcular el Retorno de inversión (ROI) bajo la siguiente fórmula.

$$- \text{ROI: } \left( \frac{\text{Utilidad Neta}}{\text{Inversión inicial}} \right) \times 100$$

Con los resultados obtenidos, el proyecto presenta un ROI de 54.95%, lo cual indica que se recupera la inversión inicial y genera un rendimiento adicional significativo, por lo que se concluye que el proyecto es rentable.

## CAPITULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El proyecto del presente trabajo fue ejecutado exitosamente, se logró implementar el proceso de gestión de contratos de alquiler dentro de NetSuite cumpliendo los requerimientos del cliente.

Durante el desarrollo de las tareas involucradas en el proyecto, se presentaron desafíos a nivel funcional y técnico que involucraron un esfuerzo adicional por parte del autor para lograr una entrega efectiva de la solución. Adicionalmente, superar estos desafíos, permitieron al autor, reforzar sus habilidades blandas y técnicas, así como también ganar mayor experiencia y generar un impacto positivo en el equipo. A continuación, se resaltan las reflexiones que deja el desarrollo del presente proyecto:

- Uno de los desafíos más importantes se originó desde el ámbito funcional, ya que no se contempló en un inicio que el módulo Advanced Billing no contaba con la flexibilidad para generar las reglas de pago como lo requería el cliente. Esto significó realizar este proceso usando SuiteScripts para poder brindar la flexibilidad necesaria. Se resalta este desafío ya que implicó que el autor asista a reuniones con los líderes y encargados, proponga ideas de solución y brinde una respuesta rápida y efectiva debido a que no se contaba con mucho tiempo para la entrega al cliente y estaba en riesgo la implementación total del proyecto.
- Por parte del lado técnico, los retos que se presentaron están relacionados al rendimiento y unidades de gobierno de los SuiteScripts. A pesar de contar con diversas herramientas útiles que permiten personalizar el ERP, es fundamental conocer las funciones y limitantes de estas herramientas, uno muy importante es el número máximo de unidades de gobierno que soporta cada tipo de Script. Gracias a la documentación oficial que brinda NetSuite podemos contar con esta información, pero al ser extensa y no contar siempre con ejemplos precisos, obligó al autor a realizar investigaciones y pruebas de contexto para confirmar que una solución pueda ser aplicada. Este desafío fue de gran importancia debido a que se tiene procesos que manejan grandes volúmenes de datos, lo que significó un análisis profundo del rendimiento del sistema y de aprovechar al máximo los SuiteScripts para tener la mejor solución.
- Adicionalmente, se presentó una necesidad del cliente para ejecutar un proceso propio del módulo estándar de NetSuite por medio de integraciones. Esto fue un desafío ya que el módulo fue diseñado para ser ejecutado por la interfaz de usuario y

al ser un producto de NetSuite, no permite modificar su código. Para superar esta limitante, se necesitaba conocer, a nivel de código, cómo se ejecuta el proceso, para lo cual se aprovechó que, aunque los scripts de la solución no se pueden modificar, en los ambientes de prueba fue posible revisar el código de los scripts y el autor pudo encontrar la estructura de la información necesaria para ejecutar el proceso desde un SuiteScript que facilitó la ejecución del proceso por medio de integración. Si bien la solución es simple, se invirtió tiempo en realizar una revisión exhaustiva de scripts, pruebas y ajustes hasta dar con la solución final.

Esta experiencia resultó ser enriquecedora ya que proporcionó al autor crecer a nivel profesional y obtener aprendizajes entre los que destacan:

- Extender conocimientos técnicos. Al revisar documentación y realizar pruebas, el autor fortaleció el dominio en la herramienta que le permitirá ofrecer soluciones cada vez más eficientes en otros proyectos.
- Extender conocimientos funciones. Estar en constante contacto con el equipo funcional permitió al autor experimentar el funcionamiento de los módulos estándar y conocer el punto de vista del usuario final, lo que brindó al autor mayor conocimiento del resultado esperado por parte de los usuarios finales y poder así brindar soluciones cada vez más efectivas.
- Desarrollar habilidades blandas. La constante interacción con el equipo funcional y líderes de equipos por medio de mensajes o participar en reuniones, permitió al autor asumir un rol más representativo dentro del proyecto y obtuvo mejoras en las capacidades de comunicación y liderazgo.

## **CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES**

### **5.1 Conclusiones**

En relación con los objetivos definidos en el presente trabajo de suficiencia profesional se detallan las siguientes conclusiones:

- La implementación del módulo personalizado para automatizar la gestión de contratos de alquiler permitió conocer a profundidad los procesos del cliente y realizar un levantamiento de requerimientos para asegurar que las soluciones desarrolladas en el ERP Oracle NetSuite estén alineadas con las necesidades del negocio. En base a este análisis se encontraron procesos críticos y reglas de negocio que no estaban cubiertas por los módulos estándar del ERP. Esta limitación hizo necesaria la personalización usando las herramientas SuiteCloud.
- La identificación y análisis de los procesos de gestión de contratos y requerimientos fiscales fue fundamental para definir correctamente la estructura a nivel de componentes del ERP y tipos de scripts a usar.
- El uso e implementación correcta de registros personalizados, campos a nivel de cuerpo y líneas de transacción, búsquedas guardadas, entre otros, resultó un factor determinante para asegurar la coherencia de información, automatización de procesos y facilitó el desarrollo de los SuiteScripts.
- El desarrollo del módulo personalizado mediante SuiteScripts permitió generar una solución integral alineada a los procesos críticos definidos por el cliente garantizando la automatización en la generación de transacciones, calendarización de facturación, actualización de rentas e integración con otros módulos del ERP.
- Las pruebas internas y con el usuario permitió validar el correcto funcionamiento de las soluciones y asegurar que cumpla con los requerimientos definidos. Estas etapas de prueba junto a la documentación técnica y funcional garantizaron la trazabilidad de cambios y/o ajustes. Esto facilita el mantenimiento futuro del sistema.

## 5.2 Recomendaciones

A partir de la experiencia obtenida por el desarrollo del proyecto, se identificaron ciertas áreas de oportunidad que podrían ser consideradas en futuras versiones o implementaciones de personalización.

- Fortalecer el proceso de levantamiento de requerimientos funcionales y técnicos. En el presente proyecto se siguió un proceso estructurado de levantamiento de requerimientos que incluye reuniones, entrevistas y validaciones documentadas con los usuarios finales. Esto permitió reducir retrabajos y evitar ambigüedades. Sin embargo, hubo casos donde no se llegó a identificar ciertos *gaps* con la funcionalidad estándar hasta una fecha cercana al término del proyecto, lo que significó realizar una investigación a fondo para brindar una solución alterna o nueva y por consecuencia, volver a priorizar tareas y reagendar fechas de entrega. Con el fin de minimizar este tipo de impactos que puedan poner en riesgo la implementación del ERP, se sugiere agregar en la planeación, tareas de pruebas de concepto para asegurar que la funcionalidad estándar cubre por completo las necesidades del cliente, y de no hacerlo, tener el tiempo necesario dentro del planeamiento para agendar y desarrollar una tarea para cubrirlo.
- Mantener actualizada la documentación técnica y funcional. El contar con un repositorio centralizado de documentación, scripts desarrollados y componentes creados, facilitó el desarrollo de actividades del proyecto, el seguimiento de nuevas tareas y la capacitación de nuevos colaboradores. En el presente trabajo se usaron las herramientas de Atlassian como Jira, Confluence y BitBucket.
- Considerar la integración con otros procesos del ERP como los reportes financieros con el fin de maximizar el valor del sistema y mejorar la toma de decisiones.

### 5.3 Fuentes de información

Alshalfi, A. (2018). ERP Systems and Its Impacts on the Business Operations. *American Journal of Information Systems*, 6(1), 1-4.

Nasirzada, N. (2025). ERP System Integration to Optimize Financial Reporting in Real Estate Management. *Universal Library of Business and Economics*, 2(1).

Vera, Á. B. (2006). Implementación de sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC. *Capic Review*, (4), 3.

Shivam Gupta, Subhas C. Misra, Ned Kock, David Roubaud, (2018) Organizational, technological and extrinsic factors in the implementation of cloud ERP in SMEs. *Journal of Organizational Change Management*, Vol. 31 Issue: 1, pp.83-102, <https://doi.org/10.1108/JOCM-06-2017-0230>

Hong, P. W., Dorasamy, M., Hong, L. J., & Malarvizhi, C. A. N. (2024). Exploring cloud enterprise resource planning and open innovation for small and medium enterprises: Insights from practitioners. *Journal of Open Innovation: Technology, Market, and Complexity*, 10(4), 100418.

Ley 29/1994, de 24 de noviembre, de Arrendamientos Urbanos, (Boletín Oficial del Estado, núm. 282, 25 de noviembre de 1994). Recuperado de <https://www.boe.es/eli/es/l/1994/11/24/29/con>

Ley 12/2023, de 24 de mayo, por el derecho a la vivienda, (Boletín Oficial del Estado, num 124, 26 de mayo de 2023). Recuperado de <https://www.boe.es/buscar/act.php?id=BOE-A-2023-12203>

Instituto Nacional de Estadística. (2025). Índice de referencia de arrendamientos de vivienda. Recuperado de [https://www.ine.es/dyngs/INEbase/operacion.htm?c=Estadistica\\_C&cid=1254736177110&menu=ultiDatos&idp=1254735976607](https://www.ine.es/dyngs/INEbase/operacion.htm?c=Estadistica_C&cid=1254736177110&menu=ultiDatos&idp=1254735976607)

Galán, J. S. (2021). Alquiler. *Economipedia*. Recuperado de <https://economipedia.com/definiciones/alquiler.html>

Junta de Andalucía. (2025). Contrato de alquiler. Recuperado de <https://www.juntadeandalucia.es/temas/vivienda-consumo/alquiler/contrato.html>

Oracle NetSuite (2025). Recuperado de <https://www.NetSuite.com/portal/es/company/why-NetSuite.shtml>

Oracle (2024). NetSuite SuiteCloud Platform. Recuperado de <https://www.NetSuite.com/portal/platform.shtml>

Oracle. (2024). ¿Qué es el ERP? Una guía completa. Recuperado de <https://www.NetSuite.com.mx/portal/mx/resource/articles/erp/what-is-erp.shtml>

Atlassian. (s. f.). Introducción a Jira. Atlassian. Recuperado de <https://www.atlassian.com/es/software/jira/guides/getting-started/introduction#what-is-jira-software>

Atlassian. (s. f.). Acerca de Confluence Recuperado de <https://www.atlassian.com/es/software/confluence/resources/guides/get-started/overview#about-confluence>

Git. (s. f.). Git. Recuperado de <https://git-scm.com/>

Microsoft. (s. f.). Visual Studio Code. Recuperado de <https://code.visualstudio.com/>

Stakić, Đ., & Đurđić, D. (2023). The role and importance of programming languages in the application of ERP solutions. Dynamics, 365, 31.

Kelly, J., Braud, L., & Huffman, M. (2010). NetSuite for dummies. John Wiley & Sons.

Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) agile & waterfall model advantages, disadvantages, and application suitability in software quality engineering.