

UNIVERSIDAD DE GUANTÁNAMO
Facultad de Ingeniería y Ciencias Técnicas

*Tema: Sistema Informático para la Consolidación y la Gestión de las Cuentas
por Cobrar y Pagar de la Empresa Aplicaciones Informáticas (Desoft).*
SCONSOL

Tesis presentada en opción al Título de Ingeniero Informático.

Autor: Manuel Vinent Guilarte
Tutor: Msc. Mary Luz Matos Utria

Guantánamo, Junio 2020
Cuba

DECLARACIÓN DE AUTORIA

Declaro que soy el único autor de este trabajo y autorizo a la Empresa de Servicios Informático División Guantánamo DESOFT y a la Universidad de Guantánamo para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 15 días del mes de junio del 2020.

Firma del Autor

Nombre del Autor

Firma del Tutor

Nombre del Tutor

AVAL DE LA ENTIDAD

OPINION DEL TUTOR

DEDICATORIA

A mi esposa Danays Boza Ferrer por su excelente comprensión y ayuda para la realización de este trabajo.

A mis padres que han estado pendiente a lo largo de estos años de la carrera para formarme como Ingeniero Informático.

AGRADECIMIENTOS

Al claustro de profesores de la Universidad de Guantánamo, en especial a mi Tutora Msc. Mary Luz Matos Utria, quienes han tenido toda la paciencia y dedicación para atender mis necesidades en el estudio de esta carrera.

A mis compañeros de trabajos que siempre tuvieron pendiente y prestaron ayuda incondicional a mis necesidades de estudio y facilitaron feliz conclusión de esta importante carrera para mi vida, especialmente a Antonio Ruban Espinal que en el momento más difícil de mi trabajo de Diploma supo ayudarme a salir adelante.

A todos MUCHAS GRACIAS

RESUMEN

El trabajo que se presenta tiene como objetivo la consolidación y la gestión de las cuentas por cobrar y pagar en la Empresa de Aplicaciones Informáticas de DESOFT con vista a su informatización mediante el desarrollo de una herramienta informática. Para el desarrollo de la investigación se utilizaron los métodos teóricos y empíricos que ayudaron a comprender el proceso de consolidación y gestión de la información de las cuentas por cobrar y pagar. Como metodología para la elaboración de la propuesta se utilizó SCRUM. La aplicación fue desarrollada utilizando el lenguaje de programación Python, OdoO como framework de desarrollo, las tecnologías HTML y el gestor de base de datos Postgres. Como resultado se obtuvo una aplicación web que facilita la Consolidación y la Gestión de las Cuentas por Cobrar y Pagar, posibilitando acceder a la información de manera rápida y segura. La misma abarca los reportes de las cuentas por cobrar y pagar, obligaciones de pago y de cobro, informes de ventas y finanzas de la Empresa de Aplicaciones Informáticas de DESOFT.

Palabras Claves: consolidación, cuentas por cobrar, cuentas por pagar, obligaciones de cobro, obligaciones de pago, aplicación, información

Abstract

The work presented is aimed at the consolidation and management of accounts receivable and payable at DESOFT's Computer Applications Company with a view to their information through the development of a computer tool. For the development of the investigation, the theoretical and empirical methods were used that helped to understand the process of consolidation and management of the information of accounts receivable and payable. SCRUM was used as the methodology for preparing the proposal. The application was developed using the Python programming language, Odoo as a development framework, HTML technologies and the Postgres database manager. As a result, a web application was obtained that facilitates the Consolidation and Management of Accounts Receivable and Payable, making it possible to access information quickly and safely. It includes the accounts receivable and payable reports, payment and collection obligations, sales and financial reports of the DESOFT Computer Applications Company.

Key words: consolidation, accounts receivable, debts to pay, collection obligations, payment obligations, applications, informations

Índice

RESUMEN	vii
Abstract	viii
Índice	ix
Índice de Figuras	xii
Índice de Tablas	xiii
INTRODUCCIÓN	1
Objetivo:	2
Campo de acción:	2
Idea a defender:	2
Métodos teóricos.	3
CAPÍTULO I. Fundamentación teórica.	5
1.1 Caracterización del objeto de estudio	5
Figura 1. Organigrama de la División Guantánamo de la Empresa DESOFT.....	5
Diagnostico de la situación actual.....	6
1.2 Descripción de la metodología y las tecnologías actuales que respaldan la propuesta. ..	7
1.2.1 Cliente – Servidor	8
1.2.2 Metodología a utilizar	9
Scrum:	9
Teoría de Scrum	9
¿Por qué esta metodología y no otra?	14
1.2.3 Lenguaje Unificado de Modelado (UML).....	14
1.2.4 Desarrollo de la aplicación Web.....	15
1.3 Tecnologías del lado del cliente	16
1.3.1 Mozilla Firefox.....	16
1.4 Lenguajes de Programación	16
1.4.1 Framework de desarrollo a utilizar	17
¿Qué es un ORM?	18
El ORM de Odoos.....	19

Figura 2. Ejemplo de Modulo en Odoo Framework	19
Funcionamiento interno de Odoo	20
Figura 3. Funcionamiento Interno del Odoo	20
1.5 Tecnologías del lado del Servidor	21
1.5.1 Sistema Gestor de Base de Datos.....	21
1.5.2 Base de Datos.....	22
1.5.3 Servidor Web.....	22
1.5.4 Herramienta CASE.	23
Conclusiones parcial del capítulo	24
Capitulo II. Análisis, diseño e implementación. Sistema Informático SCONSOL.....	25
2.1 Especificación de los requerimientos de software	25
2.1.1 Requisitos Funcionales	25
2.1.2 Requisitos no Funcionales	26
2.1.3 Pila de producto	27
Tabla 1. Product Backlog	28
2.2 Descripción del Sistema	28
Tabla 2. Especificaciones del producto	28
Tabla 3. Resumen de la característica de la aplicación	29
Tabla 4. Roles del equipo de trabajo.....	30
Usuarios del Sistema	30
2.1 Historias de Usuarios.....	31
Tabla 5. Historia de Usuario	32
Descripción de los elementos	32
2.2.2 Planificación del Sprint.....	32
Tabla 6. Sprint Backlog	34
2.3 Diseño de la Base de Datos	34
Creación y normalización de la base de datos	34
2.4 Diseño del Sistema.....	35
2.4.1 Patrones de diseño.....	35
2.4.2 Arquitectura	36
2.5 Diagrama de Despliegue	37
Figura 6. Diagrama de despliegue	38
2.6 Estudio de Factibilidad.....	38
Factibilidad Técnica	38

Factibilidad Operativa	39
Factibilidad Económica.....	39
2.6 Implementación a la Aplicación.	40
Figura 7. Prototipo de Login (SCONSOL)	40
Figura 8. Prototipo de Vista Kaban (Conexión al VERSAT)	41
Figura 9. Prototipo de Vista Formulario (Obligaciones MN)	41
Figura 10. Prototipo de Vista Detalle (Factura).....	42
Figura 11. Prototipo de Vista Lista (Cuentas por Pagar MN).....	42
Figura 12. Prototipo de Vista Genérica (Obligaciones en MLC)	43
Conclusiones parciales del capítulo.....	44
CONCLUSIONES GENERALES.....	45
RECOMENDACIONES.....	46
Referencias Bibliográfica	47
Glosario de Términos.....	51
Anexos	52

Índice de Figuras

Figura 1. Organigrama de la División Guantánamo de la Empresa DESOFT.....	5
Figura 2. Ejemplo de Modulo en Odoo Framework	19
Figura 3. Funcionamiento Interno del Odoo	20
Figura 6. Diagrama de despliegue	38
Figura 7. Prototipo de Login (SCONSOL)	40
Figura 8. Prototipo de Vista Kaban (Conexión al VERSAT)	41
Figura 9. Prototipo de Vista Formulario (Obligaciones MN)	41
Figura 10. Prototipo de Vista Detalle (Factura).....	42
Figura 11. Prototipo de Vista Lista (Cuentas por Pagar MN).....	42
Figura 12. Prototipo de Vista Genérica (Obligaciones en MLC)	43

Índice de Tablas

Tabla 1. Product Backlog	28
Tabla 2. Especificaciones del producto	28
Tabla 3. Resumen de la característica de la aplicación	29
Tabla 4. Roles del equipo de trabajo	30
Tabla 5. Historia de Usuario	32
Tabla 6. Sprint Backlog	34

INTRODUCCIÓN

Actualmente el uso de las tecnologías de la Información y la Comunicaciones (TIC) y el incremento de los volúmenes de información que hacen que los procesos se complejicen de manera acelerada, por lo que se han convertido en una necesidad inmediata para solucionar problemas cada vez más crecientes en nuestra sociedad, además realizar de manera más eficientes y dinámicas nuestras tareas diarias y de largos períodos, garantizar su durabilidad y mantenimiento. Las redes de computadoras permiten acelerar a velocidades enormes la generación, transmisión y distribución de la información en tiempos instantáneos sin distinción de formato de transporte y distancia resultando así una herramienta indispensable para gestionar eficientemente la información.

En estos momentos, los sistemas de gestión que emplean las organizaciones evolucionan gradualmente en correspondencia con el desarrollo económico, político, social y tecnológico de la sociedad, debido a que es una vía eficaz y eficiente para toma de decisiones.

El comportamiento de las cuentas por cobrar y pagar marca un distanciamiento o no de la actividad financiera con la económica por lo que la consolidación y gestión de la información de estas cuentas es de vital importancia garantizando que se cumplan lo pactado por el cliente y su empleador a fin de cumplir las pautas contratadas, en actividades de pago y de cobro.

En la Empresa de Aplicaciones Informáticas DESOFT, se prestan varios servicios informáticos, los cuales contribuyen a que se generan constantemente obligaciones por factura emitida y para garantizar el cumplimiento de estas y otras actividades se generan obligaciones por factura recibidas que se registran en el Sistema Informático VERSAT SARASOLA el cual se analizan semanalmente; la búsqueda, recopilación, consolidación y gestión de esa información se realiza de manera lenta y trabajosa, pues hay que sacar esa información y luego procesarla para su posterior análisis y toma de decisiones.

Este proceso tiene como deficiencias:

1. Complejidad en el proceso de extracción y Consolidación de la Información de las Cuentas por Cobrar y Pagar.
2. Demora en la Gestión de la Información de las Obligaciones de Cobro y Pago a fin tomar prontas decisiones económicas y financieras.
3. El proceso de gestión de la información de las cuentas por cobrar y pagar se realiza de forma semi manual pudiendo ocurrir errores de transcripción en los documentos.

Por lo anteriormente expuesto conlleva al siguiente **problema a resolver**:
¿Cómo mejorar la Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo?

Planteándonos como **Objeto de Estudio**: Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.

Objetivo: Mejorar la Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo a través de un Sistema Informático.

Campo de acción: Consolidación de la Información de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.

Idea a defender: Con la realización del sistema informático para la Consolidación y la Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT permitirá un mejor control de las cuentas por cobrar y pagar, conocer con prontitud y eficiencia las obligaciones de cobro y pago a fin de tomar prontas decisiones económicas y financieras.

Para dar cumplimiento al objetivo propuesto se trazaron las siguientes **tareas de investigación**:

1. Caracterización del Proceso de Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT.
2. Realizar un estudio de la metodología, herramienta y tecnologías a utilizar en el desarrollo del software.

3. Análisis y Diseño del Sistema Web para la Gestión y Consolidación de las Cuentas por Cobrar y Pagar
4. Implementación del Sistema Web para la Gestión y Consolidación de las Cuentas por Cobrar y Pagar.
5. Realizar pruebas de aceptación a la aplicación.
6. Realizar un estudio de la factibilidad del Sistema Web para la Gestión y Consolidación de las Cuentas por Cobrar y Pagar.

Para desarrollar el proceso de la investigación se utilizaron los siguientes métodos científicos:

Métodos teóricos.

- Análisis y síntesis: que permite interpretar la teoría de las bibliografías y documentación revisada, así como su sistematización en el proceso de elaboración y fundamentación teórica de las tecnologías en el objeto de estudio, y en la propuesta elaborada.
- Inductivo y deductivo: para razonar, llegar a conclusiones y tomar decisiones acerca de las ideas principales y acciones que conforman la aplicación propuesta.

Métodos empíricos.

- Entrevistas: al jefe de departamento y especialistas del departamento de Economía de la División Guantánamo para conocer la forma en que se realiza el proceso de consolidación y la gestión de la información de las cuentas por cobrar y pagar; las características de sus dependencias con el propósito de declarar los requisitos funcionales del sistema informático.
- Revisión documental: para realizar el análisis de los documentos por los que se rige la entidad, para realizar el proceso de gestión de las cuentas por cobrar y pagar con el objetivo de comprender su funcionamiento.
- Consultas de usuarios: para evaluar la factibilidad del sistema informático que se propone.

El presente documento se estructura en un resumen en español e inglés, dos capítulos, conclusiones, bibliografía y anexos.

En el Capítulo I: Fundamentación teórica. Se describe la situación actual de la investigación, así como la definición las herramientas y tecnologías a utilizar el desarrollo del software.

Capitulo II: Implementación del Sistema Informático. Se realiza una descripción de cómo fue implementado el software.

CAPÍTULO I. Fundamentación teórica.

1.1 Caracterización del objeto de estudio

La División Guantánamo de la Empresa DESOFT, partiendo de su fortaleza identificada tiene determinada.

OBJETO SOCIAL.

Prestar servicios de desarrollo, despliegue y soporte de aplicaciones informáticas integrales. Brindar servicios públicos de Alojamiento y Hospedaje de Aplicaciones y Sistemas Informáticos (Tecnología en las nubes). Gestionar y administrar aplicaciones informáticas en Centros de Datos.

MISIÓN.

Desarrollar y comercializar productos y servicios informáticos integrales asociados a las Tecnologías de la Información, contribuyendo al desarrollo sostenible de la sociedad.

VISIÓN.

Ser una empresa de Alta Tecnología basada en el uso intensivo de las Tecnologías de la Información con Alto Reconocimiento Social a partir de su impacto en la sociedad del conocimiento y en la gestación de una fuerte comunidad TI.



Figura 1. Organigrama de la División Guantánamo de la Empresa DESOFT

En la actividad periódica de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo se realiza el proceso de facturación el cual está dirigido por el departamento Comercial y supervisado por el de Economía a fin de que se puedan realizar correctamente las actividades de dicha entidad.

Todas estas actividades son registradas en el software VERSAT SARASOLA, el cual se encarga de tener el peso de la actividad económica de la entidad.

Las Cuentas por cobrar salen de las Obligaciones de Pago por factura emitida a las entidades clientes de acuerdo a las relaciones contractuales, así como a su vez las Cuentas por Pagar salen de las Obligaciones de Cobro por factura recibida en la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.

Todas las semanas tiene lugar una reunión con los departamentos implicados el cual analizan las cuentas por cobrar y pagar que se tienen en dicha entidad, así como el análisis de otros indicadores del proceso de facturación. En tal sentido se hace extremadamente engorroso consolidar y mostrar de manera más diversa y desglosada reportes que permitan un análisis más profundo de la situación financiera actual. También se vuelve un problema que los especialistas que realizan el acto de cobro en las diferentes empresas no cuenten con un Sistema que le permita conocer acerca de esas cuentas por cobrar y sus componentes en tiempo real.

Diagnostico de la situación actual

Después de realizar una búsqueda a través de la navegación por internet, se pudo estudiar una serie de sistemas informático que hacen referencia a la gestión económica relacionados evidentemente con las cuentas por cobrar y pagar derivadas de las obligaciones de las entidades

VERSAT SARASOLA, es un sistema de gestión contable-financiero, y se reconoce como el primer sistema cubano de gestión contable-financiera autóctono y más versátil, pues facilita trabajar con la doble moneda todavía presente en nuestra sociedad, está basado en módulos que al día de hoy dentro de los que se encuentran: Configuración, Contabilidad, Costos y Procesos, Control de Inventarios, Finanzas y Caja, Activos Fijos, Contratación y

Facturación, Nóminas, Planificación y Presupuestos, Gestión de Información, Auditoría, Logística, Punto de Ventas y otros

El software antes mencionado no da respuesta del todo a nuestra situación problemática, pues no presenta de una manera entendible las cuentas por cobrar y pagar, así como las operaciones interdependencias al haber varias entidades interactuando, es un Sistema Informático de Escritorio o Desktop lo que hace que la información sea visible en aquellas computadoras en la que la aplicación cliente se encuentre instalada. El software propuesto tiene como objetivo principal Consolidar y Gestionar la información de las Cuentas por Cobrar y Pagar, así como informaciones de venta y financiera necesarias para la toma de decisiones oportunas vía web y accesible para todas las personas de la red que se le hallan otorgado los permisos.

Como parte de la investigación y para conocer como se desarrolla actualmente la Gestión de las Cuentas por Cobrar y Pagar se realizaron entrevistas al Jefe del Departamento de Economía y a la Jefa del Departamento Comercial donde se constato la existencia de varias deficiencias:

1. Complejidad en el proceso de extracción y Consolidación de la Información de las Cuentas por Cobrar y Pagar.
2. Demora en la Gestión de la Información de las Obligaciones de Cobro y Pago a fin tomar prontas decisiones económicas y financieras.
3. El proceso de gestión de la información de las cuentas por cobrar y pagar se realiza de forma semi-manual pudiendo ocurrir errores de transcripción en los documentos.

1.2 Descripción de la metodología y las tecnologías actuales que respaldan la propuesta.

Seleccionar la metodología y las tecnologías adecuada para desarrollar el sistema informático, es de vital importancia para un desempeño favorable y mantenible en el tiempo, capaz de dar solución a nuestro problema en el menor tiempo posible, con calidad y de manera eficiente.

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

1.2.1 Cliente – Servidor

Cliente – Servidor: Según International Business Machines (IBM) es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, a través de la organización, en múltiples plataformas. El modelo soporta un ambiente distribuido en el cual los requerimientos de servicio, hechos por estaciones de trabajo o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores.

La arquitectura cliente-servidor es un caso especial de la arquitectura de tres capas en la cual la lógica del negocio o de la aplicación se ubica en una de las otras capas (en la capa del cliente o en la de base de datos).

A continuación, se muestran algunas ventajas de la arquitectura cliente-servidor:

Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).

Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de cliente-servidor que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

Fácil mantenimiento: Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible realizar actualizaciones o

reemplazos de tecnologías donde el cliente no se ve afectado por ese cambio (o se afectarán mínimamente).

1.2.2 Metodología a utilizar

Scrum: Un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

- Ligero
- Fácil de entender
- Extremadamente difícil de llegar a dominar

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos.

Teoría de Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

Reunión de Planificación del Sprint (Sprint Planning Meeting)

- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

El equipo Scrum (Scrum Team)

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

El Dueño de Producto (Product Owner)

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). La gestión de la Lista del Producto incluye:

Expresar claramente los elementos de la Lista del Producto;

Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;

Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo;

Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,

Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

El Dueño de Producto es una única persona, no un comité. El Dueño de Producto podría representar los deseos de un comité en la Lista del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través del Dueño de Producto.

Para que el Dueño de Producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Lista del Producto. No está permitido que nadie pida al Equipo de Desarrollo que trabaje con base en un conjunto diferente de requerimientos, y el Equipo de Desarrollo no debe actuar con base en lo que diga cualquier otra persona.

El equipo de desarrollo (Development Team)

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente despleables;

Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto;

Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;

Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla; y,

Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección

y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar las vitales transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse.

El Sprint

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

Lista de Producto (Product Backlog)

La Lista de Producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.

Lista de Pendientes del Sprint (Sprint Backlog)

La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de

Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint.

Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no.

¿Por qué esta metodología y no otra?

La metodología Scrum encaja perfectamente con el tipo de proyecto, las condiciones de desarrollo así como la idea que se tiene del sistema. Las razones fundamentales que se tuvieron en cuenta al escoger esta metodología es que está guiada y desarrollada por Sprint con tiempos que varían de una semana hasta un mes máximo.

El equipo de desarrollo es versátil y con conocimientos relativamente completos a fin de que existan la menor dependencia posible de agentes externos.

Se realizan continuas y completas liberaciones periódicas el cual el cliente ya puede trabajar.

1.2.3 Lenguaje Unificado de Modelado (UML).

Unified Modeling Language, por sus siglas en inglés UML, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema

(modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software tal como el XP.

1.2.4 Desarrollo de la aplicación Web.

Las aplicaciones web son sistemas informáticos que utilizan los usuarios para acceder a un servidor web mediante una red local o internet. En la actualidad se han convertidos en sistemas muy usados debido a dos razones bien fundamentadas, la primera es la gran práctica del navegador web como cliente ligero, la segunda es la facilidad que brindan para actualizar y darle mantenimiento a las aplicaciones sin tener que distribuir ni instalar software en las computadoras de los usuarios.

Una aplicación web genera una serie de páginas dinámicas en un formato estándar común como es HTML (Hypertext Markup Language: Lenguaje de Marcado de Hipertexto), o XHTML (Extensible Hypertext Markup Language: Lenguaje Extensible de Marcado de Hipertexto), soportado por los navegadores web. Utilizan lenguajes interpretados del lado del cliente, para añadir elementos dinámicos a la interfaz de usuario.

El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías web pasando desde las primeras páginas escritas en HTML almacenadas en un fichero en el servidor web hasta aquellas que se generan dinámicamente como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Dentro de las grandes ventajas que presentan las aplicaciones web están que:

Facilita el trabajo a distancia. Son fáciles de usar (no requieren conocimientos avanzados de computación).

Con una aplicación Web tendrá alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde tenga acceso a Internet y a cualquier momento.

No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador Web.

La diferencia entre una aplicación Web y un sitio Web, está dada por su utilización ya que las aplicaciones implementan lógica de negocio y los sitios presentan la información por medio de código HTML, que es interpretado y ejecutado por el navegador.

1.3 Tecnologías del lado del cliente

1.3.1 Mozilla Firefox

El navegador es una aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden, esto permite acceder a otros recursos de información alojados también en Servidores web, como pueden ser videos, imágenes, audio y archivos XML

Mozilla Firefox es un navegador web libre y gratuito que ofrece numerosas opciones de personalización, un funcionamiento excelente y que es conocido por la protección de la vida privada. Integra funcionalidades por defecto que se pueden ampliar gracias a un catálogo de extensiones muy largo. Ofrece una navegación por pestañas y un sistema de motores de búsqueda integrados y fácilmente personalizables. Se encuentra en constante actualización y cuenta con una amplia comunidad donde se desarrollan plugin y extensiones que le brindan una mayor utilidad. Además, cuenta con herramientas para depurar aplicaciones web.

1.4 Lenguajes de Programación

Un lenguaje de programación es un lenguaje artificial para representar un algoritmo de tal forma que sea comprensible tanto para el humano como para

el ordenador. Estos permiten definir de una manera sencilla una secuencia de instrucciones para resolver un problema específico.

El lenguaje escogido para el desarrollo del sistema de consolidación y gestión es el Python en su versión 3.7.

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel con un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Python es fácil de usar y de instalar, está disponible para sistemas operativos Windows, Mac OS y Linux, lo que hace que sea un lenguaje multiplataforma.

Python permite separar el programa en módulos que pueden reusarse en otros programas en Python. Viene con una gran colección de módulos estándar que puedes usar como base de los programas.

1.4.1 Framework de desarrollo a utilizar

Un marco de trabajo o framework es una estructura conceptual y tecnológica de soporte definido. Simplifica el desarrollo de las aplicaciones mediante la automatización de muchas tareas comunes, facilitando a los diseñadores y programadores dedicarse por completo a los aspectos específicos de cada aplicación.

Odoo es más que un Framework, pues está en la categoría de ERP (Enterprise Resource Planning) o Sistemas de Planificación de Recursos Empresariales, que son un conjunto de sistemas de información que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad.

Odoo posee una arquitectura de tres capas: una capa para el cliente Odoo, otra capa para el servidor Odoo y otra para la conexión y manejo de la BD.

Además sigue el patrón de diseño 'MVC' (Modelo-Vista-Controlador), con lo que separa la parte que maneja la vista de la que maneja la lógica la aplicación, usando como intermediario un controlador que captura las peticiones del cliente y las comunica al modelo, y que entrega al cliente las respuestas fabricadas por el modelo.

La peculiaridad es que los módulos de Odoo de tipo 'backend' manejan de forma diferente este patrón. Al implementar un módulo de este tipo, el programador no puede fabricar su controlador para recibir las peticiones del cliente, procesarlas, comunicárselas al modelo de la aplicación, y entregar las respuestas de éste al cliente.

En el caso de estos módulos, Odoo proporciona su propio controlador. Este controlador forma parte del núcleo de Odoo.

El núcleo de Odoo, programado en lenguaje Python, es el que dota de toda la funcionalidad necesaria a más bajo nivel para que todo funcione, incluidos los módulos programados para Odoo.

El framework de Odoo, que proporciona todas las herramientas necesarias para la construcción de módulos en Odoo, también forma parte de este núcleo.

Los módulos 'backend' son construidos usando todas las funcionalidades que proporciona este framework. Sin duda, una de las más importantes es el ORM que nos permite comunicarnos con la BD.

¿Qué es un ORM?

ORM son las siglas en inglés de 'Object Relational Mapping', en español "mapeo de objetos a bases de datos".

ORM es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos utilizado y el sistema de base de datos relacional usado en el desarrollo de nuestra aplicación sin que el programador tenga que usar SQL para manejar la BD.

El ORM de Odoo

Odoo proporciona su propio ORM que permite manejar los datos de la base de datos con una API propia. Esta API nos dice cómo hacer las operaciones sobre la BD, así como la manera de definir las columnas de cada tabla de la BD usando su propia sintaxis.

Para usar esta API debemos utilizar el lenguaje Python.

Odoo maneja el término modelo, que a partir de ahora nombraré en varias ocasiones.

Un modelo no es más que una clase programada en Python que corresponderá a una tabla de la BD. Por ejemplo:

```
class ObligationMN(models.Model):
    _name = 'sconsol.obligation.mn'
    _description = 'Obligation MN'

    id_doc = fields.Char(string='Documento', required=False)
    ueb = fields.Char(string='Unidad Básica')
    dabase = fields.Char(string='Base de datos')
    client = fields.Char(string='Cliente')
    short_name = fields.Char(string='Nombre Corto', required=False)
    nro_invoice = fields.Char(string='Nro Factura', required=False)
    date = fields.Datetime(string='Fecha', required=False)
    amount = fields.Float(string='Importe', group_operator="sum")
    paid = fields.Float(string='Pagado', group_operator="sum")
    account = fields.Char(string='Cuenta', group_operator=False)
    account_description = fields.Char(string='Descripción', required=False)
    days = fields.Integer(string='Días', required=False)
```

Figura 2. Ejemplo de Modulo en Odoo Framework

En el ejemplo anterior estamos definiendo una clase llamada ObligationMN que se mapeará en la BD en forma de una tabla de nombre 'sconsol_obligation_mn'. Vemos que el nombre de tabla corresponde al atributo '_name' de la clase separando los puntos por guiones bajos.

Funcionamiento interno de Odoo

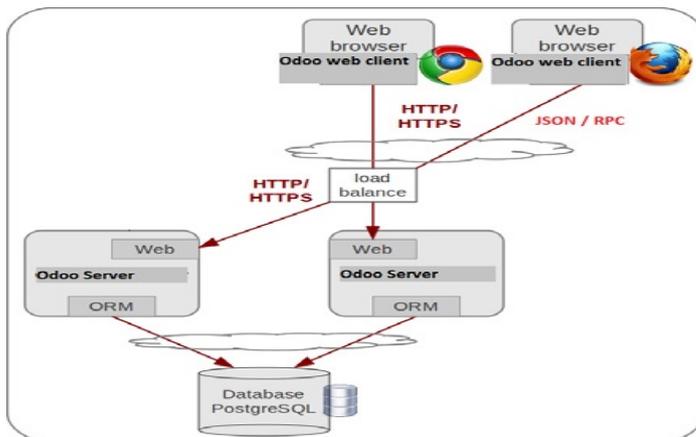


Figura 3. Funcionamiento Interno del Odoo

Como se ve en la ilustración anterior, Odoo consiste en tres componentes básicos:

Un servidor con un SGBD PostgreSQL que contiene todos los datos: datos de cada modelo de cada módulo instalado, el código de las vistas, datos de configuración del propio Odoo, etc.

Un servidor que puede funcionar en local o en cualquier proveedor de hosting online. Este componente es el que contiene el núcleo de Odoo. El núcleo de Odoo está programado en Python y es lo que hace que todo funcione. Una capa de este servidor está destinada a comunicarse con el SGBD PostgreSQL a través del motor del ORM que Odoo incorpora.

Otra capa está destinada a comunicarse con el cliente web Odoo que se ejecuta en el navegador del usuario.

El tercer componente consiste en una completa aplicación JavaScript que se ejecuta en el navegador del usuario para que el usuario pueda usar Odoo sin instalar ninguna aplicación.

Odoo proporciona un completo framework con toda una serie de técnicas y herramientas para la programación de un módulo dentro de esta plataforma.

Este framework nos proporciona todas las herramientas tanto como para construir la lógica de tratamiento de datos de nuestro módulo, así como su

interfaz gráfica, sin tener que preocuparnos de tratar con el controlador de la aplicación para manejar peticiones desde el navegador del usuario.

El componente más importante es el ORM, que nos ofrece una API orientada a objetos para manejar la BD sin usar SQL.

1.5 Tecnologías del lado del Servidor

1.5.1 Sistema Gestor de Base de Datos.

Los Sistemas de Gestión de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Es un grupo de programas que permiten crear y mantener una base de datos, además se puede definir también como “el conjunto de herramientas que suministran a todos, ya sea a un administrador, analista, programador, usuario; los medios necesarios para describir, recuperar y manipular los datos en la base de datos, manteniendo la seguridad, integridad y confiabilidad de los mismos”. Estos poseen varias ventajas y dentro de ellas mencionamos algunas a continuación:

- Proveen facilidades para la manipulación de grandes volúmenes de datos.
- Simplifican la programación de equipos de consistencia.
- Garantizan que los cambios de la base serán siempre consistentes.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Disminuyen los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

1.5.2 Base de Datos

PostgreSQL Versión 10.0

Potente sistema de base de datos, de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad y corrección. Se ejecuta en la mayoría de los sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS, Solaris, Tru64) y Windows. Tiene soporte completo para claves foráneas, combinaciones, vistas, triggers, subconsultas. Es extensible o lo que es lo mismo, posibilita que los usuarios implementen o definan nuevos tipos de datos a su comodidad. También es compatible con el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeos.

PgAdmin III

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Incluye una interfaz administrativa gráfica y herramientas de consulta SQL así como un editor de código procedural. Está diseñado para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, Mac y Solaris.

1.5.3 Servidor Web

Servidor Web: Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web.

Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

Odoo trae su propio servidor web basado en tecnologías wsgi.

1.5.4 Herramienta CASE.

Para el desarrollo de nuestra aplicación web no se emplearon herramientas para case convencionales, se uso en su lugar el generador de diagramas de clases de PyCharm versión 2020.1.

1.3 Reglas del Negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, puesto que regulan algún aspecto del negocio. (Jacobson & Rumbaugh, 2006)

1. Para realizar la consolidación de los datos debe estar instalado el VERSAT SARASOLA.
2. Se debe tener acceso a las bases de datos del VERSAT
3. Se imprimirán solo las cuentas por cobrar que no estén en litigio por parte de la entidad.
4. Los reportes se desglosarán por edades o sea de 0 a 30, de 30 a 60, de 60 a 90 y de más de 90 días.
5. Las cuentas por cobrar deben desglosarse por clientes y por unidades.

Teniendo en cuenta las deficiencias encontradas en el proceso de Gestión de las Cuentas por Cobrar y Pagar de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo se proponen las siguientes mejoras:

1. Informatizar el proceso de Gestión y Consolidación de las Cuentas por Cobrar y Pagar.
2. Garantizar la centralización de la información de las Cuentas por Cobrar y Pagar.
3. Garantizar la toma oportuna de decisiones contables-financieras a partir de la información consolidada.
4. Permitir el acceso via web a la información de las Cuentas por Cobrar y Pagar.

Conclusiones parcial del capítulo

En este capítulo se realiza la caracterización del Proceso de Gestión de las Cuentas por Cobrar y Pagar que tienen lugar en la Subdirección de Economía y Comercial en la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo, se diagnosticó el estado actual donde se evidenció que existen un conjunto de deficiencias. Se propuso un conjunto de mejoras potenciales para erradicar las deficiencias analizadas y contribuir al aumento de la eficiencia del proceso.

Se brinda, una panorámica de las tendencias y tecnologías actuales para el desarrollo del sistema, un análisis crítico de las herramientas utilizadas, teniendo en cuenta la exigencia del cliente. Se utilizó SCRUM como metodología ágil para realizar el proceso de ingeniería de desarrollo del software.

Capítulo II. Análisis, diseño e implementación. Sistema Informático SCONSOL

En el presente capítulo se realiza la descripción del Sistema Informático para la Gestión y Consolidación de las Cuentas por Cobrar. Se especifican los requerimientos de software permitiendo identificar las historias de usuarios con mayor prioridad, se realizan el análisis y diseño de casos de pruebas de la aplicación y se detectan los errores y fallas que el sistema podría generar para verificar la calidad del software. Se realiza además el estudio de factibilidad.

2.1 Especificación de los requerimientos de software

Con la especificación de los requerimientos del software, se garantiza la definición del producto que se desea construir, lo cual permite describir completamente el comportamiento de la aplicación a elaborar. Estos requerimientos son clasificados, basándose en la funcionalidad del sistema, en requisitos funcionales y no funcionales.

2.1.1 Requisitos Funcionales

Son los que el usuario solicita que efectúe la aplicación. Para la aplicación se definieron las siguientes funcionalidades:

- ***RF1: Gestionar Información de Usuario.***
- ***RF2: Gestionar conexión al VERSAT SARASOLA***
- ***RF3: Gestionar Obligaciones de Cobro***
- ***RF4: Gestionar Obligaciones de Pago***
- ***RF5: Gestionar Cuentas por Cobrar.***
- ***RF6: Gestionar Cuentas por Pagar.***
- ***RF7: Obtener Facturas Emitidas.***
- ***RF8: Obtener Facturas Recibidas.***

Nota: Se define como gestión de la información al proceso de insertar, listar, filtrar y editar.

2.1.2 Requisitos no Funcionales

Los requisitos no funcionales son las características que deben hacer al producto atractivo, utilizable, rápido y confiable.

RNF1 Seguridad:

Se deberán implementar mecanismos de seguridad para evitar que al sistema accedan personas no autorizadas; en dependencia de los niveles de privilegios será la información a la que tendrá acceso el usuario.

RNF2 Rendimiento:

El sistema debe ser eficiente, preciso, debe brindar cortos tiempos de respuesta y gran velocidad de procesamiento de la información.

RNF3 Confiabilidad:

El sistema debe garantizar que los usuarios no avanzados perciban todas sus salidas y pueden trabajar con él cómodamente.

RNF4 Usabilidad:

El software debe garantizar un acceso fácil y rápido a los usuarios. Posibilitar las alertas ante los diferentes errores que cometa el usuario.

RNF5 Interfaz de Usuario:

El diseño de interfaz debe ser sencillo, contribuyendo a acelerar la velocidad de respuesta; amigable, legible e interactivo. Cualquier usuario debe tener posibilidad de interactuar con la información sin necesidad de ayuda.

RNF6 Software requerido:

Navegadores web: Firefox v60.0 o superior, Chrome v70.0 o superior

Servidor de Base de Datos: Postgresql v10.0

Odoo Community en su versión 13

RNF7 Características del Hardware:

Para la utilización de la aplicación los usuarios requieren de una microcomputadora conectada a red con las siguientes especificaciones:

- Memoria RAM: 512MB o superior.

- Procesadores de INTEL o AMD a 1.8GHz o superior.
- Tarjeta de red o módem configurable para el protocolo TCP/IP.

Para el funcionamiento de la aplicación se necesita un servidor con las siguientes características:

- Memoria RAM: 1GB o superior.
- Procesador a 1.8 GHz o superior.
- Tarjeta de red o módem configurable para el protocolo TCP/IP. Necesita de una red estable con un ancho de banda 100Mbps.

2.1.3 Pila de producto

La pila del producto es la lista ordenada de todo aquello que el propietario de producto cree que necesita el producto. Es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de los sucesivos sprints. Representa todo aquello que esperan el cliente, los usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo debe estar reflejado en esta pila. (Pila del producto - Scrum Manager BoK, s. f.).

ID	PRIORIDAD	DESCRIPCIÓN
1	Alta	Gestionar Información de Usuario.
2	Alta	Gestionar conexión al VERSAT SARASOLA
3	Alta	Gestionar Obligaciones de Cobro
4	Alta	Gestionar Obligaciones de Pago
5	Alta	Gestionar Cuentas por Cobrar.
6	Alta	Gestionar Cuentas por Pagar.
7	Alta	Obtener Facturas Emitidas.
8	Alta	Obtener Facturas Recibidas.

Tabla 1. Product Backlog

2.2 Descripción del Sistema

Ante las insuficiencias en el proceso de consolidación y gestión de la información de las Cuentas por Cobrar y Pagar se propone el desarrollo de un sistema informático para la consolidación y gestión de la información con ese fin para la Empresa de Aplicaciones Informáticas DESOFT Guantánamo, que permita a través de la intranet y/o internet viabilizar dicha información.

Para	Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.
Quienes	Presentan deficiencias en la consolidación y gestión de las Cuentas por Cobrar y Pagar.
Nombre del Producto	SCONSOL: Sistema Informático para la Consolidación y Gestión de la información de las Cuentas por Cobrar y Pagar.
Que	Permita mantener una base de datos única y actualizada para poder gestionar la información entre todo el personal técnico y administrativo que interactúan en el proceso de Consolidación y Gestión de la información de las Cuentas por Cobrar y Pagar.
No como	Un sistema aislado, con dificultades para gestionar la información de los especialistas y personal administrativo de forma descentralizada.
El producto	Consolida y Gestiona la información de las Cuentas por Cobrar y Pagar.

Tabla 2. Especificaciones del producto

Beneficios	Descripción
------------	-------------

<p>La existencia de un sistema de consolidación y gestión de la información</p>	<p>Permite la consolidación y gestión de la información de las Cuentas por Cobrar y Pagar.</p> <p>Permite la consolidación y gestión de las informaciones referidas entre dependencias.</p>
<p>La información estará actualizada y centralizada</p>	<p>Se utiliza la Web con previa autenticación para la gestión de los especialistas, personal administrativos y además se mantiene una base de datos con niveles de seguridad donde se recoge la información referente a las Cuentas por Cobrar y Pagar de dicha entidad.</p>
<p>La aplicación permite multiusuario</p>	<p>Se puede visualizar y gestionar la información desde distintas máquinas y por varios usuarios conectados simultáneamente, ello determina que la comunicación entre especialistas sea por medio de la aplicación y vía web.</p>

Tabla 3. Resumen de la característica de la aplicación

Roles del equipo de trabajo Los roles definen las responsabilidades de cada persona que interviene en el desarrollo del producto. A continuación se presenta la organización por roles del equipo de trabajo.

Rol	Nombre(s) y Apellidos	Estatus	Descripción
Product Owner	Mary Luz Matos Utria	Profesora	<p>Responsable de Maximizar el valor del producto y el trabajo del Equipo de Desarrollo.</p> <p>Es la única persona responsable de</p>

			gestionar la Lista del Producto (Product Backlog)
Scrum Master	Antonio Ruban Espinal	Especialista B	Líder del servicio de Equipo Scrum. Ayuda a las personas externas y al Equipo Scrum a entender las iteraciones pueden ser útiles y cuales no.
Scrum TEAM	Manuel Vinent Guilarte	Estudiante	Desarrolla los módulos necesarios para su implementación
Stakeholders		Subdirector de Economía Subdirectora del Departamento Comercial	Tributan información y lógica para su posterior implementación

Tabla 4. Roles del equipo de trabajo

Usuarios del Sistema

Los usuarios de la aplicación son todas aquellas personas que interactúan de alguna manera con esta o que desempeñan algún rol específico en la operatividad de la misma. Para la aplicación los usuarios representan distintos niveles de accesibilidad, según este nivel, lo mismos tienen privilegios o

permisos que determinan el grado de operatividad de acuerdo a su rol. Estos usuarios en el propio contexto de la aplicación se clasifican en:

Administrador: Es el encargado de administrar que usuarios van a tener permisos o privilegios para la gestión de la información dentro del software. Es el único capacitado de cambiar los roles a un usuario. Además es el único que va a poder añadir o quitar usuarios del sistema.

Especialista: Este usuario es el encargado de gestionar la gran mayoría de la información que tiene el software, con el objetivo de mantener actualizado el contenido dentro del mismo y será el encargado de emitir los reportes.

2.1 Historias de Usuarios

Son las descripciones de las funcionalidades que va a tener el software.

Estas historias de usuario, serán el resultado de la colaboración entre el cliente y el equipo, e irán evolucionando durante toda la vida del proyecto.

A continuación, se presenta una muestra de las HU definidas en este proceso.

ANVERSO	
ID: 6	TÍTULO: Gestionar Cuentas por Cobrar
DESCRIPCIÓN: Como especialista quiero poder obtener por tipo de cuentas las Cuentas por Cobrar y Pagar, pudiendo elegir entre una cuenta y la otra a fin obtener información detallada de esas cuentas.	
ESTIMACIÓN: 1	
PRIORIDAD: M	DEPENDIENTE: 1, 2
REVERSO	
PRUEBAS DE ACEPTACIÓN:	
Introducir una Empresa que no tiene Cuentas por Cobrar	
Alterar dos resultado en las Cuentas	

Introducir datos incorrectos y comprobar que no sean registrados
--

Tabla 5. Historia de Usuario

Descripción de los elementos

ID: Identificador de la historia de usuario.

TÍTULO: Título descriptivo de la historia de usuario.

DESCRIPCIÓN: Descripción sintetizada de la historia de usuario.

ESTIMACIÓN: Evaluación del coste de implementación en unidades de desarrollo. Estas unidades representarán el tiempo teórico (desarrollo/hombre) que se haya estimado al comienzo del proyecto.

PRIORIDAD: Prioridad en la implementación de la historia de usuario respecto al resto de las historias de usuario. A mayor número, mayor prioridad. Otra aproximación a la priorización de tareas se hace a través del método MoSCoW:

- M – Must, se debe completar este requerimiento para finalizar el proyecto.
- S – Should, se debe completar este proyecto por todos los medios, pero el éxito del proyecto no depende de él.
- C – Could, se debería completar este requerimiento si su implementación no afecta a la consecución de los objetivos principales del proyecto.
- W – Would, se puede completar este requerimiento si sobra tiempo de desarrollo (o en futuras versiones del mismo)

DEPENDENCIAS: Una historia de usuario no debería ser dependiente de otra historia, pero a veces es inevitable. En este apartado se indicarían los IDs de las tareas de las que depende una tarea.

2.2.2 Planificación del Sprint

La pila del sprint es la lista de tareas que va a realizar el equipo en una iteración, para construir un incremento. (Pila del sprint - Scrum Manager BoK, s. f.). Cada tarea de la pila del sprint tiene asignada una persona, y la

indicación del tiempo que aún falta para terminarla. Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos complejos de gestión.

La planificación del Sprint tiene un límite de tiempo hasta ocho horas para un sprint de un mes. Para Sprints mas cortos el evento suele más corto. (What is Sprint Planning?, 2020).

Sprint 1			
Pila del Sprint: Gestionar Cuentas por Cobrar			
Inicio: 12-01-2020		Duración: 20 días	
ID	Tareas	Estado	Responsable
1	Creamos un menú y dentro del damos acceso para gestionar las Cuentas por Cobrar.	T	Manuel Vinent Guilarte
2	Creamos la vista en forma de árbol para ver las Cuentas por Cobrar por clientes y luego por unidades.	T	Manuel Vinent Guilarte
3	Creamos la vista formulario para la Creación de una nueva Cuenta por Cobrar	T	Manuel Vinent Guilarte
4	Creamos una función que creará una Cuenta por Cobrar.	T	Manuel Vinent Guilarte
5	Creamos una función que modificará una Cuenta por Cobrar.	T	Manuel Vinent Guilarte
6	Creamos una función que obtendrá una Cuenta por Cobrar.	T	Manuel Vinent Guilarte
7	Creamos una función que eliminará una	T	Manuel Vinent

	Cuenta por Cobrar.		Guilarte
--	--------------------	--	----------

Tabla 6. Sprint Backlog

2.3 Diseño de la Base de Datos

El diseño de la base de datos durante el desarrollo del sistema es una actividad relevante para asegurar la calidad del producto. Esta actividad, en sus diferentes niveles: diseño conceptual, diseño lógico y diseño físico, brindan al trabajador del sistema o al equipo de desarrollo la posibilidad de estructurar el producto de manera coherente. El diseño conceptual no está necesariamente asociado con una plataforma de implementación, sino que es más cercano a la realidad, al problema a solucionar. La base de datos ha sido implementada utilizando Postgresql 10 como gestor de bases de. La normalización de la base de datos se realiza con el objetivo general de no presentar problemas durante la operación de actualización al minimizar la redundancia en los datos, se logra así un buen diseño de la base de datos. La realización de la 2da fase supone que se ha concluido la 1ra y así sucesivamente. Tras completar cada fase se dice que la relación puede estar en:

- 1ra Forma Normal: Se dice que una relación está en 1ra Forma Normal si todos los dominios contienen valores unarios o atómicos.
- 2da Forma Normal: Una relación está en 2da Forma Normal, si estando en 1ra Forma Normal todo atributo que no sea clave, depende totalmente de la clave primaria.
- 3ra Forma Normal: Una relación está en 3ra Forma Normal, si estando en 2da Forma Normal, ningún atributo que no sea clave, depende transitiva-mente de la clave primaria.

Creación y normalización de la base de datos

Al estar usando un ORM no es necesario que nos tengamos que encargar de crear las tablas de la BD. Su tarea es hacer el trabajo sucio por nosotros,

evitando que tengamos que tratar con SQL para crear tablas o realizar operaciones de consulta, inserción, actualización o borrado.

Además, el ORM se encarga de crear para cada modelo, en su tabla correspondiente, las claves primarias (siempre una columna 'id' con valor autoincremental), las claves foráneas a partir de los campos Many2one, One2many y Many2many definidos, los índices para hacer búsquedas más eficientes, las restricciones de cada tabla indicadas en cada modelo, así como las tablas extras que se crean a partir de los campos Many2many.

Normalización

La base de datos se normalizó a la primera, segunda y tercera forma normal con el objetivo de garantizar eficiencia en el proceso de gestión, mantenimiento y robustez.

2.4 Diseño del Sistema

En el diseño se modela el sistema y se encuentra su forma, para que soporte todos los requisitos, incluyendo los no funcionales y otras restricciones relacionadas con el entorno de implementación. El diseño consiste en obtener una visión externa del sistema y constituye un refinamiento del análisis.

2.4.1 Patrones de diseño

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, contribuyendo a dar flexibilidad y extensibilidad al diseño.

Uno de los patrones más usados son los Patrones Generales de Software para Asignar Responsabilidades (GRASP), los cuales tienen como propósito general asignar responsabilidades expresadas en forma de patrones, esto se realiza aplicando algunos principios durante la preparación de los diagramas de interacción a la hora del diseño.

- MVC: El patrón MVC realiza un diseño que separa los datos y la lógica de negocio, con la finalidad de mejorar la reusabilidad, permitiendo esto que las modificaciones realizadas en las vistas influyan en menor medida en la lógica de negocio o de datos.
- Alta cohesión: El patrón Alta Cohesión indica que la información que almacena una clase debe ser coherente, de manera que todos sus métodos tengan un comportamiento bien definido.
- Bajo Acoplamiento: El patrón Bajo Acoplamiento, se utilizó con el objetivo de tratar de mantener las clases lo menos ligadas entre sí que se pueda, de tal forma que en caso de producirse una modificación en alguna de ellas se tenga la mínima repercusión posible en el resto, potenciando la reutilización y disminuyendo la dependencia entre las clases.
- Creador: El patrón Creador se encarga de identificar la clase responsable de la creación de nuevos objetos. Se utilizó en todas las clases controladoras del sistema creando diferentes objetos del modelo según los datos que se necesiten.

La aplicación de estos patrones permitió distribuir responsabilidades entre las diferentes clases definidas, de forma que no existan muchas relaciones y que no sea sobrecargada de métodos una clase en específico, las mismas implementan las responsabilidades propias de manipulación de la información.

2.4.2 Arquitectura

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. Es una representación que permite:

- 1) analizar la eficiencia del diseño para cumplir los requerimientos establecidos,
- 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil

3) reducir los riesgos asociados con la construcción del software.

Arquitectura Cliente-Servidor: Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados "servidores".

A continuación, se muestran algunas ventajas de la arquitectura cliente-servidor:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.

2.5 Diagrama de Despliegue

El diagrama de despliegue del sistema describe la distribución física en términos de nodos, que es la representación de los equipos de cómputo que se necesitan para la ejecución de la aplicación. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo. El sistema está distribuido en diferentes nodos, se especifica uno para la base de datos y otro que instalará las aplicaciones del negocio. Se instalará además un nodo cliente (PC-Cliente). La comunicación entre los nodos se efectuará mediante el

protocolo HTTP. La Figura 2.2 representa el diagrama de despliegue que muestra la estructura del sistema propuesto en cuanto a los nodos y las relaciones entre ellos.

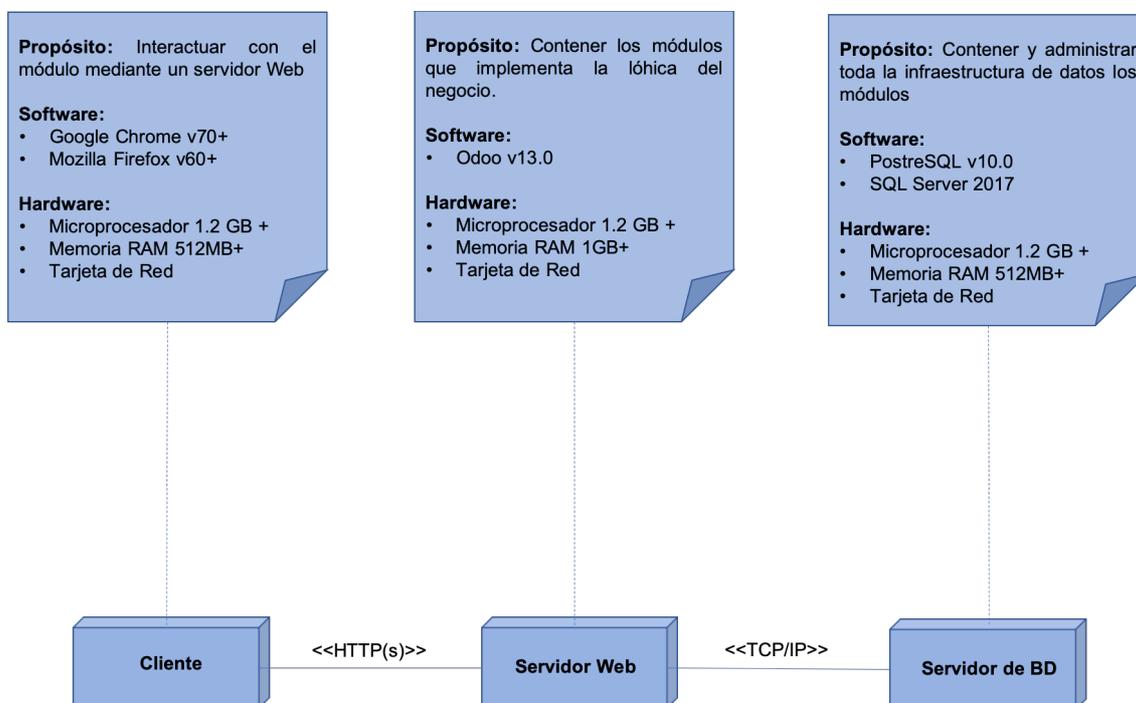


Figura 6. Diagrama de despliegue

2.4

2.6 Estudio de Factibilidad

Factibilidad Técnica

Esta tesis es factible técnicamente, ya que se tiene la disponibilidad y accesibilidad a la información para el desarrollo del aplicativo web. Cabe resaltar que el proceso que se desea automatizar cuenta con el respaldo de aplicaciones anteriormente realizadas en otras instituciones y la capacidad para realizarla, para todo esto se cuenta con herramientas como internet, libros, documentos y equipos de cómputo necesario para el funcionamiento e implementación del sistema de información.

Factibilidad Operativa

Esta tesis es factible operativamente, porque se tiene el conocimiento de la Gestión Contable Financiera de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo, y también los conocimientos necesarios para el desarrollo y la puesta en práctica del aplicativo web que serán mostrados durante el desarrollo de la tesis.

La necesidad y deseo de la consolidación y gestión de las Cuentas por Cobrar y Pagar que presentan en la empresa, expresada por el coordinador del área contable financiera y el personal involucrado con el mismo, llevó a la aceptación del aplicativo web que, de manera más sencilla y amigable, cubrirá todos los requerimientos, expectativas y proporciona la información en forma oportuna y confiable. Basándose en las conversaciones sostenidas con el personal involucrado se demostró que estos no representan ninguna oposición al cambio, por lo que es factible operacionalmente.

Factibilidad Económica

Esta tesis es factible económicamente, debido a que los autores están dispuesto a mejorar la Gestión de las Cuentas por Cobrar y Pagar, y al realizar la inversión en el desarrollo del aplicativo web para la mejora de dicho proceso se evitará gastos innecesarios por parte de la empresa. Se determinaron recursos para desarrollar, implementar y mantener en operación al sistema programado.

Costos del sistema propuesto: El sistema web para el proceso de Consolidación y Gestión de incidencias de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo, involucra los siguientes costos:

- Costo de Hardware y Software: Debido a que la organización cuenta con los equipos y recursos técnicos necesarios, para el desarrollo del sistema web, no hubo la necesidad de adquirir equipos nuevos, ni software porque se usará licencia GPL (GNU). Esta situación facilitó la puesta en marcha del proyecto por parte de la empresa. Ofreciéndole a

la empresa la posibilidad y ventaja de realizar inversiones en otros requerimientos y necesidades de la organización.

- **Costos de Recursos Humanos:** El aplicativo web propuesto no incluyó variaciones en cuanto al personal bajo cuya responsabilidad está la operación y funcionamiento del sistema. El equipo de desarrollo asumirá parte de la inversión, ya que por ser un proyecto elaborado como trabajo de grado y aporta un beneficio para la empresa, el personal encargado de impulsar el mismo como la empresa, asumirán los gastos; aspecto que favoreció aún más en el proyecto en cuestión, cabe destacar que el proceso de venta y postventa. Permitirá llevar un control de las incidencias.

2.6 Implementación a la Aplicación.

Prototipos de Interfaz de Usuario

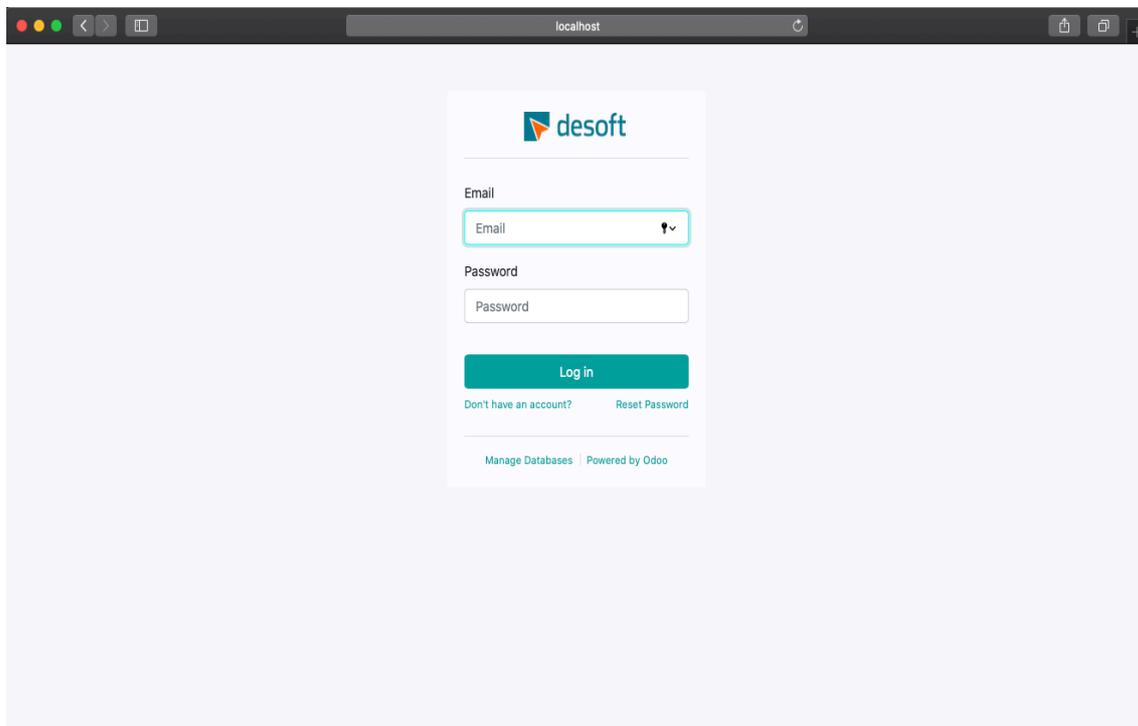


Figura 7. Prototipo de Login (SCONSOL)

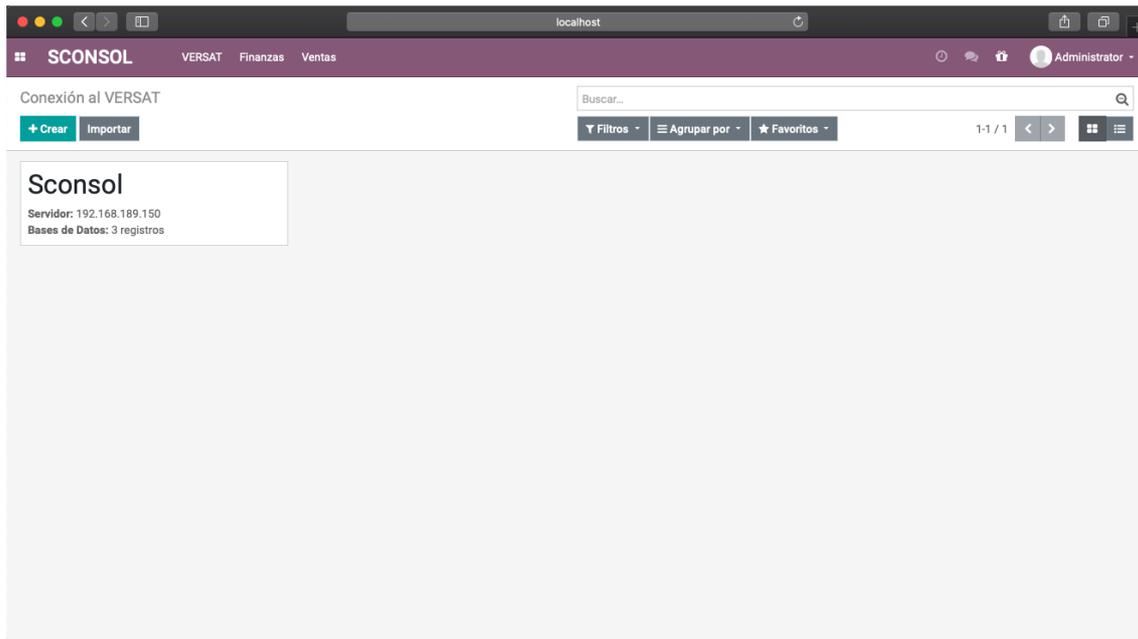


Figura 8. Prototipo de Vista Kaban (Conexión al VERSAT)

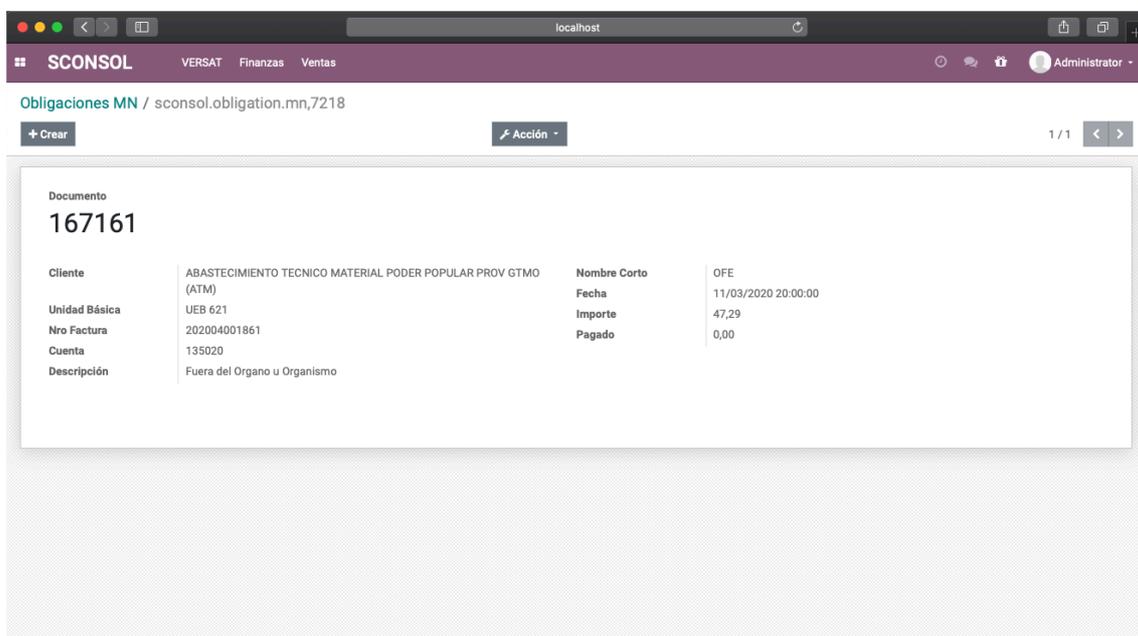
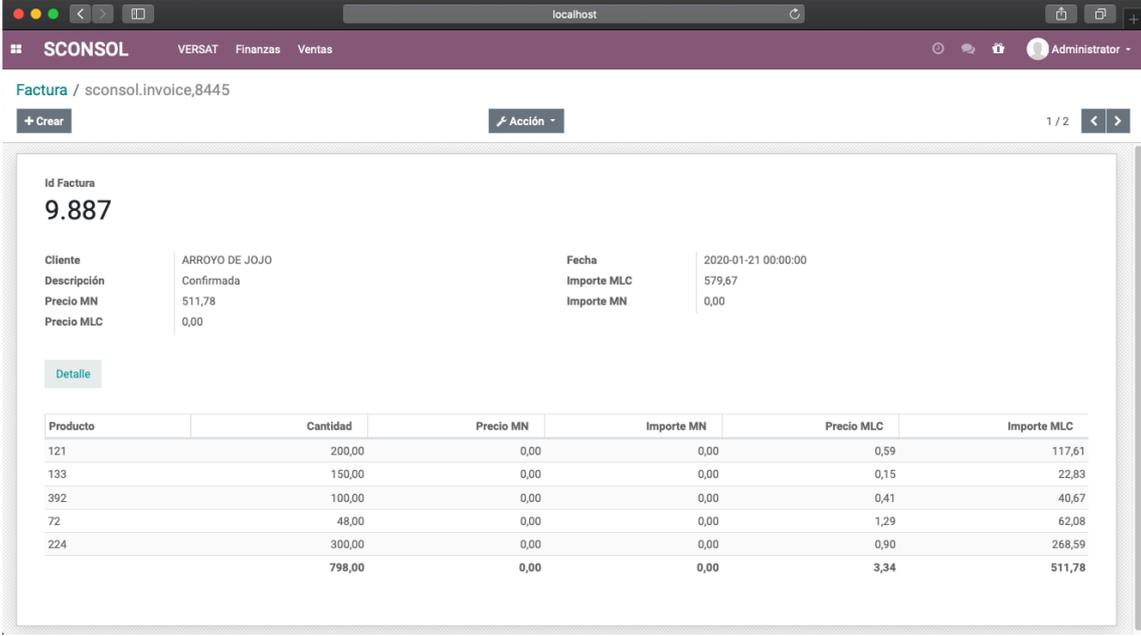


Figura 9. Prototipo de Vista Formulario (Obligaciones MN)



Factura / sconsol.invoice,8445

+ Crear Acción - 1 / 2

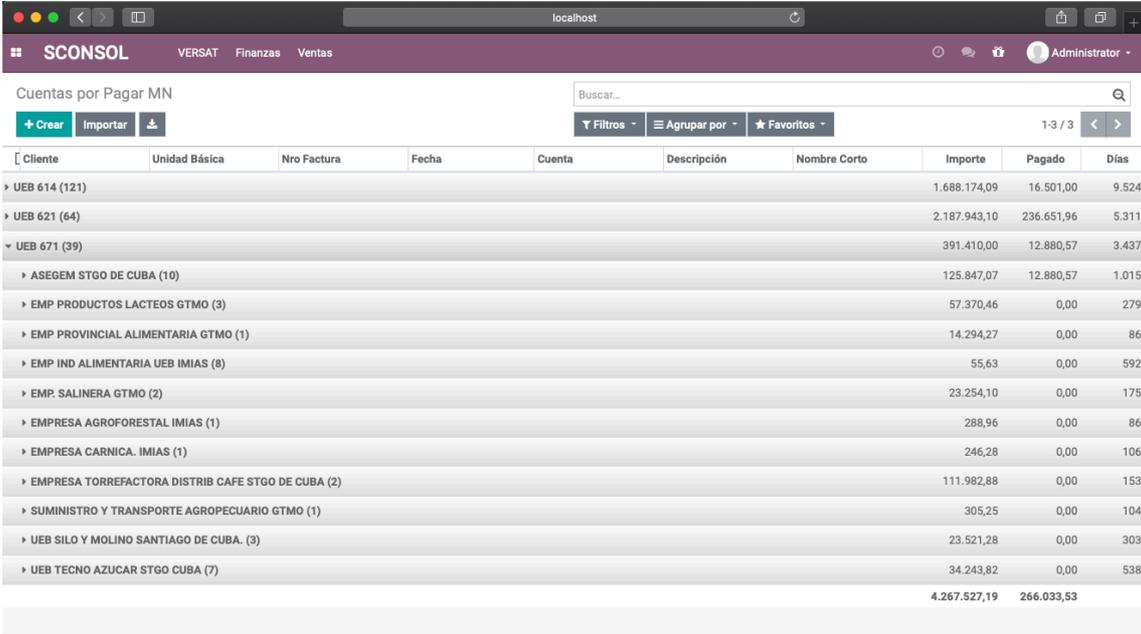
Id Factura
9.887

Ciente	ARROYO DE JOJO	Fecha	2020-01-21 00:00:00
Descripción	Confirmada	Importe MLC	579,67
Precio MN	511,78	Importe MN	0,00
Precio MLC	0,00		

Detalle

Producto	Cantidad	Precio MN	Importe MN	Precio MLC	Importe MLC
121	200,00	0,00	0,00	0,59	117,61
133	150,00	0,00	0,00	0,15	22,83
392	100,00	0,00	0,00	0,41	40,67
72	48,00	0,00	0,00	1,29	62,08
224	300,00	0,00	0,00	0,90	268,59
	798,00	0,00	0,00	3,34	511,78

Figura 10. Prototipo de Vista Detalle (Factura)



Cuentas por Pagar MN

+ Crear Importar Buscar... 1-3 / 3

Filtros - Agrupar por - Favoritos -

Cliente	Unidad Básica	Nro Factura	Fecha	Cuenta	Descripción	Nombre Corto	Importe	Pagado	Días
▶	UEB 614 (121)						1.688.174,09	16.501,00	9.524
▶	UEB 621 (64)						2.187.943,10	236.651,96	5.311
▼	UEB 671 (39)						391.410,00	12.880,57	3.437
▶	ASEGEM STGO DE CUBA (10)						125.847,07	12.880,57	1.015
▶	EMP PRODUCTOS LACTEOS GTMO (3)						57.370,46	0,00	279
▶	EMP PROVINCIAL ALIMENTARIA GTMO (1)						14.294,27	0,00	86
▶	EMP IND ALIMENTARIA UEB IMIAS (8)						55,63	0,00	592
▶	EMP. SALINERA GTMO (2)						23.254,10	0,00	175
▶	EMPRESA AGROFORESTAL IMIAS (1)						288,96	0,00	86
▶	EMPRESA CARNICA. IMIAS (1)						246,28	0,00	106
▶	EMPRESA TORREFACTORA DISTRIB CAFE STGO DE CUBA (2)						111.982,88	0,00	153
▶	SUMINISTRO Y TRANSPORTE AGROPECUARIO GTMO (1)						305,25	0,00	104
▶	UEB SILO Y MOLINO SANTIAGO DE CUBA. (3)						23.521,28	0,00	303
▶	UEB TECNO AZUCAR STGO CUBA (7)						34.243,82	0,00	538
							4.267.527,19	266.033,53	

Figura 11. Prototipo de Vista Lista (Cuentas por Pagar MN)

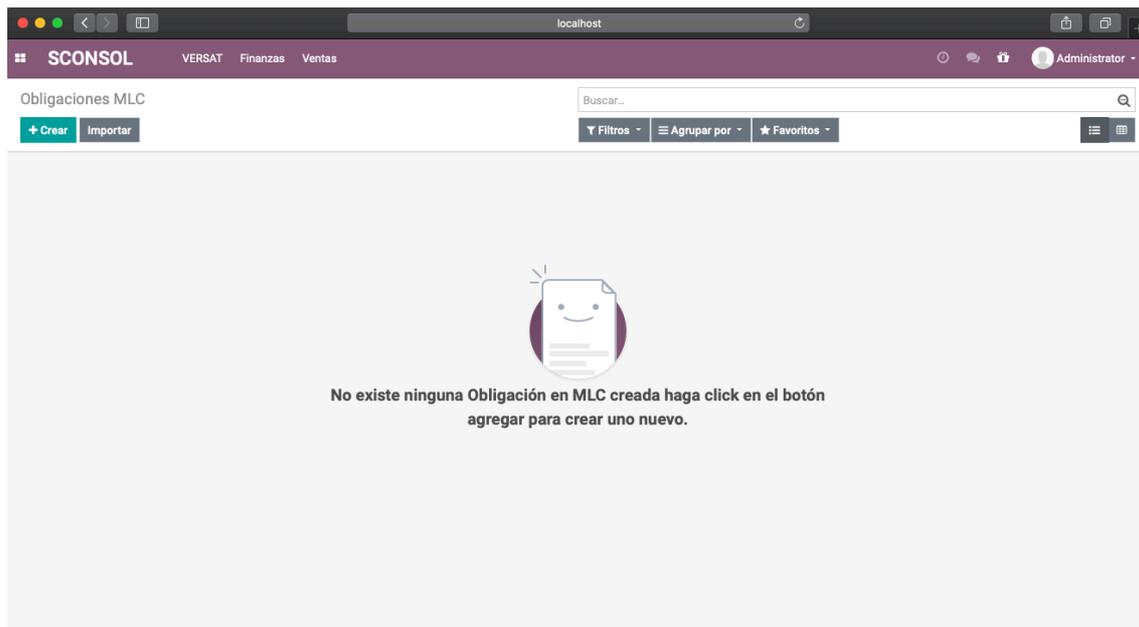


Figura 12. Prototipo de Vista Genérica (Obligaciones en MLC)

Conclusiones parciales del capítulo

En el presente capítulo se han mostrado aspectos del desarrollo de SCONSOL.

Se han expuesto las funcionalidades que lo determinan, características y beneficios. Adoptando como Metodología SCRUM, se definieron los artefactos propuestos por esta, en las fases de planificación y diseño, entre los que se destaca las Historias de Usuarios, las cuales describen como deben cumplirse los requerimientos del sistema desde la perspectiva del usuario y la prioridad de codificación de estos en la fase de codificación.

Se realizó el estudio de factibilidad quedando demostrado la viabilidad del sistema propuesto

Se concluye además que la arquitectura del sistema responde a los requerimientos no funcionales del mismo y permite cumplir con los requisitos funcionales en pos de garantizar la estructura necesaria, robusta y segura que permita la construcción del software y su posterior explotación. Fueron mencionados también aspectos generales en la concepción del diseño de la base de datos propuesta para la Sistema Web y se mostró el diseño de esta a través del modelo entidad relación.

CONCLUSIONES GENERALES

Se caracterizó el Proceso de Gestión de las Cuentas por Cobrar y Pagar que tienen lugar en la Subdirección de Economía y Comercial en la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo, llegando a la conclusión de que el mismo presentaba una serie de insuficiencias y que era necesario su informatización.

Se implementó el Sistema Informático para la Gestión y Consolidación de la Información de las Cuentas por Cobrar y Pagar optimizando el acceso a los usuarios que intervienen en el proceso y disminuyendo los tiempos para la toma de las decisiones contables-financieras de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.

RECOMENDACIONES

Analizando los resultados obtenidos en la investigación y la experiencia adquirida durante el desarrollo de la misma, se recomienda incorporar nuevas funcionalidades al sistema según además solicitudes del usuario.

Referencias Bibliográfica

1. Documentación oficial de Odoo en la Web.
<https://www.odoo.com/documentation>
2. Guido Van Rossum. El Tutorial de Python
3. Raúl González Duque. Python para todos
4. Ken Schwaber y Jeff Sutherland (2016). La Guía Definitiva de Scrum: Las Reglas del Juego
5. Manuel Trigas Gallego. GESTION DE PROYECTOS INFORMÁTICOS: Metodología Scrum.
6. Adriana Peralta – 123357(2003). Metodología SCRUM (Universidad ORT Uruguay).
7. Alberto E. Soler Mateu. Creación de un portal web con Odoo y un servidor de correo electrónico corporativo para el equipo Formula Student UPV (Trabajo de Fin de Grado)
8. Alexander Fayolle and Holger Brunn (2016). Odoo 11 Development Cookbook Second Edition.
9. Daniel Reis. Odoo 10 Development Essentials.
10. Documentación oficial en la web de Python. <https://docs.python.org>
11. Castillo, J. "Requerimientos no funcionales y Arquitectura del software."
12. Aulafacil.com "Lenguajes de Programación: Programación Web."
13. Estrategia Integral de para Desoft 2015-2019
14. Jacobson I, B. G., Rumbaugh J, (2004). El Proceso Unificado de Desarrollo de Software. A. Wesley.
15. John (2007). "Introduction to Client / Server Systems: A Practical Guide for Systems Professionals."

16. Javier Eguíluz Pérez Introducción al HTML
17. Mestras, J. P. (2009). Estructura de las Aplicaciones Orientadas a Objetos: El patrón Modelo-Vista Controlador (MVC).
18. Pressman, R. (2005). "Ingeniería de software. Un enfoque práctico."
19. Thomas Lockhart. Manual del usuario de PostgreSQL.
20. Silberschatz, A., H.F. Korth, and S. Sudarshan, FUNDAMENTOS DE BASES DE DATOS.
21. Marín, R. (2020, abril 11). Los gestores de bases de datos más usados en la actualidad.
22. Revista Digital INESEM. <https://revistadigital.inesem.es/informatica-y-tics/losgestores-de-bases-de-datos-mas-usados-en-la-actualidad>
23. Martínez Castillo, E. J. (2014, agosto 31). Lenguajes de programación del lado servidor.
24. [Blog]. <http://michelletores.mx/lenguajes-de-programacion-del-lado-servidor/>
25. Pila del producto—Scrum Manager BoK. (s. f.). Recuperado 1 de junio de 2020, de https://www.scrummanager.net/bok/index.php?title=Pila_del_producto
26. Pila del sprint—Scrum Manager BoK. (s. f.). Recuperado 1 de junio de 2020, de [https://www.scrummanager.net/bok/index.php?title=Pila del sprint](https://www.scrummanager.net/bok/index.php?title=Pila_del_sprint)
27. Reglas de Negocio – Business Rules. (2011, agosto 20). Tecnologías de la Información y Procesos de Negocios. <https://msaffirio.wordpress.com/2011/08/20/reglas-denegocio-business-rules/>
28. ReQtest. (2018, agosto 17). Test Case Design Techniques Using Test Case Software.

29. ReQtest. <https://reqtest.com/testing-blog/test-case-design-techniques/>
30. Requerimientos Funcionales y No Funcionales – Porqueria. (s. f.). Recuperado 1 de junio de 2020, de <https://ingeneriadesoftwareutmachala.wordpress.com/2017/01/20/requerimientos-funcionales-y-no-funcionales/>
31. Software—Requisitos—Tutorialspoint. (s. f.). Recuperado 1 de junio de 2020, de https://www.tutorialspoint.com/es/software_engineering/software_requirements.html
32. What is Scrum? (2020, junio 2). Scrum.Org. <https://www.scrum.org/resources/what-is-scrum>
33. What is Sprint Planning? (2020, junio 4). <https://www.scrum.org/resources/what-is-sprint-planning>
34. <http://www.scrumsense.com>
35. <http://www.proyectosagiles.org>
36. <http://www.acis.org.co>
37. <http://www.mastersoft.com.ar>
38. <http://assets.scrumfoundation.com>
39. <http://www.gestiondeproyectosit.es>
40. <http://agileee.org>
41. <http://www.pdca.es>
42. <http://msdn.microsoft.com/es-es/library/dd997578/>
43. Cómo instalar Odoo 8 (OpenERP) para Debian.
44. Especificación de requisitos de software, March 2015.
45. Installing Odoo — odoo 8.0 documentation.

46. Odoov8 vs. saas-6 vs. OCB v8: What's the difference?
47. Lenguaje unificado de modelado, June 2015.
48. Sam Williams. [(Free as in Freedom: Richard Stallman's Crusade for Free Software)]. O'Reilly & Associates, 3 2002.
49. MountainGoatSoftware.<http://www.mountangoatsoftware.com/scrum/index.php>
50. Copyright 1990. (1993). IEEE Std, IEEE Software Engineering Standard: Glossary of Software Engineering Terminology. IEEE Computer Society Press. New York: Addison-Wesley.
51. Roger, S. P. (2010). Ingeniería de Software. Un enfoque Práctico. Mexico: EDITORES, S.A.
52. Pérez., J. (2008). Introducción a XHTML. Mexico: Arlequín.

Glosario de Términos

TIC: Tecnologías de la Información y las Comunicaciones

TMS: Software para la gestión de transporte de carga

Framework: Marco de trabajo

Scrum: Metodología ágil para el desarrollo de software

Python: Lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Odo: Marco de trabajo que utiliza Python, el patrón base de su arquitectura es el MVC

MVC (Modelo Vista Controlador): Patrón de arquitectura de diseño este modelo manipula la aplicación que interactúa con la base de datos, decide cómo se van a mostrar los datos en el navegador y cuales se van a ver.

URL: Define la dirección electrónica para la navegación utilizadas por los gestores de navegación

SGBD: Sistema Gestor de Bases de Datos.

UML: Lenguaje Unificado de Modelado.

Anexos

Mapa de Procesos de la Empresa de Aplicaciones Informáticas DESOFT de Guantánamo.

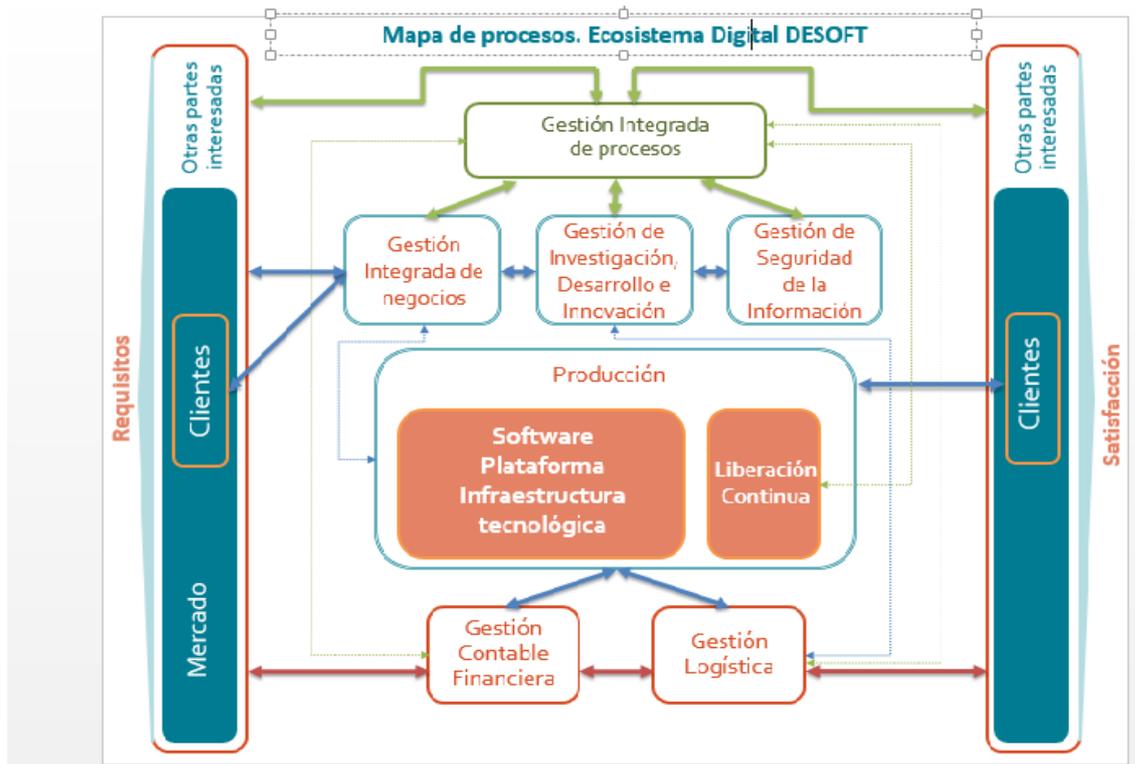


Diagrama del ciclo de vida de SCRUM

