

**ANALISIS Y DISEÑO DE UN APLICATIVO PARA EL CONTROL
PRESUPUESTAL EN LAS OFICINAS REGIONALES DEL CIAT**

PAOLA ANDREA CRUZ ARIAS

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE OPERACIONES Y SISTEMAS
PROGRAMA INGENIERÍA INFORMÁTICA
SANTIAGO DE CALI
2012**

**ANALISIS Y DISEÑO DE UN APLICATIVO PARA EL CONTROL
PRESUPUESTAL EN LAS OFICINAS REGIONALES DEL CIAT**

PAOLA ANDREA CRUZ ARIAS

**Pasantía Institucional para optar el título de
Ingeniero en informática**

Director:

**HERNANDO GONZALEZ UMAÑA
Ingeniero Agrícola**

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE OPERACIONES Y SISTEMAS
PROGRAMA INGENIERÍA INFORMÁTICA
SANTIAGO DE CALI
2012**

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniero en Informática.

ING. OLMEDO ARCILA GUZMÁN
Jurado

ING. ANTONIO LEMOS BENAVIDES
Jurado

Santiago de Cali, 23 de Noviembre de 2012

CONTENIDO

RESUMEN	12
INTRODUCCIÓN	13
1. PLANTEAMIENTO DEL PROBLEMA	15
2. JUSTIFICACIÓN	17
3. OBJETIVOS	18
3.1 OBJETIVO GENERAL	18
3.2 OBJETIVOS ESPECIFICOS	18
4. MARCO DE REFERENCIA	19
4.1 MARCO TEÓRICO	19
4.1.1 APLICACIÓN WEB	19
4.1.2 ERP	20
4.1.3 LIBRO DIARIO DE CONTABILIDAD (JOURNAL)	20
4.1.4 WEB JOURNAL	21
4.1.5 ENCUMBRANCE	22
5. METODOLOGÍA	23
5.1 FASES DEL PROYECTO	23
5.1.1 INICIO	23

5.1.2 ANALISIS	23
5.1.3 DISEÑO	23
5.1.4 CIERRE	24
6. REQUERIMIENTOS DEL SISTEMA	25
6.1 PROPÓSITO DEL SISTEMA	25
6.2 ALCANCE DEL SISTEMA	25
6.3 DESCRIPCIÓN DEL SISTEMA	25
6.4 ACTORES DEL SISTEMA	25
6.5 REQUERIMIENTOS FUNCIONALES DEL SISTEMA	26
7. CASOS DE USO Y REQUERIMIENTOS RELACIONADOS	34
8. DIAGRAMAS DE CASOS DE USO	36
9. GUIONES DE CASOS DE USO	37
10. DIAGRAMA ENTIDAD RELACIÓN	38
11. ARQUITECTURA DE LA APLICACIÓN	39
11.1 VISTA LÓGICA	41
11.2 VISTA DE DESARROLLO	42
11.3 VISTA FÍSICA	45
11.4 VISTA DE PROCESOS	46
12. POLITICAS DE USUARIOS	47

13. CONEXIONES, ESTADOS Y PROCEDIMIENTOS	50
14. DIAGRAMAS DE CLASE	53
15. DIAGRAMAS DE SECUENCIA	54
16. PLAN DE PRUEBAS	55
16.1 INTRODUCCIÓN	55
16.1.1 Propósito	55
16.1.2 Ámbito	55
16.2 REQUERIMIENTOS DE LAS PRUEBAS	55
16.2.1 Pruebas de integridad de la base de datos y de los datos	55
16.2.2 Pruebas de funcionalidad	55
16.2.3 Pruebas de interfaz de usuario	55
16.2.4 Pruebas de desarrollo	56
16.3 ESTRATEGIA DE PRUEBA	56
16.3.1 Pruebas de integridad de la base de datos y de los datos	56
16.3.2 Pruebas de funcionalidad	57
16.3.3 Pruebas de interfaz de usuario	71
16.3.4 Pruebas de desarrollo	72
17. CONCLUSIONES	73
18. RECOMENDACIONES	75

19. BIBLIOGRAFIA **76**

20. ANEXOS **77**

LISTA DE CUADROS

Cuadro 1. Relación entre casos de uso y requerimientos	32
Cuadro 2. Guión para el caso de uso CU33 Guardar Journal	35
Cuadro 3. Pruebas de integridad	56
Cuadro 4. Pruebas de funcionalidad	57
Cuadro 5. Lista de casos de prueba CU33	58
Cuadro 6. Caso de prueba # 1	60
Cuadro 7. Caso de prueba # 2	61
Cuadro 8. Caso de prueba # 3	61
Cuadro 9. Caso de prueba # 4	61
Cuadro 10. Caso de prueba # 5	62
Cuadro 11. Caso de prueba # 6	62
Cuadro 12. Caso de prueba # 7	62
Cuadro 13. Caso de prueba # 8	63
Cuadro 14. Caso de prueba # 9	63
Cuadro 15. Caso de prueba # 10	63
Cuadro 16. Caso de prueba # 11	64
Cuadro 17. Caso de prueba # 12	64
Cuadro 18. Caso de prueba # 13	64

Cuadro 19. Caso de prueba # 14	65
Cuadro 20. Caso de prueba # 15	65
Cuadro 21. Caso de prueba # 16	65
Cuadro 22. Caso de prueba # 17	66
Cuadro 23. Caso de prueba # 18	66
Cuadro 24. Caso de prueba # 19	66
Cuadro 25. Caso de prueba # 20	67
Cuadro 26. Caso de prueba # 21	67
Cuadro 27. Caso de prueba # 22	67
Cuadro 28. Caso de prueba # 23	68
Cuadro 29. Caso de prueba # 24	68
Cuadro 30. Caso de prueba # 25	68
Cuadro 31. Caso de prueba # 26	69
Cuadro 32. Caso de prueba # 27	69
Cuadro 33. Caso de prueba # 28	69
Cuadro 34. Caso de prueba # 29	70
Cuadro 35. Caso de prueba # 30	70
Cuadro 36. Caso de prueba # 31	70
Cuadro 37. Caso de prueba # 32	71

Cuadro 38. Pruebas de interfaz de usuario	71
Cuadro 39. Pruebas de desarrollo	72

LISTA DE FIGURAS

Figura 1. Diagrama de caso de uso 33 (Guardar Journal)	34
Figura 2. MER	37
Figura 3. Arquitectura 1	40
Figura 4. Arquitectura 2	41
Figura 5. Arquitectura 3	42
Figura 6. Arquitectura 4	44
Figura 7. Arquitectura 5	45
Figura 8. Arquitectura 6	45
Figura 9. Arquitectura 7	46
Figura 10. Arquitectura 8	51
Figura 11. Envio de comprobantes	52
Figura 12. Diagrama de clases caso de uso 33	53
Figura 13. Diagrama de clases secuencia de uso 33	54

RESUMEN

El proyecto Analisis y diseño de un aplicativo para el control presupuestal en las oficinas regionales del ciat se realizó con el fin de suplir una necesidad tecnologica del centro internacional de agricultura tropical relacionada con el control presupuestal fuera de ciat headquarters.

Palabras claves: CIAT, JOURNALS, PRESUPUESTO, COMPROMETIDO.

INTRODUCCIÓN

CIAT es una institución de investigación agrícola sin ánimo de lucro dedicada a encontrar soluciones científicas para remediar el problema de hambre en las zonas tropicales, desarrollando métodos sostenibles para la producción de alimentos.

Inicia sus operaciones en 1970 y hace parte de los centros de investigación de CGIAR (Consultative Group of International Agricultural Research), su sede principal está ubicada en Palmira, muy cerca a Cali (Colombia). Cuenta con el apoyo de algunos de los mejores científicos del mundo, como también, de científicos colombianos que colaboran en las distintas oficinas en África y Asia. Actualmente CIAT ha trabajado en proyectos en más de 50 países alrededor del mundo.

Las principales áreas de investigación del CIAT están relacionadas con investigación genética, fertilidad y manejo de suelos, análisis de políticas e investigación en cambio climático, análisis de información geográfica y agro biodiversidad, CIAT Colombia, está encargado de la investigación en cultivos de Frijol, Yuca, Arroz, Forrajes y Frutas tropicales, además de contar con el área de recursos genéticos quien maneja uno de los bancos de germoplasma más grandes del mundo conservando las colecciones más grandes para frijol, yuca y forrajes tropicales, los cuales han sido recolectados en más de 141 países.

El CIAT recibe sus fondos a través del CGIAR, provenientes de distintos países, organizaciones, fundaciones y universidades alrededor del mundo que contribuyen con sus donaciones.

Aunque el objetivo principal del centro es la investigación agrícola que lidera cada uno de los programas de CIAT debe existir un fuerte apoyo de la parte administrativa para poder administrar eficientemente los recursos, por esta razón existe el área de servicios corporativos la cual consiste en servicios centrales (alimentos y vivienda, operaciones de campo, compras y requisiciones, etc.), finanzas y grants management (manejo de recursos provenientes de donaciones), recursos humanos, oficina legal y unidad de sistemas de información.

La unidad de sistemas de información (USI), es la encargada de suministrar el apoyo tecnológico a todas las áreas del centro tanto administrativas como de

investigación, enriqueciendo la producción laboral de todo el personal y a la vez, permite optimizar los procesos internos, haciéndolos más ágiles.

Una de las necesidades tecnológicas de la institución es llegar a controlar, de manera sistematizada, el manejo presupuestal en las diferentes oficinas regionales en África y Asia, registrando y controlando todas las transacciones que se realicen allí, de esta manera se logrará no solo disminuir costos en licencias de software y en personal financiero, si no que habrá un control total de los movimientos financieros de las regiones monitoreado por el personal del área de finanzas desde CIAT headquarters.

1. PLANTEAMIENTO DEL PROBLEMA

Por ser el CIAT una organización internacional sin ánimo de lucro, la naturaleza pública de sus recursos financieros, requiere de políticas que permitan guiar el manejo de estos, en forma clara y transparente. Las decisiones tomadas respecto a las inversiones provenientes de estos fondos, deben tomarse centralmente desde CIAT Headquarters, en Cali, Colombia.

Actualmente, los sistemas de información de CIAT son soportados por Oracle y se cuenta con algunos de los módulos de la E-Business suite como el Oracle General Ledger de Oracle Financials. Esta aplicación es una herramienta de administración financiera, diseñada para ingresar y hacer reportes de datos de las transacciones realizadas, de manera efectiva y controlada, dando la posibilidad de obtener resultados y análisis en tiempo real.

CIAT trabaja actualmente con 24 oficinas regionales en Asia y África, donde se desarrollan diferentes proyectos financiados por una o varias entidades de la lista de donadores de CIAT. El presupuesto destinado a estos proyectos, es mayoritariamente manejado desde CIAT Headquarters y los asistentes financieros ubicados en las sedes regionales deben llevar registros, generar los libros de contabilidad y enviarlos a CIAT Headquarters en formato Excel, donde son ingresados al sistema contable centralizado. Este proceso se lleva a cabo así, debido a la imposibilidad presupuestal de adquirir licencias de Oracle para cada uno de los usuarios que requieran ingresar sus transacciones. De descartar la opción de compra de nuevas licencias para las regionales, surge la necesidad de aplicar el método de los libros en formato Excel.

Este proceso presenta varios inconvenientes:

- Pérdida de tiempo al sobrescribir una y otra vez la información de los libros de contabilidad hasta que finalmente son ingresados al sistema.
- Ausencia de controles, lo que genera errores y confusiones en el manejo del presupuesto.
- La información necesaria no está disponible oportunamente para su consulta y análisis.
- Las dificultades de presupuesto y compatibilidad no permiten utilizar otras herramientas de software contable disponibles en el mercado.

Se desea analizar y diseñar una solución que permita facilitar el manejo presupuestal en las regiones de Asia y África. Este software debe contar con características especiales que faciliten la resolución de los problemas que se generan actualmente con el método tradicional y a su vez, deben adaptarse a las condiciones tecnológicas del centro y de las sedes regionales, aprovechando los recursos tecnológicos existentes.

Una de las principales características con las que debe contar la solución, es que debe ser totalmente compatible con la E-business suite de Oracle, lo que permitiría el ingreso de la información al sistema central mediante los usuarios que cuenten con licencias. El software deberá validar las autorizaciones y trabajar en línea con Oracle E-Business suite.

Actualmente se han adquirido licencias de Oracle E-Business suite en Colombia, Kenia y Uganda y solo desde estas ubicaciones, se puede acceder al General Ledger de Oracle para autorizar transacciones, aprobar gastos según el presupuesto e ingresar información al sistema para su posterior consulta y análisis.

Se requiere que la herramienta final se ejecute online, teniendo en cuenta que sea completamente multiusuario y que permita compartir la información en tiempo real. Es prioritario que sea flexible en cuanto a las características de hardware y software de cada equipo que la utilice, teniendo en cuenta, la baja conectividad en algunas áreas remotas, lo que conlleva a la creación de un software liviano que consuma pocos recursos.

Por ser una aplicación de trabajo online, debe contar con parámetros de seguridad que garanticen la integridad, disponibilidad y confidencialidad de la información y a la vez, cumplir con las políticas y técnicas de seguridad previamente establecidas por el CIAT, como poder acceder a la información de los usuarios y sus cuentas en el Directorio Activo de CIAT.

El CIAT cuenta con su propia red VPN (virtual private network) para interconectar sus oficinas. Este recurso debe ser tenido en cuenta durante el análisis y el diseño para ser aprovechado a futuro.

2. JUSTIFICACIÓN

Es indispensable encontrar una solución para el manejo presupuestal en las oficinas regionales de CIAT, que se refleje en beneficios económicos significativos para el centro, entre ellos, ahorro de dinero en licencias Oracle, asegurando así, mayor disponibilidad de recursos para el apoyo a la investigación; debe facilitar al CIAT, tener un mayor control del manejo presupuestal, disponiendo de la información de todas las transacciones realizadas para una oportuna consulta y análisis y así, permitir una rápida toma de decisiones en cuanto al presupuesto, de acuerdo a las solicitudes recibidas con anterioridad.

Este proyecto permitirá solucionar los inconvenientes que trae consigo el método tradicional, beneficiando a los asistentes financieros de las sedes regionales y al Departamento de finanzas del CIAT Headquarters, mejorando su productividad, al hacer los procesos más fáciles, ágiles y seguros, permitiendo a su vez, un mayor control y transparencia. También traerá beneficios para la Unidad de Sistemas de Información, quienes son los encargados directos de solucionar los problemas y confusiones que surgen por error humano en el proceso actual.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Realizar el análisis y diseño de una solución basada en web, que trabaje en línea con Oracle E-Business Suite para facilitar el control de la ejecución presupuestal en las oficinas regionales de CIAT ubicadas en Asia y África

3.2. OBJETIVOS ESPECIFICOS

- Analizar en detalle, el proceso del manejo de la ejecución presupuestal en las oficinas regionales de CIAT
- Realizar el levantamiento de requerimientos del sistema.
- Realizar la debida documentación, según las metodologías utilizadas en el CIAT.
- Diseñar de la base de datos.
- Diseñar de la arquitectura.
- Diseñar la aplicación.
- Diseñar plan de pruebas.

4. MARCO DE REFERENCIA

4.1. MARCO TEÓRICO

4.1.1 Aplicación web. ¹ Las aplicaciones Web son aplicaciones basadas en navegadores de internet y son codificadas en lenguajes especiales soportados por estos navegadores, por lo tanto, se ejecutan de forma online, esto quiere decir, que no debemos instalar ninguna aplicación en el computador para acceder a ella.

Las aplicaciones web residen en un servidor y son accedidas por medio de cualquier navegador de internet. Esto trae ventajas, especialmente en aplicaciones multiusuario que necesitan estar actualizadas y donde es indispensable compartir información en tiempo real. Otra ventaja es la flexibilidad que proporciona debido a que no hay que instalar o actualizar la aplicación para cada usuario, ahorrando tiempo y espacio en el disco duro y además, funciona independientemente de las características de hardware de cada equipo o de su sistema operativo o navegador que utilice. Una gran ventaja de ejecutar el software por medio del servidor es que minimiza considerablemente el consumo de recursos para cada usuario, lo que compensa considerablemente la desventaja de trabajar siempre a través de un navegador, donde las aplicaciones están limitadas a las funcionalidades de éste, disminuyendo las posibilidades que brinda una aplicación de escritorio. La mayor desventaja es que crea una necesaria dependencia de la disponibilidad del servicio de internet y del servidor general. No debe confundirse el término aplicación web con el de página web aunque a simple vista parezcan lo mismo.

¹ FOWLER, Susan. y STANWICK, Victor. Web application design handbook, M. Kaufmann Publishers 2004.

4.1.2 Erp. ² Una solución ERP (Enterprise resource planing) integra la información administrativa de una organización, incluyendo las áreas de producción, finanzas y contabilidad, recursos humanos etc. y su principal función es facilitar el flujo de información entre las diferentes áreas. Utiliza una base de datos centralizada para el almacenamiento de toda la información y cada uno de los módulos está diseñado de manera similar y de fácil uso para los usuarios.

Para implementar un sistema ERP en una organización, se debe tener un amplio conocimiento del negocio y sus procesos. La configuración de un sistema ERP es una labor ardua debido a las diferentes características de cada organización.

Las soluciones ERP, SAP y Oracle, son unas de las más utilizadas en el momento.

4.1.3 Libro diario de contabilidad (journal). ³ Es un libro de contabilidad donde se registran diariamente todas las operaciones de carácter económico, de manera cronológica y en forma detallada, para luego trasladar los movimientos débitos y créditos a las cuentas del libro mayor (General Ledger).

Una de las principales características es que se diseña en columnas dobles para registrar en ellas el debe y el haber que resultan del resumen diario de las operaciones.

Así pues el número de columnas varía de acuerdo con las necesidades y el número de cuentas que se utilizaran en la contabilidad de la empresa, para el caso del CIAT, las cuentas se encuentran bien definidas con sus nombres, códigos y descriptores en la base de datos de ORACLE financiero.

En el diseño del libro diario existe una columna para registrar la fecha del comprobante y además, la sección donde se añade el detalle o la descripción con el número correspondiente del comprobante registrado, en el aplicativo Web Journals esta información se encuentra detallada en la cabecera de los comprobantes (Transaction).

² BECERRA, Miguel. La fecundidad [en línea]. Bogotá D.C.: ACIS, 2012 [consultado 02 de Agosto de 2011]. Disponible en Internet: <http://www.acis.org.co/fileadmin/Conferencias/ConfMiguelBecerraSep6.pdf>

³ BOLAÑOS, César A. y ALVAREZ NIÑO, Jorge, Contabilidad. Comercial: mejores recursos humanos en la empresa Norma, 1995.

En el mercado se encuentran actualmente disponibles un gran número de sistemas de información contable, estos sistemas ejecutan por lo general 3 pasos básicos, los datos se deben registrar, clasificar y resumir.

La contabilidad va más allá del proceso de creación de registros e informes. El objetivo final de la contabilidad es la utilización de esta información, su análisis e interpretación y al registrar, clasificar y resumir de manera automática la información contable.

En resumen sin los libros de contabilidad no podría llevarse un registro detallado de los movimientos contables y flujo de dinero de una organización, para CIAT llevar un control detallado del flujo de dinero es de vital importancia ya que se trabaja con dinero de donaciones y debe existir absoluta transparencia en este tipo de procesos.

Los sistemas de información agilizan, hacen más precisos los procesos y facilitan el análisis contable a los expertos.

4.1.4 Webjournal. ⁴ Es la versión web de un libro diario o Journal, al igual que un libro normal debe poder registrar información acerca del movimiento de las operaciones diarias, esto incluye la fecha y número del comprobante, las cuentas, los valores débito y crédito entre otros.

Un Web Journal, al trabajar sobre una aplicación web ofrece gran cantidad de ventajas frente a un libro consignado en papel, entre estos la posibilidad de almacenamiento, la facilidad para compartir información, el control, la precisión en los cálculos y la capacidad de interactuar con otras interfaces y la facilidad para generar reportes financieros.

En la mayoría de aplicaciones contables y financieras incluido el Oracle General Ledger de Oracle Financials. Generalmente se pueden ingresar transacciones de dos maneras, ingresar los movimientos contables al sistema de manera individual o agruparlos en un conjunto de transacciones con atributos comunes conocidos como Journal Batches. El sistema verifica que las transacciones se encuentren debidamente balanceadas antes de ser ingresadas al sistema, para así disminuir las probabilidades de error.

El aplicativo Web Journals está construido de manera que las transacciones puedan ingresarse a modo de Journal Batches, de esta manera en una sola

⁴ DARLINGTON, Taite. Oracle General Ledger User Guide, Oracle Corporation, 2003.

transacción pueden ingresarse varias líneas que pueden ser aprobadas individualmente.

Se ha decidido que el desarrollo del aplicativo contable para CIAT se construirá a modo de Web Journal ya que principalmente será utilizado para ingresar transacciones desde las diferentes regiones donde la unidad de sistemas de información no podría brindar soporte a un aplicativo de escritorio.

4.1.5 Encumbrance.⁵ Es un término muy utilizado en las aplicaciones contables, también se conoce como presupuesto comprometido o solicitud de comprometido.

Es una transacción que ocurre cuando se separa un presupuesto en un centro de costo para un propósito específico, el dinero no ha sido gastado aun pero aparece separado para que nadie más pueda gastarlo.

El propósito principal de controlar los presupuestos comprometidos es prevenir el sobre gasto de dinero y predecir el comportamiento de los flujos de caja.

En el CIAT se ha venido utilizando el sistema de comprometidos desde hace algunos años con bastante éxito ya que a final de cada año los programas deben presentar un informe de los gastos estimados para el próximo periodo, esto se debe en parte a que en la actualidad existe un seguimiento bastante riguroso en los movimientos financieros del centro ya que son indispensables para realizar proyecciones.

⁵ DARLINGTON, Taite. Oracle General Ledger User Guide, Oracle Corporation, 2003.

5. METODOLOGÍA

La metodología utilizada para el desarrollo del proyecto deberá ser consecuente con los estándares utilizados por la unidad de sistemas de información del CIAT, para la realización de los diferentes proyectos en el área informática, los cuales están basados en gran parte, en el modelo RUP (Rational Unified Process).

Esta metodología es una de las metodologías estándar para el proceso de software más utilizadas en el mundo debido a que garantiza un proceso de desarrollo disciplinado y permite establecer el cumplimiento de buenas prácticas en ingeniería de software. También plantea un modelado visual del software que es apoyado por el lenguaje de modelamiento UML (Unified Modeling Language) y sus diferentes tipos de diagramas, la metodología para la representación de estos diagramas no sigue rigurosamente el estándar UML sino que consiste en una aproximación.

5.1 FASES DEL PROYECTO

5.1.1 INICIO

En la fase de inicio o conceptualización, se recopilará la información necesaria para dar inicio al proyecto se define el modelo del negocio, el alcance, objetivos y se realiza la planeación de actividades del proyecto.

5.1.2 ANALISIS

En la fase de análisis se realiza el levantamiento de requerimientos y especificación de casos de uso, en esta fase quedaran definidas todas las funcionalidades que tendrá el sistema.

5.1.3 DISEÑO

En esta fase se realizara el proceso de diseño de la aplicación, este incluye el diseño de la base de datos, del modelo de datos, de la arquitectura, las conexiones y las pruebas, se definirán las políticas administrativas de la aplicación y se crearan los diagramas UML que se consideren importantes.

5.1.4 CIERRE

En la fase de cierre del proyecto se recopilara toda la información y los documentos generados y se creara el informe final.

6. REQUERIMIENTOS DEL SISTEMA

6.1 PROPÓSITO DEL SISTEMA

El sistema Web Journals permitirá trabajar en línea con Oracle E-Business Suite para facilitar el control de la ejecución presupuestal en las oficinas regionales de CIAT ubicadas en Asia y África, permitiendo a los usuarios reservar fondos y registrar transacciones “debito/crédito”, manejando las debidas aprobaciones.

6.2 ALCANCE DEL SISTEMA

El sistema Web Journals permitirá a los usuarios crear solicitudes de comprometido o “Encumbrances”, es decir reservar fondos para transacciones futuras y a su vez crear comprobantes relacionados con estas transacciones, una vez aprobados los comprobantes, estos deben poder ser enviados al sistema principal “Oracle” por usuarios autorizados.

6.3 DESCRIPCIÓN DEL SISTEMA

Ambiente web disponible a través de la VPN para las regiones, ingresando desde un navegador de Internet.

6.4 ACTORES DEL SISTEMA

Los actores del sistema pueden agruparse en 3 perfiles principales:

- **Web Journal Approver:** es un usuario que tiene licencia Oracle y acceso al Oracle General Ledger, son los únicos que pueden crear “Encumbrances” y que pueden aprobar e ingresar las transacciones creadas por los Journalists.
- **Journalist:** Es el usuario que ingresa las transacciones, por lo general es un asistente financiero ubicado en alguna estación regional.
- **Administrador:** Es el encargado de administrar el sistema, de crear usuarios, asignar responsabilidades. Etc.

6.5 REQUERIMIENTOS FUNCIONALES DEL SISTEMA

Los requerimientos funcionales del sistema se detallaran por módulos.

Módulo de inicio de sesión

- 1. El sistema debe permitir solo a los usuarios validos y activos ingresar a la aplicación.
- 2. El sistema debe permitir a un usuario tener acceso únicamente a las funcionalidades disponibles de acuerdo a su rol.
- 3. El sistema debe permitir al usuario finalizar su sesión en la aplicación o debe cerrarla automáticamente después de un periodo de inactividad.

Módulo de Usuario

(Solo disponible para usuarios que tengan asignados los permisos de Users administration)

- 4. El sistema debe permitir agregar usuarios que estén registrados en el directorio activo del CGIAR, si el login digitado coincide con uno existente, el sistema debe autocompletar la información en los campos Name, Email y no solicitara la entrada de un Password. Durante la creación del usuario también se debe indicar si este será o no aprobador y si no lo es se deberá escoger uno entre la lista de usuarios aprobadores, también se deberá definir el rol que tendrá el usuario en la aplicación y sus reglas de seguridad.
- 5. El sistema debe permitir agregar usuarios particulares de la aplicación que NO estén registrados en el directorio activo del CGIAR. En este caso deben definirse los campos Name, Email y Password, indicar si este será o no aprobador y si no lo es se deberá escoger uno entre la lista de usuarios aprobadores, también se deberá definir el rol que tendrá el usuario en la aplicación y sus reglas de seguridad.
- 6. Al momento de la creación de un usuario, el sistema debe poder comprobar si un usuario es un Web Journal Approver (Usuario Oracle) o no.
- 7. El sistema debe permitir visualizar una lista con la información básica de los usuarios, esta información consiste en el Login, Nombre, Email, Status, Rol y si es aprobador o no.

- 8. El sistema debe permitir visualizar la información en detalle de un usuario registrado, esta información consiste en la información básica más el nombre de su aprobador si lo tiene y los valores definidos para cada una de las reglas de seguridad.
- 9. El sistema debe permitir buscar un usuario por el nombre o parte del nombre.
- 10. El sistema debe permitir actualizar cierta información de los usuarios: Nombre, Email, Status, Rol, si es aprobador o no, nombre del aprobador si lo tiene y el valor para cada una de las reglas de seguridad, la única información que no podrá modificarse será el Login.
- 11. El sistema debe permitir eliminar un usuario mientras este no tenga ninguna transacción asociada.

Módulo de Rol

(Solo disponible para usuarios que tengan asignados los permisos de Roles administration)

- 12. El sistema debe permitir crear un nuevo rol, solo se debe definir un nombre y una descripción.
- 13. El sistema debe permitir listar la información de los roles que han sido creados hasta el momento.
- 14. El sistema debe permitir visualizar las diferentes opciones disponibles para cada rol.
- 15. El sistema debe permitir asignar una o varias de las opciones disponibles para un rol.
- 16. El sistema debe permitir desasignar una o varias de las opciones disponibles para un rol.
- 17. El sistema debe permitir modificar la descripción de un rol.
- 18. El sistema debe permitir eliminar un rol, mientras este no esté asociado con ningún usuario.

Módulo de Opciones

(Solo disponible para usuarios que tengan asignados los permisos de Options administration)

Las opciones son funcionalidades del sistema previamente creadas como por ejemplo users administration o enter journal, también pueden crearse descriptivos para nuevas opciones.

- 19. El sistema debe permitir crear una nueva opción, el usuario deberá definir un nombre para la opción, también puede definir: nombre de link, link target, URL (descriptivos opcionales).
- 20. El sistema debe permitir visualizar las opciones disponibles.
- 21. El sistema debe permitir modificar el nombre de una opción, nombre de link, link target, URL.
- 22. El sistema debe permitir eliminar una opción, mientras esta no esté relacionada con ningún rol.

Módulo Enter Encumbrance Request

(Solo disponible para usuarios que tengan asignados permisos de enter encumbrance request "Usuarios Oracle")

- 23. El sistema debe permitir al usuario ingresar los datos para crear una nueva solicitud de comprometido, los datos de la cabecera del comprobante son: nombre del comprobante, categoría, compañía, periodo, fecha, tipo de moneda, fecha y tipo de conversión. Los datos del detalle del comprobante que deben ser ingresados son: cuenta, centro de costo, auxiliar, valor, descripción, tipo y número de documento. 23
- 24. El sistema debe verificar todos los datos al momento que el usuario los ingresa, esta verificación incluye caracteres especiales (espacio, dos puntos, punto, coma, etc.), campos en blanco, formatos incorrectos, máximo de caracteres etc.
- 25. El sistema debe permitir al usuario almacenar los datos de una nueva solicitud de comprometido.
- 26. El sistema debe revisar al momento de guardar, que todos los datos sean validos, esto incluye verificación de caracteres especiales (espacio, dos puntos, punto, coma, etc.), campos en blanco, formatos incorrectos, máximo de caracteres etc.
- 27. El sistema debe poder validar la solicitud de comprometido.

- 28. El sistema debe poder insertar los datos de la solicitud almacenados en la tabla local a la interfaz GL de Oracle, una vez esta se encuentre en estado validada.

Módulo List Encumbrance

(Solo disponible para usuarios que tengan asignados permisos de list encumbrance)

- 29. El sistema debe permitir al usuario visualizar una lista con la información de las solicitudes de comprometidos relacionados con sus estaciones, incluyendo su estado actual.
- 30. El sistema debe permitir buscar solicitudes de comprometido filtrando resultados según su status, ID de transacción, ID de grupo, tipo de documento, número de documento, descripción, periodo, valor en dólares, compañía, cuenta, centro de costo y usuario Oracle.
- 31. El sistema debe permitir al usuario visualizar una solicitud de comprometido guardada con anterioridad. Si la solicitud tiene reversiones se debe visualizar en un formulario especial que muestre los detalles, si la solicitud no tiene asociadas reversiones podrá mostrarse en el mismo formulario de ingreso de solicitudes.
- 32. El sistema debe permitir al usuario modificar los datos de una solicitud de comprometido que haya creado con anterioridad. Se pueden modificar los datos de la cabecera del comprobante: nombre del comprobante, categoría, compañía, periodo, fecha, tipo de moneda, fecha y tipo de conversión y los datos del detalle del comprobante: cuenta, centro de costo, auxiliar, valor, descripción, tipo y número de documento. Mientras la solicitud se encuentre en estado Unposted, Error o Validated.
- 33. Si se encuentra algún error durante la validación de la solicitud, se mostrará al usuario una descripción de este para que el usuario pueda corregirlo fácilmente.
- 34. Se debe poder reversar parcial o totalmente una solicitud de comprometido, se debe validar que el valor a reversar sea menor o igual al comprometido original.
- 35. El sistema debe permitir la generación de un archivo tipo Excel con los datos de la solicitud para su posterior impresión o almacenamiento.
- 36. El sistema debe permitir al usuario eliminar una solicitud de comprometido. Un usuario solo podrá borrar solicitudes creadas por el mismo mientras esta se encuentre en estado Unposted, Error o Validated.

Módulo Enter Journal

(Solo disponible para usuarios que tengan asignados permisos de enter journal)

- 37. El sistema debe permitir al usuario ingresar los datos para crear un nuevo comprobante contable. los datos de la cabecera del comprobante son: nombre del comprobante, categoría, compañía, periodo, fecha, tipo de moneda, fecha y tipo de conversión, los datos de cada uno de los detalles del comprobante que deben ser ingresados son: cuenta, centro de costo, auxiliar, valor debito o valor crédito, descripción, tipo y número de documento.
- 38. El sistema debe verificar todos los datos al momento que el usuario los ingresa. Esta verificación incluye caracteres especiales (espacio, dos puntos, punto, coma, etc.), campos en blanco, formatos incorrectos, máximo de caracteres etc.
- 39. El sistema debe facilitar la adición de tantos registros contables como el usuario desee incluir en cada comprobante.
- 40. El sistema debe permitir mostrar en la misma pantalla toda la información del comprobante, la información de los registros que se esté adicionando antes de subirlo a memoria.
- 41. El sistema debe permitir guardar el comprobante contable en la base de datos local.
- 42. El sistema debe validar para cada uno de los registros contables del comprobante que se está creando o que ya se creó en memoria, que los campos asociados a una lista de valores adquieran un valor valido.
- 43. El sistema debe ofrecer la posibilidad de validar, si los valores de los registros del comprobante de tipo crédito, coinciden con el valor de los de tipo debito. Además cada uno de estos campos puede ser tanto negativo como positivo.
- 44. El sistema debe validar para cada uno de los registros contables del comprobante que se está creando o que ya se creó en memoria, que exista disponibilidad presupuestal para la transacción que lo requiera.
- 45. El sistema debe validar para cada uno de los registros contables del comprobante que se está creando o que ya se creó en memoria, que no se violen reglas de seguridad, reglas de validación cruzada y los campos requeridos por este.

- 46. Si el usuario es un usuario Oracle, el sistema debe poder insertar los datos del comprobante y sus respectivos detalles almacenados en la tabla local a la interfaz GL de Oracle financiero, una vez esta se encuentre en estado validado.
- 47. El sistema debe tomar los resultados del envío de cada comprobante a Oracle y mostrar si la transacción fue exitosa.
- 48. Si el usuario no es un usuario Oracle, el sistema debe permitir enviar el comprobante al debido aprobador para que este pueda ingresarlo al Oracle financiero cada uno de los comprobantes con sus respectivos registros que se encuentren almacenados en la base de datos local.
- 49. El sistema debe permitir a los usuarios enviar un correo electrónico automático informativo a su debido aprobador al momento de solicitar la aprobación del comprobante.

Módulo List Journal

(Solo disponible para usuarios que tengan asignados permisos de List Journal)

- 50. El sistema debe permitir al usuario visualizar una lista con la información de los comprobantes relacionados con sus estaciones, incluyendo su estado actual.
- 51. El sistema debe permitir buscar comprobantes filtrando resultados según su status, ID de transacción, ID de grupo, nombre del Journal, usuario, usuario Oracle, compañía, categoría, periodo y moneda.
- 52. El sistema debe permitir seleccionar uno de los comprobantes contables y ver toda su información incluyendo la lista de los registros que ya se hayan subido a memoria con anterioridad para este comprobante. Esta información se mostrará en el mismo formulario de ingreso de comprobantes.
- 53. El sistema debe mostrar los detalles de cada uno de los registros contables que se hallan adicionado al comprobante que se encuentre en memoria.
- 54. El sistema debe permitir la modificación de cada uno de los campos de los registros contables del comprobante que se encuentre en memoria mientras este en estado Unposted, Error o Validated.
- 55. El sistema debe permitir la generación de un archivo tipo Excel con los datos del comprobante para su posterior impresión o almacenamiento.

- 56. El sistema debe permitir la eliminación de cada uno de los campos de la cabecera los registros contables o de la línea entera del comprobante que se encuentre en memoria mientras este en estado Unposted, Error o Validated.
- 57. El sistema debe buscar si hay solicitudes de comprometido relacionadas con el comprobante que se está revisando en el momento.
- 58. El sistema debe permitir la realización de una reversión en caso de que exista alguna solicitud de comprometido relacionada con el comprobante que se está revisando en el momento.
- 59. El sistema debe permitir a los usuarios Oracle aprobar una o varias líneas de un comprobante.
- 60. El sistema debe permitir a los usuarios Oracle desaprobado una o varias líneas de un comprobante.

Módulo Reports

(Solo disponible para usuarios que tengan asignados permisos de Reports)

- 61. El sistema debe permitir a un usuario seleccionar entre varios tipos de reportes pre construidos.
- 62. El sistema debe permitir a un usuario seleccionar en que formato desea ver el reporte solicitado, las opciones deben ser HTML, PDF y Excel.
- 63. El sistema deberá permitir al usuario seleccionar el tipo de reporte que desea generar, dependiendo de este, el usuario podrá determinar algunos parámetros para su creación como por ejemplo el periodo de tiempo o el centro de costo para el cual desea obtener la información.

Módulo Oracle Responsibilities

(Solo disponible para usuarios que tengan asignados permisos de Oracle Responsibilities)

- 64. El sistema debe permitir crear una nueva responsabilidad Oracle, el usuario deberá definir el login del usuario a quien desea asignar una responsabilidad, la compañía, el flag que puede ser: Actual (Journal) o Encumbrance y debe seleccionar de entre la lista de responsabilidades que existen ya creadas en Oracle GL para cada usuario.

- 65. El sistema debe permitir listar la información de las responsabilidades Oracle existen hasta el momento.
- 66. El sistema debe permitir buscar una responsabilidad Oracle por el nombre o parte del nombre del usuario.
- 67. El sistema debe permitir eliminar una responsabilidad Oracle.

7.2 Requerimientos no funcionales del sistema

- Para el desarrollo del sistema será utilizado como estándar la tecnología J2EE para el desarrollo de aplicaciones Web en java con metodología Faces (Framework Icefaces).
- El sistema debe funcionar sobre un ambiente web, donde las interfaces tienen que estar implementadas para Internet Explorer v6.0 o mayor o para Firefox 2.0 o mayor.
- El sistema debe ejecutarse sobre el sistema operativo Windows XP o superior.
- El sistema debe manejar una serie de interfaces gráficas, las cuales deben tener un formato similar al usado en el módulo General Ledger de Oracle financiero.
- El sistema debe utilizar la base de datos de Oracle financiero, que se utiliza actualmente en el área administrativa y financiera de CIAT, para consultas y validaciones.
- El sistema debe utilizar una base de datos de Oracle propia de la aplicación, para el manejo de roles y la gestión de información contable contra el módulo General Ledger de Oracle Financiero.
- El sistema debe cargar en el menú principal y mostrar en todo momento el login y el rol del usuario que inicio una sesión.
- El sistema debe proveer una interfaz gráfica central de acuerdo con el rol con el que ha ingresado el usuario, mostrando solo a los links de las funcionalidades a las que pueda acceder.
- Al ser una aplicación que será usada a nivel internacional se usará como estándar el idioma Inglés.

7. CASOS DE USO Y REQUERIMIENTOS RELACIONADOS

Cuadro 1. Relación entre casos de uso y requerimientos

CU No.	CASO DE USO	REQUERIMIENTO RELACIONADO
1	Iniciar sesión	Requerimiento No. 1 Requerimiento No. 2
2	Finalizar sesión	Requerimiento No. 3
3	Crear usuario	Requerimiento No. 4 Requerimiento No. 5 Requerimiento No. 6
4	Listar usuarios	Requerimiento No. 7
5	Visualizar usuario	Requerimiento No. 8
6	Buscar usuario	Requerimiento No. 9
7	Modificar usuario	Requerimiento No. 10
8	Eliminar usuario	Requerimiento No. 11
9	Crear rol	Requerimiento No. 12
10	Listar roles	Requerimiento No. 13
11	Visualizar rol	Requerimiento No. 14
12	Habilitar opciones de un rol	Requerimiento No. 15
13	Deshabilitar opciones de un rol	Requerimiento No. 16
14	Modificar rol	Requerimiento No. 17
15	Eliminar rol	Requerimiento No. 18
16	Crear opción	Requerimiento No. 19
17	Listar opciones	Requerimiento No. 20
18	Modificar opción	Requerimiento No. 21
19	Eliminar opción	Requerimiento No. 22
20	Ingresar encumbrance	Requerimiento No. 23 Requerimiento No. 24
21	Guardar encumbrance	Requerimiento No. 25 Requerimiento No. 26
22	Validar encumbrance	Requerimiento No. 27
23	Enviar encumbrance a Oracle	Requerimiento No. 28
24	Listar encumbrances	Requerimiento No. 29
25	Buscar encumbrance	Requerimiento No. 30
26	Visualizar encumbrance	Requerimiento No. 31
27	Modificar encumbrance	Requerimiento No. 32
28	Visualizar error encumbrance	Requerimiento No. 33
29	Reversar encumbrance	Requerimiento No. 34
30	Imprimir encumbrance	Requerimiento No. 35

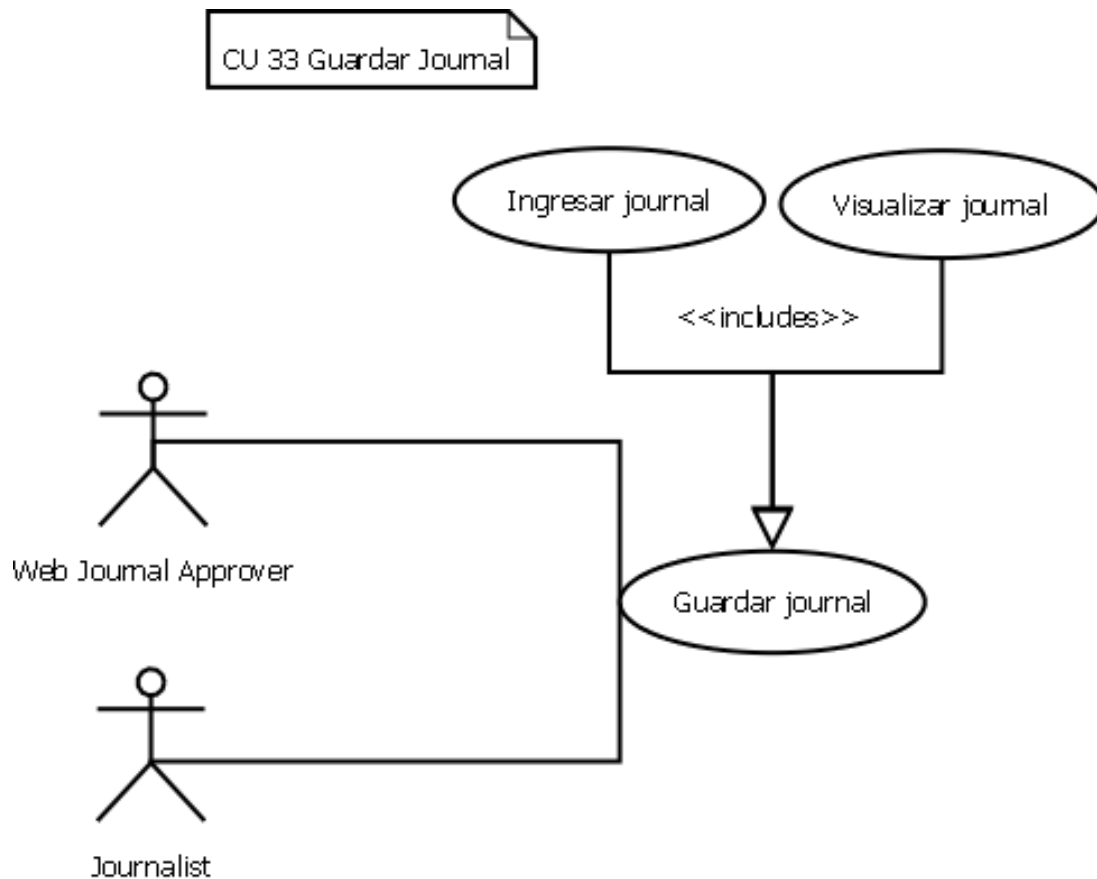
Cuadro 1. Relación entre casos de uso y requerimientos (continuación)

31	Eliminar encumbrance	Requerimiento No. 36
32	Ingresar journal	Requerimiento No. 37 Requerimiento No. 38 Requerimiento No. 39 Requerimiento No. 40
33	Guardar journal	Requerimiento No. 41
34	Validar journal	Requerimiento No. 42 Requerimiento No. 43 Requerimiento No. 44 Requerimiento No. 45
35	Enviar journal a Oracle	Requerimiento No. 46 Requerimiento No. 47
36	Enviar journal a aprobador	Requerimiento No. 48 Requerimiento No. 49
37	Listar journals	Requerimiento No. 50
38	Buscar journal	Requerimiento No. 51
39	Visualizar journal	Requerimiento No. 52 Requerimiento No. 53
40	Modificar journal	Requerimiento No. 54
41	Imprimir journal	Requerimiento No. 55
42	Eliminar journal	Requerimiento No. 56
43	Aprobar línea	Requerimiento No. 57 Requerimiento No. 58 Requerimiento No. 59
44	Desaprobar línea	Requerimiento No. 60
45	Generar reporte	Requerimiento No. 61 Requerimiento No. 62 Requerimiento No. 63
46	Crear responsabilidad Oracle	Requerimiento No. 64
47	Listar responsabilidades Oracle	Requerimiento No. 65
48	Buscar responsabilidad Oracle	Requerimiento No. 66
49	Eliminar responsabilidad Oracle	Requerimiento No. 67

8. DIAGRAMAS DE CASOS DE USO

Muestra el comportamiento del caso de uso y su relación con los actores del sistema en este caso podemos visualizar el diagrama de caso de uso para el CU33 Guardar Journal el cual se considera el más importante y crítico del sistema, los diagramas para los demás casos de uso pueden encontrarse como anexos en la versión digital.

Figura 1. Diagrama de caso de uso 33 (Guardar Journal)



9. GUIONES DE CASOS DE USO

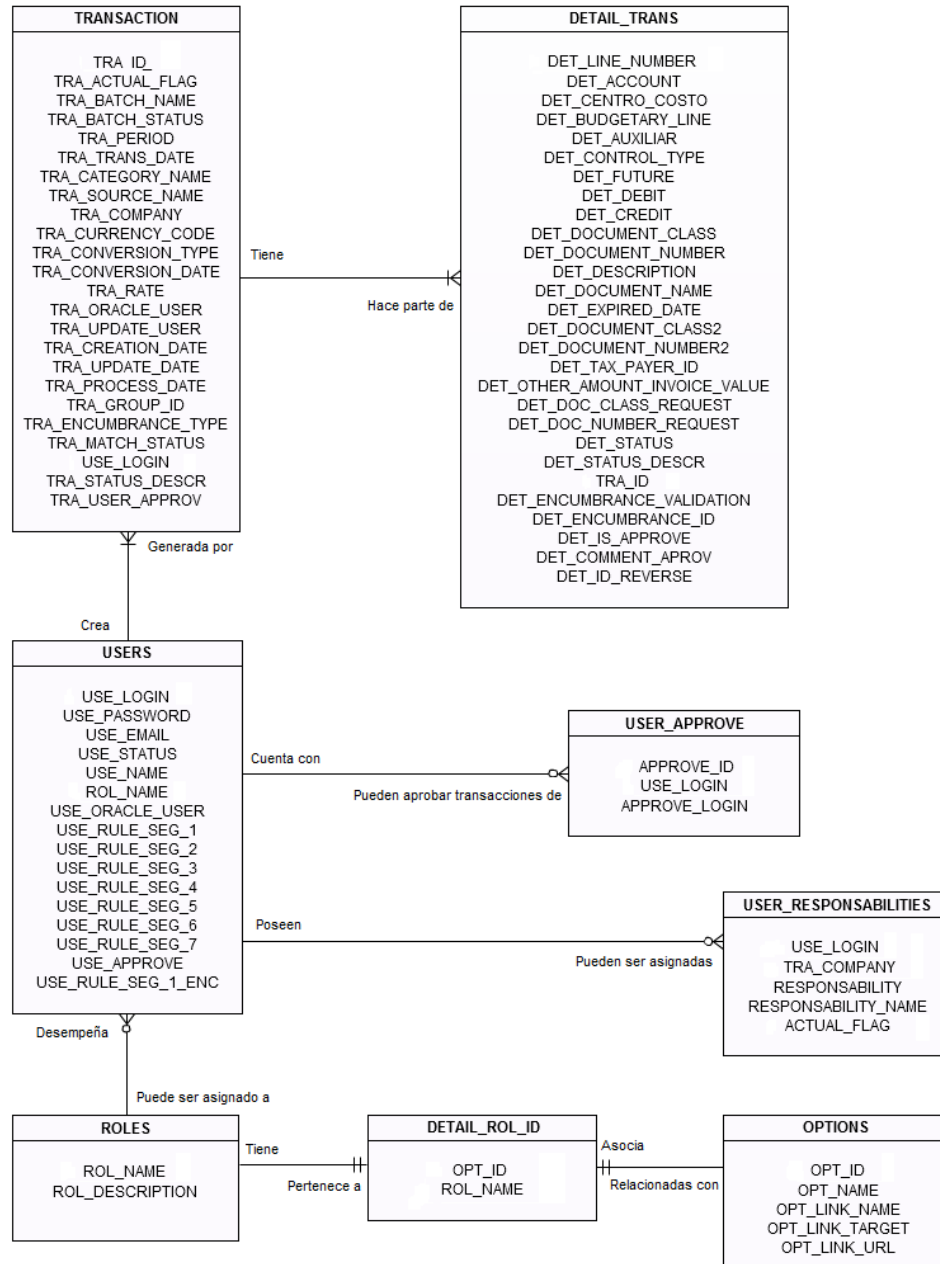
Muestra el guión para el caso de uso CU33 Guardar Journal el cual se considera el más importante y crítico del sistema, los guiones para los demás casos de uso pueden encontrarse como anexos en la versión digital.

Cuadro 1. Guión para el caso de uso CU33 Guardar Journal

No.	CU 33
Nombre	CU guardar Journal
Descripción	El usuario puede guardar un nuevo comprobante contable.
Actores	Journalist, Web Journal Approver
Precondiciones	Se debe haber iniciado una sesión CU 32 Ingresar Journal
Poscondiciones	
Guión	
Actor	Sistema
1. Presiona click en el botón guardar.	2. El sistema verifica que la información que el usuario ha ingresado es completa y correcta. 3. Se almacena un nuevo comprobante contable.
Caminos Alternos	2.1 Si la fecha del comprobante no coincide con el periodo activo, el sistema mostrará el siguiente mensaje: “Effective date does not correspond to period” 2.2 Si se encuentran errores en los datos de la cabecera del comprobante “Error in the transaction header!” 2.3 Si se encuentran errores en alguna línea de registro del comprobante, el sistema mostrará el siguiente mensaje: “Error in detail, line number”
Casos de uso relacionados	CU 32 Ingresar Journal

10. DIAGRAMA ENTIDAD RELACIÓN

Figura 2. MER



11. ARQUITECTURA DE LA APLICACIÓN

La arquitectura de la aplicación, consiste en tres capas bien definidas que son: la presentación, la lógica y el acceso a datos.

Para la capa de presentación se utilizará Icefaces 1.7 que es un framework J2EE Ajax con JSF (java Server faces) para diseñar y lograr cambios dinámicos en la interfaz.

Cada una de las pantallas de la aplicación es construida en un archivo separado de formato (JSPX).

Para cada una de las páginas JSPX, se cuenta con objetos planos que manejan los campos y los eventos asociados a los formularios. Estos objetos conocidos como “Backing Beans”, se manejan a nivel de petición Web (request).

Los formularios y pantallas, cuando requieren datos de la aplicación o que son mantenidos en memoria, hacen uso de otros objetos planos conocidos como “Managed Beans”. Estos objetos representan el Modelo, la representación de los datos que se obtiene de la lógica de dominio y el acceso a datos. Estos objetos se manejan a nivel de sesión de usuario (session).

Para la construcción de la capa lógica se utilizará Icefaces, JavaScript y J2EE 1.4 haciendo uso del IDE MyEclipse 8.6 que permite la integración de todos los elementos necesarios para el desarrollo de la aplicación.

El mecanismo de acceso a datos se lograra por medio de la especificación Hibernate 3.2, que se encargará de toda la parte del mapeo objeto-relacional y así del manejo de conexiones a la base de datos y de operaciones de inserción, actualización, eliminación y consulta implementando el patrón DAO (Data Acces Object).

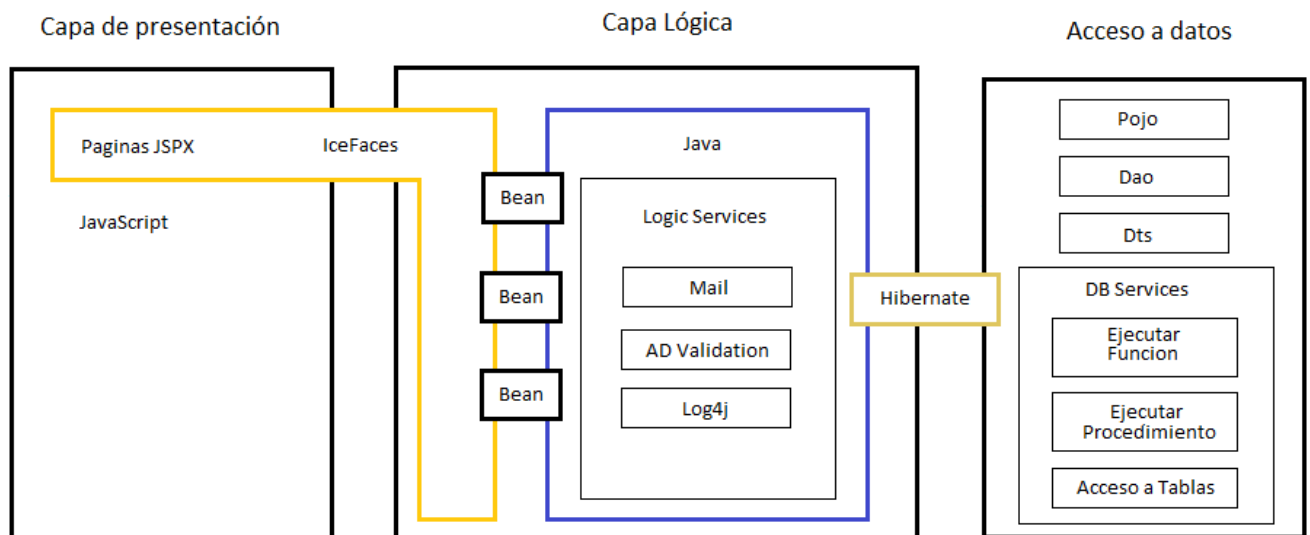
Hibernate 3.2 también permite de forma transparente la invocación a los componentes de negocio en Oracle®, esta comunicación es vital ya que las aplicaciones CIAT que trabajan en línea con Oracle® Financial requieren realizar consultas y actualizaciones de datos por medio de funciones y/o procedimientos almacenados propios de Oracle® Financial o desarrollados por el equipo del CIAT.

Para realizar modificaciones en los datos administrados por las aplicaciones en Oracle® Financial se deben utilizar tablas intermedias y sus datos son importados por medio de procesos dentro de la aplicación Oracle® Financial o por procedimientos que modifican los datos.

La obtención y almacenamiento de los datos se realiza haciendo un mapeo entre la aplicación y la base de datos mediante una técnica ORM (Object Relational Mapping) que se encarga de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Los descriptores XML sirven para definir el mapeo entre los objetos Java y la base de datos relacional. Esta mapeo sirve para traducir un dato a objeto y viceversa.

Cada tabla mapeada cuenta su respectivo DAO en donde se encuentra los métodos para guardar, modificar, consultar y eliminar. Ejemplo de este: UsersDao().save(), UsersDao().update(), etc. Este recibe como parámetros objetos planos de java.

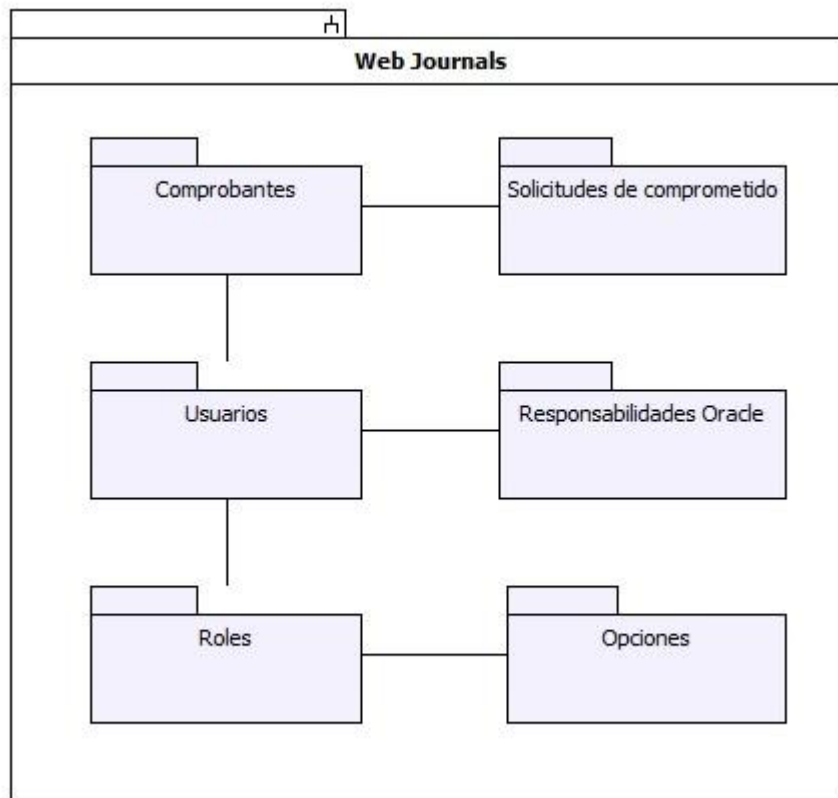
Figura 3. Arquitectura 1



11.1 Vista Lógica

La Aplicación de “Sistema de Web Journals” estará compuesta de varios módulos integrados en un solo aplicativo Web, los módulos principales son:

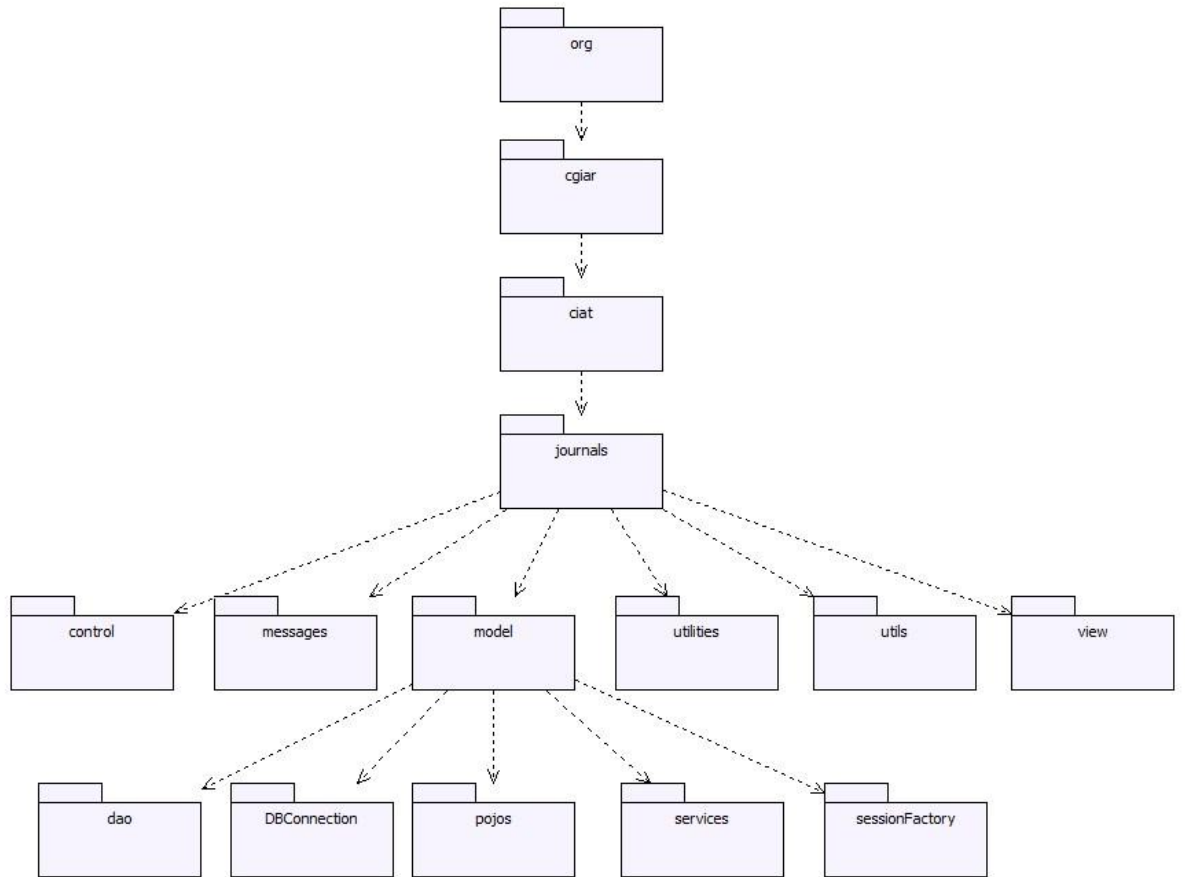
Figura 4. Arquitectura 2



11.2 Vista de desarrollo

El siguiente diagrama permite visualizar la distribución de paquetes del código de la aplicación.

Figura 5. Arquitectura 3



Descripción de los paquetes

Control: En este paquete se encuentran las clases que controlan la aplicación, aquí se crean los métodos propios de cada clase, se controlan errores y se realizan validaciones.

Messages: En este paquete se encuentra la clase MessageResources que maneja los mensajes que muestra la aplicación, esta traduce el nombre con el que trabaja el sistema al que ve el usuario en pantalla

Model:

Dao: En este paquete se albergan todas las clases que acceden a los datos (DAO), para cada una de las clases existen los métodos de búsqueda, guardado, eliminación y actualización. Estas clases son autogeneradas realizando el proceso de ingeniería inversa de Hibernate y también crean un objeto asociado directamente a cada atributo de las tablas mapeadas.

DBConnection: En este paquete se encuentran las clases que permiten las conexiones, tanto la conexión local como la conexión con la base de datos financiera.

Pojos: Este paquete contiene clases que también son autogeneradas realizando el proceso de ingeniería inversa de Hibernate contienen los métodos accesorios (getters y setters)

Services: En este paquete se encuentran las clases que prestan servicios adicionales a la aplicación, aquí se crean métodos que llaman a los métodos de las clases DAO para realizar consultas, actualizaciones, inserción y eliminado.

SessionFactory: En este paquete se encuentran las clases que permiten el acceso a las sesiones Hibernate.

Utilities: En este paquete existen varias clases cuyos métodos funcionan como utilidades para el resto de la aplicación, por ejemplo métodos accesorios especiales, manejo de permisos, sesiones y mensajes y creación de reportes.

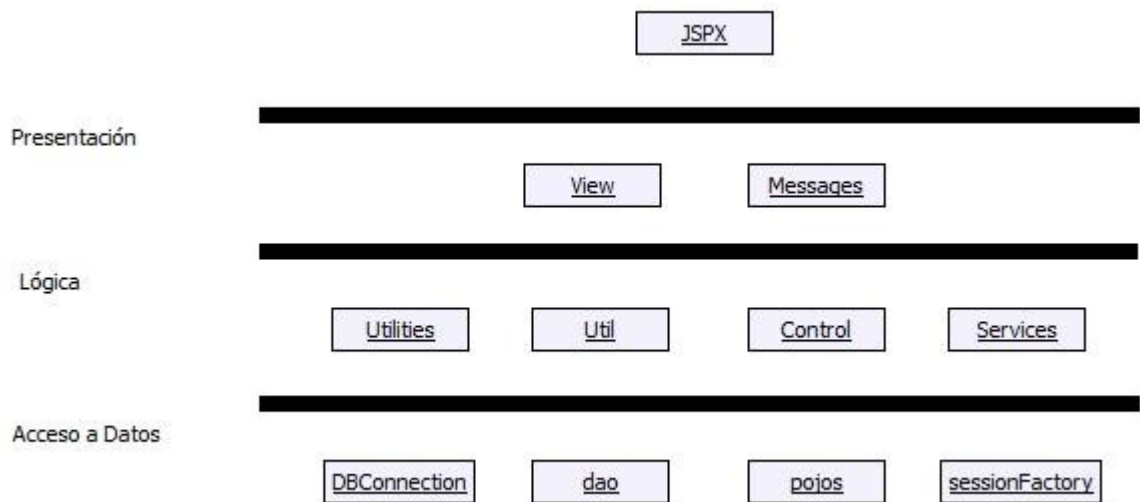
Utils: En este paquete existen varias clases que realizan procesos críticos de la aplicación, como el manejo de transacciones, el envío de correos electrónicos a

los aprobadores, el envío de información a Oracle Financials y el manejo de la navegación entre archivos JSPX.

View: En este paquete se encuentran los beans que manejan los archivos JSPX, se crean listas que almacenan información de la base de datos y las muestran en pantalla y se manejan los eventos relacionados con la interfaz gráfica.

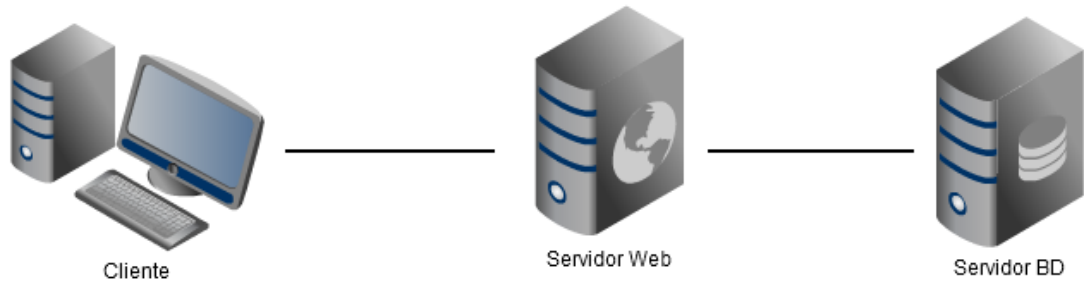
Según la descripción del contenido de cada paquete puede decirse que cada uno de ellos trabaja para una capa específica.

Figura 6. Arquitectura 4



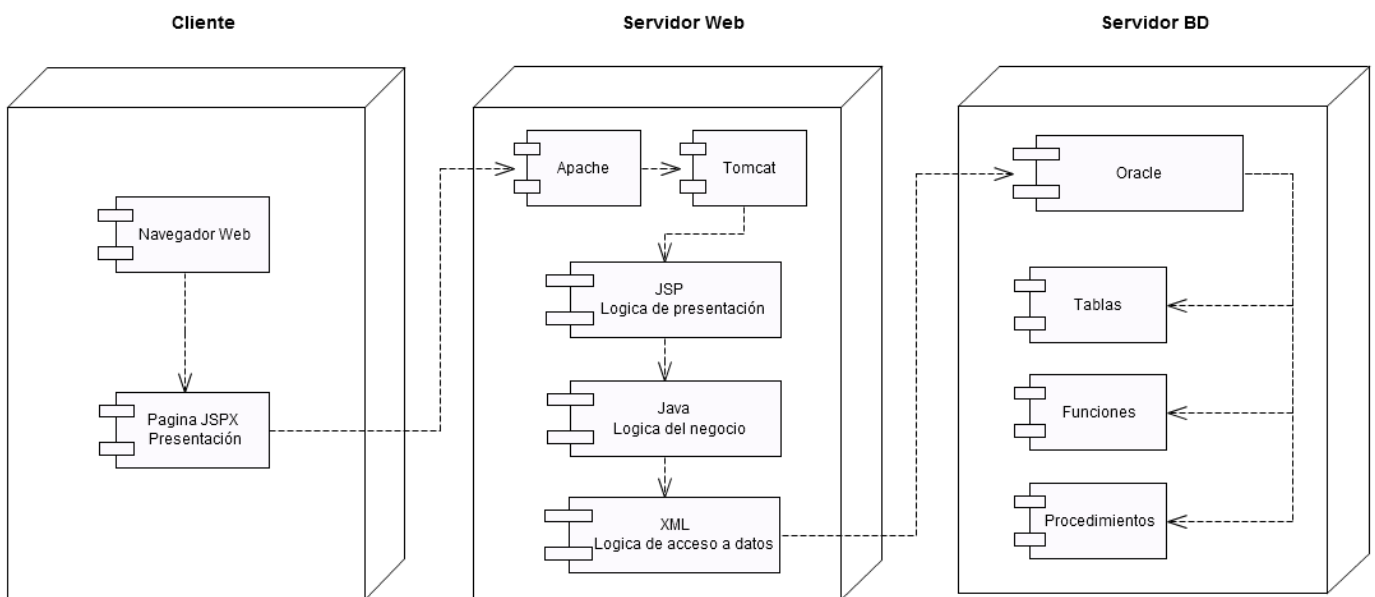
11.3 Vista física

Figura 7. Arquitectura 5



El acceso a la aplicación se hará a través de un navegador Web. El servidor Web será apache el cual realizará peticiones a tomcat como contenedor java y este estará encargado de responder peticiones de paginas JSPX y de java (.java); Estos harán uso de archivos de configuración y de mapeo en XML. Los archivos XML tendrán la configuración del mapeo de tablas en bases de datos Oracle.

Figura 8. Arquitectura 6

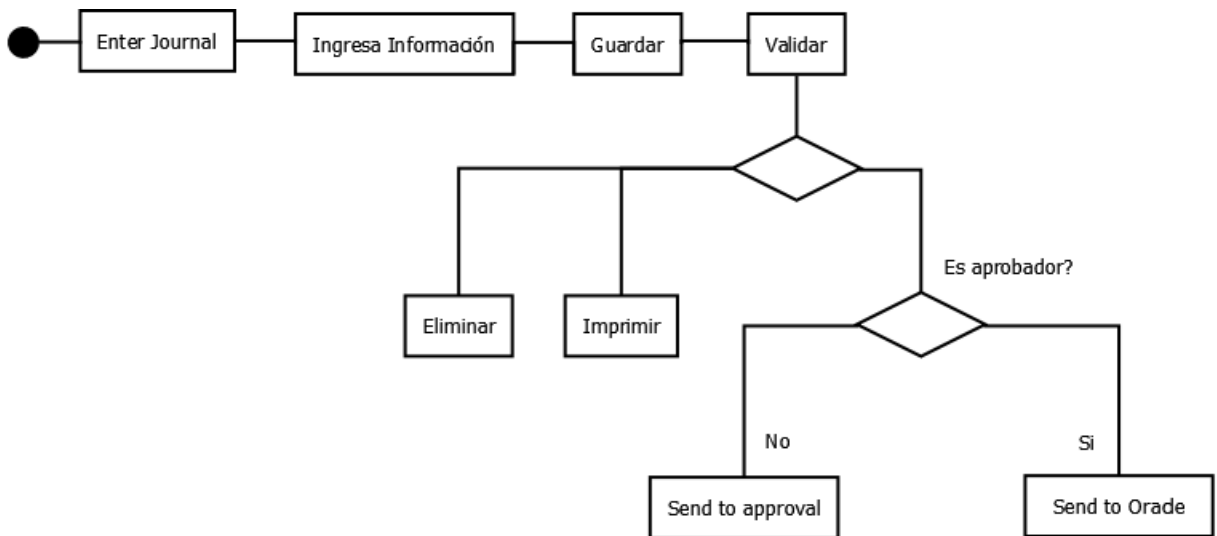


11.4 Vista de procesos

En la vista de procesos se encuentran los diagramas de actividad en este caso tenemos el diagrama de actividad de el módulo de comprobantes el cual se relaciona con el CU35 enviar journal a oracle.

Los diagramas de actividad para Inicio y Menú principal y para cada uno de los módulos de la aplicación se pueden encontrar como anexos en la versión digital del proyecto.

Figura 9. Arquitectura 7



12. POLÍTICAS DE USUARIOS

El aplicativo Web Journals permite la creación de usuarios locales y de usuarios registrados en el directorio activo de CGIAR, si el usuario existe en el directorio activo la contraseña para ingresar a la aplicación será la misma que utiliza normalmente para ingresar a su cuenta de usuario.

En el momento de la creación del usuario se deberá definir si el usuario será o no un aprobador, en caso de la creación de un usuario aprobador el sistema podrá comprobar si en efecto es un usuario Oracle, si el usuario no es un usuario Oracle podrá escoger un aprobador de la lista de usuarios aprobadores para que el apruebe sus transacciones y posteriormente las envíe a Oracle.

Se pueden eliminar usuarios de la aplicación mientras estos no tengan relacionadas transacciones, en caso contrario no se podrá borrar pero si se podrá inactivar impidiendo así su ingreso a la aplicación y almacenando sus transacciones.

Para poder acceder a las diferentes funcionalidades de la aplicación, los usuarios deben tener un Rol asignado, cada rol otorgará a un usuario los permisos necesarios para poder acceder a las diferentes opciones creadas en el sistema.

Las opciones por defecto creadas en el sistema son:

Enter Journal
Journal Entry Inquiry
Journal Reports
Enter Encumbrance Request
List Encumbrance
Users Administration
Roles Administration
Options Administration
List Encumbrance II
User responsibilities
Validate all

Las opciones disponibles para cada usuario dependen directamente del rol que el usuario tenga asignado en la aplicación.

Existen también algunos roles creados por defecto en la aplicación, pueden crearse nuevos roles asignando las opciones deseadas o también pueden modificarse los existentes.

Los roles por defecto creados en el sistema son:

Role	Description
Admin Financial	All options except administrative
Administrator 1	All options available
Administrator 2	All administrative options
Administrator 3	Only user administration
Administrator 4	User administration and OL options
Administrator valida	Validate any journal
Encumbrance 1	All encumbrance options
Journalist 1	All journal options
Journalist 2	Only journal reports
OL user	All OL options except administrative

Estos roles pueden agruparse en tres grupos principales dependiendo la función que cumplan, estos grupos son:

Administrador: Se encarga de la administración de la aplicación, incluye el manejo de los módulos de usuario, roles, opciones y responsabilidades Oracle.

Journalist: Es el usuario que ingresa las transacciones (comprobantes contables), por lo general es un asistente financiero ubicado en alguna estación regional.

Web Journal Approver: es un usuario que tiene licencia Oracle y acceso al Oracle General Ledger, son los únicos que pueden crear solicitudes de comprometidos "Encumbrances" y que pueden modificar, aprobar, desaprobado e ingresar a Oracle las transacciones creadas por los Journalists.

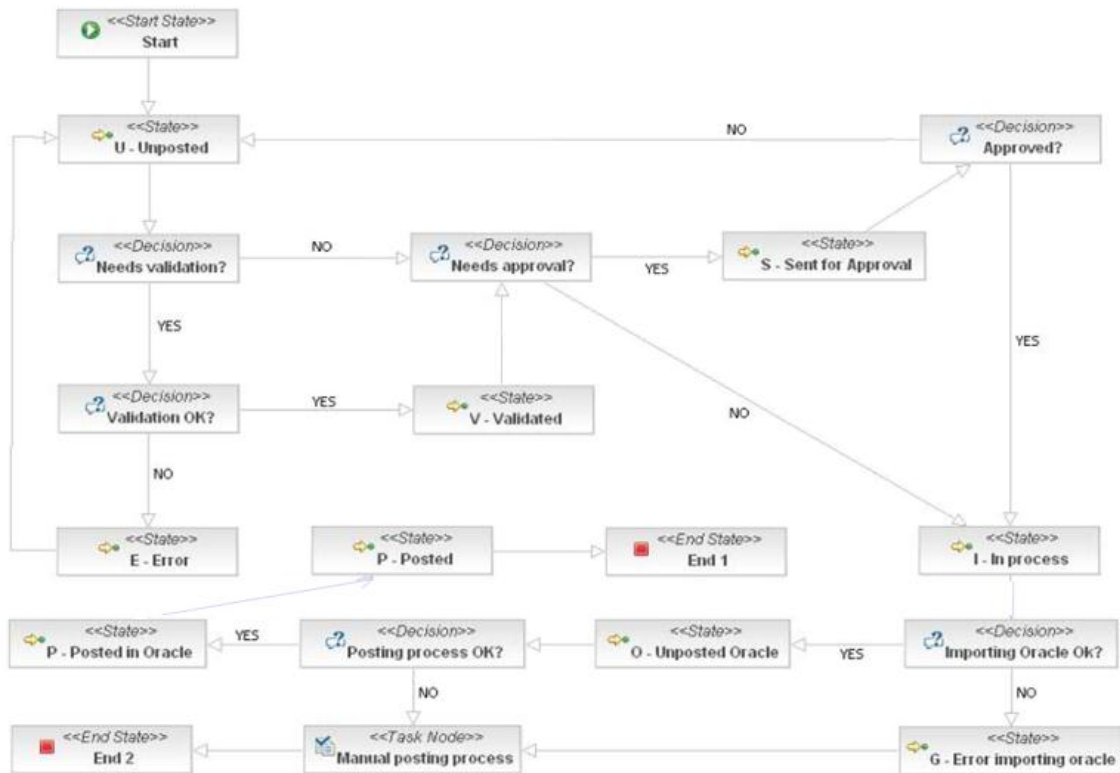
Se puede eliminar un rol mientras este no esté asociado con ningún usuario.

13. CONEXIONES, ESTADOS Y PROCEDIMIENTOS

La aplicación WebJournals permitirá a las estaciones regionales generar comprometidos y comprobantes contables de sus transacciones y que de esta manera queden almacenados y registrados en Oracle financiero de una forma transparente para ellos. El procesamiento para generar comprometidos y/o comprobantes es ingresar a la aplicación web ubicada en el url: <http://newapp.ciat.cgiar.org/journals> . El ingreso se realizará utilizando usuarios válidos del directorio activo con usuarios Oracle válidos relacionados. Estos usuarios Oracle poseen reglas de autenticación que son cargadas en la aplicación para de esta forma validar centros de costo, cuentas y permisos generales a dichos usuarios. La autenticación de los usuarios del directorio activo es posible mediante la librería CGIARAuthenticate.jar

Si se desea crear un nuevo comprobante o comprometido, el sistema mostrará únicamente la información que el usuario esté autorizado a ver y utilizar. Estos comprobantes y comprometidos son validados por cada uno de los aprobadores en las regiones y serán enviados a Oracle a través de la aplicación de WebJournals después de ser pasar por todas las validaciones correspondientes. Una vez los comprobantes se encuentran en la interfaz, el proceso automático de importación se encarga de validar estos comprobantes y de enviarlos o devolverlos según sea el caso. El diagrama de estados en la aplicación puede apreciarse en el siguiente gráfico:

Figura 10. Arquitectura 8



Los procedimientos almacenados de validación para comprometidos y comprobantes son los siguientes:

Validación presupuestal:

CALL CIATAPPS.P_VALIDACION_PPTAL_ORACLE_INIC

Validación del balanceo de un comprobante:

WEBFINANCIALS01.P_RETURN_JOURNAL_BALANCE

Validación del periodo de comprobantes

WEBFINANCIALS01.P_VALIDATE_PERIOD

Validación del periodo de comprometidos

WEBFINANCIALS01.P_VALIDATE_PERIOD_ENCUMBRANCE

Regla de validación cruzada:

CIATAPPS.P_CIA_T_CROSS_VALIDATION_RULES

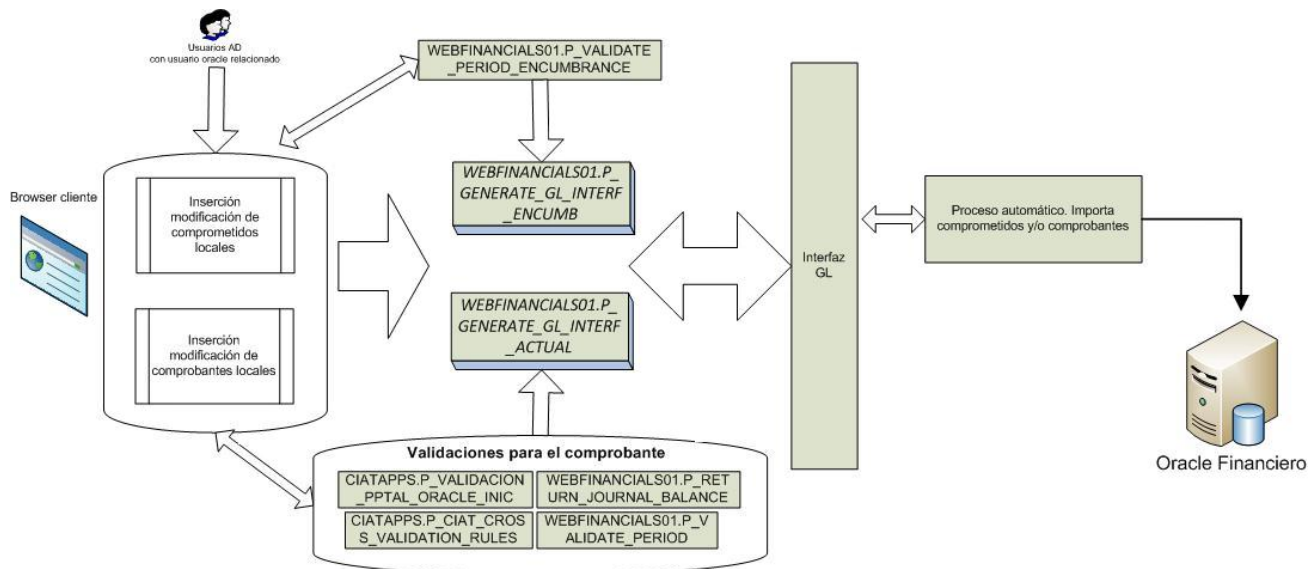
Los procedimientos almacenados para insertar comprometidos y comprobantes en la interfaz GL son los siguientes:

El procedimiento de envío de comprometidos a la interfaz GL es el siguiente:
WEBFINANCIALS01.P_GENERATE_GL_INTERF_ENCUMB

Y de envío de comprobantes contables:
WEBFINANCIALS01.P_GENERATE_GL_INTERF_ACTUAL

El siguiente diagrama ilustra el proceso de envío de comprobantes y comprometidos a Oracle financiero

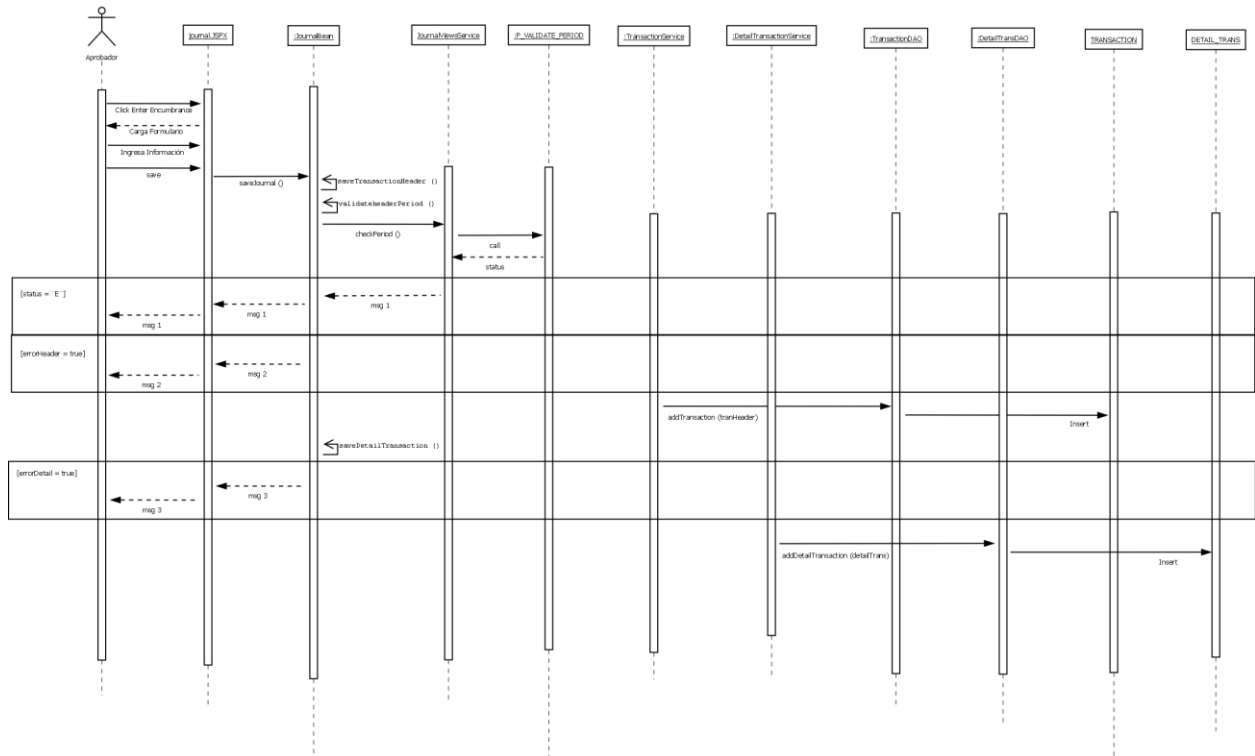
Figura 11. Envío de comprobantes



15. DIAGRAMAS DE SECUENCIA

A continuación podemos observar el diagrama de secuencia relacionado con el CU33 Guardar Journal, los diagramas de secuencia relacionados a los demás casos de uso pueden encontrarse como anexo en la versión digital del proyecto.

Figura 13. Diagrama de clases secuencia de uso 33 (Guardar Journal)



Msg 1 : “Effective date does not correspond to period”

Msg 2 : “Error in the transaction header!”

Msg 3 : “Error in detail, line number”

16. PLAN DE PRUEBAS

16.1 Introducción

16.1.1 Propósito

Este documento describe el Plan de pruebas para el aplicativo Web Journals. En concreto define los siguientes objetivos específicos:

- Identifica los elementos que se van a probar.
- Describe la estrategia de pruebas que se va a seguir en el proceso de prueba.

16.1.2 Ámbito

Este Plan de Pruebas describe las pruebas de unidad, integración y del sistema que se aplicarán al software desarrollado.

El objetivo es probar todas las funcionalidades definidas en el documento de requerimientos y en la especificación de de casos de uso.

16.2 Requerimientos de las pruebas

La lista que proporcionamos en esta sección identifica los elementos (casos de uso, requisitos funcionales y requisitos no funcionales) que son objetivos de las pruebas. Es decir, los elementos que vamos a probar.

16.2.1 Pruebas de integridad de la base de datos y de los datos:

- Verificar el acceso al sistema Web Journals.
- Verificar la recuperación correcta de las modificaciones realizadas en la base de datos.
- Verificar accesos simultáneos de lectura de datos.

16.2.2 Pruebas de funcionalidad:

Las pruebas de funcionalidad realizadas para otros casos de uso pueden encontrarse en la sección de anexos de la versión digital del proyecto.

- Verificar el caso de Guardar Journal (CU33).

16.2.3 Pruebas de interfaz de usuario:

- Verificar que la navegación a través de un conjunto de pantallas es fácil, intuitivo y lógico.

- Navegar a través de todos los casos de uso, verificando que cada interfaz de usuario se comprende fácilmente.
- Verificar que todas las interfaces de usuario siguen los estándares de GUI.

16.2.4 Pruebas de desarrollo:

- Se realizaran diferentes pruebas para evaluar tiempos de respuesta de manera local y remota y con diferentes niveles de conectividad.

16.3 Estrategia de prueba

16.3.1 Pruebas de integridad de la base de datos y de los datos.

Cuadro 3. Pruebas de integridad

Objetivos de la prueba	Comprobar que los procedimientos y métodos de acceso a la base de datos funcionan correctamente.
Técnicas	Invocar cada procedimiento o método de acceso a la base de datos con datos válidos e inválidos. Inspeccionar la base de datos para asegurar que los datos son los previstos, todos los eventos de la base de datos ocurren adecuadamente, o revisar los valores devueltos para asegurar que la recuperación de datos es correcta.
Criterios de finalización	Todos los procedimientos y métodos de acceso funcionan como se diseñaron y sin ningún error en los datos.
Consideraciones	Se debería usar bases de datos de tamaño pequeño o mínimo (limitado según el número de registros) para incrementar la visibilidad de cualquier evento no aceptable. Las pruebas pueden necesitar un entorno de desarrollo DBMS para recuperar o modificar datos directamente de la base de datos.

16.3.2 Pruebas de funcionalidad.

Las pruebas de funcionalidad se deberían centrar en cualquier requisito que pueda ser trazado directamente de los casos de uso y reglas de negocio. El objetivo de estas pruebas es verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de las reglas de negocio. Este tipo de pruebas están basadas en técnicas de caja negra, es decir, verificar la aplicación interactuando a través de las interfaces de usuario y analizando los resultados.

Cuadro 4. Pruebas de funcionalidad

Objetivos de la prueba	Asegurar la navegación correcta de la aplicación, la entrada de datos, su procesamiento y recuperación.
Técnicas	Ejecutar cada caso de uso y flujo del caso de uso con datos válidos e inválidos para verificar lo siguiente: <ul style="list-style-type: none">• Cuando se utilizan datos correctos se obtienen los resultados esperados.• Cuando se utilizan datos incorrectos se obtienen los mensajes de error o advertencias adecuadas.• Cada regla de negocio se ha aplicado correctamente.
Criterios de finalización	Todas las pruebas planificadas se han ejecutado. Todos los defectos identificados se han considerado.
Consideraciones	Ninguna.

Casos de prueba para el caso de guardar journal (CU33).

Cuadro 5. Lista de casos de prueba CU33

Entrada	Caso de Prueba
Effective date	<p>(1) Effective date no es un campo vacio</p> <p>(2) Effective date es un campo vacio (camino alternativo 2.1)</p> <p>(3) El valor ingresado en el effective date coincide con el periodo activo</p> <p>(4) El valor ingresado en el effective date no coincide con el periodo activo (camino alternativo 2.2)</p>
Conversion date	<p>(5) Conversion date no es un campo vacio</p> <p>(6) Conversion date es un campo vacio (camino alternativo 2.1)</p>
Account	<p>(7) Account no es un campo vacio</p> <p>(8) Account es un campo vacio (camino alternativo 2.1)</p>
Cost Center	<p>(9) Cost Center no es un campo vacio</p> <p>(10) Cost Center es un campo vacio (camino alternativo 2.2)</p>
Budgetary Line	<p>(11) Budgetary Line no es un campo vacio</p> <p>(12) Budgetary Line es un campo</p>

	vacio (camino alternativo 2.3)
Auxiliary	<p>(13) Auxiliary no es un campo vacio</p> <p>(14) Auxiliary es un campo vacio (camino alternativo 2.3)</p>
Future	<p>(15) Future no es un campo vacio</p> <p>(16) Future es un campo vacio (camino alternativo 2.3)</p>
Debit	<p>(17) Debit no es un campo vacio</p> <p>(18) Debit es un campo vacio (camino alternativo 2.3)</p> <p>(19) El campo Debit tiene un valor mayor que cero y menor que 1.000.000</p> <p>(20) El campo Debit tiene un valor menor o igual que cero (camino alternativo 2.3)</p>
Credit	<p>(21) Credit no es un campo vacio</p> <p>(22) Credit es un campo vacio (camino alternativo 2.3)</p> <p>(23) El campo Credit tiene un valor mayor que cero y menor que 1.000.000</p> <p>(24) El campo Credit tiene un valor menor o igual que cero (camino alternativo 2.3)</p>
Doc class	(25) Doc class no es un campo vacio

	(26) Doc class es un campo vacío (camino alternativo 2.3)
Document number	(27) Document number no es un campo vacío (28) Document number es un campo vacío (camino alternativo 2.3) (29) Document number no contiene caracteres especiales (30) Document number contiene caracteres especiales (camino alternativo 2.3)
Description	(31) Description no es un campo vacío (32) Description es un campo vacío (camino alternativo 2.3)

Cuadro 6. Caso de prueba # 1

No. Caso Prueba	1
Nombre Entrada	Efective date
Nombre Caso de Prueba o Regla asociada	Efective date no es un campo vacío
Valor Entrada	13-JUL-2011
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 3, 5, 7, 9, 11, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 7. Caso de prueba # 2

No. Caso Prueba	2
Nombre Entrada	Effective date
Nombre Caso de Prueba o Regla asociada	Effective date es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in the transaction header!
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 8. Caso de prueba # 3

No. Caso Prueba	3
Nombre Entrada	Effective date
Nombre Caso de Prueba o Regla asociada	El valor ingresado en el effective date coincide con el periodo activo
Valor Entrada	Period: APR-2012, Effective date: 02-APR-2012
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Caso de prueba 1
Poscondición	Caso de prueba 5, 7, 9, 11, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 9. Caso de prueba # 4

No. Caso Prueba	4
Nombre Entrada	Effective date
Nombre Caso de Prueba o Regla asociada	El valor ingresado en el effective date no coincide con el periodo activo
Valor Entrada	Period: AUG-2010, Effective date: 02-APR-2012
Salida Esperada	Effective date does not correspond to period
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 10. Caso de prueba # 5

No. Caso Prueba	5
Nombre Entrada	Conversion date
Nombre Caso de Prueba o Regla asociada	Conversion date no es un campo vacio
Valor Entrada	13-JUL-2011
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 7, 9, 11, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 11. Caso de prueba # 6

No. Caso Prueba	6
Nombre Entrada	Conversion date
Nombre Caso de Prueba o Regla asociada	Conversion date es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in the transaction header!
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 12. Caso de prueba # 7

No. Caso Prueba	7
Nombre Entrada	Account
Nombre Caso de Prueba o Regla asociada	Account no es un campo vacio
Valor Entrada	11110204
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 9, 11, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 13. Caso de prueba # 8

No. Caso Prueba	8
Nombre Entrada	Account
Nombre Caso de Prueba o Regla asociada	Account es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 14. Caso de prueba # 9

No. Caso Prueba	9
Nombre Entrada	Cost Center
Nombre Caso de Prueba o Regla asociada	Cost Center no es un campo vacio
Valor Entrada	DC06
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 11, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 15. Caso de prueba # 10

No. Caso Prueba	10
Nombre Entrada	Cost Center
Nombre Caso de Prueba o Regla asociada	Cost Center es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 16. Caso de prueba # 11

No. Caso Prueba	11
Nombre Entrada	Budgetary Line
Nombre Caso de Prueba o Regla asociada	Budgetary Line no es un campo vacio
Valor Entrada	000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 13, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 17. Caso de prueba # 12

No. Caso Prueba	12
Nombre Entrada	Budgetary Line
Nombre Caso de Prueba o Regla asociada	Budgetary Line es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 18. Caso de prueba # 13

No. Caso Prueba	13
Nombre Entrada	Auxiliary
Nombre Caso de Prueba o Regla asociada	Auxiliary no es un campo vacio
Valor Entrada	000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 15, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 19. Caso de prueba # 14

No. Caso Prueba	14
Nombre Entrada	Auxiliary
Nombre Caso de Prueba o Regla asociada	Auxiliary es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 20. Caso de prueba # 15

No. Caso Prueba	15
Nombre Entrada	Future
Nombre Caso de Prueba o Regla asociada	Future no es un campo vacio
Valor Entrada	000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, (17, 19) o (21, 23), 25, 27, 29, 31

Cuadro 21. Caso de prueba # 16

No. Caso Prueba	16
Nombre Entrada	Future
Nombre Caso de Prueba o Regla asociada	Future es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 22. Caso de prueba # 17

No. Caso Prueba	17
Nombre Entrada	Debit
Nombre Caso de Prueba o Regla asociada	Debit no es un campo vacio
Valor Entrada	7000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, 19, 25, 27, 29, 31

Cuadro 23. Caso de prueba # 18

No. Caso Prueba	18
Nombre Entrada	Debit
Nombre Caso de Prueba o Regla asociada	Debit es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 24. Caso de prueba # 19

No. Caso Prueba	19
Nombre Entrada	Debit
Nombre Caso de Prueba o Regla asociada	El campo Debit tiene un valor mayor que cero y menor que 1.000.000
Valor Entrada	80000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Caso de prueba 17
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, 25, 27, 29, 31

Cuadro 25. Caso de prueba # 20

No. Caso Prueba	20
Nombre Entrada	Debit
Nombre Caso de Prueba o Regla asociada	El campo Debit tiene un valor menor o igual que cero
Valor Entrada	-100
Salida Esperada	Error in detail, line number
Precondición	Caso de prueba 17
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 26. Caso de prueba # 21

No. Caso Prueba	21
Nombre Entrada	Credit
Nombre Caso de Prueba o Regla asociada	Credit no es un campo vacio
Valor Entrada	7000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, 23, 25, 27, 29, 31

Cuadro 27. Caso de prueba # 22

No. Caso Prueba	22
Nombre Entrada	Credit
Nombre Caso de Prueba o Regla asociada	Credit es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 28. Caso de prueba # 23

No. Caso Prueba	23
Nombre Entrada	Credit
Nombre Caso de Prueba o Regla asociada	El campo Credit tiene un valor mayor que cero y menor que 1.000.000
Valor Entrada	80000
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Caso de prueba 21
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, 25, 27, 29, 31

Cuadro 29. Caso de prueba # 24

No. Caso Prueba	24
Nombre Entrada	Credit
Nombre Caso de Prueba o Regla asociada	El campo Credit tiene un valor menor o igual que cero
Valor Entrada	-100
Salida Esperada	Error in detail, line number
Precondición	Caso de prueba 21
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 30. Caso de prueba # 25

No. Caso Prueba	25
Nombre Entrada	Doc class
Nombre Caso de Prueba o Regla asociada	Doc class no es un campo vacio
Valor Entrada	PV
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, (17, 19) o (21, 23), 27, 29, 31

Cuadro 31. Caso de prueba # 26

No. Caso Prueba	26
Nombre Entrada	Doc class
Nombre Caso de Prueba o Regla asociada	Doc class es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 32. Caso de prueba # 27

No. Caso Prueba	27
Nombre Entrada	Document number
Nombre Caso de Prueba o Regla asociada	Document number no es un campo vacio
Valor Entrada	PV
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, (17, 19) o (21, 23), 25, 29, 31

Cuadro 33. Caso de prueba # 28

No. Caso Prueba	28
Nombre Entrada	Document number
Nombre Caso de Prueba o Regla asociada	Document number es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 34. Caso de prueba # 29

No. Caso Prueba	29
Nombre Entrada	Document number
Nombre Caso de Prueba o Regla asociada	Document number no contiene caracteres especiales
Valor Entrada	A300
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Caso de prueba 27
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, (17, 19) o (21, 23), 25, 31

Cuadro 35. Caso de prueba # 30

No. Caso Prueba	30
Nombre Entrada	Document number
Nombre Caso de Prueba o Regla asociada	Document number contiene caracteres especiales
Valor Entrada	#300
Salida Esperada	Error in detail, line number
Precondición	Caso de prueba 17
Poscondición	No tiene caso de prueba asociado por dependencia

Cuadro 36. Caso de prueba # 31

No. Caso Prueba	31
Nombre Entrada	Description
Nombre Caso de Prueba o Regla asociada	Description no es un campo vacio
Valor Entrada	Office Internet subscription fees
Salida Esperada	Depende del resultado del caso de prueba Poscondición
Precondición	Ninguno
Poscondición	Caso de prueba 1, 3, 5, 7, 9, 11, 13, (17, 19) o (21, 23), 25, 27, 29

Cuadro 37. Caso de prueba # 32

No. Caso Prueba	32
Nombre Entrada	Description
Nombre Caso de Prueba o Regla asociada	Description es un campo vacio
Valor Entrada	“ ”
Salida Esperada	Error in detail, line number
Precondición	Ninguno
Poscondición	No tiene caso de prueba asociado por dependencia

16.3.3 Pruebas de interfaz de usuario.

Las pruebas de interfaz de usuario verifican la interacción del usuario con el sistema software. El objetivo de esta prueba es asegurar que la interfaz de usuario permite al usuario acceder y navegar a través de toda la funcionalidad de la aplicación. Además, la prueba de interfaz de usuario garantiza que las interfaces de usuario cumplen los estándares.

Cuadro 38. Pruebas de interfaz de usuario

Objetivos de la prueba	<p>Verificar los siguientes objetivos:</p> <ul style="list-style-type: none"> • La navegación a través de la aplicación refleja adecuadamente las reglas de negocio y los requisitos incluyendo ventana a ventana, campo a campo y métodos de acceso (tabulador, movimientos del ratón y teclas de función). • Las ventanas y sus características, como menús, tamaño, posición y estado cumplen los estándares.
Técnicas	Crear o modificar pruebas para cada ventana con el objetivo de verificar la correcta navegación y su estado.
Criterios de finalización	Cada ventana se ha verificado con éxito y es consistente con la versión de referencia o con los estándares utilizados.
Consideraciones	Ninguna.

16.3.4 Pruebas de desarrollo

Las pruebas de desarrollo miden tiempos de respuesta, índices de transacción y otros requisitos susceptibles al tiempo. El objetivo de estas pruebas es verificar y validar que los requisitos de rendimiento se han alcanzado.

Las pruebas de desarrollo normalmente se ejecutan varias veces usando cada vez un cargo de trabajo diferente. La prueba inicial se debería realizar con una carga normal y la segunda prueba con una carga extrema.

Cuadro 39. Pruebas de desarrollo

Objetivos de la prueba	Validar el tiempo de respuesta del sistema software para las transacciones diseñadas o funciones de negocio bajo las condiciones siguientes: <ul style="list-style-type: none">• Volumen de trabajo normal.• El peor volumen de trabajo.• Conexiones remotas.
Técnicas	Usar los procedimientos de prueba definidas para las pruebas de funcionalidad.
Criterios de finalización	Se han completado las pruebas sin ningún error y dentro de los tiempos de respuesta esperados.
Consideraciones	Ninguna.

17. CONCLUSIONES

- Es en la fase de análisis cuando deben traducirse las necesidades del cliente en un modelo de sistema lo que nos ayudará a realizar un buen diseño y un buen diseño definitivamente puede hacer la diferencia en el resultado final de un proyecto de software.
- En el desarrollo de productos de software la etapa de análisis y diseño toma gran parte de la totalidad de un proyecto pero es indispensable hacerlo de manera exhaustiva para que a futuro el sistema pueda cumplir con las necesidades de los usuarios.
- En la producción de software la calidad es un factor de vital importancia por lo cual se deben realizar pruebas para detectar errores y corregirlos antes que el software sea entregado al usuario final. Para realizar un proceso de pruebas exitoso debe realizarse un plan para optimizar los resultados del mismo.
- Podemos concluir que el análisis y diseño del aplicativo Web Journals fue exitoso ya que se sentaron las bases para realizar posteriormente el desarrollo y la implementación del mismo, el cual fue desarrollado por el equipo de analistas del área de sistemas administrativos y financieros de la unidad de sistemas de información del CIAT, gracias a esto se disminuyeron costos ya que aunque se invirtieron recursos en el desarrollo del proyecto no fue necesaria la adquisición de licencias ORACLE adicionales, además en el proceso se adquirieron conocimientos y se desarrolló material intelectual lo cual representa una ganancia para la unidad.
- Gracias a la implementación del aplicativo Web Journals se han agilizado notablemente los procesos contables tanto de las regiones como el del personal financiero de CIAT headquarters, se ha obtenido el control que se deseaba al momento de ingresar transacciones a ORACLE GENERAL LEDGER pues aunque muchas personas pueden trabajar e ingresar sus transacciones solo los aprobadores pueden realmente ingresar información a la interfaz.

- Una vez finalizado el proyecto se recopilaron las experiencias de los usuarios y se ha detectado que el módulo más utilizado en el aplicativo es el de Journals, el módulo Encumbrance no ha sido realmente muy aprovechado ya que se está comenzando a gestar la cultura del pago anticipado, es posible que esta sea una reacción ante el cambio de sistema ERP que se presentará en el año 2014 debido a la política de unificación de sistemas de todos los centros pertenecientes al CGIAR.

18. RECOMENDACIONES

Debido a la política de unificación de sistemas (OCS) de todos los centros pertenecientes al CGIAR, el ERP de Oracle Financials dejara de operar a finales del año 2014 y cambiará drásticamente la manera en que se llevan los procesos contables, sin embargo esto no quiere decir que se perderá el trabajo realizado durante el desarrollo de este proyecto, se recomienda aprovechar al máximo los resultados obtenidos aun así el aplicativo como tal salga de producción tomando como base la metodología aplicada y las experiencias obtenidas para el desarrollo de nuevos proyectos, ya que cada vez se hace más necesaria la creación de aplicativos que puedan ser usado tanto en CIAT headquarters como en las regiones y probablemente otros centros pertenecientes al CGIAR.

19. BIBLIOGRAFÍA

- [1] General Ledger data sheet en español [En línea], 2011 Disponible en internet: <http://www.oracle.com/global/es/applications/oracle%20general%20ledger11i.pdf>
- [2] ELMASRI, Ramez. NAVATHE Shamkant. Fundamentos de sistemas de bases de datos. Quinta Edición. Madrid: Pearson Educación S.A., 2007. 1012 p.
- [3] PRESSMAN, Roger. Libro de ingeniería de software un enfoque práctico. Quinta edición. Editorial McGraw-Hill 2002.
- [4] SOMMERVILLE, Ian. Ingeniería de software. Séptima edición. Editorial Pearson, Addison-Wesley. Madrid, 2005. 687 p.
- [5] Información Oracle E-Business suite [En línea], 2011 Disponible en internet: <http://www.oracle.com/us/products/applications/ebusiness/index.html>
- [6] BOLAÑOS, CESAR A. ALVAREZ NIÑO, JORGE. Contabilidad. Comercial: mejores recursos humanos en la empresa Norma, 1995.
- [7] DARLINGTON, COLLIGAN, HAYS. Oracle University: Financials Student Guide, Oracle, 2001.
- [8] FOWLER, STANWICK. Web application design handbook, Morgan Kaufmann Publishers, 2004.

20. ANEXOS

En la version digital del proyecto puede encontrarse la sección de anexos donde se encuentran la totalidad de diagramas y guiones de casos de uso, diagramas de actividad, diagramas de secuencia, clases y pruebas de funcionalidad.