



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Académico Profesional de Ingeniería de Sistemas

**Sistematización del proceso de ventas de una pyme
basada en SOA (caso de estudio DARUNE SAC)**

TESINA

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Ruben Dario FLORES PENADILLO

ASESOR

César Alberto MOLINA NEYRA

Lima, Perú

2016



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

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

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

Referencia bibliográfica

Flores, R. (2016). *Sistematización del proceso de ventas de una pyme basada en SOA (caso de estudio DARUNE SAC)*. [Tesina de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Académico Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
PROGRAMA DE ACTUALIZACIÓN PROFESIONAL 2014-II

Acta de Sustentación de Tesina

Siendo las 18:55 del día 28 de Mayo del año 2016, se reunieron los docentes designados como miembros de Jurado de la Tesina, presidido por el Ing. Winston Ignacio, Ugaz Cachay, el Ing. Joel Fernando, Machado Vicente (Miembro) y el Ing. Cesar Alberto, Molina Neyra (Miembro Asesor) para la sustentación de la Tesina intitulada: "SISTEMATIZACIÓN DEL PROCESO DE VENTAS DE UNA PYME BASADA EN SOA (CASO DE ESTUDIO DARUNE SAC)". Por el Sr. Bach, RUBEN DARIO FLORES PENADILLO; para optar el Título Profesional de Ingeniero de Sistemas.

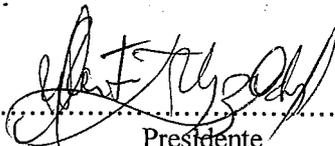
Acto seguido de la exposición de la Tesina, el Presidente invitó al graduando a dar respuesta a las preguntas establecidas por los Miembros de Jurado.

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

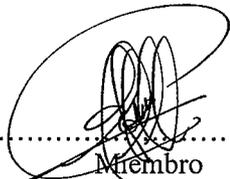
Finalmente habiéndose efectuado la calificación correspondiente por los miembros de Jurado, el graduando obtuvo la nota de 18 (En letras). Dieciocho

A continuación el Presidente del Jurado el Ing. Winston Ignacio, Ugaz Cachay declara al graduando **Ingeniero de Sistemas**.

Siendo las 19:55 horas, se levantó la sesión.


.....
Presidente

Ing. Winston Ignacio, Ugaz Cachay


.....
Miembro

Ing. Joel Fernando, Machado Vicente


.....
Miembro Asesor

Ing. Cesar Alberto, Molina Neyra

FICHA CATALOGRÁFICA

Flores Penadillo, Ruben Dario

SISTEMATIZACIÓN DEL PROCESO DE VENTAS DE UNA PYME,
BASADA EN SOA (CASO DE ESTUDIO DARUNE SAC)

Tecnologías de la Información y comunicación / Ingeniería de Software
(Lima, Perú 2016)

Tesina, Facultad de Ingeniería de Sistemas, Pregrado, Universidad
Nacional Mayor De San Marcos

Formato 28 x 20 cm Paginas #

DEDICATORIA

Dedico este trabajo a mis padres y hermanos que me apoyaron todo el tiempo con sus buenos ánimos.

A mis amigos, que siempre compartieron experiencias conmigo.

AGRADECIMIENTOS

Al Prof. Cesar Molina Neyra, asesor de este trabajo de investigación, quien me orientó en todo momento durante el desarrollo y la ejecución del mismo.

Al Gerente General de la empresa DARUNE SAC, Sr. Epifanio Penadillo Valentin, por la confianza brindada hacia mi persona para darme acceso a información confidencial, la cual me dio la oportunidad de desarrollar el presente trabajo de investigación.

A los docentes, compañeros, amigos y a todas aquellas persona que directa o indirectamente hicieron posible la realización de este estudio.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMÁTICA
ESCUELA ACADEMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS
SISTEMATIZACIÓN DEL PROCESO DE VENTAS DE UNA PYME
BASADA EN SOA (CASO DE ESTUDIO DARUNE SAC)

Autor: FLORES PENADILLO, Ruben Dario
Asesor: MOLINA NEYRA, CESAR
Título: Tesina, para optar el Título Profesional de Ingeniero de Sistemas
Fecha: Enero 2016

RESUMEN

Esta tesina ha sido realizada con el fin de sistematizar el Proceso de Ventas para una PYME, utilizando para ello una Arquitectura Orientada a Servicios (SOA), esto como solución al problema de carencia de control y organización de la información, además de obtener una respuesta rápida a nuevas necesidades de negocio. Se realizará un análisis del negocio en la empresa DARUNE SAC, nuestro caso de estudio, en la cual aplicaremos la solución planteada.

Palabra claves: Pyme, arquitectura, servicio, solución.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERIA DE SISTEMAS E INFORMÁTICA

ESCUELA ACADEMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

**SYSTEMATIZATION SALES PROCESS IN A PYME BASED SOA (STUDY
CASE DARUNE SAC)**

Autor: FLORES PENADILLO, Ruben Dario

Asesor: MOLINA NEYRA, CESAR

Título: Tesina, para optar el Título Profesional de Ingeniero de Sistemas

Fecha: January 2016

ABSTRACT

This thesis has been carried out in order to systematize the process of Sales for a PYME, using Service Oriented Architecture (SOA), this solution to the problem of lack of control and organization of information, plus get a quick response to new business requirements. An analysis of the business in the company DARUNE SAC, our case study, in which we will apply the proposed solution.

Key words: Pyme, architecture, service, solution.

ÍNDICE

ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIV
INTRODUCCIÓN	XVI
Capítulo I: PLANTEAMIENTO METODOLÓGICO	1
1.1. Antecedentes del problema	1
1.2. Definición o formulación del problema	4
1.3. Objetivos	4
1.3.1. Objetivos Generales	4
1.3.2. Objetivos Específicos	5
1.4. Justificación.....	5
1.5. Alcance.....	6
1.6. Organización de la tesina	6
Capítulo II: MARCO TEÓRICO	7
2.1. Diseño y Arquitectura de Software	7
2.1.1. Introducción	7
2.1.2. Diseño de Software	7
2.1.3. Arquitectura de Software.....	9
2.1.4. Importancia de la Arquitectura de Software en el Diseño.....	13
2.2. Arquitectura Orientada a Servicios (SOA).....	16
2.2.1. Introducción	16
2.2.2. Definición.....	20
2.2.3. Mitos y Realidades de SOA	28
2.2.4. Características de SOA.....	29

2.2.5.	Beneficios de SOA	38
2.2.6.	Ejemplo de una Arquitectura Genérica alineada a SOA	43
2.3.	Metodología RUP.....	47
2.3.1.	Características Principales del RUP	49
2.3.2.	Dimensiones del RUP	53
2.3.3.	Fases del RUP	55
2.3.4.	Iteraciones del RUP.....	66
2.3.5.	Estructura Estática del RUP	67
2.3.6.	Disciplinas y Workflows	71
2.4.	Proceso de ventas de DARUNE SAC	80
2.4.1.	Procesos Estratégicos	80
2.4.2.	Procesos Principales	81
2.4.3.	Procesos de Apoyo o Soporte.....	81
2.4.4.	Descripción del proceso de ventas de DARUNE SAC	82
Capítulo III. ESTADO DEL ARTE METODOLÓGICO		89
3.1.	Metodologías para el desarrollo de la solución	89
3.1.1.	Análisis, Diseño e Implementación de un sistema de información para una tienda de ropa con enfoque al segmento juvenil	89
3.1.2.	Análisis, Diseño e Implementación de un Sistema de Comercio Electrónico integrado con una Aplicación móvil para la reserva y venta de pasajes de una empresa de transporte interprovincial	89
113		
3.2.	Comparación de alternativas de solución	121
Capítulo IV. DESARROLLO DE LA SOLUCIÓN.....		123
4.1.	Generalidades	123
4.2.	Modelado del Negocio	123
4.2.1.	Misión y Visión.....	124

4.2.2.	Mapa de Procesos	124
4.2.3.	Modelo de casos de uso del negocio	125
4.2.4.	Modelo de análisis de negocio	129
4.3.	Captura de Requerimientos	132
4.3.1.	Requerimientos funcionales	132
4.3.2.	Requerimientos no funcionales	136
4.3.3.	Actores del sistema.....	138
4.3.4.	Diagrama de casos de uso	139
4.3.5.	Matriz de trazabilidad.....	140
4.3.6.	Especificación de casos de uso.....	140
4.4.	Análisis y Diseño del Sistema	143
4.4.1.	Modelo de dominio	147
4.4.2.	Diseño de Base de Datos	148
4.4.3.	Diseño de la Arquitectura.....	149
4.4.4.	Diagrama de componentes DARUNE SAC	150
4.5.	Implementación del Sistema	151
4.5.1.	Pantallas del sistema web DARSYS	151
4.6.	Pruebas del Sistema.....	163
4.6.1.	Plan de pruebas.....	163
4.6.2.	Resultados de pruebas	165
4.7.	Análisis costo beneficio	168
4.7.1.	Inversión Inicial.....	168
4.7.2.	Gastos concurrentes u operativos	170
4.7.3.	Beneficios.....	170
4.7.4.	Flujo de caja proyectado.....	171

4.7.5. Valor Actual Neto (VAN)	171
4.7.6. Tiempo de retorno de inversión.....	172
Capítulo V. CONCLUSIONES Y RECOMENDACIONES	173
5.1. Conclusiones	173
5.2. Recomendaciones.....	175
REFERENCIAS BIBLIOGRÁFICAS.....	177

ÍNDICE DE FIGURAS

Figura 1: Stock y flujo de empresas por años, 2010-2014. (INEI, 2015)	1
Figura 2: Stock de empresas por año, 2010-2014. (INEI, 2015)	1
Figura 3: Empresas según segmento empresarial 2013-2014. (INEI, 2015)	2
Figura 4: Peru: Empresas, según actividad económica, 2013-2014. (INEI, 2015)	2
Figura 5: Arquitectura de Software de TopLine System. (TopLine Systems, 2010)	13
Figura 6: Dos aspectos claves de las organizaciones: proceso de negocio y tecnología. (Erl, 2005)	21
Figura 7: Capa de servicios entre los procesos del Negocio y las tecnologías. (Erl, 2005)	23
Figura 8: Evolución de Arquitecturas de Software hasta SOA. (Endrei, 2007)	25
Figura 9: Bajo Acoplamiento del Servicio. (Pulier y Tylor, 2006)	31
Figura 10: Sistemas distribuidos con diferentes propietarios. (Josuttis, 2007)	36
Figura 11: Agilidad en el negocio articulada por SOA. (Accenturre, 2008)	42
Figura 12: Arquitectura de 3 capas alineada a SOA. (Bocchio, 2014)	43
Figura 13: Modelo Iterativo. (Rational, 2001)	54
Figura 14: Las fases e hitos importantes en el proceso. (Rational, 2001)	56
Figura 15: Hito - Objetivo del ciclo de vida. (Rational, 2001).....	58
Figura 16: Hito - Arquitectura del ciclo de vida. (Rational, 2001)	60
Figura 17: Hito - Capacidad operativa inicial. (Rational, 2001)	63
Figura 18: Hito - Release del sistema. (Rational, 2001)	65
Figura 19: Proceso de ventas de DARUNE SAC - AS IS. ELaboración Propia.	82
Figura 20: Venta en Tienda. Rodrigez, J. (2013)	97
Figura 21: Venta en Tienda – Catálogo en Línea. Rodrigez, J. (2013)	98
Figura 22: Catalogo de Actores. Rodrigez, J. (2013)	99
Figura 23: Paquetes de Ventas. Rodrigez, J. (2013)	100

Figura 24: Diagrama de Clases del Módulo de Ventas. Rodrigez, J. (2013)	103
Figura 25: Representación de la Arquitectura. Rodrigez, J. (2013)	104
Figura 26: Arquitectura web en 3 capas. Rodrigez, J. (2013)	105
Figura 27: Pantalla principal del Sistema de Información TSJ. Rodrigez, J. (2013)	108
Figura 28: Diagrama de Base de Datos del Sistema. Rodrigez, J. (2013)	109
Figura 29: Cuadro comparativo de algunas características de los lenguajes PHP, Ruby y Python. Rodrigez, J. (2013).....	110
Figura 30: Diagrama de Arquitectura. Becerra, C. (2013).....	117
Figura 31: Modelo físico de datos. Becerra, C. (2013)	119
Figura 32: Mapa de Procesos de DARUNE SAC. Elaboración Propia	125
Figura 33: Diagrama de casos de uso del negocio. Elaboración propia.	128
Figura 34: Diagrama de clases del CUN Realizar venta. Elaboración propia.	130
Figura 35: Diagrama de actividades del proceso de ventas DARUNE SAC. Elaboración propia.....	131
Figura 36: Diagrama de casos de uso del proceso Realizar venta. Elaboración propia.	139
Figura 37: Modelo de dominio: Elaboración propia.	147
Figura 38: Modelo Físico de la Base de Datos. Elaboración Propia.....	148
Figura 39: Arquitectura de DARUNE SAC orientada a servicios. Elaboración propia.	149
Figura 40: Diagrama de componentes Sistema Web DARUNE SAC. Elaboración propia.	150
Figura 41: Pantalla de ingreso al Sistema DARSYS. Elaboración propia.	151
Figura 42: Pantalla principal del operador de ventas. Elaboración propia.	152
Figura 43: Pantalla de generación de preventas. Elaboración propia.....	153
Figura 44: Pantalla búsqueda de productos. Elaboración propia.	154
Figura 45: Pantalla principal del operador de caja. Elaboración propia.....	155
Figura 46: Pantalla de la bandeja del operador de caja. Elaboración propia.	156
Figura 47: Pantalla de atención a una preventa. Elaboración propia.....	157

Figura 48: Pantalla de atención a una preventa. Elaboración propia. 158

Figura 49: Pantalla de pago en efectivo. Elaboración propia. 159

Figura 50: Pantalla principal del operador de despacho. Elaboración propia 160

Figura 51: Pantalla de la bandeja del operador de despacho. Elaboración propia. 161

Figura 52: Pantalla de atención de un despacho. Elaboración propia. 162

ÍNDICE DE TABLAS

Tabla 1: Arquitectura de 3 capas alineada a SOA. Bocchio (2014)	47
Tabla 2: Ficha del Proceso de Ventas de DARUNE SAC. Elaboración Propia.....	84
Tabla 3: Fases del RUP Aplicadas al Proyecto. Rodrigez, J. (2013)	92
Tabla 4: Artefactos del RUP aplicados al Proyecto. Rodrigez, J. (2013)	93
Tabla 5: Artefactos del RUP aplicados al Proyecto. Rodrigez, J. (2013)	94
Tabla 6: Requisitos del Módulo de Ventas. Rodrigez, J. (2013)	95
Tabla 7: Tecnologías para la construcción. Rodrigez, J. (2013)	112
Tabla 8: Herramientas para la construcción. Rodrigez, J. (2013)	112
Tabla 9: Cuadro de resumen para metodologías. Becerra, C. (2013).	116
Tabla 10: Comparación de alternativas de Arquitecturas para nuestra solución. Elaboración propia	122
Tabla 11: Actores de negocio. Elaboración propia.	126
Tabla 12: Casos de uso del negocio. Elaboración propia.	127
Tabla 13: Trabajadores del negocio. Elaboración propia.....	129
Tabla 14: Requerimientos funcionales del proceso de ventas. Elaboración propia.	136
Tabla 15: Requerimientos no funcionales. Elaboración propia.	137
Tabla 16: Actores del sistema. Elaboración propia.	138
Tabla 17: Matriz de trazabilidad	140
Tabla 18: Especificación de caso de uso Consultar producto. Elaboración propia.	141
Tabla 19: Especificación de caso de uso Generar ticket. Elaboración propia.....	142
Tabla 20: Diagrama de análisis del proceso de ventas DARUNE SAC - ASIS. Elaboración propia.	143
Tabla 21: Costos promedio por rol. Elaboración propia.	144
Tabla 22: Costo del proceso de ventas DARUNE SAC. Elaboración propia.	145
Tabla 23: Diagrama de análisis del proceso de ventas DARUNE SAC - TOBE. Elaboración propia.	166

Tabla 24: Costo del proceso de ventas sistematizado DARUNE SAC. Elaboración propia.	167
Tabla 25: Resumen de análisis de pruebas del proceso. Elaboración propia.	168
Tabla 26: Costos de servicios. Elaboración propia.	168
Tabla 27: Costos de materiales. Elaboración propia.	169
Tabla 28: Costos de personal. Elaboración propia.	169
Tabla 29: Costos de software. Elaboración propia.	169
Tabla 30: Costos de hardware. Elaboración propia.	170
Tabla 31: Costos concurrentes u operativos. Elaboración propia.	170
Tabla 32: Beneficios implementando la sistematización. Elaboración propia.	171
Tabla 33: Flujo de caja proyectado en el periodo de 5 semestres. Elaboración propia.	171
Tabla 34: Flujo de caja acumulado en un periodo de 5 semestres. Elaboración propia.	172

INTRODUCCIÓN

El presente trabajo de investigación tiene como objetivo principal sistematizar el proceso de venta de una PYME, para lo cual hemos tomado como caso de estudio a la empresa DARUNE SAC, dicha empresa se desarrolla en el rubro comercial, específicamente en la compra y venta de artículos para bebés, como por ejemplo pañales, biberones, lácteos, colonias, etc. al por mayor y menor. Esta sistematización será realizada bajo una arquitectura orientada a servicios (SOA), debido a que se busca optimizar el proceso actual, además de flexibilizar y reducir esfuerzos de implementaciones como soluciones a las necesidades del negocio a futuro.

Dentro del proceso de ventas de la PYME DARUNE SAC, la información acerca de los productos y clientes es tratada de manera manual, por lo que el proceso está constantemente expuesto a errores a la hora del registro y consulta de información.

La sistematización del proceso de ventas de la empresa DARUNE SAC adquiere relevancia debido a que con esto se busca minimizar la utilización de recursos que intervienen o son parte de dicho proceso, además de no solo cumplir rígidamente con las necesidades actuales del negocio, sino también la de estar preparado para cubrir oportunidades futuras de crecimiento y desarrollo de la organización, motivados por los requerimientos cambiantes del negocio, como por ejemplo la atención en línea a una solicitud de crédito o a una solicitud de reportes de compras por parte del cliente.

El tener un proceso óptimo de ventas conllevará a que los colaboradores y clientes de la organización tengan una herramienta eficaz y eficiente que cubra sus necesidades, además de permitir a la gerencia una toma de decisiones oportuna debido al acceso a la información en tiempo real.

Capítulo I: PLANTEAMIENTO METODOLÓGICO

1.1. Antecedentes del problema

El Directorio Central de Empresas y Establecimientos (DCEE) elaborado por el Instituto Nacional de Estadística e Informática (INEI) registró al 31 de diciembre de 2014, un millón 883 mil 531 unidades económicas, mayor en 5.9% respecto a similar periodo del año anterior.

Concepto	2010	2011	2012	2013	2014	Var. % 2014/13
Stock al inicio del periodo	1 270 414	1 374 278	1 478 220	1 634 504	1 778 377	8,8
Altas ^{1/}	223 027	252 559	311 669	286 311	262 973	-8,2
Bajas ^{2/}	-101 088	-120 066	-144 508	-153 514	-163 729	6,7
Otros Ingresos y salidas ^{3/}	18 075	28 551	-10 877	11 076	5 910	
Stock al final del periodo	1 374 278	1 478 220	1 634 504	1 778 377	1 883 531	5,9
Tasa de altas (%) ^{4/}	16,2	17,1	19,1	16,1	14,0	
Tasa de bajas (%) ^{5/}	7,4	8,1	8,8	8,6	8,7	

Figura 1: Stock y flujo de empresas por años, 2010-2014. (INEI, 2015)

En la siguiente figura se puede notar gráficamente el incremento en el stock de las empresas en el Perú durante el periodo 2010-2014.

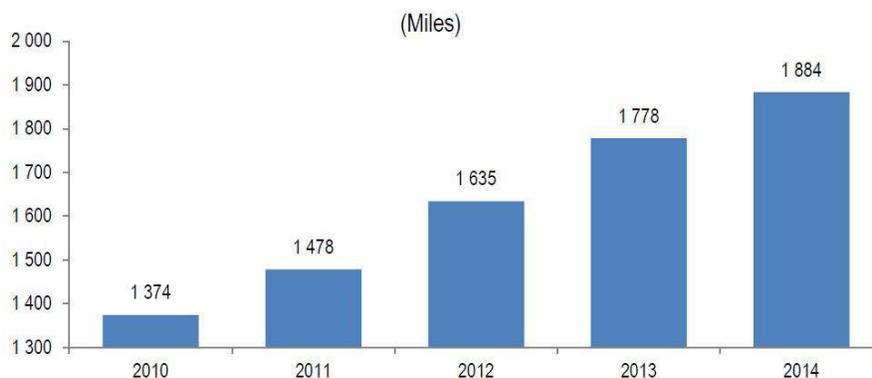


Figura 2: Stock de empresas por año, 2010-2014. (INEI, 2015)

Como se puede apreciar en la figura 2, el número de empresas en el Perú ha ido aumentando en los últimos 5 años, para un mayor detalle a continuación se muestra un cuadro que divide a las empresas en el Perú según el segmento empresarial.

Segmento empresarial	2013	2014	Estructura porcentual 2014	Var % 2014/13
Total	1 778 377	1 883 531	100,00	5,91
Micro empresa	1 689 366	1 787 857	94,92	5,83
Pequeña empresa	71 442	77 503	4,11	8,48
Mediana y gran empresa	11 195	11 380	0,60	1,65
Administración pública	6 374	6 791	0,36	6,54

Figura 3: Empresas según segmento empresarial 2013-2014. (INEI, 2015)

Como podemos notar en el gráfico anterior las pequeñas empresas presentan un incremento porcentual mayor respecto de los demás segmentos empresariales.

Adicionalmente, en el año 2014 la actividad económica con mayor número de empresas ha sido la del Comercio y reparación de vehículos automotores y motocicletas.

Actividad económica	2013	2014	Estructura porcentual 2014	Var % 2014/13
Total	1 778 377	1 883 531	100,00	5,91
Agricultura, ganadería, silvicultura y pesca	35 723	36 435	1,93	1,99
Explotación de minas y canteras	9 897	15 660	0,83	58,23
Industrias manufactureras	157 001	161 887	8,59	3,11
Electricidad, gas y agua	4 237	4 399	0,23	3,82
Construcción	48 745	49 747	2,64	2,06
Comercio y reparación de vehículos automotores y motocicletas	824 742	879 475	46,69	6,64
Transporte y almacenamiento	86 283	90 687	4,81	5,10
Actividades de alojamiento y de servicio de comidas	142 579	155 174	8,26	8,83
Información y comunicaciones	52 839	53 987	2,87	2,17
Servicios profesionales, técnicos y de apoyo empresarial	179 006	188 732	10,02	5,43
Otros servicios 1/	237 325	247 348	13,13	4,22

Figura 4: Perú: Empresas, según actividad económica, 2013-2014. (INEI, 2015)

En la actualidad, el incremento de las PYME en el sector comercial ha generado una pugna entre dichas unidades económicas, esto por posicionarse en un lugar privilegiado dentro de un mercado altamente competitivo, cambiante y lleno de desafíos; sin embargo la mayoría de empresas de este tipo limitan sus oportunidades y no logran crear ventajas competitivas, como por ejemplo procesar de una manera más rápida los pedidos de los clientes o efectuar una venta en un tiempo menor al de la competencia, esto por diversos factores, uno de los más importantes y en la mayoría de casos, es la falta de organización sobre la información interna (entre las áreas de la empresa) y externa (clientes y proveedores).

Para las organizaciones es de suma importancia que el flujo de la información sea cada vez más rápido, ya que así será más fácil identificar los problemas en el menor tiempo posible y con ello se podrán tomar decisiones oportunas para dar con las soluciones.

El proceso de ventas de la empresa DARUNE SAC actualmente presenta problemas en:

- La verificación del stock de los productos.
- Emisión del precio correcto de los productos.
- El tiempo en realizar la validación de las proformas y los comprobantes de pagos.
- El tiempo de realizar las proformas y los comprobantes de pagos.
- La calidad de la información registrada en las proformas y los comprobantes de pagos.

Por otro lado las empresas que hoy en día cuentan con procesos sistematizados buscan abarcar nuevas tendencias tecnológicas para cubrir así en mayor porcentaje la demanda e incrementar su mercado.

Esta cobertura de nuevas tecnologías habitualmente es muy costosa debido a que la sistematización que actualmente poseen se enfocó en la aplicación final para las necesidades del negocio en ese momento, por lo que no es flexible a nuevas necesidades del negocio.

Ante esto, se plantea la sistematización del Proceso de Ventas para una PYME (caso DARUNE SAC), enfocado a una Arquitectura Orientada a Servicios (SOA), la cual mejorará el proceso mencionado, y se tendrá la ventaja de ser escalable e interoperable.

1.2. Definición o formulación del problema

¿Mejorará el Proceso de Ventas de la PYME DARUNE SAC con la Sistematización basada en una Arquitectura Orientada a Servicios (SOA)?

1.3. Objetivos

1.3.1. Objetivos Generales

Sistematizar el Proceso de Ventas de una PYME basado en una Arquitectura Orientada a Servicios (SOA). (caso de estudio DARUNE SAC)

1.3.2. Objetivos Específicos

- Realizar el modelado del negocio.
- Realizar la captura de los requerimientos funcionales y no funcionales.
- Definir la arquitectura del sistema web tal que permita la interoperabilidad con la aplicación móvil.
- Implementar un sistema web que soporte el proceso de ventas definido.
- Implementar una aplicación móvil con una interfaz que le permita al usuario conectarse al sistema web y solicitar un crédito.
- Realizar el análisis costo beneficio para el presente proyecto.

1.4. Justificación

Lograr que el Proceso de ventas de la PYME DARUNE SAC se sistematice, permitirá un mayor control, acceso rápido y en tiempo real, a la información requerida.

Si no se sistematiza el proceso de ventas de la PYME DARUNE SAC, no se reducirán los costos operativos como por ejemplo el sueldo al personal que realiza actividades manualmente, además de dejar ir nuevas oportunidades de negocio como por ejemplo una mayor inversión - ganancia sobre un producto determinado debido a su alta demanda, esto en consecuencia de falta de control, organización y acceso rápido a la información.

Al no sistematizar el proceso de ventas, para nuestro caso de estudio, la empresa DARUNE SAC, la información no será manejada en tiempo real, por lo que no se podrá tomar decisiones oportunas.

Si no se implementa la solución planteada, orientada a servicios, la adaptación a cambios y/o mejoras futuras debido a las nuevas necesidades del negocio como por ejemplo vender en línea un producto, será de un costo muy alto.

1.5. Alcance

La solución propuesta abarca la sistematización del proceso de ventas de una PYME en el sector comercial peruano, dicha sistematización se aplicará a nuestro caso de estudio, la empresa DARUNE SAC.

1.6. Organización de la tesina

A continuación se describe la organización de este trabajo de investigación, el presente es dividido en 5 capítulos, el capítulo I trata del Planteamiento Metodológico, el capítulo II, el Marco Teórico, en el capítulo III desarrollamos el Estado del Arte Metodológico, en el capítulo IV, se describe el Desarrollo de la Solución y por último en el capítulo V las Conclusiones y Recomendaciones. Al final de esta tesina podrá encontrar las referencias bibliográficas.

Capítulo II: MARCO TEÓRICO

2.1. Diseño y Arquitectura de Software

2.1.1. Introducción

Se presenta el estado actual del conocimiento en Diseño y Arquitectura de Software, que constituye la base para el trabajo de investigación realizado. Si bien la metodología propuesta está enfocada a desarrollos basados en un solo estilo de Arquitectura de Software orientado a Servicios o Service Oriented Architecture (SOA), el trabajo realizado y el propio enfoque surge a partir de los conceptos, estándares y procesos de Diseño y Arquitectura de Software que se han venido desarrollando desde los inicios de la Ingeniería de Software, y que hoy en día son ampliamente aceptados.

2.1.2. Diseño de Software

En el trabajo de investigación realizado por Delgado (2007, p.50) el cual hace referencia a que según el estándar 610.12 de la IEEE (1990) el diseño de software se define como “el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o componente” así como “el resultado de ese proceso”.

Asimismo hace referencia a SWEBOK (2004) que establece que “visto como un proceso, el diseño de software es la actividad del ciclo de vida de la Ingeniería de Software en la cual los requerimientos del software son analizados para producir una descripción de la estructura interna del software que sirve como base para su construcción. Más precisamente, un diseño de software (el resultado) debe describir la Arquitectura de Software, esto es, como el software se descompone y organiza en componentes e interfaces entre estos componentes. También debe describir los componentes a un nivel de detalle que permitan su construcción. El diseño de software juega un rol importante en el desarrollo de software: permite la producción de varios modelos que forman una especie de plano de la solución a ser implementada. Se pueden analizar y evaluar estos modelos para determinar si permitirán cumplir los requerimientos. Se pueden también examinar y evaluar varias soluciones alternativas y concesiones a realizar. Finalmente, se pueden utilizar los modelos resultantes para planificar las actividades subsecuentes de desarrollo, además de utilizarlos como entrada y punto de partida para la construcción y verificación”.

Delgado (2007, p.50) en su trabajo de investigación cita a Sommerville (2002) quien define que “Un diseño de software es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y, algunas veces, los algoritmos detallados. Los diseñadores no obtienen inmediatamente un diseño detallado, sino que lo desarrollan de manera iterativa a través de diversas versiones diferentes. El proceso de diseño incluye agregar formalidad y detalle durante el desarrollo del diseño, y regresar a los diseños anteriores para corregirlos. ... incluye el desarrollo de varios modelos del sistema con diferentes niveles de abstracción. En cuanto se

descompone el sistema, se descubren los errores y omisiones de las etapas previas. Esta retroalimentación permite mejorar los modelos de los diseños previos. Durante todo el proceso de diseño se detalla cada vez más la especificación. El resultado final del proceso son especificaciones precisas de los algoritmos y estructuras de datos a implementarse.”

De las definiciones anteriores, podemos observar que tanto el diseño como solución a los requerimientos planteados para un sistema, como el proceso de diseño para la obtención de dicha solución, son elementos complejos donde deben interactuar distintos involucrados con distintos intereses, desde el cliente, los analistas de requerimientos, los propios diseñadores hasta los implementadores y verificadores del software. El proceso de diseño además no se puede realizar en forma lineal y abarcando todos los aspectos planteados de una única vez, sino que debe ser atacado en forma progresiva para ir ganando conocimiento del problema y así evaluar distintas opciones que provean la mejor solución para el mismo, modificando y mejorando los modelos del sistema realizados.

2.1.3. Arquitectura de Software

No existe una única definición de Arquitectura de Software, pero el concepto es compartido en términos generales por la mayoría de las definiciones.

Según la publicación en la revista SG BUZZ (2010), el concepto de arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos

propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo. Al igual que en la ingeniería civil, las decisiones críticas relativas al diseño general de un sistema de software complejo deben de hacerse desde un principio. El no crear este diseño desde etapas tempranas del desarrollo puede limitar severamente el que el producto final satisfaga las necesidades de los clientes. Además, el costo de las correcciones relacionadas con problemas en la arquitectura es muy elevado. Es así que la arquitectura de software juega un papel fundamental dentro del desarrollo.

Delgado (2007, p.51) en su trabajo de investigación cita a Clements (2002), el cual se plantea que hace ya más de tres décadas, en 1974 Parnas observó que el software consiste de varias estructuras que definió como descripciones parciales que muestran el sistema como una colección de partes y relaciones entre éstas, identificando la estructura de módulos como unidades de trabajo, la estructura “uses” entre módulos como dependencia de su corréctitud y la estructura de procesos como proveedores de trabajo computacional. Ya en la década de los noventa, en 1992 Perry y Wolf reconocen que “Análogamente a la arquitectura de edificios, se propone el siguiente modelo de Arquitectura de Software = {Elementos, Formas, Razonamientos/Restricciones}, esto es, una Arquitectura de Software es un conjunto de elementos arquitectónicos (o si se prefiere, de diseño) que tienen una forma particular.”

La definición de Arquitectura de Software dada por Garlan (1996), el cual es citado en el trabajo de investigación de Delgado (2007, p.51) proviene de un artículo publicado previamente en 1993 y establece que “La Arquitectura de Software va más allá de los algoritmos y las estructuras de datos de computación, diseñar y especificar la estructura global

del sistema emerge como un nuevo tipo de problema. Aspectos estructurales incluyen organización y estructura de control global, protocolos de comunicación, sincronización, y acceso a datos; asignación de funcionalidad a elementos de diseño; distribución física; composición de elementos de diseño; escalabilidad y performance; y selección entre alternativas de diseño.”

Delgado (2007, p.51) cita en su trabajo de investigación a Bass, Clements, Kazman (2003, p.560) que retoman la definición dada en la primer edición del libro en 1998 “La Arquitectura de Software de un programa o sistema de computación es la estructura o estructuras del sistema, que comprende elementos de software, la propiedades visibles externamente de esos elementos, y las relaciones entre estos. Por “propiedades visibles externamente” nos referimos a las asunciones que otros elementos pueden hacer de un elemento, como ser los servicios que provee, características de performance, manejo de faltas, uso de recursos compartidos, entre otros.”, sustituyendo lo que antes se mencionaba como “componente” por “elemento”, lo que permite mayor amplitud en la definición.

La definición dada por Jacobson, Booch, Rumbaugh (1999, p.463) citados en el trabajo de investigación de Delgado (2007, p.51) establece que “Una Arquitectura es el conjunto de decisiones significativas sobre la organización de un sistema de software, la selección de elementos estructurales y sus interfaces de los cuales se compone el sistema, conjuntamente con su comportamiento el cual se especifica en las colaboraciones entre estos elementos, la composición de estos elementos estructurales y de comportamiento en subsistemas de

progresivamente mayor tamaño, y el estilo de arquitectura que guía esta organización, esos elementos y sus interfaces, sus colaboraciones y su composición.”

De las definiciones anteriores se puede extraer como factor común el hecho de que la Arquitectura de Software representa un primer nivel de descomposición que muestra como se organiza el sistema en términos de elementos de software, sus interfaces y las relaciones entre éstos. Este hecho hace que la definición de la Arquitectura de Software de una aplicación revista gran importancia como parte fundamental del Diseño de Software.

Para mayor comprensión de lo mencionado en las definiciones a continuación se muestra en la figura 2, un ejemplo de Arquitectura de Software de 3 capas.

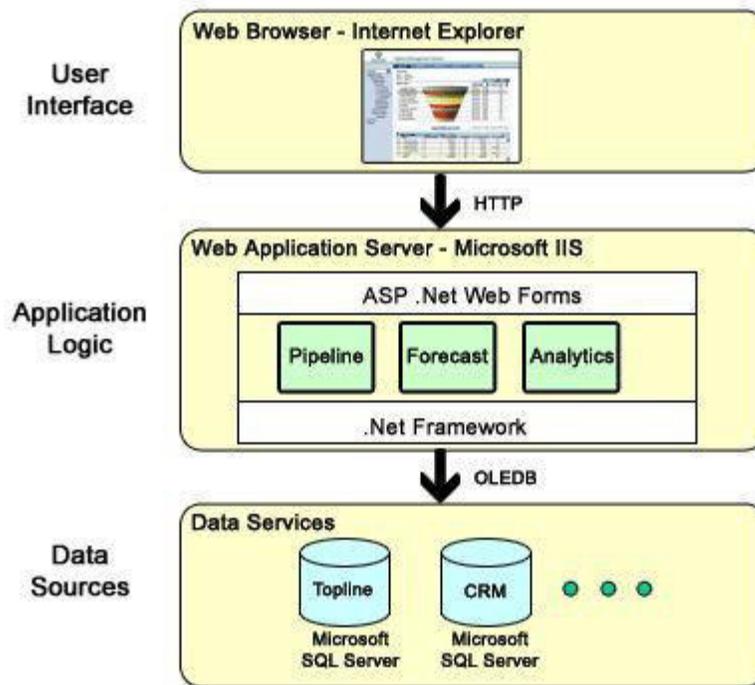


Figura 5: Arquitectura de Software de TopLine System. (TopLine Systems, 2010)

2.1.4. Importancia de la Arquitectura de Software en el Diseño

En el trabajo de investigación de Delgado (2007, p.51) cita a Clements, Bass, Garlan (2002), quienes plantean que “sin una arquitectura apropiada para el problema que se está resolviendo y también en el caso en que se tenga definida la arquitectura pero ésta no sea bien comprendida ni comunicada, en otras palabras bien documentada, el proyecto puede fracasar. La arquitectura presenta un plano del sistema que abarca tanto el sistema en construcción como el proyecto de desarrollo, definiendo las unidades de trabajo que deben realizar los equipos tanto de diseño como de implementación. La documentación de la arquitectura con suficiente detalle, sin ambigüedades, y organizada de forma que otros puedan encontrar rápidamente la información

necesaria, es tan importante como definirla correctamente. Posiblemente el concepto más importante asociado a la documentación de la Arquitectura es el de “vistas”, representación de un conjunto de elementos del sistema y las relaciones asociadas con ellos.”

Por otro lado Bass, Clements, Kazman (2003, p.560), citado en el trabajo de investigación Delgado (2007, p.51), se responde específicamente a la pregunta “porqué es importante la Arquitectura de Software”, “desde el punto de vista técnico, por tres razones fundamentales: comunicación entre los involucrados, decisiones tempranas de diseño y abstracción reusable y transferible del sistema”. A continuación se presenta un resumen de los principales conceptos planteados para cada una de las razones fundamentales identificadas:

- Como vehículo de comunicación entre los involucrados en el desarrollo, desde clientes, director de proyecto, implementadores, verificadores, entre otros, donde cada uno está interesado en distintas características del sistema afectadas por la arquitectura, y ésta provee entonces un lenguaje común donde estos intereses distintos pueden expresarse, negociarse y ser resueltos a un nivel que es intelectualmente manejable incluso para sistemas complejos y de gran tamaño.
- Representa el conjunto de decisiones tempranas de diseño, las más difíciles de tomar correctamente y las más difícil de cambiar luego en el proceso de desarrollo. Define restricciones en la implementación en cuanto a los elementos que se deben desarrollar, la forma en que deben interactuar y realizar las responsabilidades asignadas. Los atributos de calidad del sistema son permitidos o inhibidos por la Arquitectura de Software definida.

Las estrategias para obtener atributos de calidad como performance, modificabilidad, seguridad, entre otras, se aplican principalmente a nivel arquitectónico. Es posible predecir este cumplimiento en base a las decisiones arquitectónicas tomadas utilizando métodos de evaluación de Arquitecturas.

- Como un modelo reusable y transferible, permite reutilizar decisiones arquitectónicas, transfiriendo por lo tanto las consecuencias de estas decisiones, lo que resulta importante por ejemplo, en el marco de líneas de productos de software, donde la Arquitectura define que aspectos están fijos para todos los miembros y cuales son variables. Permite realizar análisis “implementar, comprar, reutilizar” en base a las definiciones realizadas. La recolección de estilos/patrones arquitectónicos reconocidos permite que “menos sea más” ya que si bien los elementos de software pueden combinarse en forma ilimitada, restringir estas combinaciones a determinadas elecciones permite ganar en varios aspectos como minimizar la complejidad del diseño, promover la reutilización de software, entender y comunicar más fácilmente la Arquitectura definida, brindar mayor capacidad de análisis y acortar el tiempo de diseño.

Según Delgado (2007, p.51), de las reflexiones presentadas se puede observar que la importancia de la Arquitectura de Software en el Diseño de Software va más allá de ser únicamente relativa a aspectos técnicos que si son de los más importantes. La Arquitectura de Software constituye la base de las definiciones que se realizan como parte del Diseño de Software, provee las bases para el desarrollo del sistema, identificando los elementos de

software a ser construidos, sus características y relaciones, permite identificar oportunidades de reúso.

Sirve como forma de comunicación tanto en el equipo de desarrollo como con los involucrados en el mismo, incluyendo cliente y gerentes del proyecto, la utilización de estilos/patrones permite reutilizar decisiones de diseño y facilita la comprensión de la Arquitectura definida, así como su comunicación. Tener una Arquitectura de Software definida permite razonar sobre las decisiones de diseño tomadas y sus consecuencias, y evaluarlas contra los requerimientos especificados para los atributos de calidad de forma de comprobar su cumplimiento identificando riesgos asociados a las decisiones y definiciones tomadas. En desarrollos de la complejidad y tamaño que presentan las aplicaciones actuales, contar con una Arquitectura de Software definida, documentada y comprendida por todos los involucrados constituye la base para el éxito del proyecto.

2.2. Arquitectura Orientada a Servicios (SOA)

2.2.1. Introducción

La capacidad para responder de una forma rápida a los cambios y hacer óptimo los procesos de negocio es un factor clave para la competitividad y desarrollo de las organizaciones. La agilidad de estas puede verse afectada si se apoya en entornos de Tecnologías de la Información que no pueden responder de una forma flexible a los cambios que afectan las actividades del negocio.

Según la publicación (Microsoft Corporación, 2016) las empresas necesitan poder interconectar los procesos, personas e información tanto con la propia organización como -atravesando sus fronteras- con subsidiarias y socios comerciales. La falta de integración entre los componentes de IT –sistemas, aplicaciones y datos- hace difícil obtener una respuesta rápida y efectiva ante los cambios que afectan de forma natural a los negocios. La inflexibilidad genera costes, reduce la capacidad de respuesta ante los clientes, compromete el cumplimiento con las normativas legales y afectan negativamente a la productividad de los empleados. En suma, una deficiente integración es uno de los problemas más importantes a los que las organizaciones deben hacer frente para mantener su competitividad y garantizar su crecimiento. La Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture) supone una estrategia general de organización de los elementos de IT, de forma que una colección abigarrada de sistemas distribuidos y aplicaciones complejas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible.

Un proyecto SOA bien ejecutado permite alinear los recursos de IT de forma más directa con los objetivos de negocio, ganando así un mayor grado de integración con clientes y proveedores, proporcionando una inteligencia de negocio más precisa y más accesible con la cual se podrán adoptar mejores decisiones, y ayuda a las empresas a optimizar sus procesos internos y sus flujos de información para mejorar la productividad individual. El resultado neto es un aumento muy notable de la agilidad de la organización. Si bien una adopción de SOA bien planificada y ejecutada puede mejorar la capacidad de respuesta de las organizaciones, no todos los esfuerzos de orientación a servicios han resultado satisfactorios

Cada vez más las organizaciones dependen de su infraestructura de IT para alcanzar sus objetivos. Pero en un entorno competitivo como el actual, aprovechar las oportunidades de negocio exige moverse con rapidez. Sin embargo, con frecuencia las Tecnologías de Información no permiten estas respuestas rápidas ni disponen de la flexibilidad necesaria para competir de forma efectiva. Un alto porcentaje de las ineficiencias organizativas tienen un mismo origen: el predominio de procesos manuales con un nivel de error elevado, sistemas ineficaces para compartir la información en el seno de la organización; la incapacidad de hacer un correcto seguimiento de los procesos de negocio desde principio a fin; el cumplimiento con las normativas legales aplicables obliga a manejar grandes cantidades de información y en formatos complicados de utilizar; y finalmente, las ineficiencias propias del servicio a clientes. Allí donde se produce cualquiera de estas situaciones, el impacto sobre la productividad de los empleados es negativo, y se pone en riesgo la capacidad de crecimiento y competencia de la propia empresa. En la raíz de todas estas deficiencias está la información. No es un problema de escasez de información –de hecho, la información es el activo de más rápido crecimiento dentro de las organizaciones- sino de la imposibilidad de presentar la información de forma sencilla y útil a los usuarios y directivos de una manera coherente y sistemática. En última instancia, esto se debe a que las aplicaciones de línea de negocio y otras aplicaciones antiguas normalmente dan soporte a funcionalidades avanzadas –en áreas como la gestión financiera, marketing, control de clientes, etc.- pero sin poder compartir información entre ellas y, por consiguiente, no pueden aportar una visión general de los procesos de negocio cuando éstos abarcan varias áreas funcionales. Para lograr que la información se mueva a través de sistemas distintos (y generalmente incompatibles), dentro de las fronteras de la organización o a través de ellas, siempre es necesaria la intervención humana. Hasta ahora esta intervención se ha

hecho de dos formas. La primera es la de volver a introducir a mano los datos entregados por un sistema en otro distinto e incompatible. La segunda ha sido programar una interfaz especialmente diseñada para permitir la transferencia de información entre dos aplicaciones incompatibles. Ambas soluciones son costosas e ineficientes debido en el primer caso a que se trata de un esfuerzo de duplicación de datos donde el error humano es frecuente, y en el segundo, porque conseguir un buen nivel de interoperabilidad e integración no solo es muy difícil a la hora de desarrollar una solución, sino porque su mantenimiento posterior es una verdadera pesadilla: cualquier cambio en cualquier aplicación puede hacer fallar la interfaz de transferencia e impedir la comunicación entre ellas. Lo que se necesita es una herramienta basada en estándares para integrar sistemas y aplicaciones heterogéneos sobre una serie de plataformas y protocolos de comunicación heterogéneos, así como una metodología bien establecida para lograr el nivel óptimo de integración, de manera que la infraestructura subyacente facilite –en lugar de impedir- los cambios posteriores que puedan surgir como respuesta a la evolución en las necesidades de la empresa.

2.2.2. Definición

Delgado (2007, p.71) en su trabajo de investigación cita a Erl (2005) el cual menciona que el concepto de SOA está enfocado en la definición de una infraestructura del Negocio, con el término servicio lo que se está denotando es un servicio del Negocio como puede ser realizar una reservación aérea o acceder la base de datos de la compañía de un cliente. El servicio provee operaciones del Negocio como hacer una reservación, cancelar una reservación o recuperar el perfil de un cliente. Sin embargo también hay servicios técnicos de infraestructura que proveen operaciones como comienzo de transacción o modificación de datos. Una SOA debe desacoplar aplicaciones del Negocio de servicios técnicos y hacer que la Organización sea independiente de una implementación técnica específica o infraestructura.

Según Erl (2005) la lógica colectiva que define y guía una empresa es una entidad siempre en evolución que cambia constantemente en respuesta a influencias externas e internas. Desde la perspectiva de TI, esta lógica empresarial puede ser dividida en dos mitades importantes: lógica del negocio y lógica de la aplicación. Cada una existe en un mundo por si mismo, y cada una representa una parte necesaria de la estructura organizacional contemporánea. La lógica del negocio es una implementación documentada de los requerimientos del negocio que se origina en el área del negocio de la empresa.

Además menciona que la lógica del negocio está generalmente estructurada en procesos que expresan esos requerimientos, así como cualquier restricción asociada, dependencia e influencias externas. La lógica de la aplicación es una implementación automatizada de la

lógica del negocio organizada en varias soluciones tecnológicas. La lógica de la aplicación expresa flujos de los procesos del negocio mediante sistemas adquiridos o desarrollados específicamente por la infraestructura de TI de la Organización, con restricciones de seguridad, capacidad técnica y dependencias con vendedores. En la figura 2.1 se muestran estos conceptos.

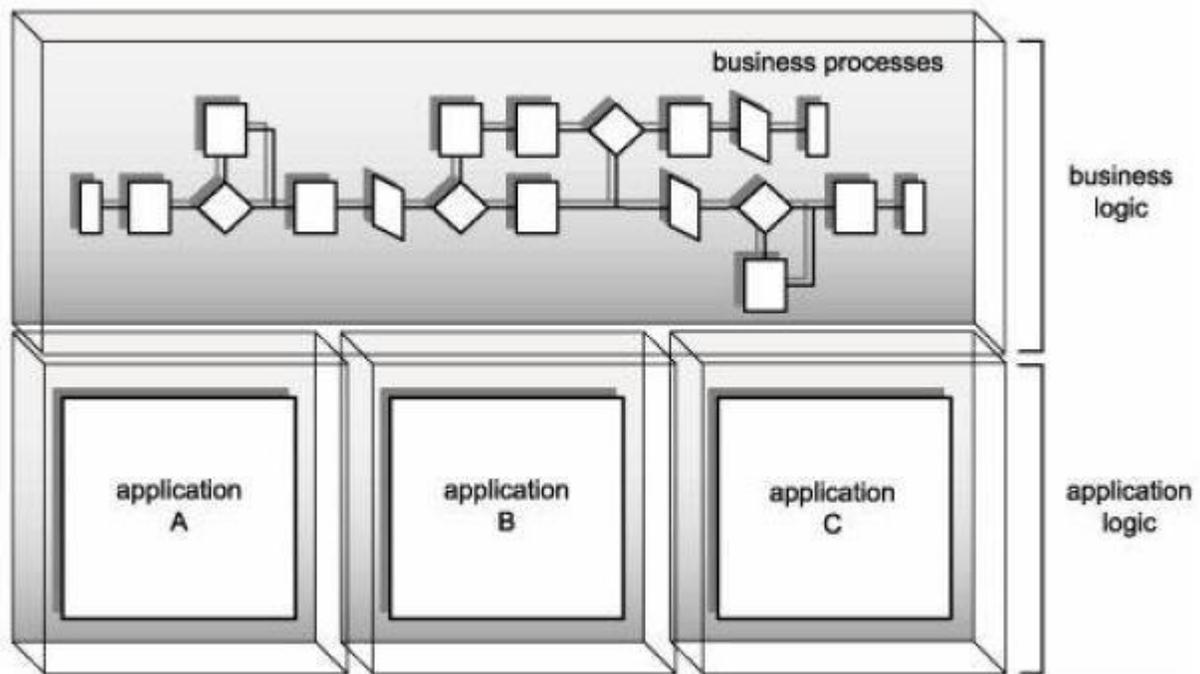


Figura 6: Dos aspectos claves de las organizaciones: proceso de negocio y tecnología. (Erl, 2005)

Erl (2005) menciona que los conceptos introducidos por la orientación a servicios son realizados mediante la introducción de servicios. Los servicios establecen una forma de abstracción de alto nivel entre las capas tradicionales del negocio y la aplicación. Al posicionarse entre medio, los servicios pueden encapsular la lógica de la aplicación física así como la lógica de los procesos del negocio. Los servicios modularizan la empresa, formando

unidades de lógica independientes que existen mediante una capa común de conectividad. Los servicios pueden ser estratificados de forma que servicios padre puedan encapsular servicios hijos. Esto permite que la capa de servicios a su vez consista de múltiples capas de abstracción en un nivel físico, los servicios son desarrollados y distribuidos en ambientes propietarios, mientras son individualmente responsables por el encapsulamiento de la lógica de la aplicación específica. En la figura 2.2 se muestra como los servicios en la capa de interface de servicios representan lógica de la aplicación realizada en distintas plataformas.

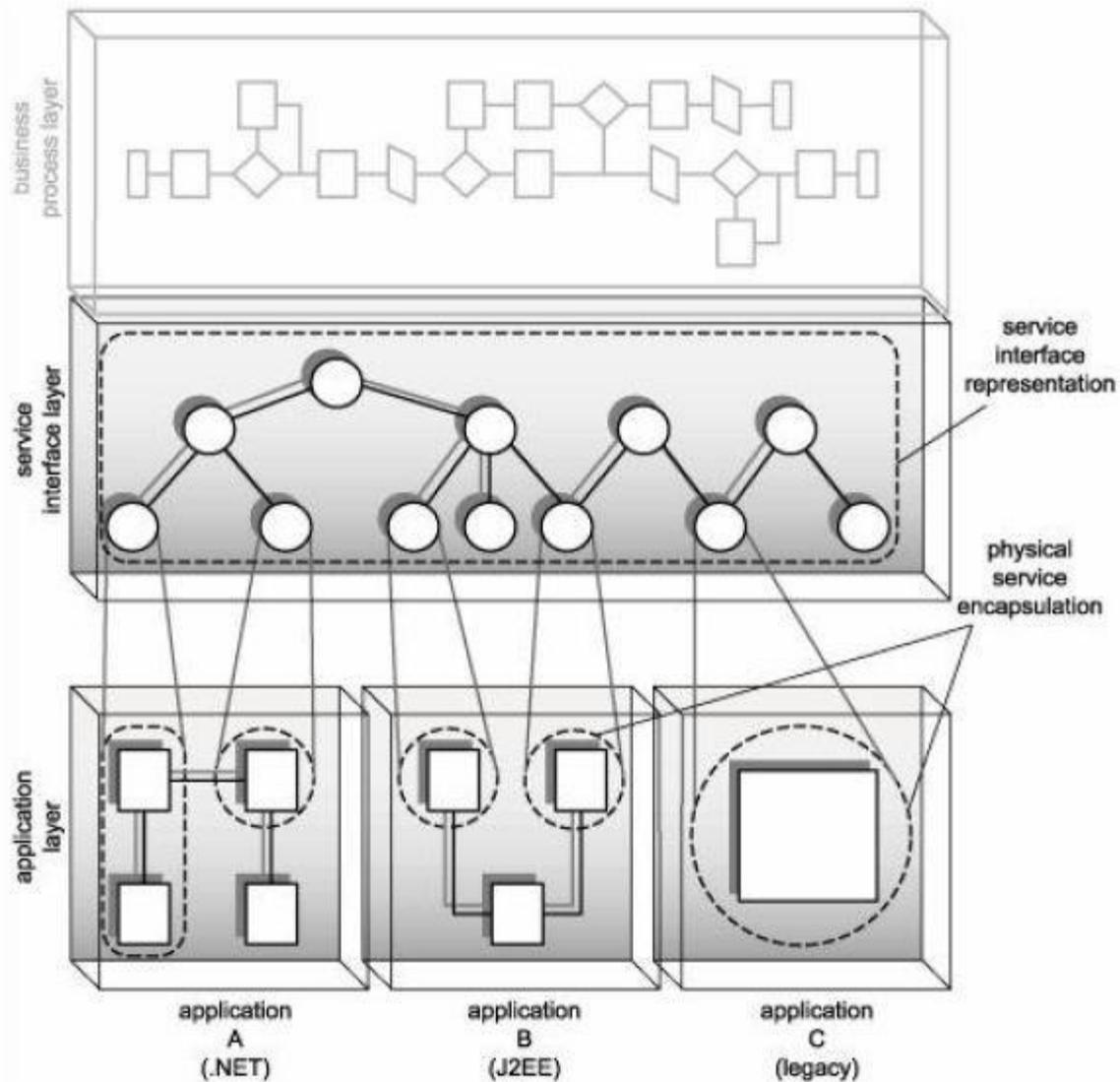


Figura 7: Capa de servicios entre los procesos del Negocio y las tecnologías. (Erl, 2005)

Endrei (2007) el cual es citado en el trabajo de investigación de Delgado (2007, p.72) menciona que los problemas de heterogeneidad, interoperabilidad y requerimientos cambiantes, son solucionados por una SOA que provee una plataforma para la construcción de aplicaciones basadas en servicios con las características de bajo acoplamiento, ubicación transparente de

servicios e independencia de protocolos. Un consumidor de servicios no debe preocuparse por un servicio particular con el que comunicarse debido a que la infraestructura por debajo, el bus de servicios, puede hacer la elección en representación del consumidor. La infraestructura esconde la mayor cantidad de detalles técnicos posible al consumidor, y diferentes tecnologías de implementación como J2EE [J2EE] o .NET [MNET] no afectan a los usuarios de una SOA. También debe ser posible sustituir una implementación de un servicio por una mejor si la hay disponible, con mejores características de calidad de servicio, según la metadata que describe al servicio.

También menciona que en las últimas décadas se han experimentado cambios importantes en las tecnologías y paradigmas de desarrollo y uso de aplicaciones, que han impactado fuertemente en la forma en que las Organizaciones realizan e implementan su negocio. Por un lado el acceso cada vez mayor a Internet pone a disposición información y software, permite comunicación entre distintas Organizaciones, sus proveedores y clientes, ampliando tanto los requerimientos del negocio como sus necesidades y realización. Las mejoras en la tecnología continúan en forma acelerada, incrementando a su vez el ritmo de los cambios en los requerimientos. El negocio debe adaptarse rápidamente para sobrevivir en el ambiente dinámico actual, y la infraestructura de TI debe permitir la habilidad de adaptación del negocio. Esta evolución desde las tecnologías al negocio y viceversa, presenta una parte importante en la evolución de las Arquitecturas de Software desde los primeros sistemas monolíticos hasta la visión actual de orientación a servicios, esto se muestra en la siguiente figura.

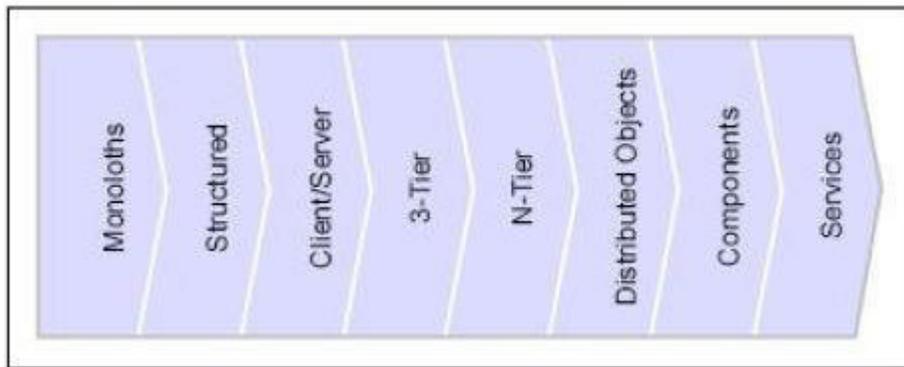


Figura 8: Evolución de Arquitecturas de Software hasta SOA. (Endrei, 2007)

No existe una única definición para Service Oriented Architecture (SOA), las que sin embargo comparten algunos conceptos base. A continuación citaremos definiciones de algunos autores para poder tener una visión más clara de lo que significa SOA.

Según la publicación de la Object Management Group (OMG, 2016) la Arquitectura Orientada a Servicios (SOA) es una arquitectura de software donde la funcionalidad se agrupa en torno a los procesos de negocio y empaquetado como servicios interoperables. SOA también describe la infraestructura de TI que permite a diferentes aplicaciones para intercambiar datos entre sí a medida que participan en los procesos de negocio. El objetivo es una articulación flexible de servicios con sistemas operativos, lenguajes de programación y otras tecnologías que subyacen a las aplicaciones. SOA separa funciones en unidades distintas, o servicios, que se hacen accesibles a través de una red a fin de que se pueden combinar y volver a utilizar en la producción de aplicaciones de negocios. Estos servicios se comunican entre sí mediante el paso

de datos de un servicio a otro, o mediante la coordinación de una actividad entre dos o más servicios.

Según la publicación en Wikipedia (2016) La Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos. Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio las cuales incluyen facilidad y flexibilidad de integración con sistemas legados, alineación directa a los procesos de negocio reduciendo costos de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad.

Para el autor Hewitt (2009, p.9) SOA es un tipo de arquitectura que utiliza servicios como bloques de construcción para facilitar a la empresa la integración y reutilización de componentes al poseer bajo acoplamiento.

Bocchio (2014, p.1) en su trabajo de investigación cita a Erl (2005) quien define a SOA como una representación de una arquitectura abierta, extensible y federada basada en composición, que promueve la orientación a los servicios interoperables e independientes de los proveedores, los cuales pueden ser identificados en catálogos con gran potencial de reutilización e implementados como Servicios Web. SOA puede establecer una abstracción de la lógica del negocio y la tecnología, resultando en un bajo acoplamiento entre dominios.

En el mismo trabajo de investigación Bocchio (2014, p.1) cita a Newcomer & Lomow (2005) quienes dicen que SOA es el producto de la evolución de plataformas de tecnología que se

solían utilizar con frecuencia, preservando las características exitosas de las arquitecturas tradicionales. Podemos entender a las arquitecturas SOA como un estilo de diseño que guía los aspectos de creación y uso de servicios de negocio a través de su correspondiente ciclo de vida. Además define y aprovisiona la infraestructura de tecnologías de la información que permite a diferentes aplicaciones intercambiar datos y participar en los procesos de negocio, independientemente del sistema operativo o de los lenguajes de programación con los cuales estos servicios (y aplicaciones) fueron desarrollados e implementados.

Josuttis (2007) quien es citado por Bocchio (2014, p.1) en su trabajo de investigación observa que muchas definiciones de SOA incluyen el término “Servicios Web”, sin embargo, es necesario hacer la distinción de estos conceptos y aclarar que SOA no es lo mismo que Servicios Web puesto que SOA, a diferencia de los Servicios Web, define y trata un paradigma, en tanto que los Servicios Web son solo una forma posible de consumir la infraestructura utilizando una estrategia de implementación específica.

De las definiciones previamente citadas podemos inferir que SOA es un concepto, un paradigma de arquitectura que permite el tratamiento de procesos de negocio distribuidos de sistemas nuevos sistemas heterogéneos que se encuentran bajo el control o responsabilidad de diferentes propietarios, siendo su concepto clave principal el de los servicios, la interoperabilidad entre lenguajes y bajo acoplamiento.

2.2.3. Mitos y Realidades de SOA

Alrededor de SOA están asociados varios mitos y realidades, los cuales son importantes entender antes de profundizar en el tema, según la publicación en (Wikipedia, 2016) La Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture). A continuación se presenta una lista de mitos con sus respectivas realidades.

Mitos

- a) SOA es una tecnología.
- b) Las SOA requieren de servicios web.
- c) SOA es nuevo y revolucionario.
- d) SOA garantiza la alineación de TI y el negocio.
- e) Una arquitectura de referencia SOA reduce riesgo de implementación.
- f) SOA requiere una revisión completa de la tecnología y procesos de negocios.
- g) Necesitamos construir una SOA.

Realidades

- a) SOA es una filosofía de diseño independiente de cualquier proveedor, producto, tecnología o industria. Las necesidades de SOA varían de una compañía a otra, por tanto la adquisición de una arquitectura SOA de otra compañía no será la solución apropiada para su propia compañía.
- b) SOA se puede realizar a través de servicios web pero los servicios web no son un requisito necesario para implementar SOA.

- c) EDI, CORBA y DCOM son ejemplos conceptuales de orientación de servicios.
- d) SOA no es una metodología.
- e) No hay dos SOA iguales. Una arquitectura de referencia SOA puede no ofrecer la mejor solución para su organización.
- f) SOA debe ser gradual y construirse sobre sus inversiones actuales.
- g) SOA es un medio, no un fin.

2.2.4. Características de SOA

Bocchio (2014) en su trabajo de investigación nos cita a varios autores para describir las características de SOA, en esta sección se presentan dichas características, tales como Bajo Acoplamiento, transparencia de Red, reusabilidad y granularidad, sistemas distribuidos, diferentes propietarios y heterogeneidad.

2.2.1.1. Bajo Acoplamiento

Stevens (1974), el cual es citado por Bocchio (2014, p.7) señala que el acoplamiento es el grado en que cada módulo de programa depende de cada uno de los otros módulos de programa. El bajo acoplamiento generalmente se correlaciona con la alta cohesión y viceversa.

Bocchio (2014, p.7) también cita a Pulier y Tylor (2006), los cuales destacan que los servicios web poseen la característica de estar basados en una estructura flexible mediante la

cual, una vez que una pieza de software ha sido expuesta como un servicio web, es relativamente fácil de mover a otro equipo, puesto que la funcionalidad se encuentra abstraída de la interfaz que define el contrato de sus operaciones.

La figura 5 ilustra el bajo acoplamiento del servicio. En la parte 1 del dibujo, un miniordenador accede a un servicio web que se ha expuesto en un mainframe. Digamos que, sin embargo, el propietario de la unidad central quiere remplazar la máquina antigua con un nuevo servidor Sun. Como vemos en la parte 2, la máquina Sun sustituye a la unidad central, pero la minicomputadora, que es el consumidor del servicio web no tiene conocimiento de esta sustitución. La minicomputadora se sigue comunicando por medio del protocolo SOAP. Es completamente transparente para el consumidor del servicio (cliente) que la interfaz del servicio esté implementada en una computadora central, en un equipo Windows o en cualquier otra plataforma. Una vez que la unidad central ha sido sustituida por la máquina Sun, la minicomputadora continúa accediendo al servicio sin percibir este cambio. En las partes 3 y 4 de la figura antedicha el proceso de sustitución de ordenadores continúa. El propietario de la minicomputadora la remplaza con un PC. El PC que está equipado con su propia interfaz SOAP puede acceder fácilmente a los servicios web expuestos por la máquina Sun. Si por cualquier razón el propietario del servicio decidiera sustituir la máquina Windows por otro servidor Sun, no habría problema alguno; una vez más se podría acceder al servicio sin necesidad de realizar ninguna modificación a éste. A continuación se presenta la figura 5, que ilustra el bajo acoplamiento de servicios.

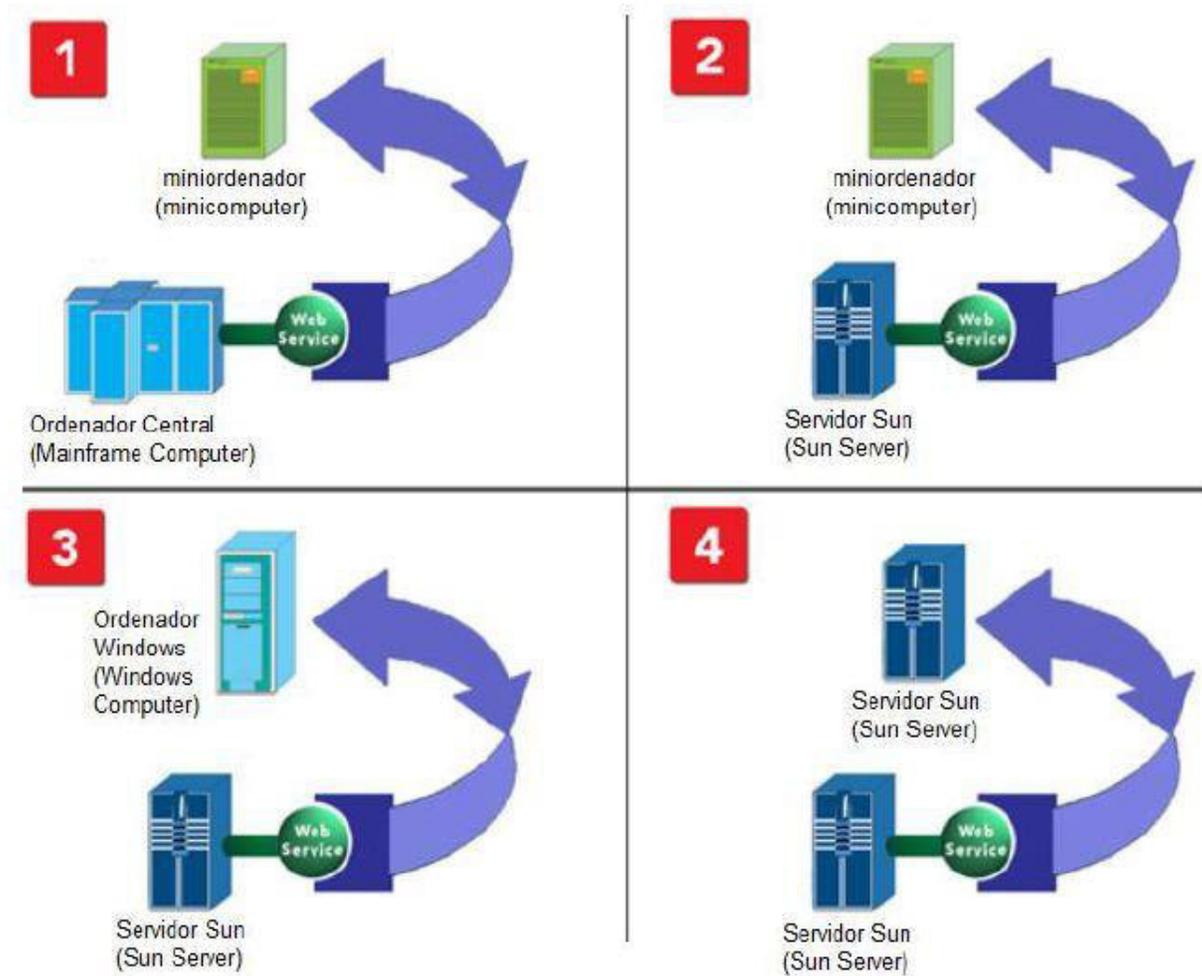


Figura 9: Bajo Acoplamiento del Servicio. (Pulier y Tylor, 2006)

2.2.1.2. Transparencia de Red

Bocchio (2014, p.8) en su trabajo de investigación cita a Pulier y Tylor (2006), quienes argumentan que debido a que el acoplamiento entre los servicios web es "flojo" y que los consumidores y los proveedores de servicios de Internet envían mensajes entre sí mediante protocolos abiertos de Internet, los servicios web ofrecen transparencia total de la red a los

que los emplean. Transparencia de la red se refiere a la capacidad de un servicio web para estar activos en cualquier parte de una red, o grupo de redes, sin tener ningún impacto en su capacidad de funcionar. Debido a que cada servicio web tiene su propia URL, servicios web tienen una flexibilidad similar a sitios web en Internet. De la misma manera que no hay diferencia en qué parte del mundo un sitio web está alojado para poder ser navegado, un servicio web se puede encontrar en cualquier equipo que esté conectado a la red y se comunica con protocolos de Internet. Cuando navegamos por ejemplo Amazon.com para comprar un libro, no existe ninguna necesidad de conocer dónde residen las aplicaciones a las cuales estamos accediendo con nuestro navegador, lo único que se necesita saber es la dirección web.

Un mismo servicio web puede estar situado en dos dominios diferentes. Si por alguna razón el dominio A se encontrara no disponible, entonces el consumidor del servicio podría acceder al servicio web desde el dominio B de manera alternativa. Lo único que tiene que ocurrir es la modificación de la dirección (URL) del servicio web en el documento WSDL, de manera que se establezca un vínculo entre el cliente (consumidor del servicio) y la nueva dirección del servicio web en el dominio B.

Dada nuestra amplia experiencia con la Internet en los últimos años, la propiedad de transparencia de la red puede no parecer tan importante, pero en realidad se trata de un aspecto fundamental para el futuro de la informática. La combinación de acoplamiento débil y la transparencia de red presentan nada menos que una revolución en la informática empresarial, no porque se trate de una idea novedosa, sino debido a que la infraestructura y

los estándares han llegado por fin para que sea una realidad. Las empresas han gastado fortunas en los últimos años en la gestión de interfaces que hacen factible la interoperabilidad de los equipos en entornos distribuidos. Algunas empresas Norteamericanas gastan cientos de miles de millones de dólares por año en tecnologías de la información.

2.2.1.3. Reusabilidad y granularidad

Bocchio (2014, p.9) en su trabajo de investigación cita a Carter (2007), quien manifiesta que un servicio es un software reutilizable y auto-contenido, independiente de las aplicaciones y las plataformas de computación en las cuales se ejecuta. Los servicios tienen interfaces bien definidas y permiten una correspondencia entre las tareas de negocio y los componentes exactos de TI necesarios para ejecutar la tarea. Los servicios SOA se centran en tareas de nivel de negocio, actividades e interacciones. La relación entre los servicios y los procesos de negocios es crítica. Un proceso de negocio es un conjunto de tareas de negocios relacionados que abarcan personas, sistemas e información para producir un resultado o producto específico. Con SOA, un proceso se compone de un conjunto de servicios.

Antes de SOA, el foco estaba en sub-tareas técnicas. Puede haber oído que la gente llama a esto un buen "nivel de granularidad" o bajo "grado de abstracción." Por la simplicidad de SOA, sabemos que es más que una simple relación uno-a-uno entre los pasos de un proceso (como la comprobación de una calificación crediticia) y los servicios que están diseñados para apoyar ese proceso de negocio flexible.

Cada empresa tiene una visión diferente respecto de la granularidad que requieren sus servicios en función de su diseño de negocios. En pocas palabras, la granularidad es la cantidad de función que un servicio expone. Por ejemplo, un servicio de granularidad fina proporciona unidades más pequeñas de un proceso de negocio, y un servicio de granularidad gruesa proporciona una tarea de negocio más amplia que contiene un mayor número de pasos.

Sandy Carter opina que los servicios no pueden ser demasiado grandes o demasiado pequeños, sino que deben tener la granularidad adecuada. Diseñar y decidir la granularidad de los servicios es una cuestión clave que conduce al éxito. Si el servicio es demasiado grande, será menos reutilizable. Si el servicio es demasiado pequeño, provoca una disminución en el rendimiento y mala asignación de tareas entre los negocios y los servicios que los apoyan.

Determinar qué tan grande o pequeño debe diseñarse un servicio es función qué tan atómica es la composición de su función.

Es importante observar que este concepto de servicios es una de las claves para hacer que SOA sea el lenguaje de los negocios. La mayoría de los líderes empresariales no comprenden el valor de SOA. En su lugar, se centran -y con razón- en el problema en cuestión. Debido a estos servicios de negocio, este lenguaje y la vinculación de los servicios de negocio y SOA se convierten en una pieza fundamental para la solución del problema, y constituyen la futura misión estratégica.

2.2.1.4. Sistemas Distribuidos

Bocchio (2014, p.10) en su trabajo de investigación cita a Josuttis (2007), quien explica que a medida que las empresas crecen se vuelven más y más complejas, y cada vez se involucran más empresas y sistemas. Hay una integración y un cambio constantes. SOA es muy adecuado para tratar con sistemas distribuidos complejos. De acuerdo con el modelo de referencia SOA de OASIS, es un paradigma para "organizar y utilizar capacidades distribuidas.

NOTA:

Una definición más adecuada para "capacidades distribuidas" en IT sería "sistemas distribuidos", o como dice la definición de SOA de Wikipedia: "nodos de una red" o "recursos de una red."

SOA permite a las entidades que necesitan ciertas capacidades distribuidas, localizar y hacer uso de esas capacidades. En otras palabras, facilita las interacciones entre los proveedores de servicios y sus consumidores, lo que permite la realización de funcionalidades de negocio.

2.2.1.5. Diferentes Propietarios

En su trabajo de investigación Bocchio (2014, p.11) cita a Josuttis [2007] quien recuerda que la definición de SOA del modelo de Referencia de OASIS dice que las capacidades distribuidas "pueden estar bajo el control de diferentes dominios de propiedad." Este es un punto muy importante que a menudo es suprimido en definiciones de SOA. Esta es una de

las claves para ciertas propiedades de SOA, y una razón importante por la que SOA no es sólo un concepto técnico.

SOA incluye prácticas y procesos que se basan en el hecho de que las redes de los sistemas distribuidos no son controladas por los propietarios individuales. Diferentes equipos, diferentes departamentos o incluso diferentes empresas pueden gestionar diferentes sistemas. Por lo tanto, diferentes plataformas, programas, prioridades, presupuestos, etc. deben ser tenidos en cuenta. Este concepto es clave para la comprensión de SOA y de grandes sistemas distribuidos en general.

A continuación se presenta la figura 2.2, que ilustra un conjunto de sistemas distribuidos con diferentes propietarios.

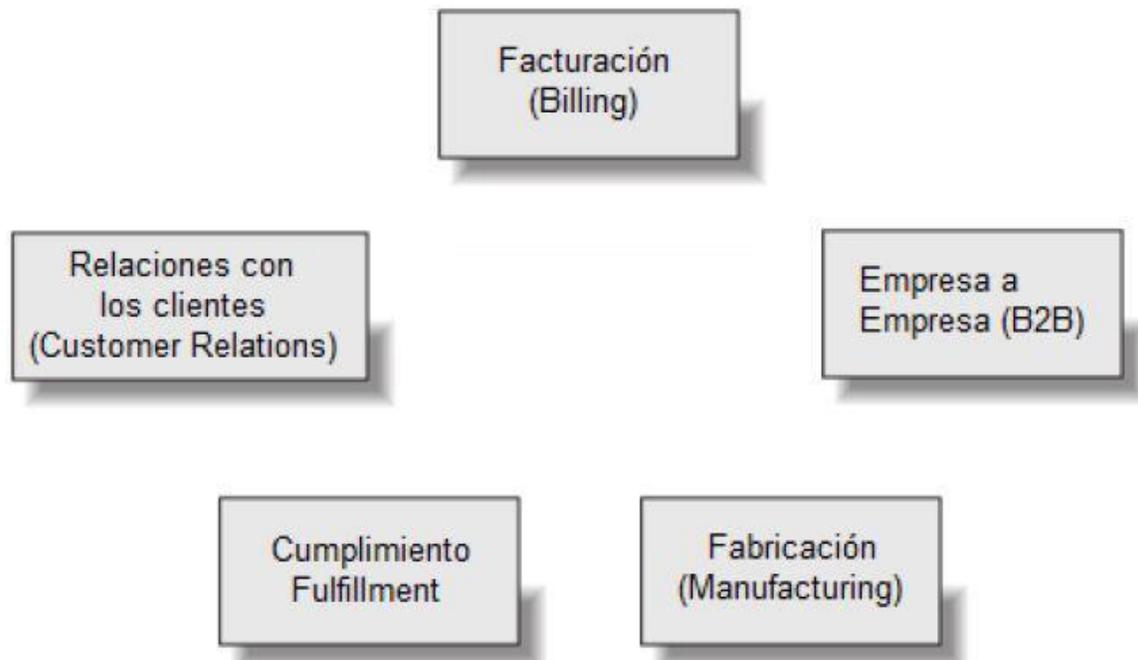


Figura 10: Sistemas distribuidos con diferentes propietarios. (Josuttis, 2007)

Las formas de lidiar con problemas y realizar modificaciones en los entornos con diferentes propietarios y en entornos donde se dispone de control total, pueden variar. No se puede implementar la funcionalidad y modificar el comportamiento de la misma manera en grandes sistemas como en sistemas más pequeños. Una consideración importante es que "la política" entra en juego: hay que comprometerse con los demás, y hay que aceptar que existen diferentes prioridades y soluciones. Porque no se puede controlar todo, hay que aceptar que no siempre puede ser capaz de hacer las cosas a su manera.

2.2.1.6. Heterogeneidad

En su trabajo de investigación Bocchio (2014, p.12) cita a Josuttis (2007) quien explica que una diferencia muy importante entre los sistemas pequeños y grandes es la falta de armonía. Todos lo sabemos por experiencia (aunque podríamos no estar de acuerdo acerca de si se trata de un fenómeno natural o el resultado de un mal diseño). Los grandes sistemas utilizan distintas plataformas, diferentes lenguajes de programación (y paradigmas de programación), e incluso diferentes componentes middleware. Se trata de un lío de mainframes, ejércitos de SAP, bases de datos, aplicaciones J2EE, pequeños motores de reglas, etc. En otras palabras, son heterogéneos.

En el pasado, se han propuesto una gran cantidad de métodos para resolver el problema de la integración de sistemas distribuidos mediante la eliminación de la heterogeneidad: "Vamos a armonizar, y todos los problemas habrán desaparecido," "Vamos a sustituir todos los sistemas con aplicaciones J2EE", "Vamos a usar CORBA en todas partes", "Usemos serie

MQ," y así sucesivamente. Pero todos sabemos que estos métodos no funcionan. Grandes sistemas distribuidos con diferentes propietarios son heterogéneos.

Esto es un hecho, y por lo tanto algo que debemos aceptar la hora de diseñar soluciones distribuidas de gran tamaño.

2.2.5. Beneficios de SOA

Según la publicación de (Accenture, 2008) existen cinco factores importantes que aumentan el interés del equipo ejecutivo y sobre todo, de los responsables de desarrollo, por la arquitectura SOA, los cuales presentamos a continuación:

- **La arquitectura SOA ayuda a mejorar la agilidad y flexibilidad de las organizaciones**

Las empresas deben ser capaces de crear y producir nuevos productos y servicios para unos clientes y ciudadanos que son cada vez más exigentes. El aumento de la colaboración con los clientes y proveedores, y la mayor capacidad para interpretar los datos de los clientes, proporcionan a las organizaciones los medios necesarios para interpretar los cambios del mercado de una forma más precisa y rápida. Lo que necesitan en estos momentos es conseguir que sus procesos de negocio sean capaces de ser adaptados al menos al mismo ritmo. Este dinamismo exige un nuevo conjunto de capacidades tecnológicas que permitan adaptar rápidamente los sistemas informáticos.

El pensamiento tecnológico tradicional, que normalmente intentaba crear una aplicación nueva para cada proceso nuevo, nunca ha sido capaz de generar tal agilidad. Las

aplicaciones se desarrollaban normalmente en momentos diferentes, con diferentes intenciones, plataformas, conjuntos de usuarios y niveles de servicio, y suponían diferentes ciclos de mantenimiento, mejoras y presupuestos. Haciendo un análisis retrospectivo, no nos sorprende que los esfuerzos por integrar las aplicaciones y los sistemas de una organización pudieran ser tan laboriosos y costosos de implantar y mantener.

Por el contrario, la arquitectura SOA se centra en las capacidades, no en las aplicaciones. SOA contempla la arquitectura de toda la empresa, incluidos los procesos de negocio y las tecnologías de la información. Además, el alto nivel de desacoplamiento e interoperabilidad proporcionado por la arquitectura SOA permite un alto grado de reutilización (interno y externo) y de parametrización. Todo ello redundando en una mayor facilidad y flexibilidad para adaptar y mejorar los procesos de las organizaciones según los cambios de prioridad del negocio.

- **La arquitectura SOA permite una “personalización masiva” de las tecnologías de la información**

La personalización masiva es un concepto que se ha tomado prestado de los procesos de fabricación, donde al combinar de distinta manera los módulos estándar, se puede dar forma a un producto individualizado dentro de la infraestructura masiva de producción. Mediante la arquitectura SOA se puede aplicar el mismo principio a la tecnología de una organización y, como consecuencia, a los procesos de negocio habilitados por dicha tecnología. Así por ejemplo, en una gran compañía de telecomunicaciones, la arquitectura

SOA ha permitido acelerar el proceso de creación e integración de nuevos servicios, y abaratar sus costes, lo que ha permitido desarrollar complejas políticas de precios y contratación mejor adaptadas a segmentos específicos de clientes.

Frecuentemente, la información necesaria para desarrollar nuevos servicios o productos ya existe dentro de los sistemas de la organización, y la arquitectura SOA crea una forma más fácil y rápida (y más barata) de acceder a ella que en el pasado, y de utilizarla en la unidad que gestiona los clientes. Ésta es, en efecto, la capacidad de personalizar productos y servicios a gran escala, mientras se utiliza la misma infraestructura servidora o transaccional (“de back-end”).

- **La arquitectura SOA permite la simplificación del desarrollo de soluciones mediante la utilización de estándares de la industria y capacidades comunes de industrialización**

La arquitectura SOA desacopla los tres componentes de una aplicación: presentación, orquestación de procesos y lógica de negocio, a la vez que estandariza la comunicación entre cada una de las capas. Todo ello favorece a que el proceso de construcción se pueda dividir y por lo tanto industrializar más fácilmente.

Además, las empresas se pueden focalizar en los componentes de mayor valor como los procesos y externalizar o comprar el resto de componentes.

Por ejemplo, una gran entidad bancaria ha decidido concentrarse en la definición de los procesos de la entidad, delegando la construcción (o compra) de los componentes, presentación y servicios, a factorías de software.

- **La arquitectura SOA permite aislar mejor a los sistemas frente a los cambios generados por otras partes de la organización (protección de las inversiones realizadas)**

Al organizar los sistemas en módulos más pequeños (servicios) se reduce notablemente el impacto de los cambios. Por otra parte, durante las últimas décadas, las organizaciones han realizado fuertes inversiones en sus infraestructuras tecnológicas. A través de la creación de un modelo flexible que pueda reconfigurarse en función de las necesidades del negocio, la arquitectura SOA reutiliza, de un modo efectivo, los distintos sistemas tecnológicos actuales, por ejemplo, identificando la funcionalidad bajo los sistemas tecnológicos actuales y encapsulándolos en servicios que pueden ser utilizados por diferentes aplicaciones y procesos.

Al respecto, las principales compañías que ofrecen herramientas de “discovery” están reorientando sus productos para ofrecer la identificación de reglas de negocio y servicios de los sistemas actuales, para facilitar su evolución hacia SOA.

- **La arquitectura SOA permite alinear y acercar las áreas de tecnología y negocio**

SOA cubre la brecha entre la visión del negocio y la de sistemas, estableciendo un marco de diálogo con un lenguaje común: los procesos de negocio.

- ✓ Las áreas de negocio se centran en la definición de los procesos de acuerdo a la estrategia y el modelo de negocio de la compañía.

- ✓ El área de tecnología implementa los procesos a partir de la utilización de servicios existentes y la creación de nuevos cuando es necesario.

Cuando el negocio requiere cambios en los procesos existentes, éstos se realizan de forma flexible y ágil, pues están implementados mediante tecnología estándar y servicios reutilizables. Además, por primera vez, hay una definición común de las aplicaciones: los procesos, que tanto el área de tecnología como el área de negocio comparten y entienden.

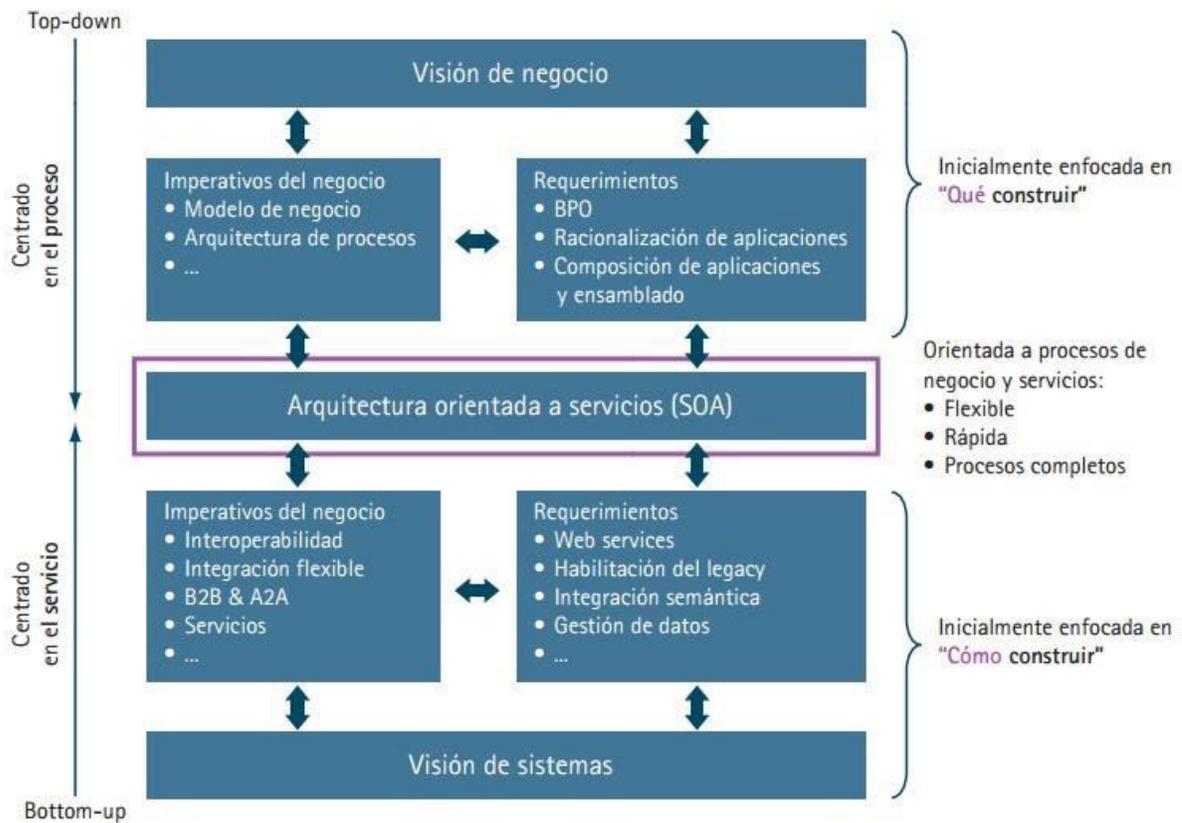


Figura 11: Agilidad en el negocio articulada por SOA. (Accenture, 2008)

2.2.6. Ejemplo de una Arquitectura Genérica alineada a SOA

Bocchio (2014, p.5) en su trabajo de investigación nos presenta un diagrama el cual citamos en la figura 2.3, dicho diagrama modela una arquitectura genérica de 3 capas alineada a SOA.

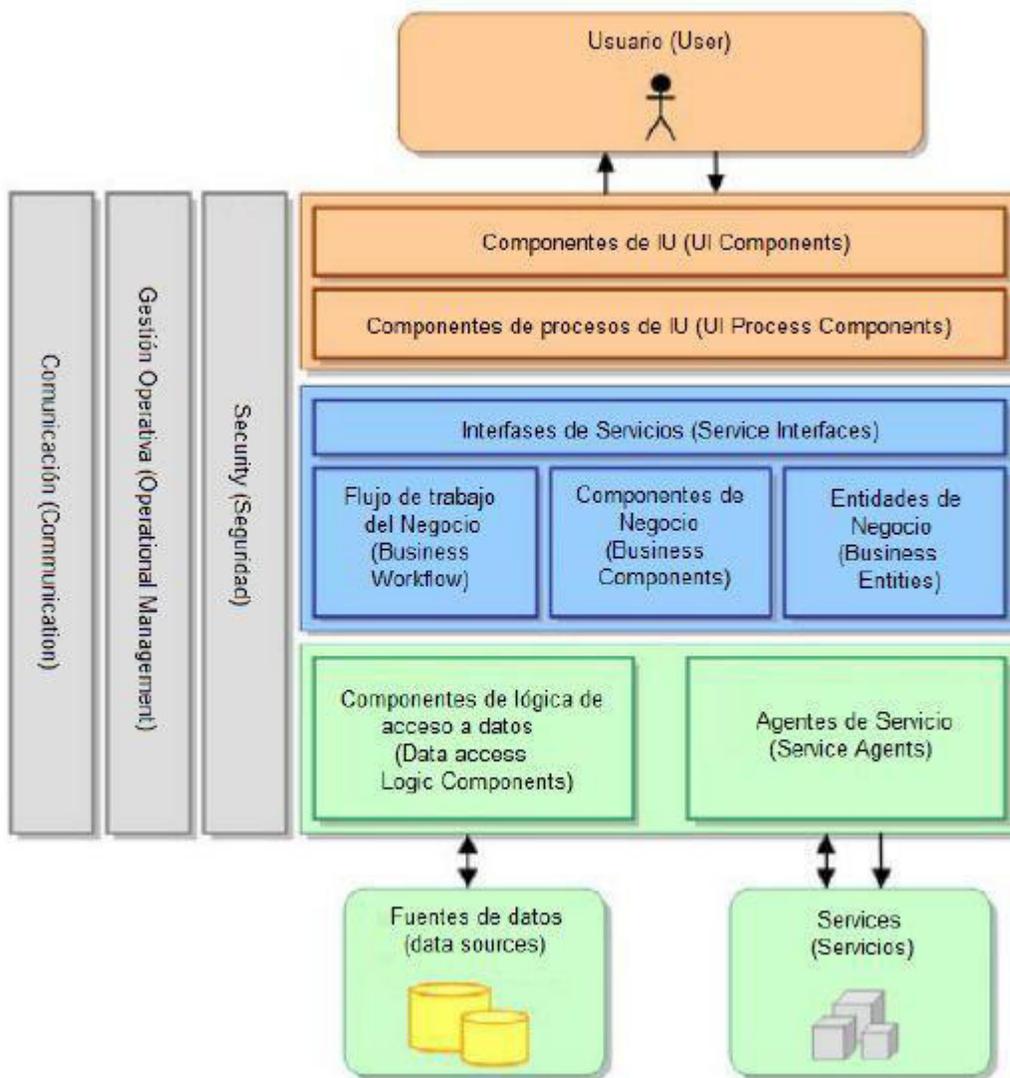


Figura 12: Arquitectura de 3 capas alineada a SOA. (Bocchio, 2014)

A continuación se presenta la tabla 1, la cual describe cada uno de los componentes que integran arquitectura genérica presentada en la figura 8:

Termino	Definición
Agentes de Servicio	Este componente (Service Agents) permite a las implementaciones de los servicios de negocio acceder a otras instancias de servicios, con el propósito de reutilizar funcionalidades, testear minimizar las pruebas funcionales (las funcionalidades reutilizables se encuentran encapsuladas en componentes y se implementan una única vez), minimizar el mantenimiento (cuando se implementan mejoras sobre un componente de servicios reutilizables, todos los servicios que refieran a este servicio podrán beneficiarse con las mejoras implementadas).
Componentes de IU	Se refiere a los componentes de interfaz de usuario que serán implementados y desplegados en esta arquitectura

Componentes de Lógica de Acceso a Datos	Estos componentes contienen la lógica de acceso a datos (Data Access Logic Components), abstrayendo y separando concretamente la capa de lógica del negocio de la capa de acceso a datos (separación y asignación de responsabilidades por capas).
Componentes de Negocio	Corresponden a la implementación de los servicios de negocio, comprendiendo fundamentalmente las funcionalidades y operaciones de negocio requeridas. Implementan las operaciones definidas en las interfaces del negocio (Service Interfaces).
Componentes de Procesos de IU	Hace referencia a los componentes que implementan procesos competentes a la capa de interfaz de usuario.
Entidades de Negocio	Corresponden a las entidades del modelo de dominio (Business Entities). Son utilizadas tanto por los servicios como por los componentes de acceso a datos.

Flujo de Trabajo del Negocio	Este componente encapsula e implementa mecanismos de flujos de trabajo (Business Workflow) que permiten formalizar y organizar los procesos del negocio para el modelo de dominio en cuestión.
Fuentes de datos	Algunas de las fuentes de datos posibles (Data Sources) podrían ser una base datos relacional, archivos de datos, un data grid, bases de datos NoSQL, etc.
Interfaces de Servicios	Las interfaces de servicios funcionan como contratos, definiendo formalmente las operaciones (capacidades) que brindarán los servicios basados en esta arquitectura.
Seguridad, gestión operativa, Comunicaciones	Estos componentes (Security, Operational Management, Communication) operan de manera transversal porque su funcionamiento debe considerarse en todas las capas. La aplicación podría utilizar también

	componentes para realizar la administración de excepciones, autorizar a los usuarios a realizar ciertas tareas y comunicarse con otros servicios y aplicaciones.
Usuarios	Se refiere a los usuarios que harán uso de la aplicación.

Tabla 1: Arquitectura de 3 capas alineada a SOA. Bocchio (2014)

2.3. Metodología RUP

La Compañía Rational (2001) estableció que el Rational Unified Process (RUP) es un Proceso de Ingeniería de Software, el cual proporciona un enfoque disciplinado para asignar tareas y responsabilidades de desarrollo dentro de una organización. Su objetivo es garantizar la producción de alta calidad software que satisfaga las necesidades de sus usuarios finales, dentro de un tiempo predecible y presupuestado.

El Rational Unified Process mejora la productividad del equipo, proporcionando cada miembro del equipo, con fácil acceso a una base de conocimientos con las directrices, las plantillas y los mentores de herramientas para todas las actividades críticas de desarrollo.

El Rational Unified Process es una guía para saber cómo utilizar eficazmente el Lenguaje Unificado de Modelado (UML). El UML es un lenguaje estándar de la industria que nos permite interpretar claramente los requisitos, arquitecturas y diseños. El UML fue creado originalmente por Rational Software, y ahora es mantenido por el Grupo de Gestión de organización de estándares de objetos (OMG).

El Rational Unified Process es apoyado por herramientas que automatizan gran parte del proceso. Se utilizan para crear y mantener los diversos artefactos-modelos, en particular, del proceso de ingeniería del software: modelado visual, programación, pruebas, etc. Ellos tienen un valor incalculable en el apoyo a toda la contabilidad asociada con la gestión del cambio, así como la gestión de la configuración que acompaña a cada iteración.

El Rational Unified Process es un proceso configurable. Ningún proceso es adecuado para todo el desarrollo de software.

El Proceso Unificado encaja en equipos de desarrollo pequeños, así como en las organizaciones de desarrollo de gran tamaño. El Proceso Unificado se basa en una arquitectura de proceso simple y clara que proporciona uniformidad a través de una familia de procesos. Sin embargo, se puede variar para adaptarse a diferentes situaciones. Contiene un kit de desarrollo, proporcionando soporte para configurar el proceso para adaptarse a las necesidades de una organización determinada.

El Rational Unified Process capta muchas de las mejores prácticas en el desarrollo de software moderno en una forma que es adecuado para una amplia gama de proyectos y organizaciones. La implementación de estas mejores prácticas utilizando el Rational Unified Process como su guía ofrece a equipos de desarrollo de una serie de ventajas importantes. En la siguiente sección, se describen las seis mejores prácticas fundamentales del Rational Unified Process.

2.3.1. Características Principales del RUP

A continuación se presentan las características principales de Rational Unified Process (RUP) mencionadas por Compañía Rational (2001).

2.3.1.1. Desarrollo iterativo

Dado los sistemas de software sofisticados de hoy en día, no es posible primero definir secuencialmente todo el problema, diseñar toda la solución, construir el software y luego probar el producto al final. Se requiere enfoque iterativo que permite una comprensión cada vez mayor del problema a través de refinamientos sucesivos, y crecer de forma incremental una solución efectiva a través de múltiples iteraciones. El Rational Unified Proces apoya un enfoque iterativo para el desarrollo que se ocupa de los temas de mayor riesgo en cada etapa del ciclo de vida reduciendo significativamente el perfil de riesgo de un proyecto. Este enfoque iterativo ayuda a atacar el riesgo a través de lanzamientos frecuentes progresivos, ejecutables que permiten la participación del usuario final y la retroalimentación continua.

Debido a que cada iteración termina con una versión ejecutable, el equipo de desarrollo se mantiene enfocado en producir resultados, y frecuente comprobaciones de estado ayudan a asegurar que el proyecto se mantiene en la fecha prevista. Un enfoque iterativo también hace que sea más fácil adaptarse a los cambios tácticos en los requisitos, características o el cronograma.

2.3.1.2. Administración de requisitos

El Rational Unified Process describe cómo obtener, organizar y documentar requerimientos funcionales y restricciones. Las nociones de casos de uso y escenarios en el proceso ha demostrado ser una excelente manera de capturar requisitos funcionales y así mismo asegurar que éstas impulsen el diseño, implementación y pruebas de software, por lo que es más probable que el sistema final cumpla las necesidades de los usuarios finales.

2.3.1.3. Uso de arquitectura basada en componentes

El proceso se centra en el desarrollo temprano y línea base de una sólida ejecución de arquitectura, antes de comprometer recursos para el desarrollo. En él se describe cómo diseñar una arquitectura resistente que es flexible, que se adapte a cambios, que sea intuitivamente comprensible, y promueva más la reutilización del software. El Rational Unified Process apoya el desarrollo de software basado en componentes.

Los componentes son módulos no triviales, subsistemas que cumplen una función clara. El Rational Unified Process ofrece un enfoque sistemático para la definición de una arquitectura utilizando componentes nuevos y existentes. Estos se ensamblan en una arquitectura bien definida.

2.3.1.4. Modelado visual del software

El proceso muestra el modelo visual del software para capturar la estructura y comportamiento de las arquitecturas y componentes. Esto le permite ocultar los detalles y escribir código usando gráficas y bloques de construcción, las abstracciones visuales ayudan a comunicar los diferentes aspectos de su software; ver cómo los elementos del sistema encajan entre sí; asegurarse de que los componentes son acordes con su código; mantener la coherencia entre un diseño y su aplicación; y promover la comunicación sin ambigüedades.

2.3.1.5. Verificación de calidad del software

El pobre rendimiento de las aplicaciones y la escasa fiabilidad son factores comunes que inhiben drásticamente la aceptabilidad de las aplicaciones de software de hoy en día. Por lo tanto, la calidad debe ser revisada con respecto a los requisitos basados en la fiabilidad, funcionalidad, rendimiento de las aplicaciones y el rendimiento del sistema. Rational Unified Process le ayuda en la planificación, diseño, implementación, ejecución y evaluación de estas pruebas. La evaluación de la calidad está integrada en el proceso, en todas las actividades, con la participación de todos los participantes, esto con el objetivo de obtener mediciones y

criterios, y no sean tratados como una ocurrencia tardía o una actividad separada realizada por un grupo independiente.

2.3.1.6. Control de cambios

La capacidad para gestionar el cambio nos brinda la certeza de que cada cambio es aceptable, y ser capaz de seguir los cambios es esencial en un entorno en el que el cambio es inevitable. El proceso describe cómo controlar, rastrear y monitorear los cambios para permitir el desarrollo iterativo con éxito. También le guía en cómo establecer espacios de trabajo seguros para cada desarrollador, proporcionando el aislamiento de los cambios realizados en otras áreas de trabajo y mediante el control de los cambios de todos los artefactos de software (por ejemplo, modelos, códigos, documentos, etc.).

Según lo que establece el RUP (De Carlo et al., 2004) los elementos del RUP son:

- **Actividades**, son los procesos que se llegan a determinar en cada iteración. En concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.
- **Roles**, definen el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas. Las

responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos.

- **Artefactos**, son un producto, es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final. Un artefacto puede ser cualquiera de los siguientes (RUP, 2002): un documento, un modelo, y un elemento del modelo.

2.3.2. Dimensiones del RUP

El proceso del desarrollo de software puede describirse en dos dimensiones, o a lo largo de dos ejes, esto según la publicación de la compañía Rational (2001).

- El eje horizontal representa el tiempo y muestra los aspectos dinámicos del proceso siendo expresados en términos de ciclos, fases, iteraciones e hitos.
- El eje vertical representa los aspectos estáticos del proceso, cómo se describe en términos de actividades, artefactos, empleados y flujos de trabajo.

A continuación presentamos una imagen descriptiva de lo antes mencionado.

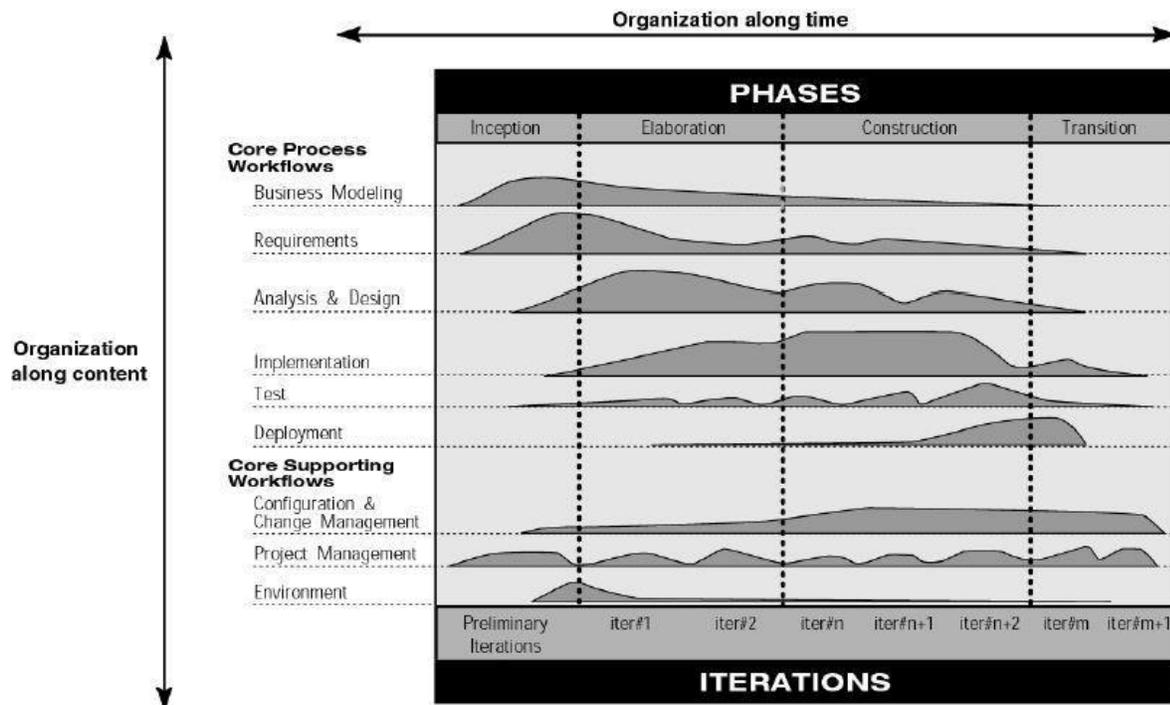


Figura 13: Modelo Iterativo. (Rational, 2001)

La Compañía Rational (2001) divide los proyectos en pequeños ciclos o iteraciones a través de cada una de las fases por las que pasa el proyecto, las cuales son establecidas claramente, cada una desarrollada en una o más iteraciones que ejecutan actividades definidas para cada flujo de trabajo en cualquier proceso de desarrollo. Concretamente el RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y las cuales se definen de acuerdo al nivel de madurez que alcanzan los productos que se van obteniendo en cada actividad ejecutada. La terminación de cada fase ocurre en el hito correspondiente a cada una, donde se evalúa que se hayan cumplido los objetivos de la fase en cuestión. Y desde la terminación de la fase de inicio se puede ya determinar la factibilidad tanto

operativa como económica del proyecto, la cual nos lleva a tomar la decisión de continuarlo o no realizarlo.

2.3.3. Fases del RUP

Esta es la organización dinámica del proceso a lo largo del tiempo. Según la publicación de la compañía Rational (2001) el proceso unificado (RUP) se repite a lo largo del mismo, en una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye en una versión del producto y cada ciclo de vida del software está dividido en cuatro fases:

- a) Fase de Inicio (Inception)
- b) Fase de Elaboración (Elaboration)
- c) Fase de Construcción (Construction)
- d) Fase de Transición (Transition)

Cada fase, al concluir, cuenta con un punto de control claramente definido que es conocido como hito, que normalmente es un conjunto de metas definidas.

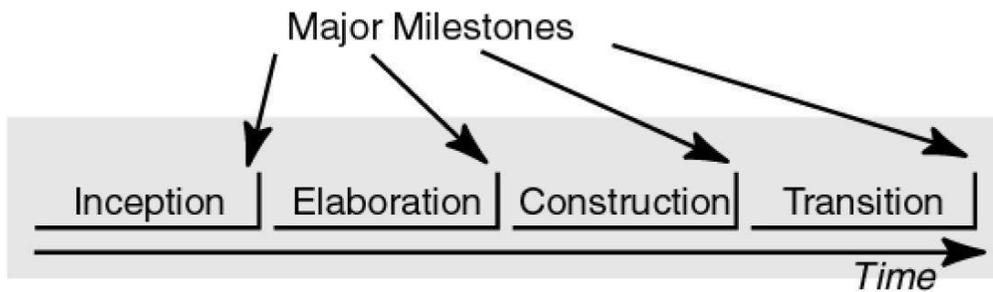


Figura 14: Las fases e hitos importantes en el proceso. (Rational, 2001)

A continuación se detalla cada fase y el propósito específico de cada una, esto según la publicación de la compañía Rational (2001).

2.3.3.1. Fase de Inicio

Durante esta fase se establece el caso de negocio para el sistema y se limita el alcance del proyecto. Para ello, se identifican todas las entidades externas, con las cuales el sistema interactuará (actores) y se define la naturaleza de esta interacción a alto nivel. Esto incluye identificar todos los casos de uso y describir los más significativos. El caso de negocio incluye criterios de éxito, riesgo, análisis y estimación de los recursos necesarios, así como, un plan preliminar de fases e iteraciones (plan grueso) que muestre las fechas de los principales puntos de control. El artefacto clave que sintetiza estos considerandos se llama: *Visión*.

Los objetivos de esta fase son:

- Establecer el ámbito de software y las condiciones de los límites del proyecto, incluidas una visión operativa, criterios de aceptación y lo que debe y no debe contener, el producto.
- Discriminar los casos de uso más importantes del sistema, los principales casos de ejemplo de las operaciones de los que dependerán las principales concesiones del diseño.
- Exhibir y tal vez demostrar, al menos una arquitectura posible contra alguno de los principales casos de ejemplo.
- Estimar el coste global y la planificación de todo el proyecto (y estimaciones más detalladas para la fase de elaboración)
- Estimar los riesgos potenciales (las causas de incertidumbre)
- Preparar el entorno de soporte para el proyecto.

El resultado de la fase es:

- Un documento panorámico: una visión general de los requerimientos esenciales del proyecto, características clave y principales exigencias.
- Un modelo de caso de uso inicial.
- Un glosario inicial del proyecto.
- Un caso de negocio inicial, que incluya contexto del negocio, criterios de éxito y presupuesto financiero.
- Una determinación inicial de riesgo.
- Un plan grueso del proyecto, que muestre fases e iteraciones.
- Un modelo del negocio, si fuera necesario.

- Si es posible un prototipo inicial

La imagen a continuación muestra el punto de control (hito) al término de esta fase:

Objetivos del ciclo de vida.

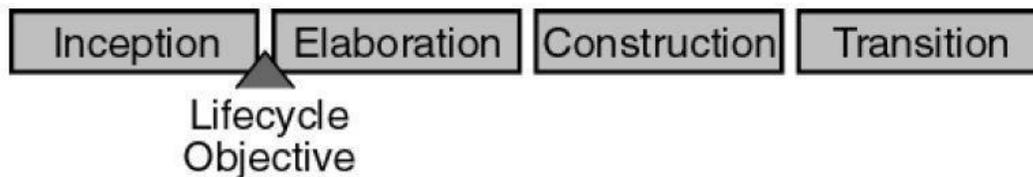


Figura 15: Hito - Objetivo del ciclo de vida. (Rational, 2001)

Los criterios de evaluación para esta fase de Inicio son:

- Acuerdo entre los participantes sobre la definición de alcance y costo/cronograma estimado.
- Comprensión de los requerimientos de alto nivel, evidenciados por la fidelidad a los casos de uso primarios.
- Credibilidad del costo/cronograma estimado, prioridades, riesgos y proceso de desarrollo.
- Una solución conceptual.

El proyecto puede ser cancelado o considerablemente repensado si no logra pasar a este hito.

2.3.3.2. Fase de Elaboración

El propósito de esta fase es analizar el dominio del problema, establecer una base de arquitectura sólida, desarrollar el plan del proyecto, y eliminar los mayores elementos de riesgo. Las decisiones sobre arquitectura deben ser hechas con comprensión del sistema completo, el alcance, la funcionalidad principal y los requerimientos no funcionales (conocidos como atributos de calidad).

Las actividades de esta fase garantizan que la arquitectura, los requerimientos y los planes están suficientemente estables, y el riesgo mitigado, como para poder determinar previsiblemente el costo y el cronograma para completar el desarrollo.

En esta fase se construye un prototipo arquitectónico, que incluye la implementación de los casos de uso más críticos y transversales. Este esfuerzo debería incluir, por lo menos, casos de uso identificados en la fase de conceptualización, los que exponen típicamente los principales riesgos técnicos del proyecto. Mientras un prototipo evolutivo de un componente producido con calidad es siempre la meta, esto no excluye el desarrollo de uno o más prototipos exploratorios, descartables para mitigar riesgos específicos, tales como negociaciones de diseño/requerimientos, estudio de factibilidad de componentes, o demostraciones a los inversionistas, clientes y usuarios finales.

El resultado de la fase es:

- Un modelo de casos de uso (completo por lo menos en un 80%), habiendo sido ya identificados todos los casos de uso y actores, y habiéndose desarrollado la especificación de la mayoría de los casos de uso.
- Requerimientos suplementarios que capturen los requerimientos no funcionales y cualquier requerimiento que no esté asociado a un caso de uso específico.
- Una descripción de la arquitectura de software.
- Un prototipo de arquitectura ejecutable.
- Una lista de riesgos revisada y el caso de negocio revisado.
- Un plan de desarrollo para todo el proyecto, incluyendo el plan grueso, que muestre iteraciones y criterios de evaluación para cada iteración.

La imagen a continuación muestra el punto de control (hito) al término de esta fase:
Arquitectura del ciclo de vida.

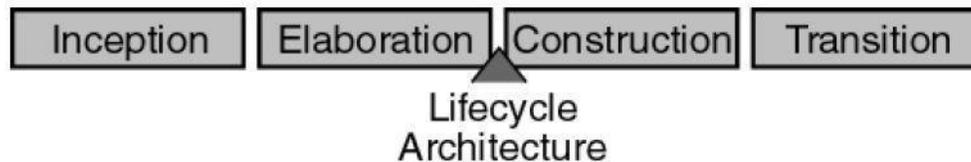


Figura 16: Hito - Arquitectura del ciclo de vida. (Rational, 2001)

Al final de la fase de elaboración está el segundo punto de control importante del proyecto: El punto de control de la arquitectura del ciclo de vida. En este punto se examinan los objetivos y alcances del sistema, la elección de la arquitectura y la resolución de los principales riesgos.

Los principales criterios de evaluación para la fase de elaboración incluyen las respuestas a estas preguntas:

- ¿Es estable la visión del producto?
- ¿Es estable la arquitectura?
- ¿Muestra el prototipo arquitectónico ejecutable que los elementos principales de riesgo han sido identificados y resueltos?
- ¿Está suficientemente detallado y afinado el plan para la primera iteración de construcción?
- ¿Está apoyado en bases de estimación creíbles?
- ¿Todos los participantes están de acuerdo en que se puede obtener la visión actual si se ejecuta el plan para desarrollar el sistema completo, en el contexto de la arquitectura actual?

2.3.3.3. Fase de Construcción

Durante la fase de construcción todos los componentes restantes y característica de la aplicación son desarrollados e integrados al producto, y todas sus funcionalidades son enteramente probadas. Esta fase es, en un sentido, un proceso de manufactura, en el cual se pone el acento en la administración de recursos y el control de las operaciones para optimizar costos, tiempos y calidad.

En este sentido, la atención se traslada del desarrollo de la propiedad intelectual durante la conceptualización y elaboración, al desarrollo de productos instalables durante la construcción y la transición.

En muchos proyectos es conveniente poder realizar actividades en paralelo. Estas actividades paralelas pueden acelerar significativamente la disponibilidad de versiones instalables; pueden también incrementar la complejidad de la administración de recursos y la sincronización del flujo de tareas. Una arquitectura robusta y un plan comprensible están altamente relacionados. En otras palabras, una de las cualidades críticas de la arquitectura es su facilidad de construcción.

Esta es una de las razones por las cuales el desarrollo balanceado de la arquitectura y el plan son acentuados durante la fase de elaboración.

El resultado de la fase de construcción es un producto listo para ser puesto en manos del usuario final. Consiste, como mínimo, en:

- El producto de software integrado en las plataformas adecuadas.
- Los manuales del usuario.
- Una descripción de la versión vigente.

La imagen a continuación muestra el punto de control (hito) al término de esta fase: Capacidad operativa inicial.

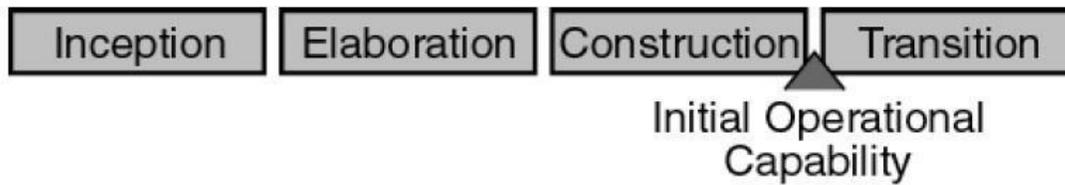


Figura 17: Hito - Capacidad operativa inicial. (Rational, 2001)

Al final de la fase de construcción está el tercer punto de control del proyecto: capacidad operativa inicial. En este punto se decide si el software, los lugares y los usuarios están listos para operar sin exponer el proyecto a altos riesgos. Esta versión es llamada a menudo versión beta.

Los criterios de evaluación para la fase de construcción incluyen la respuesta a estas preguntas:

- ¿Es este release del producto, lo suficientemente maduro y estable para ser instalado en la comunidad usuaria?
- ¿Están todos los recursos listos para la transición hacia la comunidad usuaria?

2.3.3.4. Fase de Transición

El propósito de esta fase es transferir el sistema a la comunidad usuaria. Una vez que el sistema fue entregado al usuario final, habitualmente surgen cuestiones que requieren desarrollo de nuevas versiones, corrección de ciertos problemas o conclusión de facilidades que habían sido pospuestas.

Se ingresa en la fase de transición cuando un release está lo suficientemente maduro como para ser instalado en el dominio del usuario final. Esto requiere, típicamente, que algún subconjunto utilizable del sistema haya sido completado en un aceptable nivel de calidad y que la documentación del usuario esté disponible, de modo que la transición al usuario permita obtener resultados positivos para todas las partes.

Esto incluye:

- Beta testing para validar el nuevo sistema contra las expectativas del usuario.
- Operación paralela con un sistema heredado que está siendo reemplazado.
- Conversión de las bases de datos operacionales.
- Entrenamiento de usuarios y del equipo de mantenimiento.

La fase de transición se centra en las actividades requeridas para poner el software en manos de los usuarios. Típicamente, incluye varias iteraciones, como versiones beta, versiones de disponibilidad general, así como, reparación de errores y versiones de mejoramiento. Se consume considerable esfuerzo en desarrollar la documentación orientada al usuario, entrenamiento de usuarios, apoyo a los usuarios durante la utilización inicial del sistema, y reaccionar ante la retroalimentación del usuario. En este punto del ciclo de vida, sin embargo, la retroalimentación del usuario debe ser limitada a cuestiones de sintonía, configuración, instalación y utilizabilidad.

Los objetivos primarios de la fase de transición incluyen:

- Obtener la autonomía del usuario.
- Obtener el acuerdo de los participantes de que la instalación ha sido completa y que es consistente con los criterios de evaluación de la visión.
- Perfeccionar el producto final.

La imagen a continuación muestra el punto de control (hito) al término de esta fase: Capacidad operativa inicial.

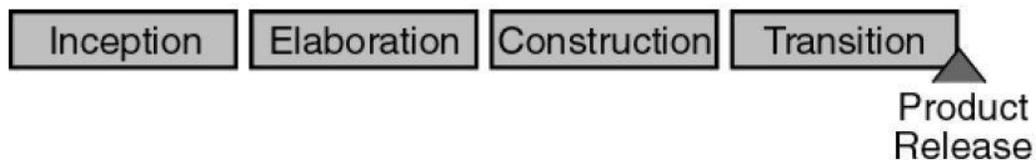


Figura 18: Hito - Release del sistema. (Rational, 2001)

Al final de la fase de transición está el cuarto punto importante de control del proyecto: release del sistema. En este punto, se decide si los objetivos han sido alcanzados, y si se podría comenzar otro ciclo de desarrollo. En algunos casos este punto de control puede coincidir con el final de la fase de conceptualización para el próximo ciclo.

Los criterios de evaluación primarios para la fase de transición implican la respuesta a la siguiente pregunta: ¿Está satisfecho el usuario?

2.3.4. Iteraciones del RUP

Según la documentación en Rational (2001) cada fase en el Rational Unified Process puede subdividirse en iteraciones. Una iteración es un completo bucle de desarrollo dando como resultado una liberación (interna o externa) de un producto ejecutable, un subconjunto del producto final en fase de desarrollo, que crece de forma incremental a partir de una iteración a otra para convertirse en el sistema final.

Beneficios de un enfoque iterativo

En comparación con el proceso tradicional de cascada, el proceso iterativo tiene las siguientes ventajas:

- Los riesgos se mitigan
- El cambio es más manejable
- Mayor nivel de reutilización
- El equipo del proyecto puede aprender en el camino
- Mejor calidad general

2.3.5. Estructura Estática del RUP

Según IBM Rational Software (2006) un proceso describe quién está haciendo qué y cómo. El Rational Unified Process se representa usando 3 elementos de modelado primarios:

- **Roles - el quién**
- **Tareas - el cómo**
- **Productos de Trabajo - el qué**

2.3.5.1. Roles

El término ROL es usado para denominar a los puestos a los cuales se pueden asignar personas dentro de un proyecto de desarrollo de software. Cada ROL es responsable de un conjunto completo de tareas, como las tareas necesarias para el análisis de procesos de negocio. Una persona puede asumir las responsabilidades de varios roles.

Un ROL será responsable de hacer una serie de tareas y será responsable de la confección de una serie de Productos de Trabajo.

Es como un “sombrero” que la persona usa durante el proyecto:

Una persona puede tener varios sombreros

Es el rol que desempeña en un momento dado

Responsabilidades:

- Hacer una serie de tareas.
- Ser el responsable de una serie de productos de trabajo.

2.3.5.2. Tareas

Una tarea describe una unidad de trabajo. Cada tarea es realizada por funciones específicas.

Tareas por lo general, se definen como una serie de pasos que implican la creación o actualización de uno o más artefactos.

Algunos ejemplos de tareas son:

- Encontrar actores y casos de uso: Una tarea que realiza el papel de analista de sistemas, la cual consiste en identificar los requisitos funcionales de alto nivel en términos de actores y casos de uso.
- Describir la distribución: Una tarea que realiza el papel de arquitecto de software, la cual consiste en describir la distribución de software a través de múltiples procesadores.

2.3.5.3. Productos de Trabajo

Son piezas de información que son producidas, modificadas o usadas por un proceso. Son el resultado parcial o final, que es producido y usado durante el proyecto.

Un producto de trabajo es una abstracción general que representa un resultado del proceso.

Entre éstos se tiene a los siguientes

- **Artefactos:** Son productos de trabajo tangibles y bien definidos, que las tareas consumen, producen o modifican. Pueden estar compuestos por otros artefactos. En general, los artefactos no son documentos. La forma más eficaz y pragmática de gestionar los artefactos del proyecto es mantenerlos dentro de la herramienta adecuada que se haya utilizado para crearlos y gestionarlos.

- Ejemplos:
 - ✓ Una especificación de caso de uso almacenada en Microsoft Word
 - ✓ Un modelo de diseño almacenado en Rational Software Architect.
 - ✓ Un plan de proyecto almacenado en Microsoft Project

- **Entregable:** Es un producto de trabajo que proporciona una descripción y una definición del empaquetado de otros productos de trabajo y puede entregarse a una parte interna o externa. Por lo tanto, un entregable agrega otros productos de trabajo y se utilizan para

representar una salida de un proceso que tiene valor, material o información de otro tipo para un cliente u otra parte interesada.

- **Resultado:** Describe productos de trabajo intangibles que son un resultado o un estado, como un servidor instalado o una red optimizada. Como la aparición de un resultado suele documentarse de manera informal (por ejemplo, mediante actas o memorándums), los resultados también pueden utilizarse para describir productos de trabajo que no estén definidos formalmente.

Las tareas tienen productos de trabajo de entrada y de salida. Los roles utilizan productos de trabajo para realizar tareas y producir otros productos de trabajo mientras realizan las tareas.

Un Producto de trabajo puede ser un documento, un modelo o un elemento de modelo. Se encuentran agrupados por cada disciplina:

- ✓ Business Modeling Set
- ✓ Requirements Set
- ✓ Analysis & Design Set
- ✓ Implementation Set
- ✓ Test Set
- ✓ Deployment Set
- ✓ Project Management Set
- ✓ Configuration & Change Management Set
- ✓ Environment Set

2.3.6. Disciplinas y Workflows

A continuación presentamos los conceptos de disciplinas y workflows según IBM Rational Software (2006).

2.3.6.1. Disciplinas

El término es una categorización de tareas basada en las similitudes de las preocupaciones y la cooperación del esfuerzo de trabajo.

La agrupación de tareas en disciplinas es fundamentalmente una ayuda para comprender el proyecto desde una perspectiva de cascada tradicional. Aunque es más común realizar tareas simultáneamente en varias disciplinas (por ejemplo, ciertas tareas de requisito se realizan en estrecha coordinación con las tareas de análisis y diseño), la separación de estas tareas en disciplinas diferentes supone sencillamente una forma eficaz de organizar el contenido, lo que facilita la comprensión.

El RUP define 9 disciplinas:

- Modelado empresarial o de Negocio
- Requisitos
- Análisis y diseño
- Implementación
- Prueba
- Despliegue

- Gestión de cambios y configuración
- Gestión de proyectos
- Entorno

2.3.6.2. Workflows

- **Core Workflows:** Representan una división de los workers y actividades en agrupaciones lógicas.

Existen seis workflows de ingeniería de procesos

- ✓ Modelado del Negocio
- ✓ Requerimientos
- ✓ Análisis y Diseño
- ✓ Implementación
- ✓ Test
- ✓ Desarrollo

Además de tres workflows de apoyo o soporte:

- ✓ Configuración y Administración de Cambios
- ✓ Administración del Proyecto
- ✓ Ambiente

A continuación se describirán los workflow:

✓ **Workflow de Modelado de Negocio**

Tiene las siguientes características:

- Describe los procesos de una organización
- Consiste de actores del negocio y casos de uso del negocio
- El resultado es un modelo de casos de uso del negocio

Detalles de Workflows:

- Evaluar la Organización o Negocio.
- Describir el proceso actual.
- Definir el Negocio.
- Explorar la automatización de procesos.
- Desarrolla el modelo de Dominio.

✓ **Workflow de Requerimientos**

Describe la secuencia de actividades para realizar el proceso de ingeniería de requerimientos del proyecto. Este proceso abarca desde la captura y formulación de requerimientos, su seguimiento durante el desarrollo, la administración de los cambios y su impacto, hasta la validación y aceptación final. El resultado es un modelo de casos de uso.

Detalle de Workflows:

- Analizar el problema.
- Conocer las necesidades de los stakeholder.
- Definir el sistema.
- Gestionar el ámbito del sistema.
- Gestionar cambios de requisitos.
- Perfeccionar la definición del sistema.

✓ **Workflow de Analisis y Diseño**

Muestran como se verá el sistema en la implementación. El modelo de casos de uso es base para el diseño.

Las finalidades de análisis y diseño son:

- Transformar los requisitos en un diseño del sistema en creación
- Evolucionar una arquitectura sólida para el sistema
- Adaptar el diseño para que se ajuste al entorno de implementación

Detalle de Workflows:

- Definir una arquitectura candidata.
- Realizar la síntesis arquitectónica.
- Perfeccionar la arquitectura.
- Identificación de servicios.
- Analizar el comportamiento.
- Componentes de diseño.
- Diseñar la base de datos.
- Especificación de servicios.

✓ **Workflow de Implementación**

La finalidad de la implementación es:

- Definir la organización del código, en términos de los subsistemas de implementación organizados en capas.

- Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros)
- Probar y desarrollar componentes como unidades.
- Integrar los resultados producidos por los implementadores individuales (o equipos) en un sistema ejecutable.

La disciplina de implementación limita su ámbito a la forma en que las clases individuales deben pasar por la prueba de unidad. La prueba del sistema y la de integración, se describen en la disciplina de prueba.

✓ **Workflow de Prueba**

La prueba de disciplina actúa como proveedor de servicios de otras disciplinas en muchos aspectos. Las pruebas se centran, principalmente, en la evaluación o la valoración de la calidad del producto, hecho que se lleva a cabo mediante las prácticas:

- Buscar y documentar los defectos en la calidad del software.
- Opinar sobre la calidad percibida del software.
- Validar y demostrar las suposiciones efectuadas en las especificaciones de diseño y requisitos con una demostración concreta.
- Validar que el producto de software funciona, según lo diseñado.
- Validar que los requisitos se han implementado de forma adecuada.

Existe una interesante diferencia entre Probar y las otras disciplinas de RUP - básicamente, las tareas de Probar son encontrar y exponer los puntos débiles del producto de software. Es interesante porque, para obtener mayores beneficios, necesita una filosofía general diferente de la que se utiliza en las disciplinas de Requisitos, Análisis & diseño e Implementación. Una sutil diferencia es que estas tres disciplinas se centran en la completitud, mientras que la Prueba se centra en la falta de ésta.

Un buen esfuerzo de prueba se orienta con preguntas como las siguientes:

- ¿Cómo puede fallar este software?
- ¿En qué posibles situaciones podría fallar este software de forma previsible?

✓ **Workflow de Despliegue**

La disciplina de despliegue describe tres modalidades de despliegue del producto:

- Instalación personalizada.
- Oferta de producto "comercializable" .
- Acceso al software a través de Internet.

En cada instancia, se enfatiza la prueba del producto en el lugar de desarrollo, seguido por una prueba de versión beta, antes de que el producto se publique para los clientes.

Aunque las actividades de despliegue tienen su punto álgido en la fase de transición, algunas de las actividades se dan en fases más tempranas para planificar y preparar el despliegue.

✓ **Workflow de Configuración y Administración de Cambios**

Un sistema de Gestión de Configuración es esencial para controlar los numerosos productos de trabajo que producen las muchas personas que trabajan en un proyecto común. El control ayuda a evitar confusiones costosas y garantiza que los productos de trabajo resultantes sean conflictivos debido a algún problema de los tipos siguientes:

- Actualización simultánea.
- Notificación limitada.
- Versiones múltiples.

Un sistema de CM es útil para la gestión de las múltiples variantes de evolución de los sistemas de software, para realizar el seguimiento de las versiones que se utilizan en determinadas compilaciones de software, llevar a cabo compilaciones de programas individuales o de releases completos, de acuerdo con especificaciones de versión definidas por el usuario y la aplicación de las políticas de desarrollo específicas del sitio.

✓ **Workflow de Gestión de Proyectos**

Esta disciplina se centra en la planificación del proyecto, la gestión del riesgo, la supervisión del progreso y la métrica.

El objetivo de esta sección es facilitar la tarea al proporcionar algún contexto para la gestión de proyectos. No se trata de una receta para el éxito, sino que presenta una forma de gestión para el proyecto que mejorará notablemente la posibilidad de entregar un software satisfactorio.

Características:

- Provee una estructura para administrar los proyectos de software.
- Proporciona guías prácticas para planificar, proveer de personal, ejecutar y monitorear proyectos.
- Provee una estructura para administrar riesgos.
- Sin embargo, esta disciplina de Rational Unified Process (RUP) no trata de abordar todos los aspectos de la gestión de proyectos; por ejemplo, no cubre aspectos como los siguientes:
 - Gestión del personal: contratación, formación, adiestramiento.
 - Gestión del presupuesto: definición, asignación, etc.
 - Gestión de los contratos con los proveedores y los clientes.

✓ **Workflow de Entorno (Ambiente)**

La finalidad de la disciplina de entorno es proporcionar un entorno de desarrollo de software que dé soporte al equipo de desarrollo. Además, proporciona el entorno de soporte para un proyecto. De esa forma, da soporte a todas las otras disciplinas.

2.4. Proceso de ventas de DARUNE SAC

En este apartado describiremos el proceso de ventas actual, también llamado AS-IS, de la empresa DARUNE SAC, para una mayor información se listarán los procesos de la empresa, estos los listaremos por su categoría dentro de la organización los cuales son:

2.4.1. Procesos Estratégicos

Los procesos estratégicos son aquellos establecidos por la Alta Dirección y definen cómo opera el negocio y cómo se crea valor para el cliente / usuario y para la organización.

Soportan la toma de decisiones sobre planificación, estrategias y mejoras en la organización.

Proporcionan directrices, límites de actuación al resto de los procesos, como tales se identificó al siguiente proceso:

- Planificación Estratégica

2.4.2. Procesos Principales

Son aquellos directamente ligados a los servicios que se prestan, y por tanto, orientados al cliente/usuario y a requisitos. Como consecuencia, su resultado es percibido directamente por el cliente/usuario (se centran en aportarle valor).

A modo de resumen, los procesos claves constituyen la secuencia de valor añadido del servicio desde la comprensión de las necesidades y expectativas del cliente /usuario hasta la prestación del servicio, siendo su objetivo final la satisfacción del cliente /usuario, como tales se identificaron a los siguientes procesos:

- Compras
- Ventas
- Distribución
- Cobranzas

2.4.3. Procesos de Apoyo o Soporte

Los procesos de apoyo son los que sirven de soporte a los procesos claves. Sin ellos no serían posibles los procesos claves ni los estratégicos. Estos procesos son, en muchos casos, determinantes para que puedan conseguirse los objetivos de los procesos dirigidos a cubrir las necesidades y expectativas de los clientes / usuarios., como tales se identificó al siguiente proceso:

- Finanzas
- Gestión de RRHH
- Control de stock y aprovisionamiento

2.4.4. Descripción del proceso de ventas de DARUNE SAC

A continuación se muestra el diagrama de flujo del proceso de ventas de la empresa DARUNE SAC, dicho diagrama fue realizado con el fin de tener una visión gráfica del proceso y de poder apreciar de una forma más a detalle la relación de las actividades que se llevan a cabo dentro de dicho proceso. Para esta realización fue necesaria la colaboración del Jefe del Área de Ventas.

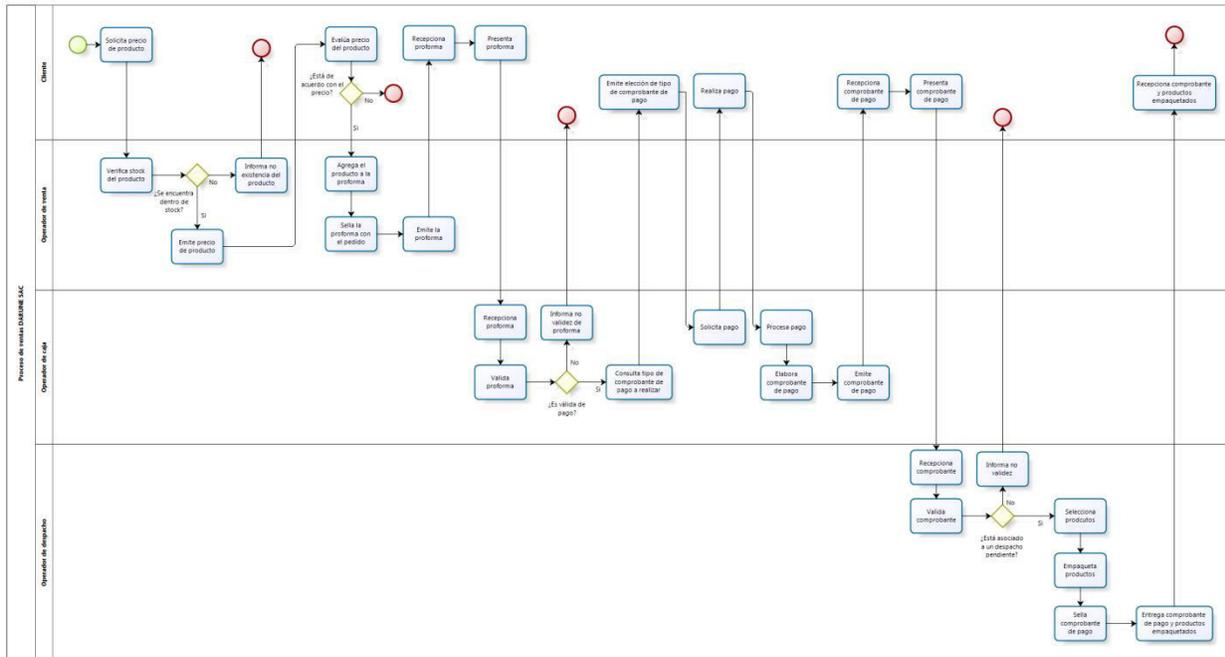


Figura 19: Proceso de ventas de DARUNE SAC - AS IS. ELaboración Propia.

2.4.1.1. Ficha de Proceso

A continuación se muestra la ficha del proceso de ventas de DARUNE SAC.

FICHA DEL PROCESO	Código: PCDV - Versión 1.0
Fecha de elaboración: 7/10/2015	Fecha de aprobación: 8/12/2015
Nombre del proceso	Proceso de ventas
Descripción	Este proceso se lleva a cabo debido a la interacción del personal interno de la organización y el cliente, esto motivado por la necesidad de adquirir algún producto de la empresa comercial DARUNE SAC por este último.
Misión/objetivo	Realizar ofertas oportunas y de calidad mediante una negociación ganar – ganar, así como entablar relaciones comerciales permanentes de entera satisfacción para nuestros clientes en toda la etapa del proceso.
Responsable	Jefe del Área de Ventas. Sra. Nancy Tito Ponce
Destinatario	Cliente / Gerente General

Inicio/Fin	El inicio del proceso está marcado por la solicitud de oferta de un potencial cliente, el proceso termina con la entrega del producto al cliente.
Entradas	<ul style="list-style-type: none"> – Solicitud de ofertas – Compras e Inventarios
Salidas	<ul style="list-style-type: none"> – Comprobante de pago – Proformas / Cotizaciones – Pedido
Indicadores	<ul style="list-style-type: none"> – Tiempo medio empleado en consultar el precio de un producto. – Tiempo medio empleado para la emisión de información a la sección de despacho. – Tiempo medio empleado en la emisión de una proforma o cotización a un potencial cliente.
Registros	<ul style="list-style-type: none"> – Informes mensuales
Procedimientos asociados	Procedimiento de evaluación de crédito.
Aplicación informática	No aplica

Tabla 2: Ficha del Proceso de Ventas de DARUNE SAC. Elaboración Propia

2.4.1.2. Indicadores del Proceso

Según la UCA (2007), la medición es un requisito de la gestión, lo que no se mide no se puede gestionar, y por lo tanto no se puede mejorar.

Por lo que a continuación se detallan los indicadores de la tabla anterior, cabe señalar que según la publicación de la UCA (2007), un indicador es una magnitud asociada a una característica (del resultado, del proceso, de las actividades, de la estructura, etc.) que permite, a través de su medición en periodos sucesivos y por comparación, evaluar periódicamente dicha característica y verificar el cumplimiento de los objetivos establecidos.

- **Tiempo medio empleado en consultar el precio de un producto**

Código	IPCDV1
Nombre	Tiempo medio empleado en consultar el precio de un producto.
Descripción	Mide el tiempo en el cual un potencial cliente obtiene la información del precio de un producto específico.
Unidad de medida	Minutos
Cálculo	$(\sum \text{Tiempo empleado en consultar el precio de un producto durante 30 días}) / 30$
Fuente	Área de ventas

Grado de fiabilidad	Fiabilidad total
Responsable	Jefe del Área de Ventas. Sra. Nancy Tito Ponce
Periodicidad	Mensual (30 días)
Versión	1.0
Fecha de realización de la ficha	8/10/2015

- **Tiempo medio empleado para la emisión de información a la sección de despacho**

Código	IPCDV3
Nombre	Tiempo medio empleado para la emisión de información a la sección de despacho.
Descripción	Mide el tiempo en el cual el operador de caja emite la información para el despacho al área de despacho.
Unidad de medida	Minutos
Cálculo	$(\sum \text{Tiempo empleado para la emisión de información del operador de caja a la sección de despacho durante 30 días}) / 30$
Fuente	Área de ventas
Grado de fiabilidad	Fiabilidad total

Responsable	Jefe del Área de Ventas. Sra. Nancy Tito Ponce
Periodicidad	Mensual (30 días)
Versión	1.0
Fecha de realización de la ficha	8/10/2015

- **Tiempo medio empleado en la emisión de una proforma o cotización a un potencial cliente**

Código	IPCDV4
Nombre	Tiempo medio empleado en la emisión de una proforma o cotización a un potencial cliente.
Descripción	Mide el tiempo en el cual es emitida una proforma o cotización a un cliente potencial.
Unidad de medida	Minutos
Cálculo	$(\sum \text{Tiempo empleado en emitir una proforma o cotización a un potencial cliente durante 30 días}) / 30$
Fuente	Área de ventas
Grado de fiabilidad	Fiabilidad total
Responsable	Jefe del Área de Ventas. Sra. Nancy Tito Ponce
Periodicidad	Mensual (30 días)

Versión	1.0
Fecha de realización de la ficha	8/10/2015

Capítulo III. ESTADO DEL ARTE METODOLÓGICO

3.1. Metodologías para el desarrollo de la solución

Para realizar el desarrollo del presente trabajo se han tomado en cuenta los trabajos de investigación de Johanna Elizabeth Rodríguez Torres y Carlos Alfredo Becerra Rodríguez, en los cuales definen una arquitectura particular para poder desarrollar sus soluciones, las cuales considero afines a la solución propuesta de este trabajo de investigación.

A continuación mostramos los principales puntos de los trabajos mencionados, los cuales nos proporcionaran información detallada acerca de la arquitectura a desarrollar para nuestro caso de estudio.

3.1.1. Análisis, Diseño e Implementación de un sistema de información para una tienda de ropa con enfoque al segmento juvenil

Este trabajo de investigación fue desarrollado por Johanna Elizabeth Rodríguez Torres, como Tesis para optar el Título de Ingeniero Informático en la Pontificia Universidad Católica del Perú, en Lima en mayo el año 2013.

A continuación se citara el resumen del trabajo de investigación mencionado, el cual nos dará un mayor panorama.

3.1.1.1. Resumen

En este trabajo de tesis se presenta el desarrollo de un sistema de información que permite gestionar las ventas y el almacén de ventas, de esta manera se ayuda a organizar, controlar y administrar los productos con los que cuenta la empresa q fue tomada como modelo, automatizando sus actividades primarias y mejorando la interacción con sus clientes. El sistema presenta los siguientes módulos: El módulo de ventas, El módulo de inventario de ventas y el módulo de catálogo en línea.

Para lograr los objetivos del presente proyecto, se propone formalizar las reglas del negocio, la elaboración de un prototipo de la posible solución, la definición de la arquitectura y la validación del sistema.

A continuación se citaran los puntos relevantes de este trabajo de investigación para nuestra solución, los cuales son los que detallan la metodología y las herramientas utilizadas en el trabajo mencionado.

3.1.1.2. Análisis

- **Metodología aplicada al desarrollo de la solución**

Para la gestión de proyecto se utilizarán las buenas prácticas para la gestión de proyectos definida por el PMI, delimitando los grupos de procesos de iniciación, planificación, ejecución, monitoreo y cierre del proyecto.

Cabe mencionar que no se utilizarán todas las áreas de conocimiento definidas en el PMBOK para el proyecto como el proceso de gestión de los recursos humanos, dado que finalmente el éste será ejecutado por una sola persona, así como el proceso de gestión de las adquisiciones, debido a que no se requerirá ningún equipo adicional además del software libre. La descripción de las herramientas utilizadas en cada fase de RUP, se encuentran en la tabla que se presenta a continuación.

Fases	Descripción
Fase de Inicio	Se define como actividad inicial la formulación del alcance del proyecto, en la cual se observa el flujo del negocio para recopilar información sobre cómo se está trabajando con este tipo d herramientas en la actualidad. Adicionalmente, se recopilan los requerimientos mediante entrevistas y trabajo de campo. Este trabajo se basará en entrevistas al personal de la empresa TSJ.
Fase de Planificación	Se definirán la arquitectura del sistema que se va a implementar, los componentes del sistema y las entidades involucradas.
Fase de Ejecución	Puesta en marcha de la codificación e

	integración con otros componentes. Se realizará en tres iteraciones. Una vez implementada la solución, se pondrá en marcha el plan de pruebas unitarias y de interconexión de la aplicación.
--	--

Tabla 3: Fases del RUP Aplicadas al Proyecto. Rodrigez, J. (2013)

Los artefactos generados por cada una de las fases detalladas anteriormente se detallan en la siguiente tabla:

Fases	Artefactos
Fase de Inicio	<ul style="list-style-type: none"> – Documento de visión – Plan del proyecto – Catálogo de requisitos
Fase de Planificación	<ul style="list-style-type: none"> – Especificación de requisitos del software. – Documento de la arquitectura – Documento de estándares de programación – Documento de la interfaz gráfica – Documento del análisis y diseño

Fase de Ejecución	<ul style="list-style-type: none"> – Plan de pruebas – Versión beta de la herramienta – Manuales de usuario
-------------------	--

Tabla 4: Artefactos del RUP aplicados al Proyecto. Rodriguez, J. (2013)

• **Herramientas que se utilizarán**

Las herramientas que se utilizaron en el desarrollo del presente proyecto de investigación fueron opensource. A continuación se presenta la siguiente tabla con dichas herramientas.

Herramienta	Descripción
Modelador de Base de Datos	Para la modelación de la base de datos se usa Erwin, herramienta propia del motor de base de datos para estas tareas de modelado. Una de las principales razones por la cual se escoge esta herramienta es la experiencia que se tiene en el uso de ella, lo que reduce la curva de aprendizaje, además de que es óptimo para trabajar con cualquier gestor de base de datos.
Motor de la base de datos	Se escogió el motor de base de datos MySQL por ser un gestor de licencia libre y

	también por la experiencia obtenida en su uso.
Lenguaje de programación	Se escogió el lenguaje PHP. Las razones principales por las cuales se escogió esta herramienta es que PHP permite hacer todo lo requerido para el sistema y por la experiencia que se tiene en el uso de ella.

Tabla 5: Artefactos del RUP aplicados al Proyecto. Rodrigez, J. (2013)

- **Identificación de requisitos**

La implementación del sistema de información requiere indicar los requerimientos funcionales y no funcionales que necesitan tomarse en consideración para definir una solución al problema planteado.

- ✓ **Catálogo de requisitos funcionales**

Este catálogo está constituido por las principales funcionalidades que permiten a la tesista organizar sus tareas. Para tal efecto, se realizó el levantamiento de la información directamente con la asistente y gerente general de la empresa TSJ.

Ac continuación presentaremos los requisitos para el módulo de venta:

Módulo de Ventas			
Número	Descripción del requerimiento	Prioridad	Tipo
VEN001	El sistema permite el mantenimiento de producto.	3	Exigible
VEN002	El sistema permite generar el comprobante de pago.	3	Deseable
VEN003	El sistema permite el mantenimiento de las promociones.	3	Exigible
VEN004	El sistema permite asociar y desasociar las promociones y los productos.	3	Deseable
VEN005	El sistema permite el mantenimiento de proveedores.	3	Exigible
VEN006	El sistema permite asociar y desasociar proveedores a los productos.	3	Exigible
VEN007	El sistema permitirá consultar documentos de pago.	3	Deseable
VEN009	El sistema permite registrar el pago en efectivo o con tarjeta	3	Exigible
VEN010	El sistema permite el pago de los productos mediante soles o dólares.	3	Exigible
VEN011	El sistema permite la apertura y cierre de caja.	3	Deseable
VEN013	El sistema permite imprimir el documento la venta realizada.	3	Deseable
VEN018	El sistema permite emitir promociones a través de la web para un grupo de clientes seleccionados.	3	Deseable

Tabla 6: Requisitos del Módulo de Ventas. Rodrigez, J. (2013)

- **Análisis de la solución**

- ✓ **Procesos de Negocio**

En esta sección se detallan las principales funcionalidades que se desarrollan en la empresa TSJ. Para tal efecto, se realizó el levantamiento de la información directamente con los empleados del negocio. Adicionalmente se explica un flujo que se tomará en cuenta para el desarrollo del sistema TSJ que hace que se automaticen tareas desarrolladas al día a día.

- **Módulo de Ventas**

Este proceso consta de dos perspectivas. La venta desde una tienda específica mostrada en la figura siguiente, y la tienda cuando un cliente ya ha reservado un producto por la web mostrado en la figura subsiguiente.

Cuando se realiza la venta por medio de las tiendas, se debe registrar al cliente para que se establezca un contacto y una relación. De esta forma se registran sus productos preferidos y se le puede mandar información relacionada. Luego de registrarlo se genera la orden de venta con los productos que se encuentren en la tienda, para que se finalice la venta con la generación del documento de venta, en todos los casos solo boleta. En casos extremos que el cliente no desee llevarse la mercadería se procede a realizar una nota de crédito y se realiza la devolución de la o las prendas. Si bien es cierto que los

productos se pueden reservar por la web, tiene un plazo, para que no afecte al proceso de ventas, Este mecanismo solamente funciona con clientes preferenciales que ya han realizado reservas con anterioridad, y son clientes frecuentes.

Estos clientes recibirán un código de reserva, a su correo y tendrá una duración de 1 día, luego de ese plazo la reserva será eliminada. Paralelamente el administrador de la tienda recibirá un correo que le especifique la reserva, para que la mercadería este en tienda en el momento que el cliente se acerque a solicitar el producto.

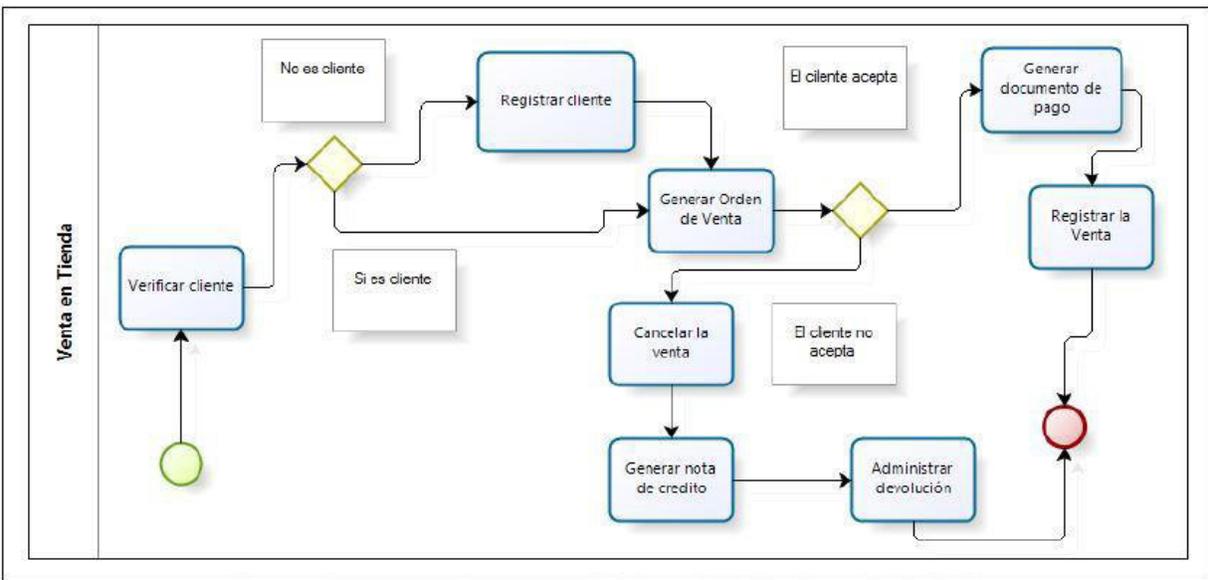


Figura 20: Venta en Tienda. Rodrigez, J. (2013)

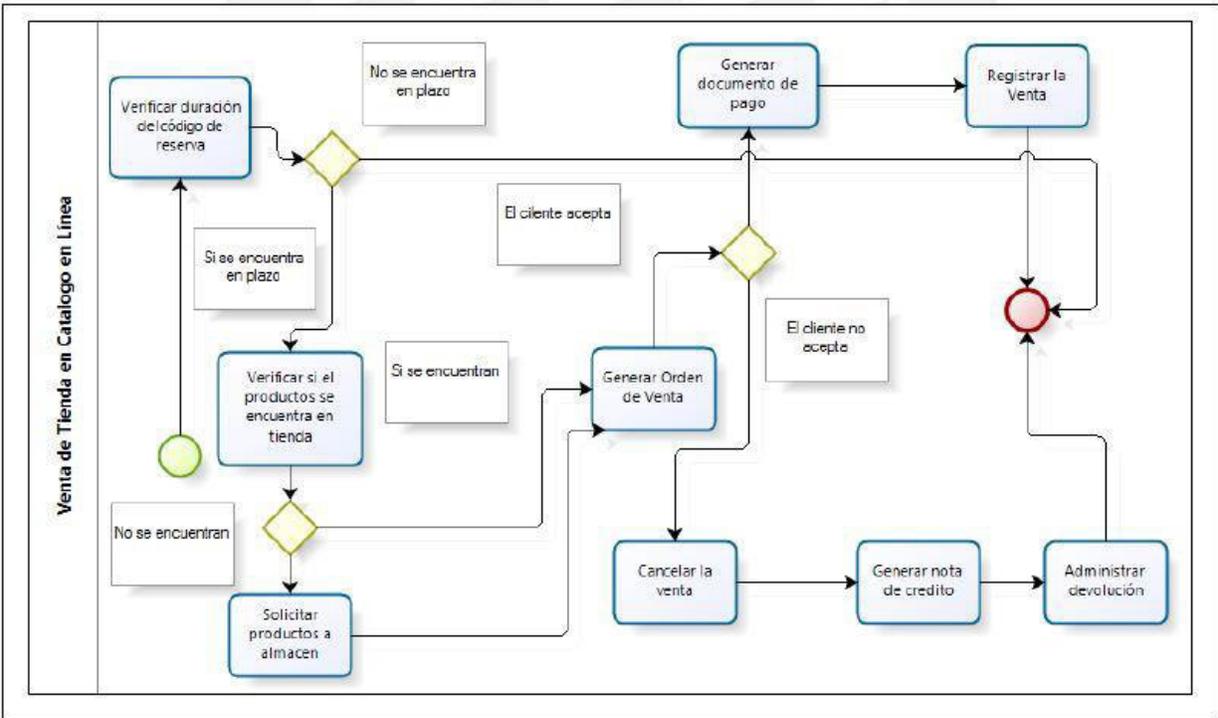


Figura 21: Venta en Tienda – Catálogo en Línea. Rodrigez, J. (2013)

✓ **Catálogo de Actores**

En la siguiente figura se puede visualizar el gráfico jerárquico de los actores que intervienen en el proyecto.

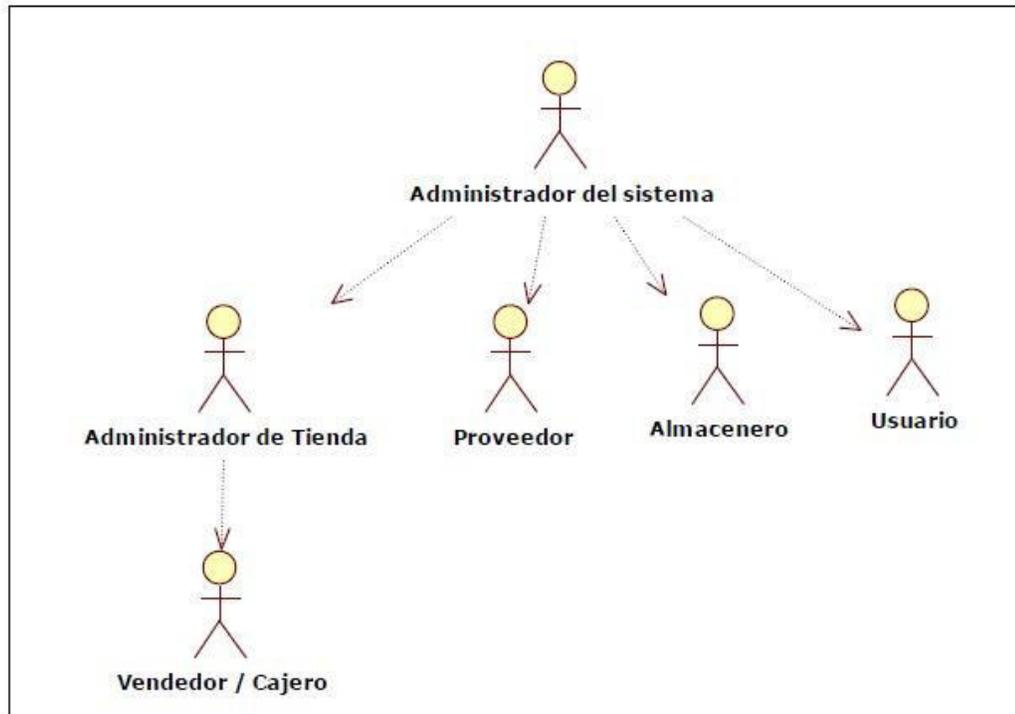


Figura 22: Catalogo de Actores. Rodrigez, J. (2013)

- **Administrador del sistema:** Encargado de generar permisos y restricciones del sistema por autorización del administrador de las tiendas, además de configurar los datos del sistema.
- **Administrador de la Tienda:** Responsable de administrar los datos de los empleados de las tiendas, cuadro de caja, control de ingreso y salida de productos del sistema, actualización del catálogo en línea. Responsable de la administración de ventas de productos, establecer promociones de acuerdo al tipo de cliente.

- **Vendedor/Cajero:** Responsable de interactuar con el cliente, registra las ventas (productos), ya sea al contado o con las tarjetas de débito o crédito, así mismo informar al cliente de las promociones existentes.
- **Proveedores:** Responsable de abastecer a las tiendas con los productos exclusivos.
- **Almacenero:** Responsable de la distribución de los productos a las tiendas y viceversa. De la correcta ubicación de los productos en el almacén ordenados por estacionalidad.

✓ **Casos de Uso por Paquetes**

A continuación se muestra una imagen con el paquete del Proceso de Ventas

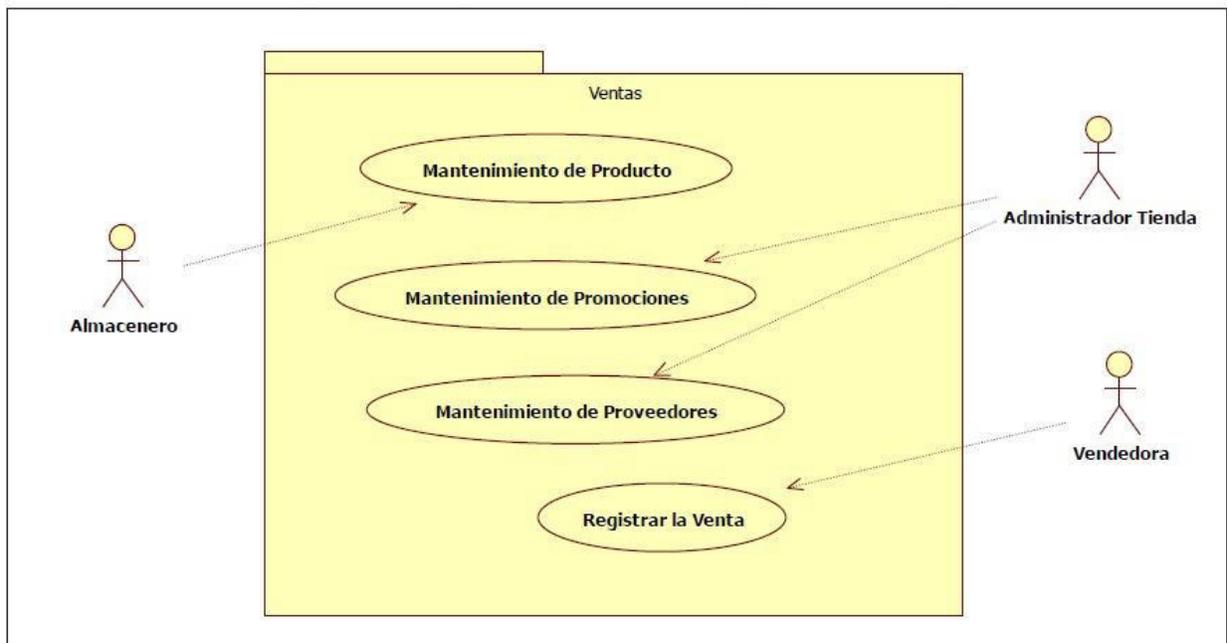


Figura 23: Paquetes de Ventas. Rodriguez, J. (2013)

✓ Especificación de Requisitos del Sistema

Las especificaciones de todos los casos de uso presentados se encuentran en el anexo “Especificación de Requisitos del Sistema”. A continuación se detalla el caso de uso Registrar comentario de un producto en el catálogo.

Registrar comentario de producto en catalogo	
Descripción:	Consiste en que tanto personas registradas, puedan ser clientes pasivos de tal manera que den a conocer si algún producto les atrae.
Actores:	Público en general
Precondición:	El usuario del sistema debe haber iniciado sesión. El usuario del sistema debe contar con los permisos necesarios para realizar esta tarea.
Flujo principal: Registrar comentario de producto en catálogo	
<ol style="list-style-type: none">1. El usuario selecciona del menú nueva colección una opción y luego selecciona el producto. El sistema muestra la imagen del producto expandida al lado derecho y en la parte inferior un cuadro para ingresar sus comentarios.2. El usuario ingresa un breve comentario y selecciona la opción Guardar.3. El sistema muestra un mensaje de confirmación “¿Seguro que desea guardar?”.<ol style="list-style-type: none">3.1. Si el usuario acepta, el sistema verifica los datos ingresados<ol style="list-style-type: none">3.1.1. Si el registro se realiza satisfactoriamente, el sistema muestra el comentario automáticamente en la parte inferior de la imagen del producto.3.1.2. Si el registro no se realiza satisfactoriamente, se muestra los errores que la causaron.	

<p>3.2. Si el usuario no acepta, se repite el proceso a partir del paso 4.</p> <p>4. El usuario selecciona la opción Regresar.</p> <p>5. El sistema retorna al menú nueva colección.</p> <p>6. Los pasos del 1 al 4 son repetidos para comentario que el usuario desee registrar. Cuando el usuario haya finalizado de registrar los comentarios, el caso de uso termina.</p>	
<p>Post condición:</p>	<p>Se registró correctamente el comentario del producto.</p>

3.1.1.3. Diseño

En este capítulo tiene la finalidad de presentar la estructura interna del sistema, mostrando a través de los diagramas análisis (diagramas de clases) como es que van a interactuar los diferentes elementos que participan en los procesos del sistema así como sus respectivas características. Se presentaran dichos diagramas tanto de manera general como para cada uno de los módulos con los cuales se está trabajando.

Adicionalmente se define la arquitectura que se utilizó para implementar la solución y se brinda información sobre las capas en las que se dividirá la solución. Asimismo, se definen los criterios para elaborar el diseño de las interfaces de las diversas pantallas de la aplicación.

- **Diagrama de Clases de Análisis**

A continuación se presentan el diagrama del proceso de ventas, esto debido a que es de interés para este trabajo de investigación.

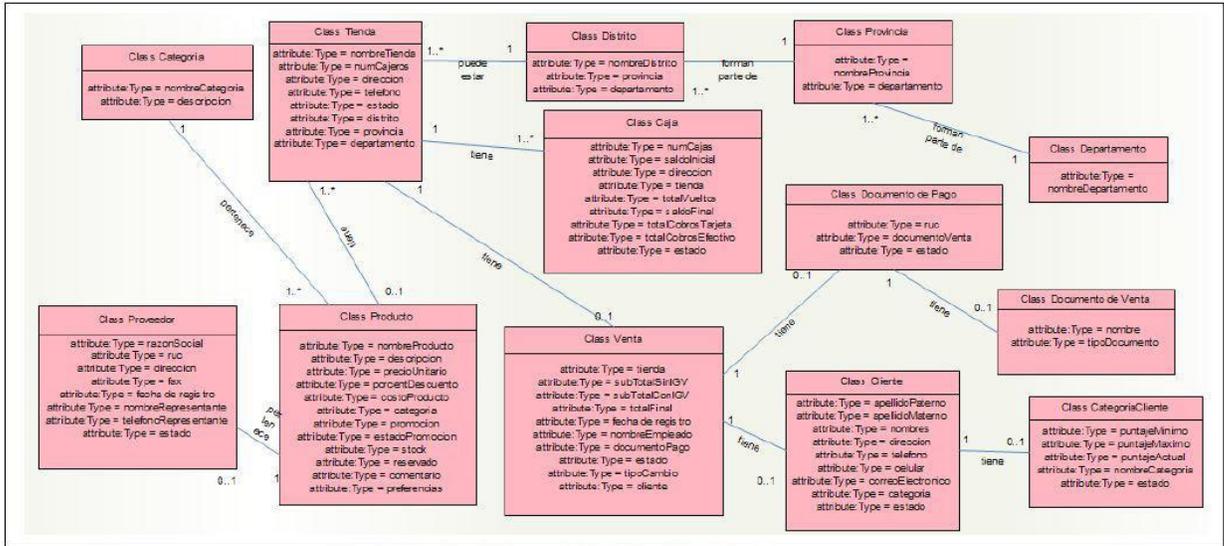


Figura 24: Diagrama de Clases del Módulo de Ventas. Rodriguez, J. (2013)

- **Arquitectura de la Solución**

En esta sección se describen los puntos resaltantes acerca de la arquitectura.

- ✓ **Representación de la arquitectura**

El proyecto que se desarrolla es un sistema web en la cual los usuarios podrán acceder mediante un servidor web a través de Internet la data se encuentra almacenada en un servidor, para lo cual se presenta la siguiente figura. En otras palabras, es una aplicación de software que se codifica con un lenguaje soportado por los navegadores web.

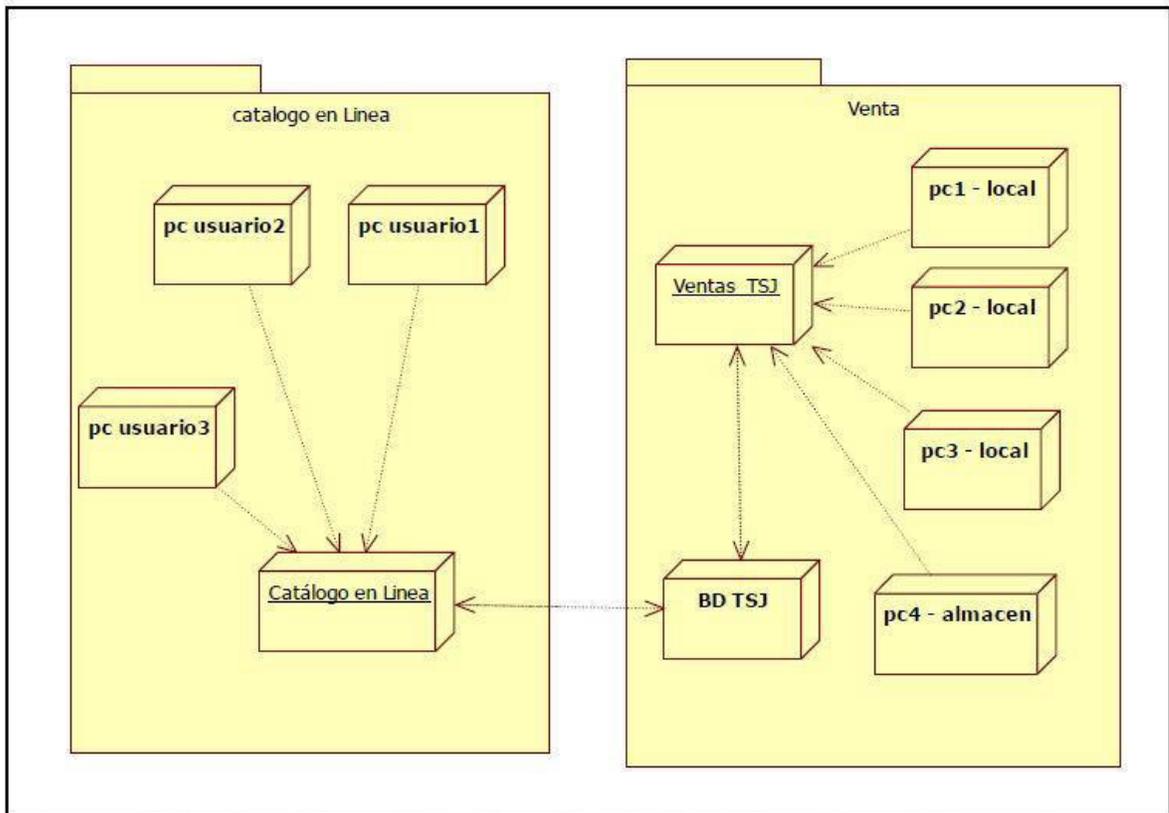


Figura 25: Representación de la Arquitectura. Rodrigez, J. (2013)

- **Arquitectura en 3 capas**

El sistema implementado adoptó tres niveles, dividido en 3 capas con un reparto claro de funciones.

- ✓ Una capa para la presentación de la interfaz GUI.
- ✓ Otra donde se encuentra el modelado del negocio.
- ✓ Otra para definir los objetos y para el almacenamiento de la información.

Estos niveles se ven reflejados en los frameworks diseñados para aplicaciones web basados en la estructura MVC de la arquitectura de patrones del modelo 2 de 3 capas. Integran la interfaz web y el modelo en un mismo servidor, aunque conservan su independencia funcional y separan el almacenamiento de datos. Es la distribución en capas más común en las aplicaciones web según la cita de FRAWRK (2012).

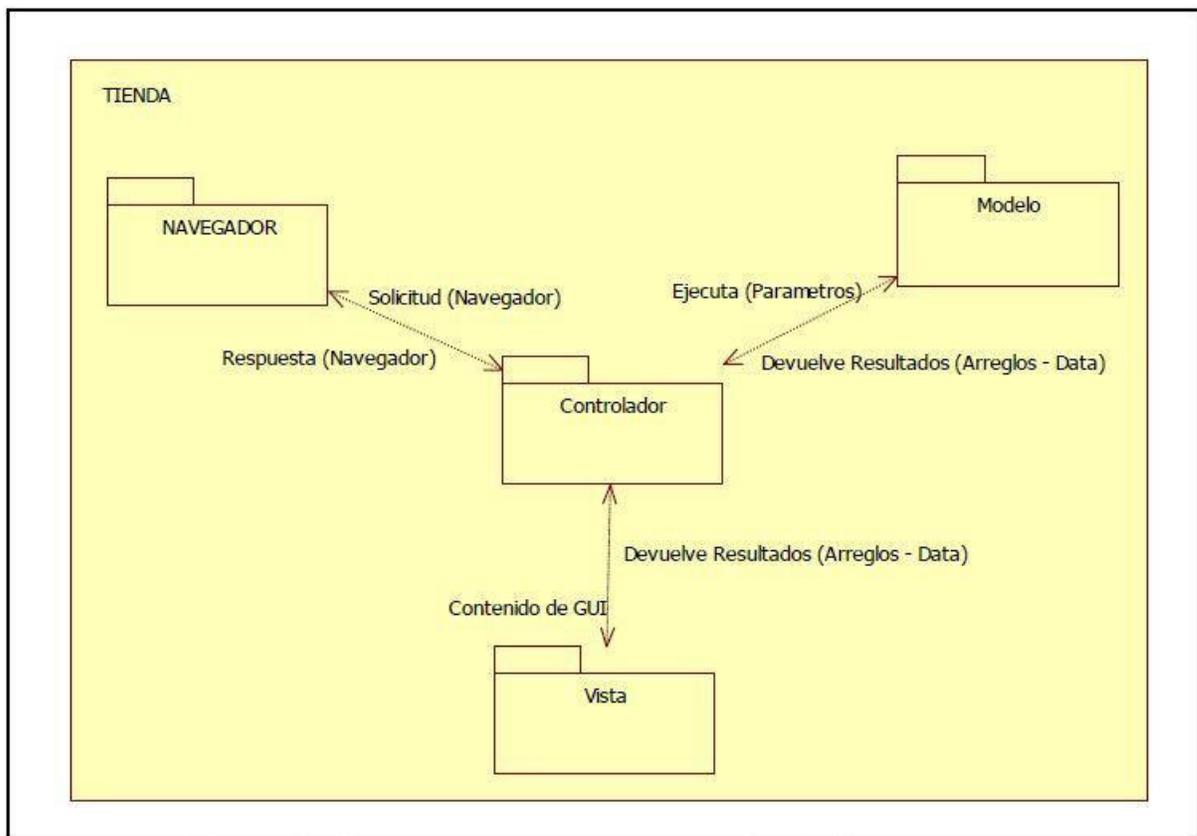


Figura 26: Arquitectura web en 3 capas. Rodrigez, J. (2013)

La capa Entidad es el objeto que representa los datos del programa. Normalmente se trata de cada una de las entidades presentes en la BD. Maneja los datos y controla todas sus transformaciones. La capa Entidad no tiene conocimiento específico de los controladores o

de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y de notificar a las vistas cuando cambia el modelo.

La capa Interfaz/Navegador – Catálogo/Venta es el objeto que maneja la presentación visual de los datos representados por la capa Entidad. Genera una representación visual de la capa Entidad y muestra los datos al usuario. Interactúa con la capa Entidad a través de una referencia. Este comportamiento se puede desarrollar en los siguientes lenguajes: java jsp, php, rhtml, etc.

La capa Lógica es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por la capa Entidad. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información de la capa Entidad o por alteraciones de las capas Interfaz o Navegador. Interactúa con la capa Entidad a través de una referencia.

- **Ventajas que presenta la arquitectura MVC**

Como para el desarrollo del presente proyecto se ha considerado trabajar con una arquitectura web, se presenta una serie de ventajas que presenta este modelo escogido, utilizando el lenguaje PHP.

Se pueden mencionar las siguientes ventajas:

- ✓ Hay una clara separación entre los componentes de un programa, lo cual permite implementarlos por separado

- ✓ Hay un API muy bien definido; cualquiera que use el API podrá reemplazar el modelo, la vista o el controlador sin aparente dificultad.

La conexión entre el modelo y sus vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución.

Si se observa que uno de los componentes posteriormente funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

- **Diseño de la interfaz gráfica**

En el diseño de la interfaz gráfica se ha tenido presente algunos criterios que fueron cubiertos en el desarrollo de las pantallas, entre ellos tenemos:

- ✓ El diseño de la interfaz gráfica cumplió los requisitos establecidos en el presente documento de tesis.
- ✓ Para facilitar el uso se conservará el mismo menú y barra de título en cada funcionalidad del sistema y se cambiará el contenido del cuerpo de la página.
- ✓ Las pantallas de la interfaz se visualizarán con una resolución de 1024 x 768 píxeles.
- ✓ Los controles serán los básicos: Labels, TextBox, CheckBox, RadioButton, Button, entre otros, distribuidos de tal manera que reflejen orden y buena distribución.

En Como resultado, a continuación se presenta la siguiente pantalla, pantalla principla del sistema.

✓ **Pantalla principal**

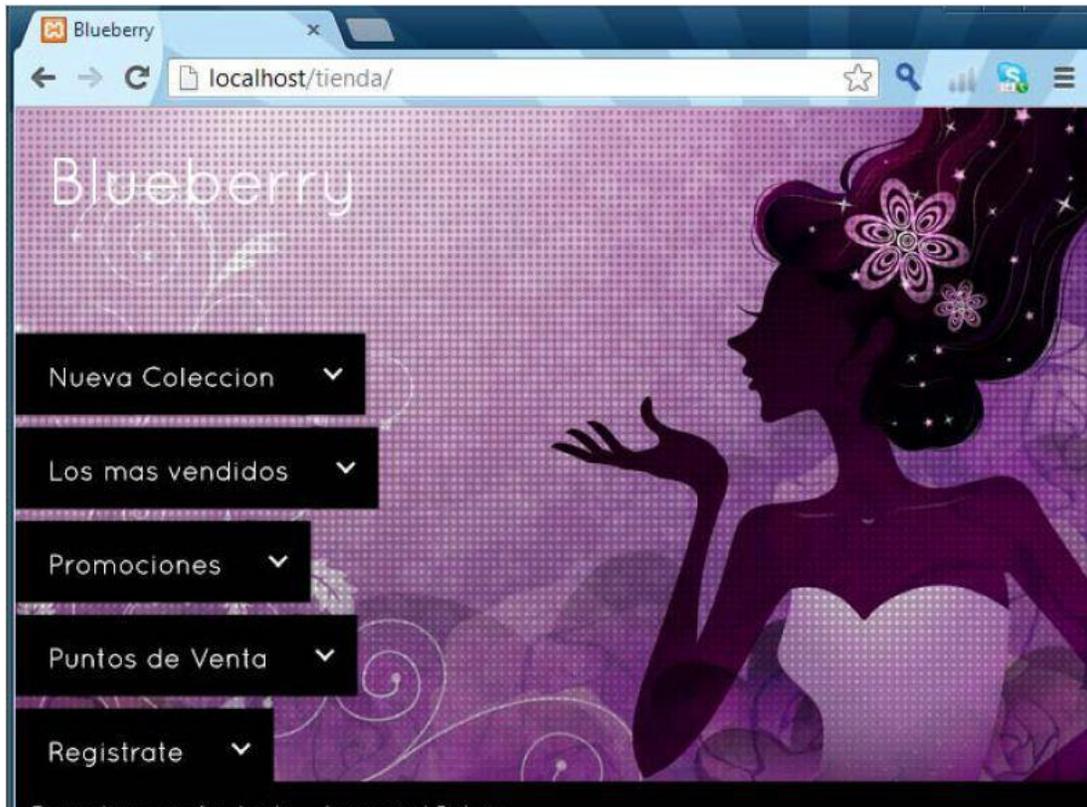


Figura 27: Pantalla principal del Sistema de Información TSJ. Rodrigez, J. (2013)

• **Diagrama de clases de diseño**

En esta sección se muestra el diagrama de clases de diseño del sistema de información de una tienda de ropa juvenil, que muestra la especificación para las clases de software que se utilizará en la implementación del sistema y la relación que existe entre ellas.

✓ **Tecnologías de construcción**

Para el proyecto citado, en la evaluación para elegir la tecnología de desarrollo, se manejaron 3 alternativas: PHP, Python y Ruby de los cuales se determinó que PHP era el más conveniente para este proyecto, a continuación se muestra una imagen con algunas características de las tecnologías antes mencionadas.

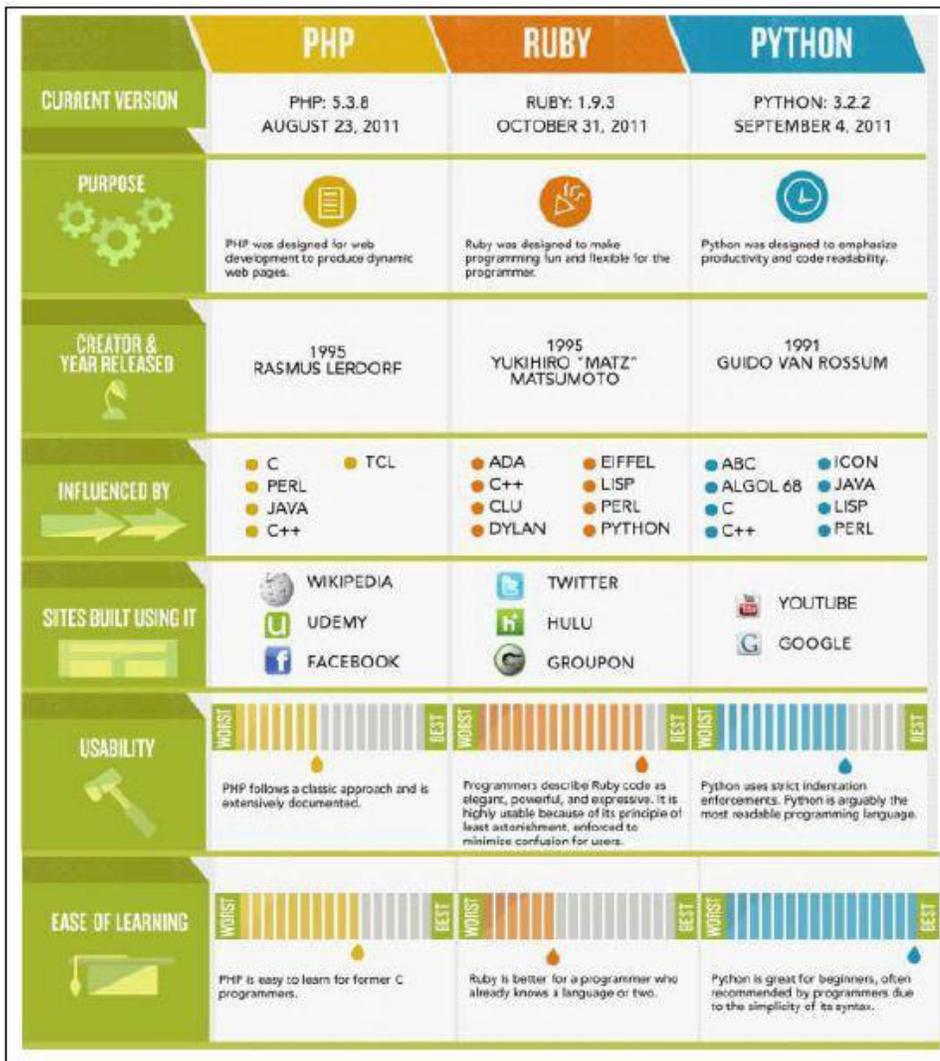


Figura 29: Cuadro comparativo de algunas características de los lenguajes PHP, Ruby y Python. Rodriguez, J. (2013)

Asimismo para el motor de base de datos se pensó trabajar con PostgreSQL o MySQL, decidiendo, previa análisis comparativo, trabajar con MySQL debido a sus características.

Para la selección del servidor se elige el servidor Apache Tomcat, dejando la alternativa de GlassFish, por su robustez, rendimiento y su capacidad para trabajar con PHP y la base de datos de forma eficiente.

✓ **Herramientas de construcción**

Son un conjunto de programas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

Para la gestión de la base de datos MySQL se optó por la herramienta PhpMyAdmin, esto debido a su facilidad de manejo.

Para la realización del modelado de la base de datos se utilizó ERwin 4.0 SP1, esta herramienta nos ayuda a crear de manera visual una base de datos.

Para el modelado de UML, se utilizó la herramienta StartUML, la cual es una herramienta de modelado de software basada en el Lenguaje Modelado Unificado, posee la riqueza suficiente para crear un modelado del sistema.

A continuación se presenta dos tablas que enmarcan a las tecnologías y herramientas usadas respectivamente para el desarrollo del sistema propuesto como solución.

Tecnologías para la construcción		
Tecnología	Descripción	Versión
Lenguaje de programación	PHP	5.5.4
Base de datos	MySQL	5.5.27
Servidor web y de aplicaciones	Apache Tomcat	2.4.3

Tabla 7: Tecnologías para la construcción. Rodrigez, J. (2013)

Herramientas para la construcción		
Herramientas	Descripción	Versión
Entorno integral de desarrollo	Notepad++	5.9.4
Modelado base de datos	ERwin	4.0
Gestor de base de datos MySQL	phpMyAdmin	3.5.2.2
Modelado UML	StarUML	2.0

Tabla 8: Herramientas para la construcción. Rodrigez, J. (2013)

- **Pruebas**

En este punto se detallan las pruebas que se harán para comprobar el correcto funcionamiento de la solución, con la finalidad de controlar la calidad del producto final. Se sabe que no hay forma de asegurar que un software esté 100% libre de errores. Sin embargo, si existen métodos para encontrarlos mediante una serie de pruebas. Existen distintos tipos de pruebas según las necesidades de verificación requeridas. Los escogidos para este sistema fueron los siguientes:

- ✓ Pruebas Unitarias
- ✓ Pruebas de Integración
- ✓ Pruebas del Sistema
- ✓ Pruebas de Requisitos Tecnológicos
- ✓ Pruebas de Certificación

3.1.2. Análisis, Diseño e Implementación de un Sistema de Comercio Electrónico integrado con una Aplicación móvil para la reserva y venta de pasajes de una empresa de transporte interprovincial

Este trabajo de investigación fue desarrollado por Carlos Alfredo Becerra Rodríguez, como Tesis para optar el Título de Ingeniero Informático en la Pontificia Universidad Católica del Perú, en Lima en noviembre el año 2013.

3.1.2.1. Resumen

El presente proyecto de fin de carrera corresponde al análisis, diseño e implementación de un Sistema de Comercio Electrónico integrado con una aplicación móvil para la reserva y venta de pasajes de una empresa de transporte interprovincial.

Es importante resaltar que se realizó un análisis de los principales problemas de una empresa mediana del sector de transporte interprovincial peruano.

Para la implementación del proyecto se consideró dos plataformas de desarrollo: web y móvil. La aplicación web se encargará principalmente del comercio electrónico (venta y reserva de pasajes), mientras que la aplicación móvil; de la consulta de información de los pasajes comprados o reservados.

3.1.2.2. Metodologías para la gestión del proyecto y el producto

En la siguiente tabla se puede observar un cuadro resumen sobre las metodologías y herramientas que se utilizaron para la gestión del proyecto y la elaboración del producto.

	Metodología/ Herramienta	Justificación
Proyecto	PMBOK	Porque posee las mejores prácticas dentro de la gestión de proyectos. Además porque define ciertas disciplinas bien detalladas de tal manera que se

		pueda gestionar de manera adecuada el desarrollo del proyecto. Los procesos que se considerarán son: Gestión de la Integración del Proyecto, Gestión del Alcance del Proyecto, Gestión del Tiempo del Proyecto, Gestión de Riesgos del Proyecto y Gestión de la Calidad del Proyecto.
Producto (Resultado Esperado)		
Documentación que contenga los diagramas de los procesos actuales y los que serán soportados por la alternativa de solución propuesta.	BPMN	Porque es una notación gráfica estandarizada que permite el modelado de los procesos de negocio. Para el presente proyecto es ideal porque se utilizará para modelar los procesos de negocio soportados por la solución planteada.
Arquitectura del sistema web y móvil, la cual mostrará la integración entre los componentes, los patrones de diseño y los estilos arquitectónicos a utilizar.	XP	Porque es un marco de trabajo para la gestión y desarrollo de software bajo un entorno de desarrollo ágil de software. Además por su simplicidad y flexibilidad para obtener el producto deseado. Es ideal porque está orientado a grupos pequeños de desarrollo, en este caso, solo una persona. También por las buenas prácticas que utiliza: desarrollo iterativo, pruebas unitarias continuas, refactorización, etc. Para obtener los resultados esperados orientados al producto software se utilizarán las siguientes fases: Fase de Exploración, Fase de Planificación y Fase de Iteraciones. El diseño de la
<i>Middleware</i> que contiene el conjunto de <i>web services</i> que serán consumidos por el sistema web y la aplicación móvil.	XP	
Prototipos de las interfaces del sistema web y la aplicación móvil.	XP	

		arquitectura del sistema forma parte de una de las actividades de la fase de Exploración. El resto de entregables se relacionan con la fase de planificación e iteración ya que es allí donde se elabora el cronograma de los entregables de tal manera que
--	--	---

Tabla 9: Cuadro de resumen para metodologías. Becerra, C. (2013).

3.1.2.3. Análisis y Diseño

- **Arquitectura**

En esta sección se describe, a alto nivel, los componentes que conforman el sistema, cómo interactúan entre sí y el conjunto de patrones que guían estas composiciones. Como se trata de un sistema orientado a servicios (SOA), con mayor razón debe haber una separación adecuada de los componentes de manera que el sistema pueda ser escalable. Como ya se explicó en el marco teórico, una arquitectura SOA se refiere a una especificación para proporcionar servicios que sean independientes de la plataforma en que se desarrolla para su uso en aplicaciones distribuidas.

- ✓ **Componentes principales**

Como ya se explicó anteriormente, el sistema está conformado por tres capas principales: cliente, servidor y datos. En la figura presentada a continuación también se puede apreciar la interacción de los componentes que conforman dichas capas.

De forma general, el cliente representa al navegador web y los dispositivos móviles, el servidor representa al servidor donde está alojado la web y los servicios web, y los datos representan a la base de datos. A continuación se explicará más a detalle en qué consiste cada uno de los componentes.

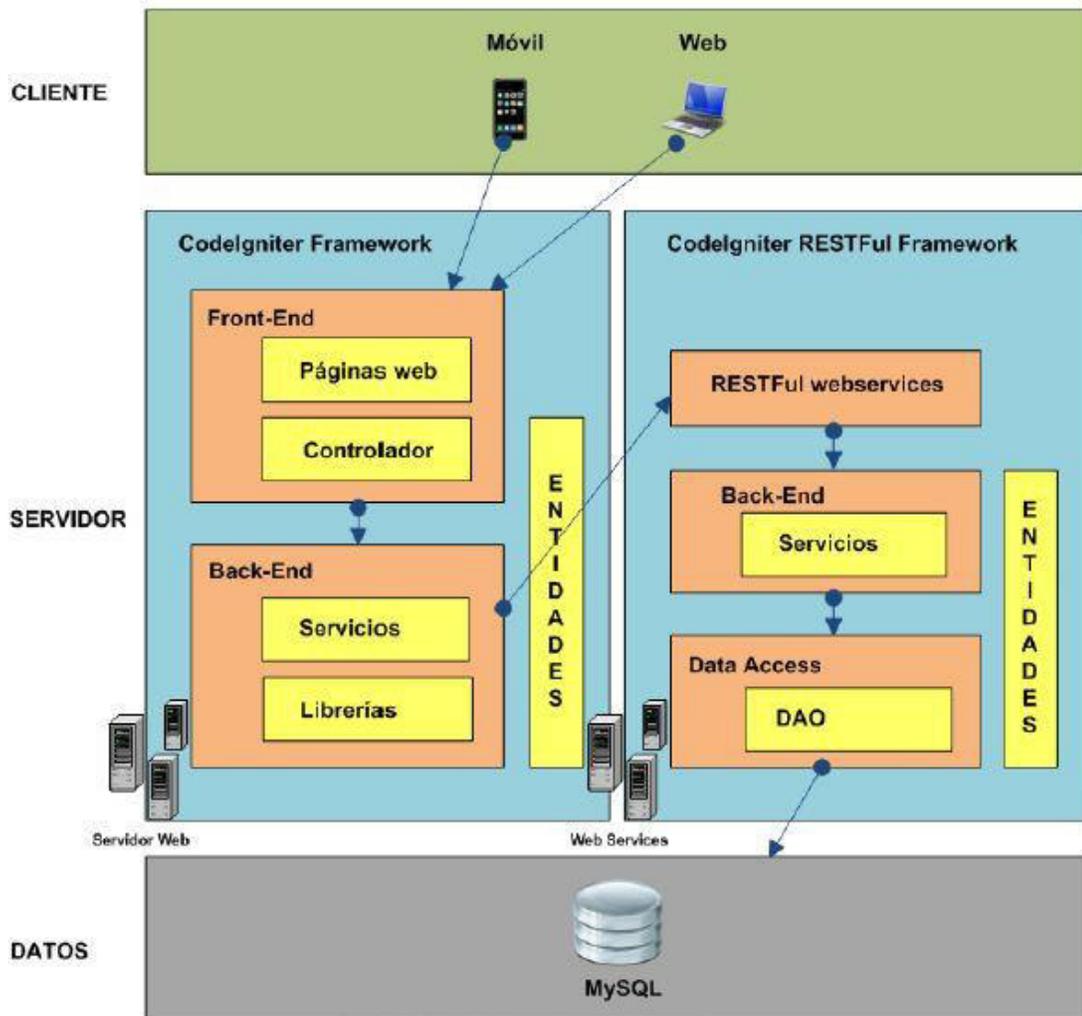


Figura 30: Diagrama de Arquitectura. Becerra, C. (2013).

✓ **Patrones de diseño de Arquitectura**

El patrón de Diseño a utilizar para el este sistema web y móvil será el Modelo- Vista- Controlador (MVC), pues las capas que lo conforman se separan de manera adecuada adaptándose a un sistema cliente-servidor. A continuación, en la Figura 4-2 se explicará de manera gráfica el funcionamiento de este patrón de diseño.

• **Modelo de Datos**

En esta sección se presenta una imagen con el modelo físico de datos para nuestra solución.

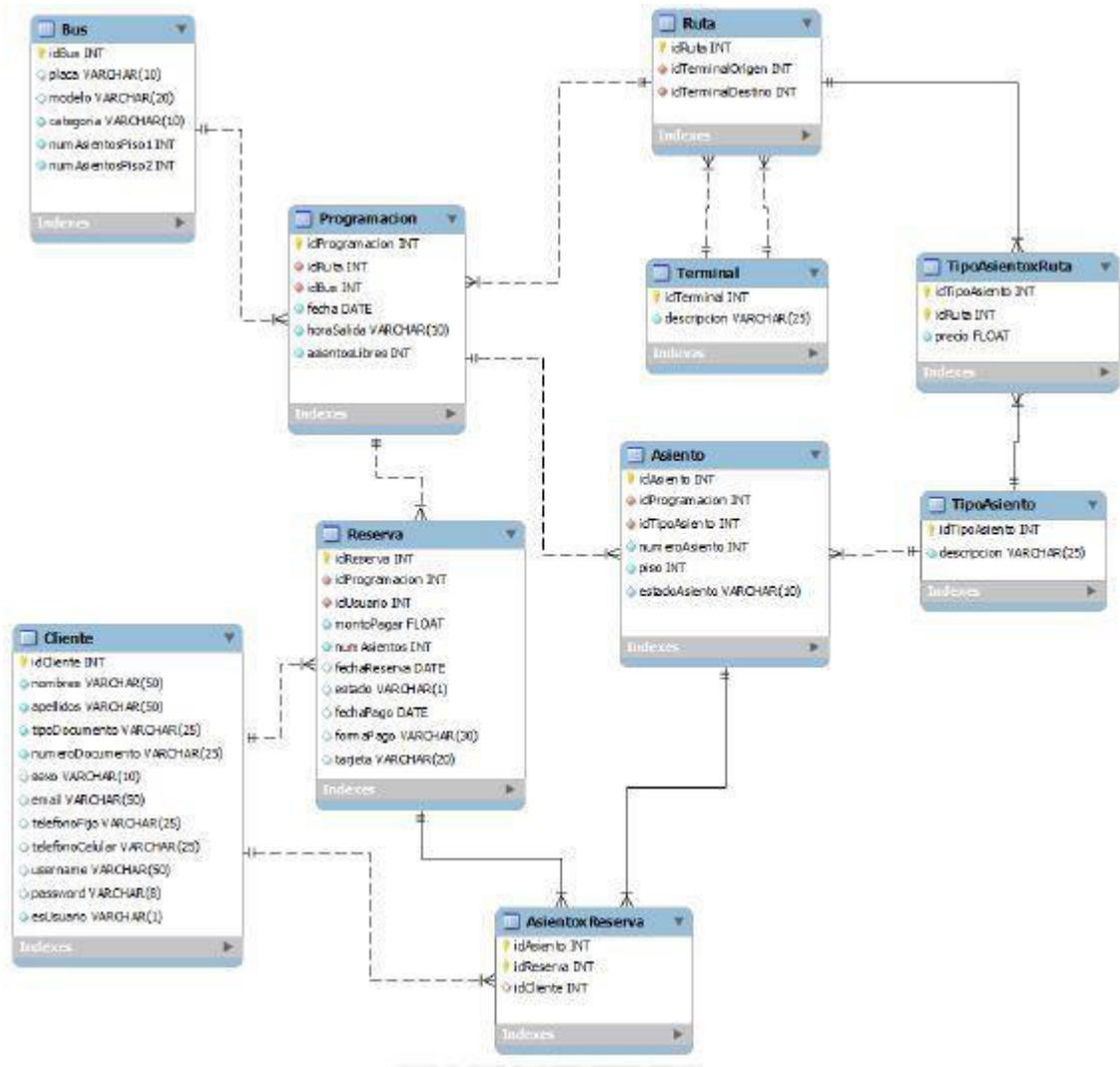


Figura 31: Modelo físico de datos. Becerra, C. (2013).

3.1.2.4. Construcción y Pruebas

En el presente apartado se describirá a gran escala, las herramientas y tecnologías que se utilizaron para la construcción del producto, se describe además los tipos de pruebas que se realizaron durante la construcción del mismo.

- **Herramientas, Plataformas y Tecnologías**

A continuación se listará el conjunto de herramientas software, plataformas y tecnologías web que se emplearon para el desarrollo del sistema. Todas las tecnologías pertenecen al grupo de software libre bajo licencia GNU GPL (General Public License).

- ✓ **Plataformas:** web, móvil (web mobile)
- ✓ **Lenguajes - scripting/web:** PHP, javascript, HTML5, CSS3
- ✓ **Servidor de Aplicaciones:** Apache
- ✓ **Webservices:** REST
- ✓ **Frameworks:** CodeIgniter, JQuery, JQuery Mobile, Bootstrap
- ✓ **Base de Datos:** MySQL
- ✓ **Herramientas de Desarrollo:** NetBeans, Subversion, MySQL WorkBench
- ✓ **Herramientas de Diseño de Interfaces:** Balsamiq

- **Pruebas**

La metodología empleada para la construcción del producto sugiere realizar pruebas constantes, por lo cual este trabajo implemento las pruebas respectivas, se usaron 2 tipo de pruebas básicas.

- ✓ Pruebas Unitarias
- ✓ Pruebas de Integración

3.2. Comparación de alternativas de solución

En esta sección se realizará una comparación de las metodologías para el desarrollo de los trabajos de investigación descritos en la sección 3.1.1 y 3.1.2.

Nuestra propuesta de solución se centra en una arquitectura bien definida, por lo que nos enfocaremos en las propuestas de arquitectura, como solución, de los trabajos de investigación antes citados.

A continuación presentaremos un cuadro comparativo donde se podrá notar las principales diferencias de las arquitecturas propuestas en los trabajos de investigación antes mencionados.

Criterio	Arquitectura Orientada a Servicios	Arquitectura de 3 niveles
Mejora en los tiempos de realización de cambios en procesos	Alto	Medio
Facilidad para evolucionar a modelos de negocios basados en tercerización	Alto	Bajo

Facilidad para abordar modelos de negocios basados en colaboración con otros entes (socios, proveedores)	Alto	Medio
Facilidad para la integración de tecnologías disímiles	Alto	Bajo
Mejora en la toma de decisiones	Alto	Alto
Aplicaciones flexibles	Alto	Medio
Aplicaciones reutilizables y adaptables	Alto	Medio
Reducción de costes	Alto	Medio
Riesgo de migración	Bajo	Alto

**Tabla 10: Comparación de alternativas de Arquitecturas para nuestra solución.
Elaboración propia**

Del cuadro comparativo anterior podemos apreciar que orientar nuestra arquitectura a los servicios es la mejor opción, con ello tendremos una solución óptima para desarrollar nuestro trabajo.

Capítulo IV. DESARROLLO DE LA SOLUCIÓN

4.1. Generalidades

En el presente capítulo se desarrollará la solución, para ello se utilizará el Proceso Unificado Rational (RUP), se desarrollará siguiendo las siguientes disciplinas: Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, y Despliegue, debido a la magnitud y el tiempo requerido para la finalización del trabajo de investigación solo se realizarán algunos productos de trabajo del RUP.

4.2. Modelado del Negocio

La empresa DARUNE SAC, la PYME de nuestro caso de estudio, se desarrolla en el rubro comercial, específicamente en la venta al por mayor y menor de artículos para bebés como pañales, biberones, leches enlatadas en polvo, colonias, etc.

La empresa cuenta con 3 puntos de ventas y 2 almacenes en el distrito de la Victoria, esta empresa tiene 11 años en el mercado, con lo cual tiene una amplia cartera de clientes y mucha información asociada al negocio, asimismo como área clave del negocio, tenemos al área de ventas, la cual es el centro de la organización ya que dicha área es la que se encarga de dar o no dar respuesta a las necesidades de sus clientes o potenciales clientes.

4.2.1. Misión y Visión

- **Misión:** Brindar al cliente una amplia variedad de productos y artículos para bebés, logrando que este pueda satisfacer la gran mayoría de sus necesidades en un mismo lugar y a un precio muy accesible. Siempre teniendo en cuenta a la calidad como el factor más importante.
- **Visión:** Ser una empresa líder en nuestro rubro, brindando bajos costos y excelente calidad en los productos que comercializamos.

4.2.2. Mapa de Procesos

A continuación se presenta el mapa de procesos de la empresa DARUNE SAC, el cual es un inventario gráfico de los procesos de esta organización, el mismo que fue diseñado basado en información fidedigna del gerente de la empresa DARUNE SAC, y siguiendo la guía publicada por la UCA (2007).



Figura 32: Mapa de Procesos de DARUNE SAC. Elaboración Propia

4.2.3. Modelo de casos de uso del negocio

- **Actores de negocio**

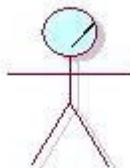
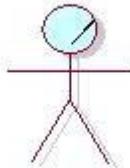
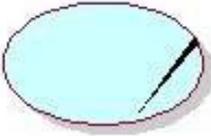
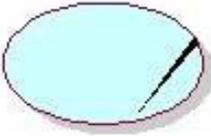
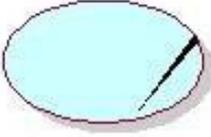
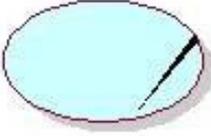
Ítem	Actor de negocio	Descripción
AN001	 Cliente	Se encarga de realizar la petición de oferta de productos.
AN002	 Proveedor	Se encarga de proveer los productos que se soliciten de parte de la organización.

Tabla 11: Actores de negocio. Elaboración propia.

- **Casos de uso de negocio**

Ítem	Caso uso de negocio	Descripción
CUN001	 <p data-bbox="751 667 930 699">Realizar venta</p>	Este proceso permite realizar las ventas de los productos en base a un pedido del cliente.
CUN002	 <p data-bbox="735 961 946 993">Distribuir pedido</p>	Este proceso permite distribuir un pedido a los clientes previa venta.
CUN003	 <p data-bbox="751 1224 930 1255">Realizar cobro</p>	Este proceso permite realizar los cobros a los clientes que solicitaron crédito en determinada venta.
CUN004	 <p data-bbox="735 1591 946 1623">Realizar compra</p>	Este proceso permite realizar las compras de mercadería a los proveedores de la empresa para mantener el stock de los productos.

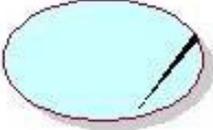
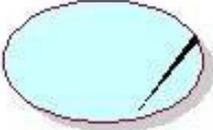
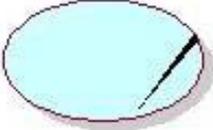
CUN005	 <p>Abastecer mercadería</p>	Este proceso permite realizar el abastecimiento de mercadería a los puntos de venta desde los almacenes de la empresa.
CUN006	 <p>Administrar finanzas</p>	Este proceso permite administrar los fondos de la organización para un adecuado funcionamiento y equilibrio de los diversos entes que la integran.
CUN007	 <p>Administrar personal</p>	Este proceso permite la selección, la realización de nóminas y la evaluación del personal que pertenece o participa dentro de la organización

Tabla 12: Casos de uso del negocio. Elaboración propia.

- Diagrama de casos de uso del negocio

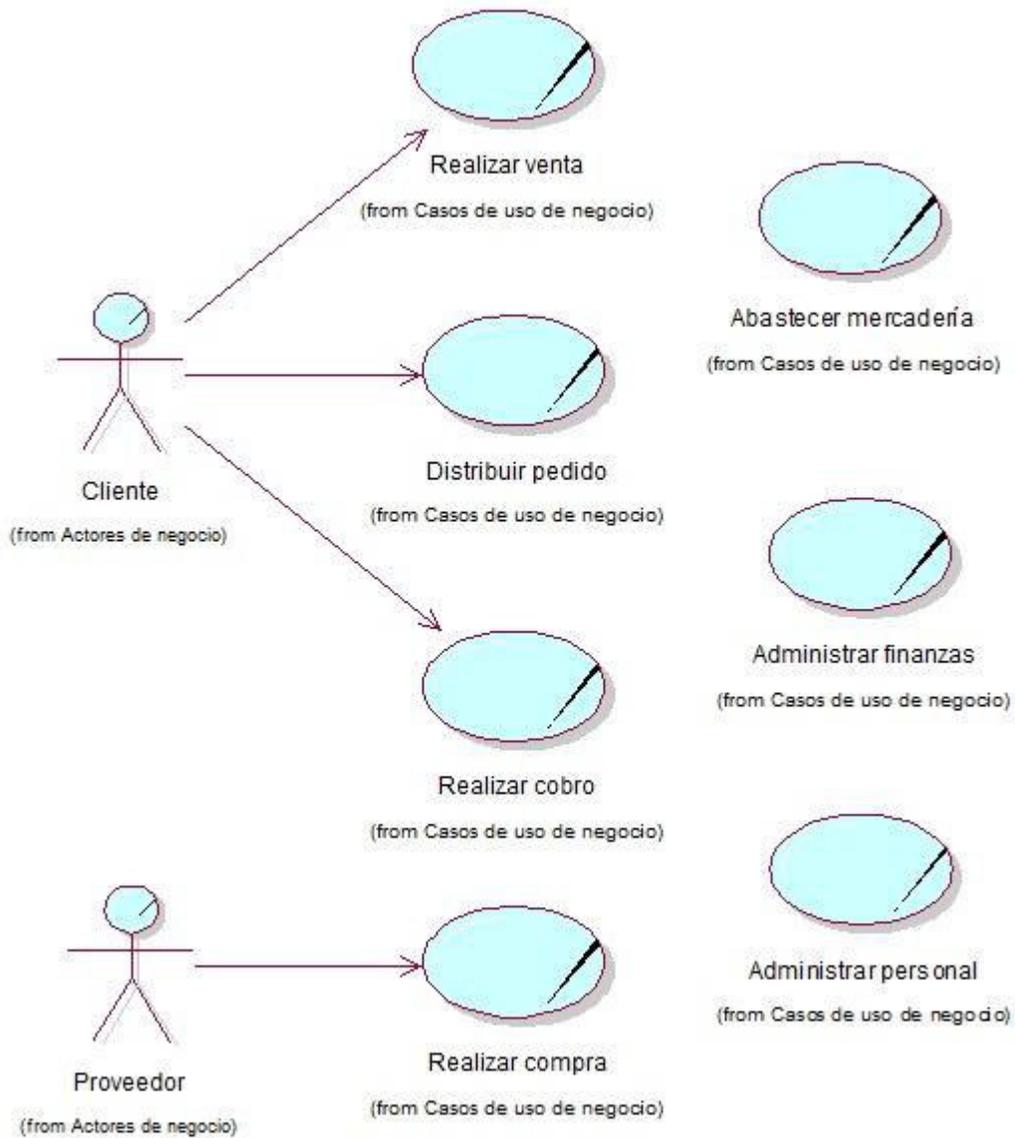


Figura 33: Diagrama de casos de uso del negocio. Elaboración propia.

4.2.4. Modelo de análisis de negocio

- **Trabajadores del negocio**

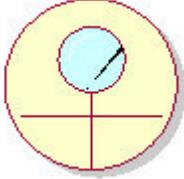
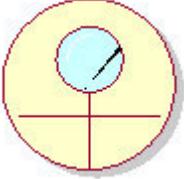
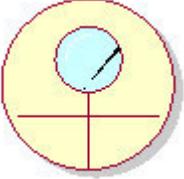
 <p data-bbox="277 661 529 695">Operador de ventas</p>	 <p data-bbox="696 661 919 695">Operador de caja</p>	 <p data-bbox="1068 661 1357 695">Operador de despacho</p>
---	---	---

Tabla 13: Trabajadores del negocio. Elaboración propia.

- **Diagrama de clases del caso de uso de negocio realizar ventas**

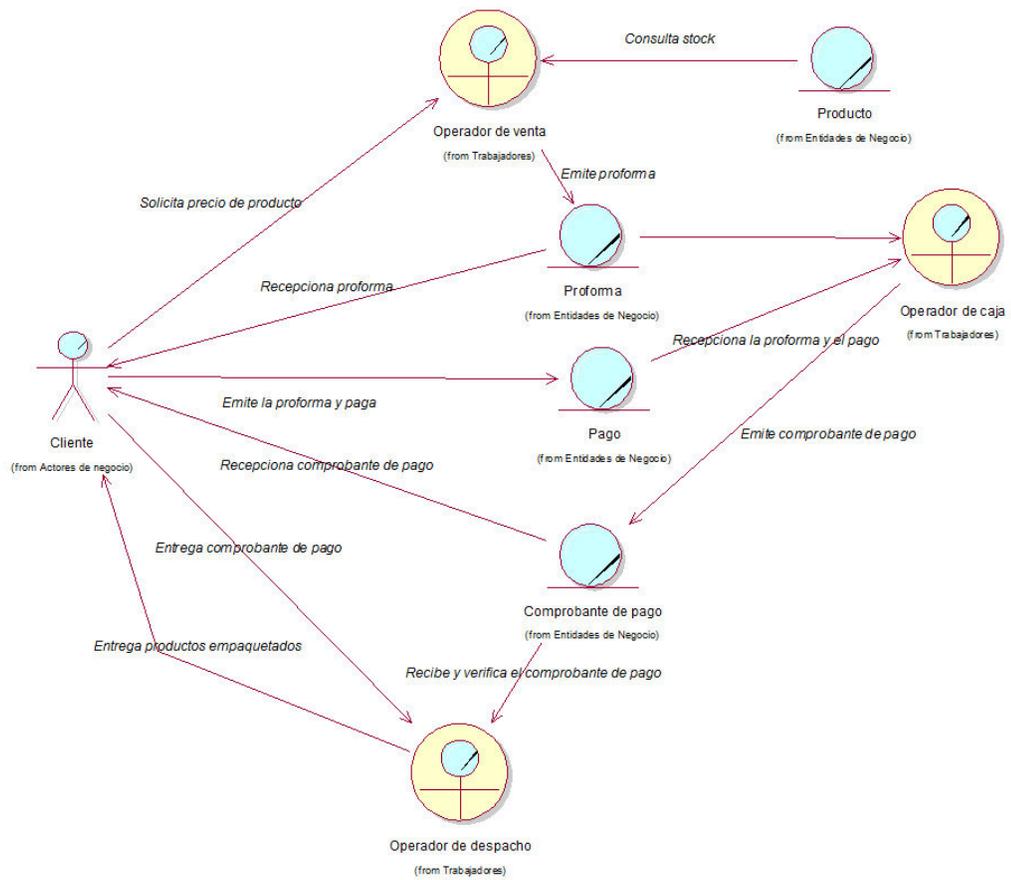


Figura 34: Diagrama de clases del CUN Realizar venta. Elaboración propia.

4.3. Captura de Requerimientos

A continuación se muestran los requerimientos principales de los usuarios de la empresa DARUNE SAC, para ello fue necesario realizar entrevistas a los usuarios del área respectiva, así tenemos para el caso de uso de negocio Realizar venta.

4.3.1. Requerimientos funcionales

Caso de uso del negocio	Código	Nombre del requerimiento funcional	Descripción	Prioridad
Realizar venta	RF001	Validar usuario	El sistema debe permitir la validación del acceso a los empleados.	Alta
	RF002	Consultar stock de producto	El sistema debe permitirle al operador de venta y al operador de caja verificar si existe mercadería de un producto determinado.	Alta
	RF003	Buscar producto por código	El sistema debe permitirle al operador de venta y al operador de caja buscar un producto por código.	Media

RF004	Agregar productos al pedido	El sistema debe permitir al operador de venta y al operador de caja la selección de uno o varios productos a la vez para agregarlos al pedido.	Alta
RF005	Modificar cantidad de producto en el pedido	El sistema debe permitirle al operador de caja y al operador de venta modificar la cantidad de un producto en el pedido antes de que el pago sea realizado.	Media
RF006	Seleccionar documento de preventa	El sistema debe permitirle al operador la selección entre generar un ticket de venta o un proforma para el pedido.	Media
RF007	Eliminar producto de pedido	El sistema debe permitirle al operador de caja y al operador de venta eliminar un producto del pedido antes de que el pago sea realizado.	Media

RF008	Generar ticket de venta	El sistema debe permitir al operador de venta generar tickets de venta para su emisión a los cliente	Alta
RF009	Consultar ticket pendientes de pago	El sistema debe permitir al operador de venta consultar cuantos tickets se encuentran pendientes de pago.	Media
RF010	Buscar ticket por serie y número de ticket	El sistema debe permitir al operador de caja buscar un ticket ingresando la serie y número correspondiente.	Alta
RF011	Validar ticket	El sistema debe permitir que el operador de caja pueda validar el ticket ingresando filtros para su búsqueda.	Alta

RF012	Seleccionar el tipo de comprobante de pago a generar	El sistema debe permitir al operador cajero seleccionar el tipo de comprobante que desea generar.	Media
RF013	Generar comprobante de pago	El sistema debe permitir que el operador de caja genere comprobantes de pago automáticamente.	Alta
RF014	Registrar venta	El sistema debe permitir que el operador registre la venta realizada.	Alta
RF015	Editar pedido	El sistema debe permitir que el operador de caja modifique el pedido antes de procesar el pago.	Baja
RF016	Elegir método de pago	El sistema debe permitir que el operador de caja pueda seleccionar el o los métodos de pagos solicitados por el cliente.	Media

	RF017	Buscar comprobante de pago por serie y número	El sistema debe permitir que el operador busque un comprobante de pago ingresando su serie y número.	Alta
	RF018	Validar comprobante de pago	El sistema debe permitir que el operador de despacho pueda validar el comprobante de pago ingresando filtros para su búsqueda.	Alta
	RF019	Generar comprobante de pago	El sistema permitirá imprimir el comprobante de pago de acuerdo a la elección del cliente.	Alta

Tabla 14: Requerimientos funcionales del proceso de ventas. Elaboración propia.

4.3.2. Requerimientos no funcionales

Código	Nombre del requerimiento no funcional	Descripción	Categoría
RN001	El sistema debe estar restringido	Solo podrán hacer uso del sistema los usuarios que cuenten con los permisos correspondientes.	Funcionalidad

RN002	El sistema debe visualizarse y funcionar en cualquier computadora.	El sistema funciona en cualquier computadora que tenga un browser instalado, con cualquier sistema operativo, sin necesidad de internet.	Funcionalidad
RN003	El sistema debe tener una interfaz amigable	La interfaz del sistema debe ser de fácil uso y adaptación de los usuarios, así como el uso de múltiples mensajes de confirmación	Usabilidad
RN004	El sistema debe contar con botones amigables en el menú principal	El menú principal de la aplicación tendrá botones con iconos amigables de 30x30 pixeles.	Usabilidad
RN005	El sistema estará disponible en un 95%	El sistema estará disponible al 95 por ciento entre las 9:00 AM y las 9:00 PM	Confiabilidad
RN006	Obligatoriedad de campos	El sistema mostrara mensajes restrictivos si no se ingresan los datos obligatorios.	Fiabilidad
RN007	Mensajes de error	El sistema mostrara mensajes de error cuando se ingresen datos incorrectos.	Fiabilidad

Tabla 15: Requerimientos no funcionales. Elaboración propia.

4.3.3. Actores del sistema

Ítem	Actor del sistema	Descripción
AS001	 Operador de venta	Se encarga de generar los pedidos en base a una solicitud del cliente.
AS002	 Operador de caja	Se encarga de procesar los pagos y generar los comprobantes de pagos.
AS003	 Operador de despacho	Se encarga de realizar el despacho después de la realización de una venta.

Tabla 16: Actores del sistema. Elaboración propia.

4.3.4. Diagrama de casos de uso

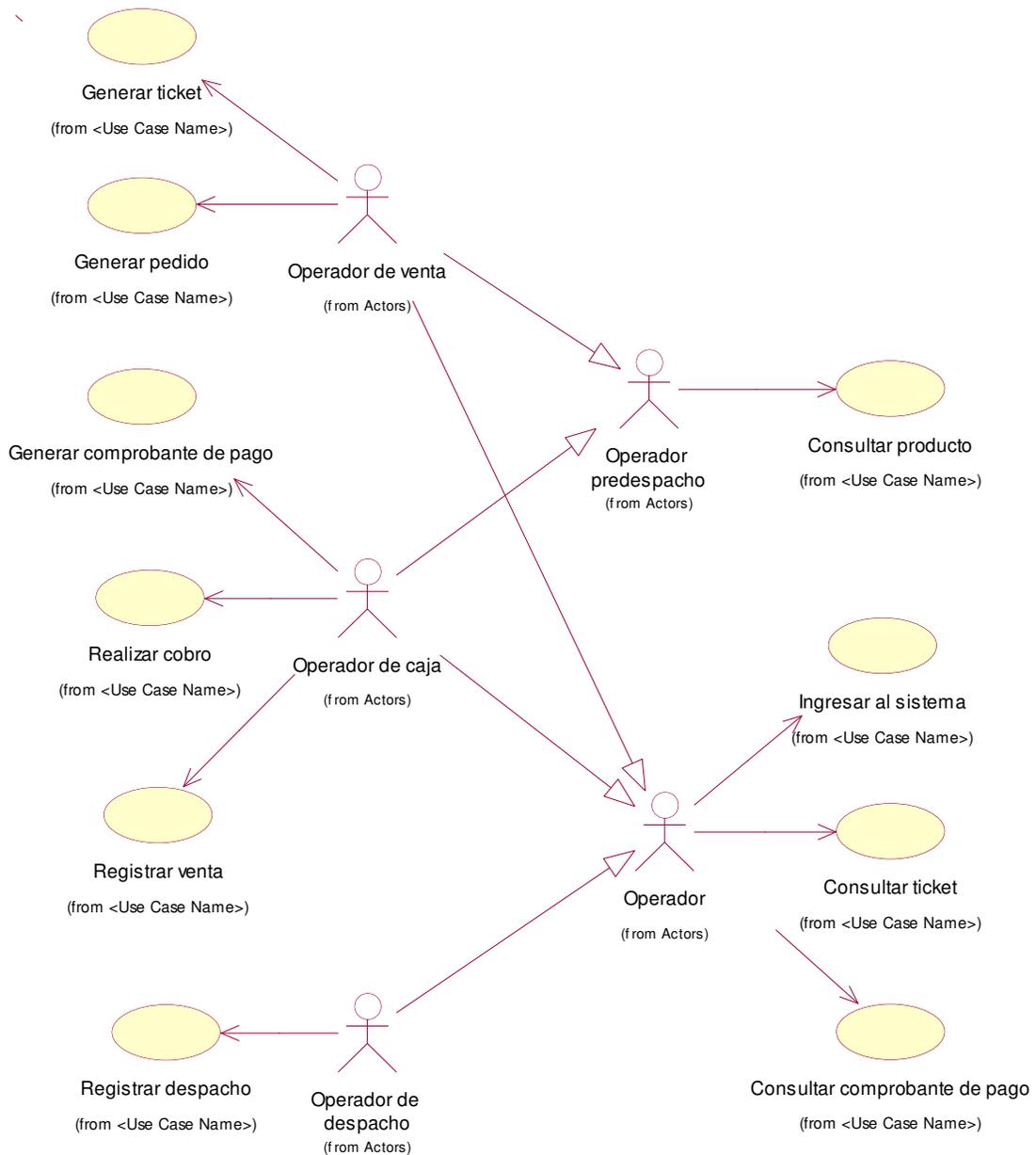


Figura 36: Diagrama de casos de uso del proceso Realizar venta. Elaboración propia.

4.3.5. Matriz de trazabilidad

Actor	Requerimientos	CU01	CU02	CU03	CU04	CU05	CU06	CU07	CU08	CU09	CU10
		Ingresar al Sistema	Consultar producto	Generar pedido	Generar ticket	Consultar ticket	Generar comprobante de pago	Registrar venta	Consultar comprobante de pago	Realizar cobro	Registrar despacho
Operador	RF001	Validar usuario									
Operador de venta, operador de caja	RF002	Consultar stock de producto	X								
Operador de venta, operador de caja	RF003	Buscar producto por código		X							
Operador de venta, operador de caja	RF004	Agregar productos al pedido			X						
Operador de venta, operador de caja	RF005	Modificar cantidad de producto en el pedido			X						
Operador de venta	RF006	Seleccionar documento de preventa			X						
Operador de venta, operador de caja	RF007	Eliminar producto de pedido			X						
Operador de venta	RF008	Generar ticket de venta				X					
Operador de venta	RF009	Consultar ticket pendientes de pago					X				
Operador de venta, operador de caja, operador de despacho	RF010	Buscar ticket por serie y número de ticket					X				
Operador de caja	RF011	Validar ticket					X				
Operador de caja	RF012	Seleccionar el tipo de comprobante de pago a generar						X			
Operador de caja	RF013	Generar comprobante de pago						X			
Operador de caja	RF014	Registrar venta							X		
Operador de caja	RF015	Editar pedido			X						
Operador de caja	RF016	Elegir método de pago								X	
Operador de despacho	RF017	Buscar comprobante de pago por serie y número							X		
Operador de despacho	RF018	Validar comprobante de pago							X		
Operador de despacho	RF019	Registrar despacho									X

Tabla 17: Matriz de trazabilidad

4.3.6. Especificación de casos de uso

4.3.5.1. Especificación de caso de uso Consultar producto

Caso de uso	Consultar producto	
Código	CU02	
Objetivo	Permite consultar la información de un producto.	
Precondiciones	<ul style="list-style-type: none"> ▪ El usuario ha iniciado sesión en el sistema. ▪ El usuario selecciono la opción Búsqueda de producto. 	
Post condiciones	<ul style="list-style-type: none"> ▪ El usuario visualiza la información de acuerdo a los filtros ingresados. 	
Actores	<ul style="list-style-type: none"> ▪ Operador de venta ▪ Operador de caja 	
Flujo Principal	Pasos	
	1.	El sistema muestra la interfaz de búsqueda de productos.

Caso de uso	Consultar producto	
	2.	El actor ingresa los filtros según criterio.
	3.	El sistema muestra la lista de productos encontrados en la grilla Resultado de búsqueda
Flujo Alternativo	Acción	
	1	En el paso 3, el sistema muestra el mensaje: No se ha registrado ningún producto, esto debido a que no hay ninguna coincidencia en la base de datos según los filtros ingresados.
Frecuencia	Diaria	
Requisito Funcional, No Funcional	RF002., RF003, RN001, RN003.	

Tabla 18: Especificación de caso de uso Consultar producto. Elaboración propia.

4.3.5.2. Especificación de caso de uso Consultar producto

Caso de uso	Generar ticket	
Código	CU04	
Objetivo	Permite generar un ticket con el cual el cliente pagará en caja el pedido realizado.	
Precondiciones	<ul style="list-style-type: none"> ▪ El usuario ha iniciado sesión en el sistema. ▪ El usuario selecciono la opción Operaciones de venta del menú principal. ▪ El usuario agregó productos al pedido. 	
Post condiciones	<ul style="list-style-type: none"> ▪ El usuario obtiene el ticket impreso 	
Actores	<ul style="list-style-type: none"> ▪ Operador de venta 	
Flujo Principal	Pasos	

Caso de uso	Generar ticket	
	1.	El usuario selecciona el tipo de documento ticket de venta en la sección Documento de venta.
	2.	El usuario selecciona el botón Generar documento de venta.
	3.	El usuario confirma la generación del ticket de venta.
	4.	El sistema muestra el mensaje: El ticket de venta se ha generado satisfactoriamente con la serie y el número correlativo correspondiente.
	5.	El usuario acepta el mensaje.
Flujo Alternativo	Acción	
	1.	En el paso 3, el sistema muestra la interfaz de los productos del pedido, debido a que se seleccionó “No” en la confirmación.
	2.	En el paso 4, el sistema muestra un mensaje con la descripción del error si el ticket no se genera correctamente.
Frecuencia	Diaria	
Requisito Funcional, No Funcional	RF008, RN001, RN003.	

Tabla 19: Especificación de caso de uso Generar ticket. Elaboración propia.

4.4. Análisis y Diseño del Sistema

En este punto además de realizar lo correspondiente a la disciplina de Análisis y diseño del RUP realizaremos un Diagrama de Análisis de Procesos para el proceso de ventas actual de la empresa DARUNE SAC, este cuadro se basa en las actividades del proceso que se pueden visualizar en el punto 4.2.4 del presente documento.

Ítem	Descripción de actividades	Duración (seg)	Sin valor	 Operación	 Inspección	 Transporte	 Demora o espera	 Almacenamiento	Automatizable
1	Solicitar precio de producto	5		X					
2	Verificar stock	120			X				Si
3	Emitir precio de producto	5				X			
4	Evaluar precio de producto	5			X				
5	Emitir conformidad de precio	3		X					
6	Agregar producto al pedido	20		X					Si
7	Sellar proforma	3	X	X					
8	Emitir proforma con pedido	3				X			
9	Recepcionar proforma con pedido	2		X					
10	Presentar proforma con pedido	2				X			
11	Recepcionar la proforma con pedido	2		X					
12	Validar proforma con pedido	15			X				Si
13	Consultar el tipo de pago a realizar	3		X					
14	Emitir elección de comprobante de pago	3		X					
15	Solicitar pago	2		X					
16	Realizar pago	10		X					
17	Procesar pago	10		X					
18	Registrar venta	15		X					Si
19	Elaborar comprobante de pago	30		X					Si
20	Emitir comprobante de pago	3				X			
21	Recepcionar comprobante de pago	2		X					
22	Presentar comprobante de pago	2				X			
23	Recepcionar comprobante	2		X					
24	Validar comprobante de pago	15			X				Si
25	Seleccionar productos	20		X					
26	Sellar comprobante de pago	3	X	X					
27	Entregar comprobante y productos empaquetados	10				X			
28	Recepcionar comprobante y productos empaquetados	5		X					
	TOTAL (seg)	320	2	18	4	6			6
					TOTAL (min)	5.33			Automatizable
									Sin valor

Tabla 20: Diagrama de análisis del proceso de ventas DARUNE SAC - ASIS. Elaboración propia.

Para poder evaluar el costo de este proceso, primero debemos definir los costos por cada minuto que invierte un trabajador que interviene en dicho proceso.

Rol	Costo mensual (S/.)	Costo promedio por hora (S/.)	Costo promedio por minuto (S/.)
Operador de venta	1200	5.00	0.08
Operador de caja	1500	6.25	0.10
Operador de despacho	1000	4.17	0.07

Tabla 21: Costos promedio por rol. Elaboración propia.

Datos:

- a) **Rol:** Es el cargo que desempeña el personal en la empresa, los cuales forman parte en este caso del proceso de venta.
- b) **Costo mensual:** Es el pago al mes que se le brinda a un empleado por sus servicios por un tiempo de 48 horas semanales.
- c) **Costo promedio:** Es el promedio por horas de acuerdo al pago mensual del personal. Esto se genera dividiendo el costo mensual entre los días trabajados al mes (30) y dividiendo este resultado entre las horas trabajadas al día (8).
- d) **Costo promedio por minuto:** Es el promedio por minuto de acuerdo al costo promedio por hora, se obtiene de dividir el costo promedio por hora entre la cantidad de minutos en una hora (60).

Una vez ya identificados estos costos promedio se puede evaluar el costo por del proceso de ventas de la empresa DARUNE SAC.

Datos de la tabla de costos por proceso:

- a) **Descripción de las actividades:** Es el detalle de cada una de las actividades de este proceso.

- b) **Duración (seg):** Es el tiempo en minutos que se emplea por cada actividad por un determinado rol.
- c) **Costo promedio:** Es el costo promedio por minuto del rol que interviene en la actividad, el cual sale del cuadro anterior “costo promedio por rol”.
- d) **Costo por actividad:** Es el resultado del producto del costo total de acuerdo a la cantidad de minutos que dura cada actividad y el costo promedio por minuto de dicha actividad.

Ítem	Descripción de actividades	Duración (seg)	Costo promedio rol/actividad (min)	Costo por actividad (S/.)
1	Solicitar precio de producto	5		
2	Verificar stock	120	0.08	0.16
3	Emitir precio de producto	5	0.08	0.01
4	Evaluar precio de producto	5		
5	Emitir conformidad de precio	3		
6	Agregar producto al pedido	20	0.08	0.03
7	Sellar proforma	3	0.08	0.00
8	Emitir proforma con pedido	3	0.08	0.00
9	Recepcionar proforma con pedido	2		
10	Presentar proforma con pedido	2		
11	Recepcionar la proforma con pedido	2	0.1	0.00
12	Validar proforma con pedido	15	0.1	0.03
13	Consultar el tipo de pago a realizar	3	0.1	0.01
14	Emitir elección de comprobante de pago	3		
15	Solicitar pago	2	0.1	0.00
16	Realizar pago	10		
17	Procesar pago	10	0.1	0.02
18	Registrar venta	15	0.1	0.03
19	Elaborar comprobante de pago	30	0.1	0.05
20	Emitir comprobante de pago	3	0.1	0.01
21	Recepcionar comprobante de pago	2		
22	Presentar comprobante de pago	2		
23	Recepcionar comprobante	2	0.07	0.00
24	Validar comprobante de pago	15	0.07	0.02
25	Seleccionar productos	20	0.07	0.02
26	Sellar comprobante de pago	3	0.07	0.00
27	Entregar comprobante y productos empaquetados	10	0.07	0.01
28	Recepcionar comprobante y productos empaquetados	5		
	TOTAL (seg)	320		0.39
			Cliente	
			Operador de venta	
			Operador de caja	
			Operador de despacho	

Tabla 22: Costo del proceso de ventas DARUNE SAC. Elaboración propia.

De la figura anterior podemos notar que el costo del proceso de ventas actual de la empresa DARUNE SAC es en promedio es de S/. 0.39. El propósito de este análisis será el de comparar el proceso una vez implementada la sistematización.

4.4.1. Modelo de dominio

A continuación el modelo de dominio

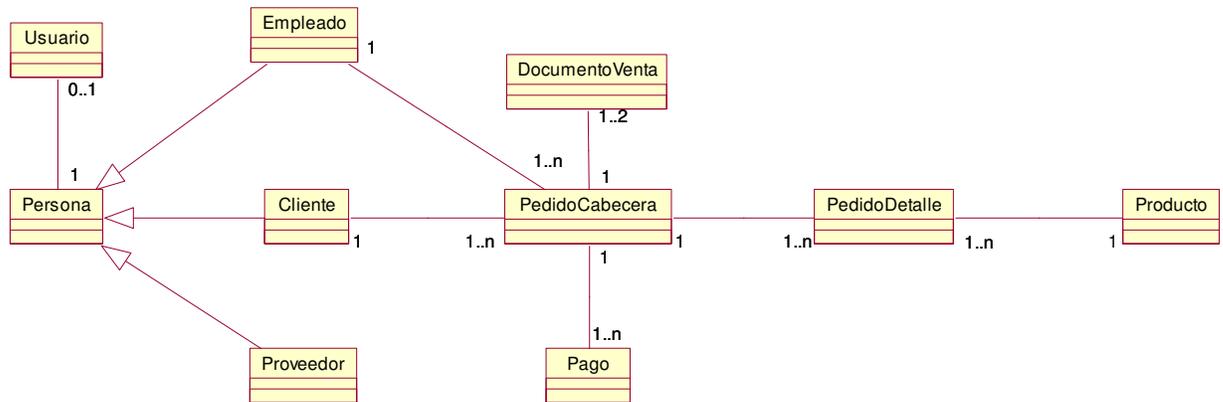


Figura 37: Modelo de dominio: Elaboración propia.

4.4.3. Diseño de la Arquitectura



Figura 39: Arquitectura de DARUNE SAC orientada a servicios. Elaboración propia.

4.4.4. Diagrama de componentes DARUNE SAC

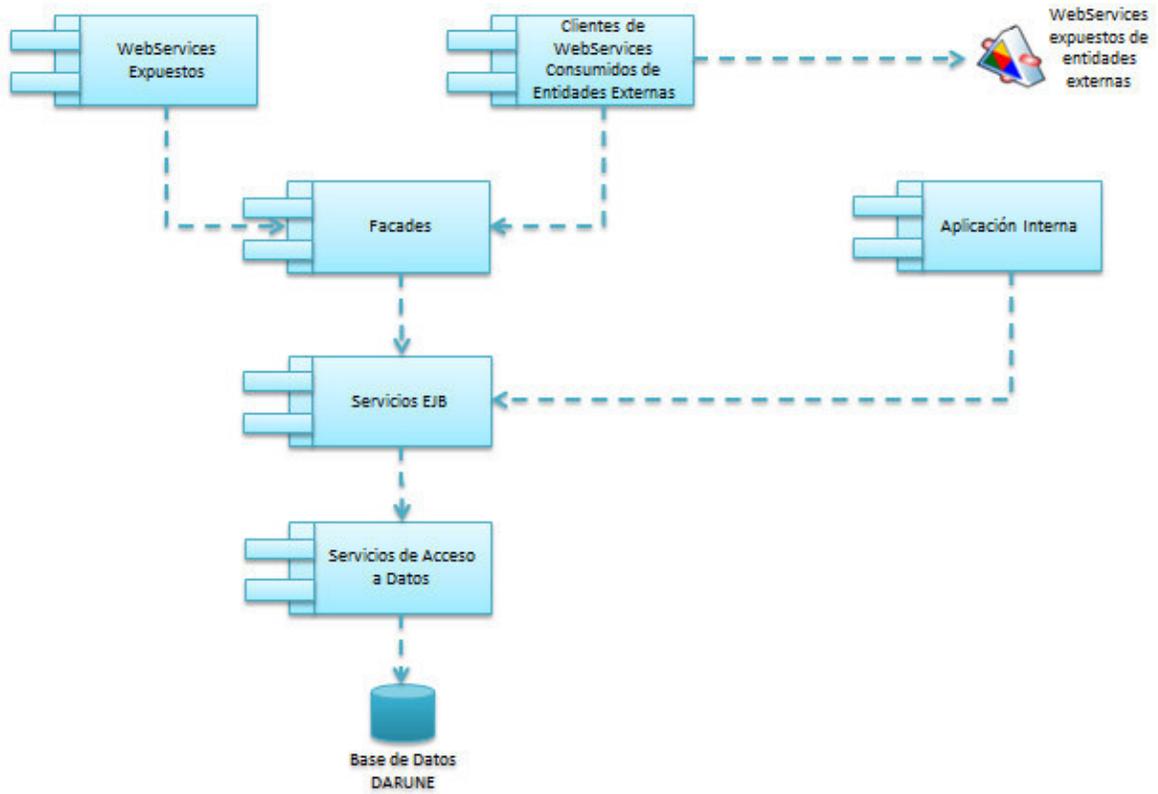


Figura 40: Diagrama de componentes Sistema Web DARUNE SAC. Elaboración propia.

4.5. Implementación del Sistema

4.5.1. Pantallas del sistema web DARSYS

4.5.1.1. Pantalla de ingreso al sistema web DARSYS

La siguiente pantalla permite el ingreso a los empleados de la empresa DARUNE SAC al sistema DARSYS.

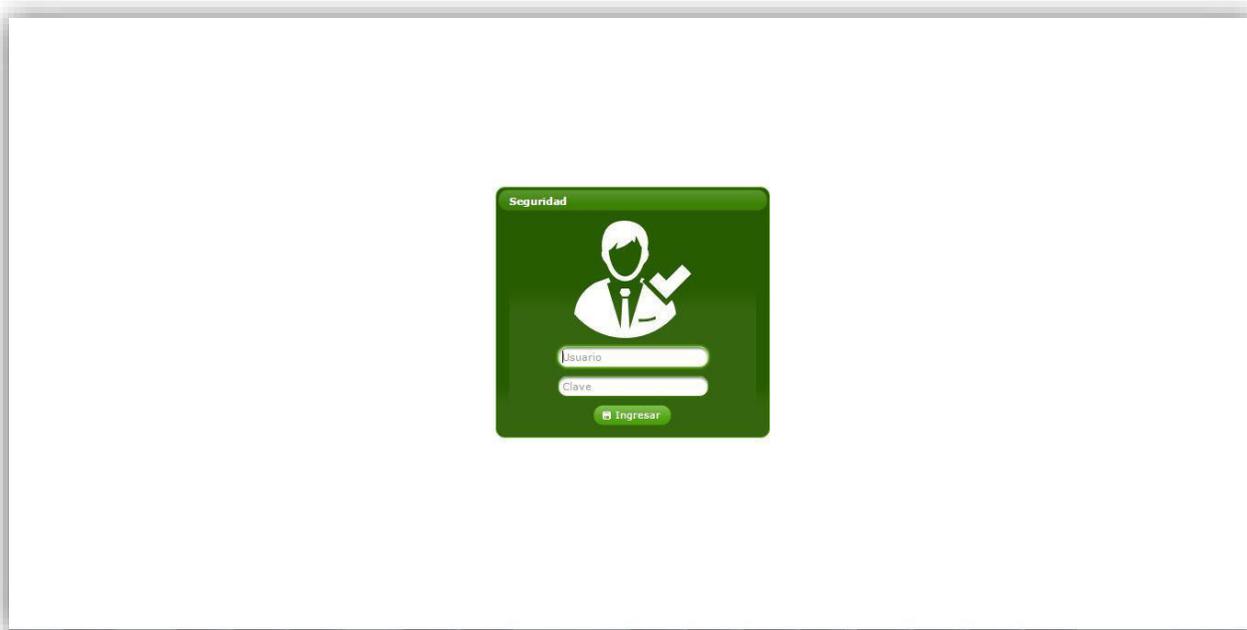


Figura 41: Pantalla de ingreso al Sistema DARSYS. Elaboración propia.

4.5.1.2. Pantalla principal del operador de ventas

La siguiente pantalla permite al operador de ventas realizar las actividades correspondientes de acuerdo a su perfil.

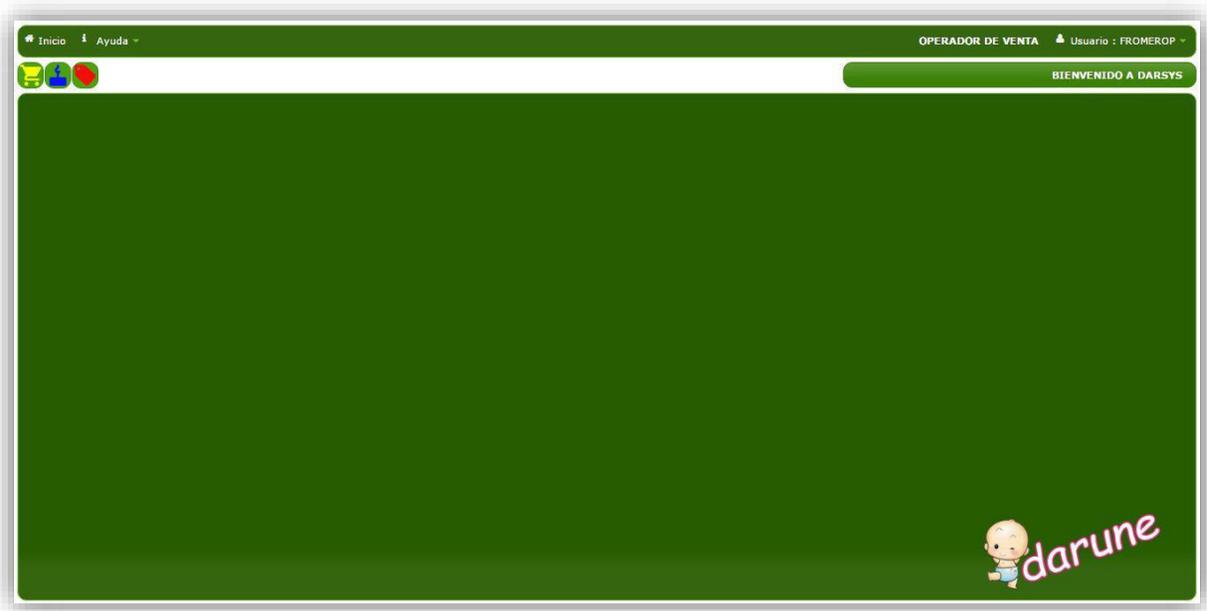


Figura 42: Pantalla principal del operador de ventas. Elaboración propia.

4.5.1.3. Pantalla de generación de preventas

La siguiente pantalla permite que el operador de ventas genere pedidos, dando como resultado un ticket de venta o proforma de acuerdo a lo solicitado por el cliente.

The screenshot displays a web application interface for sales generation. At the top, there is a navigation bar with 'Inicio' and 'Ayuda' on the left, and 'OPERADOR DE VENTA' and 'Usuario : FROMEROP' on the right. Below this is a section titled 'SECCIÓN - OPERACIONES DE VENTA'. The main area is divided into several sections: 'CLIENTE' with fields for 'Nombre o Razón Social' and 'Tipo de Cliente' (set to 'OCASIONAL'); 'PRODUCTO' with a search field; 'DOCUMENTO DE VENTA' with fields for 'Tipo de Documento' (set to 'TICKET DE VENTA'), 'Serie', and 'Número de Documento'; and 'PEDIDO' which contains a table of items. The table has columns for 'Cantidad', 'Código', 'Marca', 'Categoría', 'Descripción', 'Precio', and 'SubTotal'. Below the table, there are buttons for 'ESTADO DE DOCUMENTO - SIN ESTADO' (with a '+ Nuevo Registro' button) and 'TICKET PENDIENTES DE PAGO' (with a 'Cantidad' of 5 and a refresh icon). On the right, there is a 'GENERAR DOCUMENTO DE PREVENTA' button with a printer icon and a 'TOTAL S/. 108.80' display.

Cantidad	Código	Marca	Categoría	Descripción	Precio	SubTotal
1	6	NINET	BIBERON	BIBERON ACINTURADO JIRAFÁ CON ASAS DE SILICONA X 9 OZ	36.00	36.00
1	9	NINET	BIBERON	BIBERON BARNEY 3D X 5 OZ	19.00	19.00
2	19	AVENT	BIBERON	BIBERON NATURAL TWIN PACK X 260 ML	26.90	53.80

Figura 43: Pantalla de generación de preventas. Elaboración propia.

4.5.1.4. Pantalla búsqueda de productos

La siguiente pantalla permite buscar y seleccionar un producto de acuerdo a lo solicitado por el cliente con el fin de agregarlo al pedido.

The screenshot shows a product search interface with a green theme. At the top, there's a header "BÚSQUEDA DE PRODUCTOS". Below it, a section titled "FILTROS PARA BÚSQUEDA" contains search fields for "Descripción", "Codigo", "Marca" (set to "NINET"), and "Categoría" (set to "Seleccione..."). A "Limpiar Filtros" button is also present. The main area, titled "RESULTADO DE BÚSQUEDA", shows a table of 10 products. The table has columns for "Código", "Marca", "Categoría", "Descripción", "Precio", and "Stock". The first two rows are highlighted in yellow. Below the table, there are pagination controls and "Agregar" and "Cancelar" buttons.

	Código	Marca	Categoría	Descripción	Precio	Stock
<input type="checkbox"/>	1	NINET	BIBERON	BIBERON CRISTAL LATEX (SURT) X 5 OZ	25.00	60
<input type="checkbox"/>	2	NINET	BIBERON	BIBERON CRISTAL LATEX (SURT) X 9 OZ	29.00	50
<input type="checkbox"/>	3	NINET	BIBERON	BIBERON ACINTURADO ZEBRA CON ASAS SILICONA X 9 OZ	32.00	30
<input checked="" type="checkbox"/>	4	NINET	BIBERON	BIBERON ACINTURADO ZEBRA CON ASAS SILICONA X 8 OZ	30.00	55
<input type="checkbox"/>	5	NINET	BIBERON	BIBERON ACINTURADO ZEBRA CON ASAS SILICONA X 5 OZ	27.00	48
<input type="checkbox"/>	6	NINET	BIBERON	BIBERON ACINTURADO JIRAFRA CON ASAS DE SILICONA X 9 OZ	36.00	25
<input checked="" type="checkbox"/>	7	NINET	BIBERON	BIBERON ACINTURADO JIRAFRA CON ASAS DE SILICONA X 8 OZ	34.00	36
<input type="checkbox"/>	8	NINET	BIBERON	BIBERON ACINTURADO JIRAFRA CON ASAS DE SILICONA X 5 OZ	30.00	12
<input type="checkbox"/>	9	NINET	BIBERON	BIBERON BARNEY 3D X 5 OZ	19.00	12
<input type="checkbox"/>	10	NINET	BIBERON	BIBERON BARNEY 3D X 9 OZ	21.00	23

Figura 44: Pantalla búsqueda de productos. Elaboración propia.

4.5.1.5. Pantalla principal del operador de caja

La siguiente pantalla permite al operador de caja realizar las actividades correspondientes de acuerdo a su perfil.



Figura 45: Pantalla principal del operador de caja. Elaboración propia.

4.5.1.6. Pantalla de la bandeja del operador de caja

La siguiente pantalla permite que el operador de caja pueda visualizar todas las atenciones de acuerdo a los filtros ingresados, así como también seleccionar y atender una preventiva.

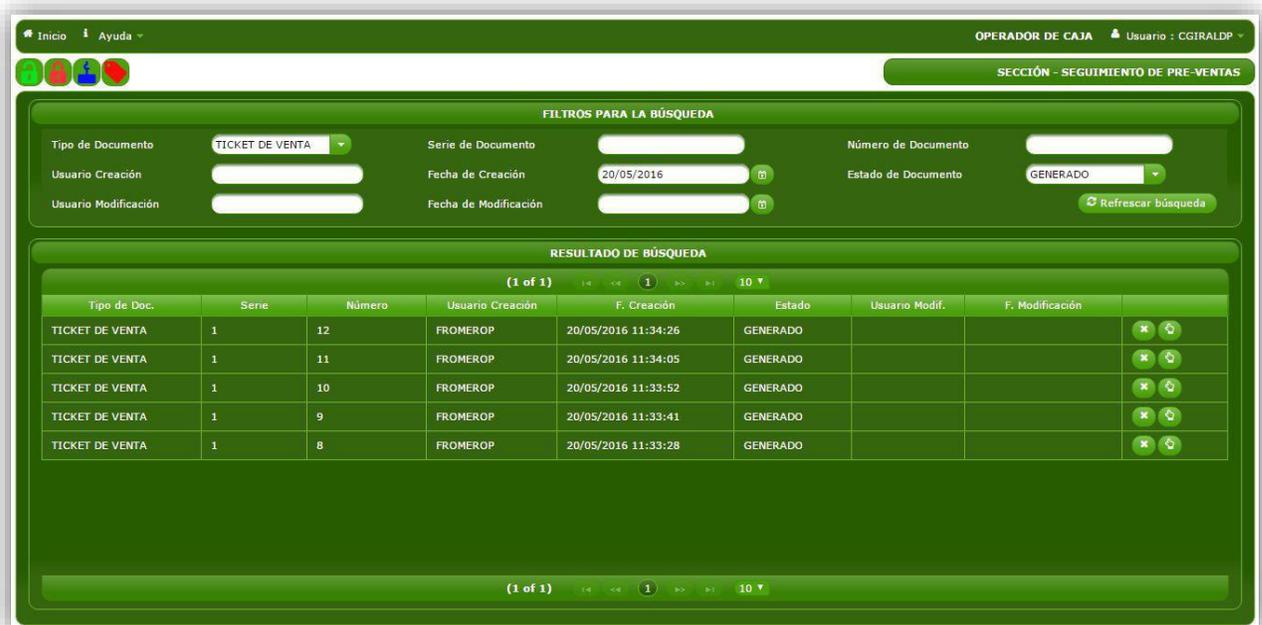


Figura 46: Pantalla de la bandeja del operador de caja. Elaboración propia.

4.5.1.7. Pantalla de atención a una preventa

La siguiente pantalla permite que el operador de caja realice el procesamiento de un pago de acuerdo al pedido generado en la preventa.

Inicio | Ayuda

OPERADOR DE CAJA | Usuario : CGIRALDP

SECCIÓN - OPERACIONES DE VENTA

CLIENTE

Nombre o Razón Social | Tipo de Cliente

PRODUCTO

Búsqueda de producto

DOCUMENTO DE VENTA

Tipo de Documento: BOLETA DE VENTA | Serie | Número de Documento

PEDIDO

Cantidad	Código	Marca	Categoría	Descripción	Precio	SubTotal	
1	1	NINET	BIBERON	BIBERON CRISTAL LATEX (SURT) X 5 OZ	25.00	25.00	-
1	2	NINET	BIBERON	BIBERON CRISTAL LATEX (SURT) X 9 OZ	29.00	29.00	-
1	3	NINET	BIBERON	BIBERON ACINTURADO ZEBRA CON ASAS SILICONA X 9 OZ	32.00	32.00	-

ESTADO DE DOCUMENTO - SIN ESTADO

Preventas | Editar venta

DOCUMENTO ORIGEN

Serie: 1 | Número: 12

PROCESAR PAGO

TOTAL S/. 86.00

Figura 47: Pantalla de atención a una preventa. Elaboración propia.

4.5.1.8. Pantalla de procesamiento de pagos

La siguiente pantalla permite que el operador de caja seleccione el medio de pago de acuerdo a lo solicitado por el cliente.

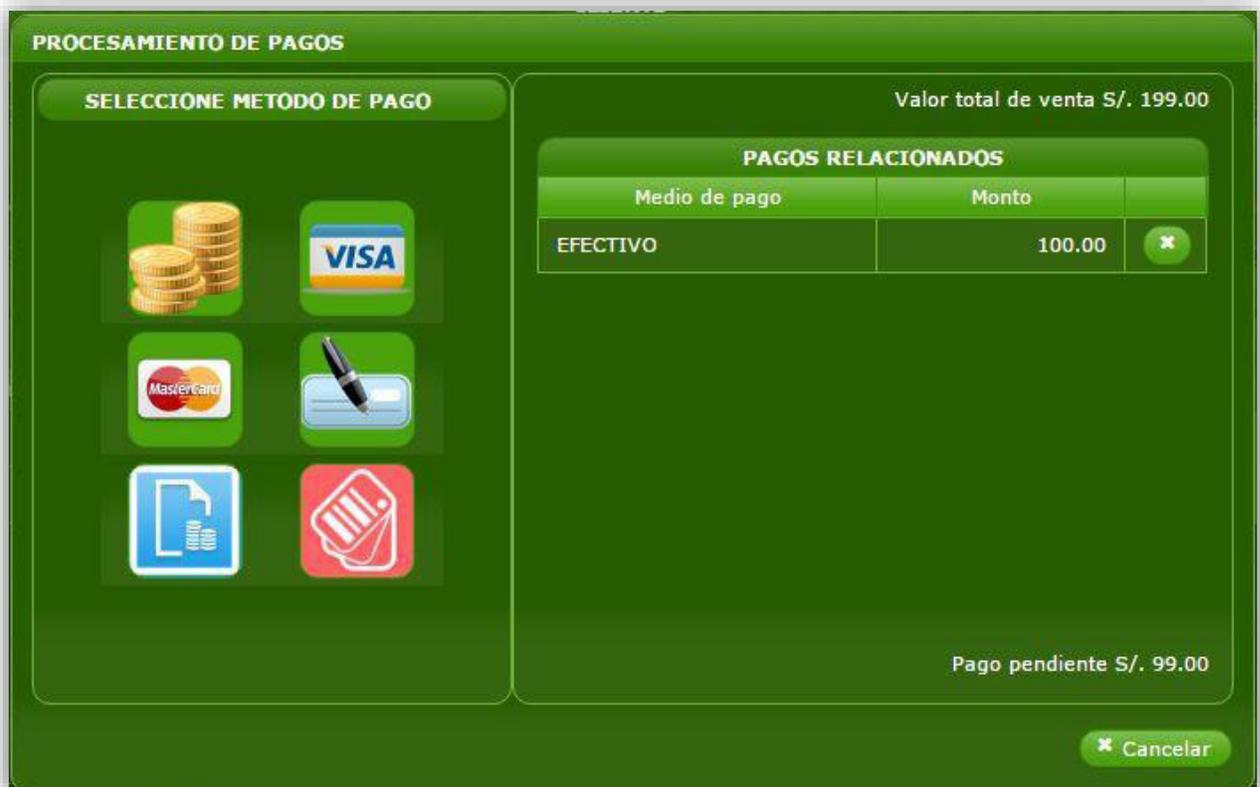


Figura 48: Pantalla de atención a una preventa. Elaboración propia.

4.5.1.9. Pantalla de pago en efectivo

La siguiente pantalla permite el procesamiento de pago en efectivo.



The image shows a software interface for processing a cash payment. The title bar at the top is green and contains the word "EFFECTIVO" in white. Below the title bar, there are two input fields. The first is labeled "Pago de cliente" and contains the value "450.00". The second is labeled "Vuelto" and contains the value "351.00". At the bottom right of the interface, there are two buttons: "Procesar" with a checkmark icon and "Cancelar" with an 'X' icon. The entire interface has a dark green background.

Figura 49: Pantalla de pago en efectivo. Elaboración propia.

4.5.1.10. Pantalla principal del operador de despacho

La siguiente pantalla permite al operador de caja realizar las actividades correspondientes de acuerdo a su perfil.

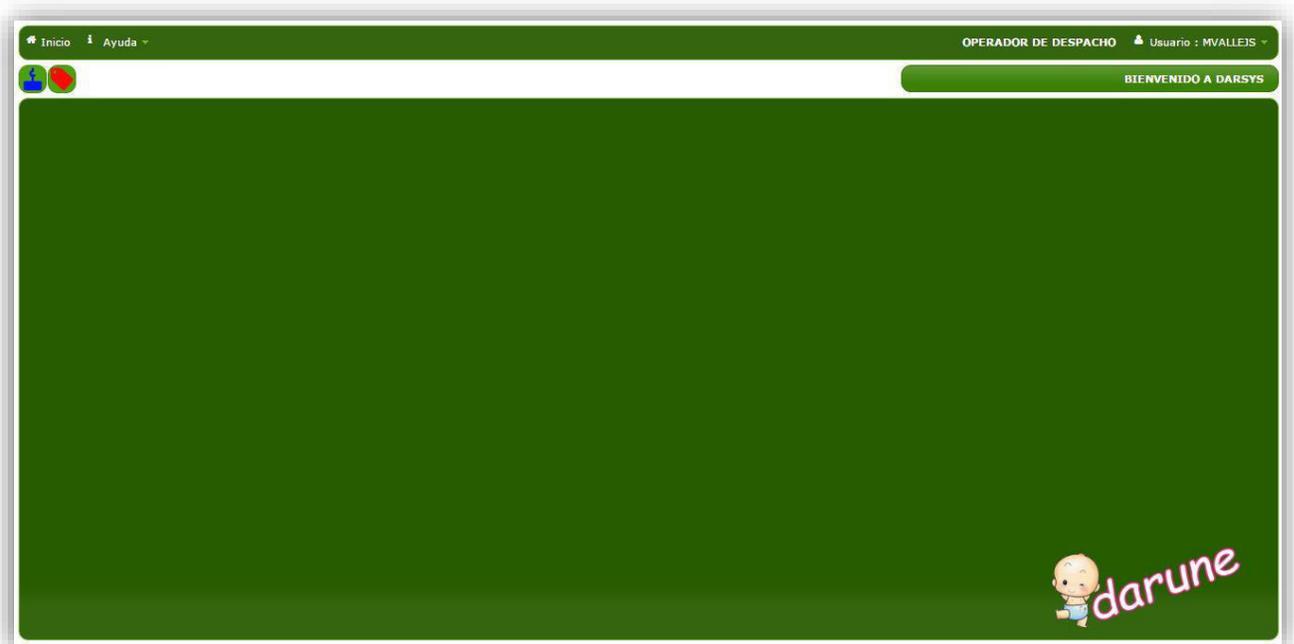


Figura 50: Pantalla principal del operador de despacho. Elaboración propia

4.5.1.11. Pantalla de la bandeja del operador de despacho

La siguiente pantalla permite al operador de despacho ver las ventas realizadas para su posterior atención, de acuerdo a los filtros ingresados.

The screenshot displays a web application interface for a dispatcher operator. At the top, there is a navigation bar with 'Inicio' and 'Ayuda' on the left, and 'OPERADOR DE DESPACHO' and 'Usuario : MVALLEJS' on the right. Below this is a section titled 'SECCIÓN - SEGUIMIENTO DE VENTAS'. The main area is divided into two parts: 'FILTROS PARA LA BÚSQUDA' and 'RESULTADO DE BÚSQUDA'. The filter section includes fields for 'Tipo de Documento' (set to 'BOLETA DE VENTA'), 'Serie de Documento', 'Número de Documento', 'Usuario Creación', 'Fecha de Creación' (20/05/2016), 'Estado de Documento' (set to 'GENERADO'), 'Usuario Modificación', and 'Fecha de Modificación'. A 'Refrescar búsqueda' button is also present. The results section shows a table with 5 rows of data, each representing a 'BOLETA DE VENTA' document. The table has columns for 'Tipo de Doc.', 'Serie', 'Número', 'Usuario Creación', 'F. Creación', 'Estado', 'Usuario Modif.', and 'F. Modificación'. Each row also has a small circular icon with a refresh symbol on the right side. The table is paginated, showing '(1 of 1)' results.

Tipo de Doc.	Serie	Número	Usuario Creación	F. Creación	Estado	Usuario Modif.	F. Modificación
BOLETA DE VENTA	1	7	CGIRALDP	20/05/2016 11:43:15	GENERADO		
BOLETA DE VENTA	1	6	CGIRALDP	20/05/2016 11:43:05	GENERADO		
BOLETA DE VENTA	1	5	CGIRALDP	20/05/2016 11:42:53	GENERADO		
BOLETA DE VENTA	1	4	CGIRALDP	20/05/2016 11:42:35	GENERADO		
BOLETA DE VENTA	1	3	CGIRALDP	20/05/2016 11:40:31	GENERADO		

Figura 51: Pantalla de la bandeja del operador de despacho. Elaboración propia.

4.5.1.12. Pantalla de atención de un despacho

La siguiente pantalla permite al operador la atención de un despacho de una venta pendiente del mismo.

The screenshot shows a software interface for handling a shipment. The interface is green-themed and includes sections for Client, Product, Document of Sale, and Order. A table lists items with columns for Quantity, Code, Brand, Category, Description, Price, and SubTotal. The total amount is S/. 172.80.

CLIENTE

Nombre o Razón Social: Tipo de Cliente:

PRODUCTO

Búsqueda de producto:

DOCUMENTO DE VENTA

Tipo de Documento: BOLETA DE VENTA Serie: 1 Número de Documento: 7

PEDIDO

(1 of 1) 10

Cantidad	Código	Marca	Categoría	Descripción	Precio	SubTotal
1	6	NINET	BIBERON	BIBERON ACINTURADO JIRAFÁ CON ASAS DE SILICONA X 9 OZ	36.00	36.00
1	9	NINET	BIBERON	BIBERON BARNEY 3D X 5 OZ	19.00	19.00
2	19	AVENT	BIBERON	BIBERON NATURAL TWIN PACK X 260 ML	26.90	53.80
1	4	NINET	BIBERON	BIBERON ACINTURADO ZEBRA CON ASAS SILICONA X 8 OZ	30.00	30.00
1	7	NINET	BIBERON	BIBERON ACINTURADO JIRAFÁ CON ASAS DE SILICONA X 8 OZ	34.00	34.00

(1 of 1) 10

ESTADO DE DOCUMENTO - GENERADO

Ventas

DOCUMENTO ORIGEN

Serie 1 Número 8

DESPACHAR VENTA

TOTAL S/. 172.80

Figura 52: Pantalla de atención de un despacho. Elaboración propia.

4.6. Pruebas del Sistema

En este punto se realizarán las pruebas al proceso después de la sistematización.

4.6.1. Plan de pruebas

4.6.1.1. Objetivos

- Identificar la información existente en el Proyecto y los componentes que deben ser probados.
- Listar los principales requisitos a probar.
- Definir las estrategias de prueba que deben emplearse.
- Identificar los recursos necesarios que puedan requerirse.
- Listar las actividades a realizar en el proceso de pruebas.
- Listar los artefactos entregables del proceso de pruebas

4.6.1.2. Características que no serán probadas

- Requisitos funcionales: No aplica porque es un sistema nuevo, se probará todo.
- Requisitos no funcionales: No aplica porque es un sistema nuevo, se probará todo.

4.6.1.3. Criterios de aceptación/falla

En esta actividad se va a describir las categorías que pueden ser asignadas a los resultados de pruebas en un caso de prueba.

- **Éxito:** El resultado de la prueba indica es conforme al resultado esperado.
- **Acceptable:** El resultado de la prueba indica que el sistema difiere de la especificación aceptada pero es aceptable, no son necesarios cambios en la aplicación, pero requiere un cambio en la especificación funcional.
- **Tolerable:** El resultado de la prueba es incorrecto, podría ser aceptada pero la falla encontrada deberá ser rectificadas en el periodo de tiempo acordado.
- **Intolerable:** El resultado de la prueba es incorrecto, y la falla debe ser corregida antes de concluir la fase de prueba.
- **Error:** El resultado de la prueba observado es correcto, pero el resultado esperado de acuerdo a los scripts de pruebas son incorrectos. Seguidamente se mencionan los tipos de error:
 - **Error crítico:** produce una parada en el sistema y la imposibilidad de continuar con las pruebas.
 - **Error grave:** se produce si en un área crítica del sistema se obtiene un fallo, lo que degrada mucho la calidad del sistema
 - **Error leve:** se produce si en un área no crítica del sistema se obtiene un fallo.
 - **Error mínimo:** se produce ante una salida inesperada que a simple vista no llame mucho la atención.

4.6.1.4. Criterios de suspensión y reanudación

Las pruebas se suspenderán ante los siguientes supuestos:

- Existe un fallo crítico en el sistema y produce que no se pueda continuar evaluando.

- Falla la base de datos para la ejecución de las pruebas.
- Falla un caso de prueba de mayor criticidad.

Las pruebas se reanudarán en los siguientes supuestos:

- Se corrigen los fallos críticos.
- Se restaura el funcionamiento de la base de datos.

4.6.2. Resultados de pruebas

En este punto se realizará un Diagrama de Análisis de Procesos para el proceso de ventas sistematizado de la empresa DARUNE SAC, con lo cual notaremos las mejoras respecto al proceso inicialmente analizado en el punto 4.4.

Ítem	Descripción de actividades	Duración (seg)	 Operación	 Inspección	 Transporte	 Demora o espera	 Almacenamiento
1	Solicitar precio de producto	5	X				
2	Verificar stock	10		X			
3	Emitir precio de producto	5			X		
4	Evaluar precio de producto	5		X			
5	Emitir conformidad de precio	3	X				
6	Agregar producto al pedido	5	X				
7	Emitir proforma con pedido	3			X		
8	Recepcionar proforma con pedido	2	X				
9	Presentar proforma con pedido	2			X		
10	Recepcionar la proforma con pedido	2	X				
11	Validar proforma con pedido	5		X			
12	Consultar el tipo de pago a realizar	3	X				
13	Emitir elección de comprobante de pago	3	X				
14	Solicitar pago	2	X				
15	Realizar pago	10	X				
16	Procesar pago	10	X				
17	Registrar venta	3	X				
18	Elaborar comprobante de pago	4	X				
19	Emitir comprobante de pago	3			X		
20	Recepcionar comprobante de pago	2	X				
21	Presentar comprobante de pago	2			X		
22	Recepcionar comprobante	2	X				
23	Validar comprobante de pago	5		X			
24	Seleccionar productos	20	X				
25	Entregar comprobante y productos emp	10			X		
26	Recepcionar comprobante y productos	5	X				
	TOTAL (seg)	131	16	4	6		
			TOTAL (min)	2.18			Actividad automatizada

Tabla 23: Diagrama de análisis del proceso de ventas DARUNE SAC - TOBE. Elaboración propia.

De la misma manera en que se realizó el análisis del proceso de ventas inicial de la empresa DARUNE SAC o proceso de ventas ASIS, se realizará el siguiente cuadro en el cual se estimará el costo total para el proceso de ventas DARUNE SAC ya sistematizado.

Ítem	Descripción de actividades	Duración (seg)	Costo promedio rol/actividad (min)	Costo por actividad (S/.)
1	Solicitar precio de producto	5		
2	Verificar stock	10	0.08	0.01
3	Emitir precio de producto	5	0.08	0.01
4	Evaluar precio de producto	5		
5	Emitir conformidad de precio	3		
6	Agregar producto al pedido	5	0.08	0.01
7	Emitir proforma con pedido	3	0.08	0.00
8	Recepcionar proforma con pedido	2		
9	Presentar proforma con pedido	2		
10	Recepcionar la proforma con pedido	2	0.1	0.00
11	Validar proforma con pedido	5	0.1	0.01
12	Consultar el tipo de pago a realizar	3	0.1	0.01
13	Emitir elección de comprobante de pago	3		
14	Solicitar pago	2	0.1	0.00
15	Realizar pago	10		
16	Procesar pago	10	0.1	0.02
17	Registrar venta	3	0.1	0.01
18	Elaborar comprobante de pago	4	0.1	0.01
19	Emitir comprobante de pago	3	0.1	0.01
20	Recepcionar comprobante de pago	2		
21	Presentar comprobante de pago	2		
22	Recepcionar comprobante	2	0.07	0.00
23	Validar comprobante de pago	5	0.07	0.01
24	Seleccionar productos	20	0.07	0.02
25	Entregar comprobante y productos empaquetados	10	0.07	0.01
26	Recepcionar comprobante y productos empaquetados	5		
TOTAL (seg)		131		0.13

Tabla 24: Costo del proceso de ventas sistematizado DARUNE SAC. Elaboración propia.

A comparación del costo del proceso inicial se puede verificar que presenta una reducción de S/. 0.26 nuevos soles por cada venta realizada, es decir ahora el nuevo costo será de S/. 0.13 nuevos soles por este proceso.

En base al análisis y a las pruebas realizadas podemos notar que se logró reducir el tiempo empleado en la realización de una venta, de 5.33 minutos a 2.18 minutos, es decir se llegó a reducir el tiempo en un 59%, además de eliminar 2 actividades que no le aportaban valor al negocio.

Actividad		Proceso inicial	Proceso sistematizado
Operación		18	16
Inspección		4	4
Trasporte		6	6
Espera		0	0
Almacenamiento		0	0
Total tiempo (min)		5.33	2.18
Total costo (S/.)		0.39	0.13

Tabla 25: Resumen de análisis de pruebas del proceso. Elaboración propia.

4.7. Análisis costo beneficio

4.7.1. Inversión Inicial

4.7.1.1. Costos de servicios

Ítem	Descripción	Monto al mes	Número de meses	Subtotal
1	Transporte	S/. 100.00	4	S/. 400.00
2	Internet	S/. 40.00	4	S/. 160.00
3	Luz	S/. 30.00	4	S/. 120.00
4	Telefonía	S/. 50.00	4	S/. 200.00
			TOTAL	S/. 880.00

Tabla 26: Costos de servicios. Elaboración propia.

4.7.1.2. Costos de materiales

Ítem	Descripción	Cantidad	Unidad	Precio por unidad	Subtotal
1	Impresiones	600	Hojas	S/. 0.20	S/. 120.00
2	Fotocopias	150	Hojas	S/. 0.05	S/. 7.50
3	Papel Bond	1	Millar	S/. 20.00	S/. 20.00
4	Insumos de oficina	1	Kit	S/. 50.00	S/. 50.00
				TOTAL	S/. 197.50

Tabla 27: Costos de materiales. Elaboración propia.

4.7.1.3. Costos de personal

Ítem	Descripción	Meses	Días por mes	Horas por día	Costo por hora	Subtotal
1	Análisis y Diseño	3	20	4	S/. 17.00	S/. 4,080.00
2	Implementación	2.5	20	5	S/. 13.00	S/. 3,250.00
3	Instalación y pruebas	2	20	4	S/. 14.00	S/. 2,240.00
4	Capacitación	1	20	4	S/. 17.00	S/. 1,360.00
					TOTAL	S/. 10,930.00

Tabla 28: Costos de personal. Elaboración propia.

4.7.1.4. Costos de software

Ítem	Descripción	Subtotal
1	Eclipse Java EE IDE Luna 4.4.2	S/. 0.00
2	MySQL Workbench 6.3	S/. 0.00
3	Android Studio 2.1	S/. 0.00
2	Servidor de aplicaciones Jboss AS 7.1	S/. 0.00
3	XAMPP Control Panel v3.2.1	S/. 0.00
3	Base de Datos MySQL 5.6.20	S/. 0.00
4	Lenguaje de programación JAVA	S/. 0.00
5	Sistema Operativo Ubuntu 14.04	S/. 0.00
6	Browser	S/. 0.00
	TOTAL	S/. 0.00

Tabla 29: Costos de software. Elaboración propia.

4.7.1.5. Costos de hardware

Ítem	Descripción	Cantidad	Precio unitario	Subtotal
1	PC Core i3 (D) 500 GB/(M) 4GB/ Monitor 18.5	2	S/. 1,600.00	S/. 3,200.00
2	Impresora Matricial (Bol/Fac)	1	S/. 1,200.00	S/. 1,200.00
3	Impresora Térmica	1	S/. 700.00	S/. 650.00
			TOTAL	S/. 5,050.00

Tabla 30: Costos de hardware. Elaboración propia.

4.7.2. Gastos concurrentes u operativos

Ítem	Descripción	Costo al mes	Costo semestral
1	Mantenimiento de licencias	S/. 0.00	S/. 0.00
2	Costos de papel térmico (tickets)	S/. 7.00	S/. 42.00
3	Mantenimiento de servidores	S/. 65.00	S/. 390.00
		TOTAL	S/. 432.00

Tabla 31: Costos concurrentes u operativos. Elaboración propia.

4.7.3. Beneficios

En este apartado se listan los beneficios que han sido obtenidos teniendo en cuenta el resumen de análisis de pruebas del proceso que se puede ver en la Tabla 20 del presente documento.

Reducción de costos por atención			
Número promedio de ventas en un día	Ahorro en una venta	Ahorro en un mes	Ahorro en un semestre
120	S/. 0.26	S/. 936.00	S/. 5,616.00
Reducción de costos por eliminación de proformas			
Costo de proformas (1 millar)	Ahorro en una venta	Ahorro en un mes	Ahorro en un semestre
S/. 115.00	S/. 0.12	S/. 414.00	S/. 2,484.00
		TOTAL	S/. 8,100.00

Tabla 32: Beneficios implementando la sistematización. Elaboración propia.

4.7.4. Flujo de caja proyectado

	Semestre 0	Semestre 1	Semestre 2	Semestre 3	Semestre 4	Semestre 5	TOTAL
Inversión inicial							
Costos de servicios	S/. 880.00						S/. 880.00
Costos de materiales	S/. 197.50						S/. 197.50
Costos de personal	S/. 10,930.00						S/. 10,930.00
Costos de software	S/. 0.00						S/. 0.00
Costo de hardware	S/. 5,050.00						S/. 5,050.00
Total de inversión inicial	S/. 17,057.50						S/. 17,057.50
Costos operativos o concurrentes							
Mantenimiento de licencias de software		S/. 0.00					
Costo de papel térmico para ticket		S/. 42.00	S/. 210.00				
Mantenimiento de servidores		S/. 390.00	S/. 1,950.00				
Total de costos operativos o concurrentes		S/. 432.00	S/. 2,160.00				
Gastos totales	S/. 17,057.50	S/. 432.00	S/. 19,217.50				
Beneficios							
Reducción de costos por tiempo de atención		S/. 5,616.00	S/. 28,080.00				
Reducción de costos en proformas		S/. 2,484.00	S/. 12,420.00				
Total bruto beneficios		S/. 8,100.00	S/. 40,500.00				
Total neto beneficios		S/. 7,668.00	S/. 38,340.00				
FLUJO DE CAJA NETO	-S/. 17,057.50	S/. 7,668.00					

Tabla 33: Flujo de caja proyectado en el periodo de 5 semestres. Elaboración propia.

4.7.5. Valor Actual Neto (VAN)

A continuación se calculará el valor actual neto en base al flujo de caja presentado en la Tabla 28.

$$VAN = -17057.50 + \frac{7668}{1.15} + \frac{7668}{1.3225} + \frac{7668}{1.520875} + \frac{7668}{1.74900625} + \frac{7668}{2.01135719}$$

Efectuando el cálculo tendremos que el valor actual neto es igual a S/. 8646.825292

Del valor calculado, notamos que es positivo, por lo que el proyecto es viable, ya que la inversión produciría ganancias por encima de la rentabilidad exigida.

4.7.6. Tiempo de retorno de inversión

También conocido como el periodo de recuperación de la inversión (PRI). Es muy importante estimar en que tiempo la inversión será retornada, este cálculo lo haremos en base al flujo de caja acumulado proyectado en el periodo de 5 semestres.

	Semestre 0	Semestre 1	Semestre 2	Semestre 3	Semestre 4	Semestre 5
FLUJO DE CAJA NETO	-S/. 17,057.50	S/. 7,668.00	S/. 7,668.00	S/. 7,668.00	S/. 7,668.00	S/. 7,668.00
FLUJO DE CAJA NETO ACUMULADO	S/. -17,057.50	S/. -9,389.50	S/. -1,721.50	S/. 5,946.50	S/. 13,614.50	S/. 21,282.50

Tabla 34: Flujo de caja acumulado en un periodo de 5 semestres. Elaboración propia.

Podemos notar que después del semestre 2 y terminando el semestre 3, ya se terminó de pagar la inversión e incluso se obtiene un excedente de S/. 5,946.50, por lo que para obtener el tiempo de retorno de inversión interpolaremos el valor del semestre 2 y el semestre 3.

$$PRI = 2 + \frac{1721.50}{1721.50 + 5946.50}$$

Después de realizar los cálculos correspondientes se obtiene que el periodo en el cual se terminará de recuperar la inversión realizada será de 2.22 semestres o 13.32 meses.

Capítulo V. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Luego de desarrollar el presente trabajo de investigación se obtuvieron las siguientes conclusiones:

- Se logró sistematizar el proceso de ventas de la PYME de nuestro caso de estudio, basándonos en una arquitectura orientada a servicios. Esto fue posible debido a que se cumplieron los objetivos específicos trazados, los cuales garantizaron el éxito del presente trabajo de investigación.
- Se logró mejorar el tiempo total del proceso de ventas de la PYME de nuestro caso de estudio basándonos en una arquitectura orientada a servicios, ya que el tiempo del mencionado proceso se redujo de 5.33 a 2.18 minutos, lo que representa una reducción del 59% de mencionado indicador.
- Se logró demostrar que el proceso de ventas de la PYME DARUNE SAC genera ganancias después de 13.32 meses una vez realizada la sistematización, dicho tiempo es en el que retorna la inversión realizada por la empresa, esto en base al análisis costo beneficio realizado.
- Se logró modelar el negocio, esto se llevó a cabo con la elaboración del Modelo de Casos de Uso del Negocio y el Modelo de Análisis de Negocio: artefactos definidos dentro de la disciplina de Modelado de Negocio del RUP.

- Se logró identificar los requerimientos funcionales y no funcionales asociados al proceso de ventas de nuestro caso de estudio, esto se llevó a cabo con la elaboración de la Especificación de Casos de Usos y Modelos de Casos de Usos definidos para este trabajo de investigación.
- Se logró definir la arquitectura del sistema web para que permita la interoperabilidad con la aplicación móvil, esto en base a que se dividió la estructura del sistema web en componentes relacionados y posicionados en capas.
- Se logró la implementación del sistema web, el cual soporta el proceso de ventas definido, así como también la implementación del aplicativo móvil el cual se conecta al sistema web y realiza una solicitud de crédito. Esto en base a la implementación de un middleware que se encarga de exponer los servicios web para que sean consumidos por aplicaciones en diferentes plataformas.
- Se logró realizar el análisis costo beneficio del proyecto dando como resultado la viabilidad de la propuesta, ya que se demostró con este análisis que la inversión producirá ganancias por encima de la rentabilidad exigida.

5.2. Recomendaciones

En base al trabajo desarrollado presento las siguientes recomendaciones que estoy seguro servirá como consejos a quienes realicen trabajos similares o afines.

- El alcance del presente trabajo de investigación se enfocó en la implementación de un fragmento del proceso de ventas de la empresa DARUNE SAC, una PYME que se desarrolla el ámbito comercial, por lo que se recomendaría que se abarque el proceso contemplado en el análisis en su totalidad.
- Debido a que se maneja un universo amplio y variado de productos para la venta dentro de la organización DARUNE SAC, es recomendable implementar una interfaz que nos permita conectar con un lector de códigos de barras, esto haría más eficiente las consultas de información de un producto determinado.
- La solución desarrollada para el proceso de ventas ya descrito se basó directamente en cubrir actividades de los operadores dentro de la organización, vale decir que se implementó una aplicación a la cual podemos caracterizar como privada, por lo que es recomendable realizar una aplicación pública, con las etapas que esta requiera, con el fin de exponerla y/o dirigirla a potenciales clientes. En dicha aplicación ellos podrían realizar actividades como consultas de diferente ámbito según el análisis lo defina.

- Si bien es cierto dentro del presente trabajo de investigación se desarrolló una pequeña aplicación móvil, específicamente una aplicación que hace uso de un servicio de la organización DARUNE SAC, la cual está desarrollada para dispositivos que poseen el sistema operativo Android, es recomendable que se implemente una aplicación para dispositivos móviles dirigida a los potenciales clientes, la cual pueda ser instalada en los dispositivos que cuenten con los sistemas operativos móviles más usados, los cuales son Android y iOS.

REFERENCIAS BIBLIOGRÁFICAS.

1. Bocchio, F. (2014). Modelo Cloud Computing como Alternativa para Escalabilidad y Recuperación de Desastres. Tesis de Maestría en Ingeniería de Sistemas de Información. Buenos Aires: Universidad Tecnológica Nacional de Argentina.
2. Delgado, A. (2007). Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture). Tesis de Maestría en Informática. Montevideo: Universidad de la Republica.
3. Rodrigez, J. (2013). Análisis, Diseño e Implementación de un sistema de información para una tienda de ropa con enfoque al segmento juvenil. Tesis de Titulación, Lima: Pontifica Universidad Católica del Perú.
4. Becerra, C. (2013). Análisis, Diseño e Implementación de un Sistema de Comercio Electrónico integrado con una Aplicación móvil para la reserva y venta de pasajes de una empresa de transporte interprovincial. Tesis de Titulación, Lima: Pontifica Universidad Católica del Perú.
5. Hewitt, E. (2009). Java SOA Cookbook, Internacional: O'Reilly Media
6. Wikipedia. (2016). Arquitectura Orientada a Servicios. Obtenida el 3 de enero del 2016, de https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

7. Object Management Group (OMG). (2013). Obtenida el 3 de enero del 2016, de <http://www.omg.org/technology/readingroom/SOA.htm>
8. SG BUZZ. (2010). Obtenida el 15 de diciembre del 2015, de <http://sg.com.mx/revista/27/arquitectura-software#.VqAhtvnhDIU>
9. TopLine Systems (2010). Obtenida el 20 de diciembre del 2015, de <http://www.topline-systems.com/solutionPlatform/architecture.asp>
10. Accenture. (2008). Arquitectura Orientada a Servicios (SOA) Cómo reformular la Arquitectura Corporativa para alcanzar el alto rendimiento. Centro de Alto Rendimiento de Accenture, consultada el 2 de enero del 2016, de http://www.kybele.etsii.urjc.es/docencia/IS_LADE/2012-2013/Material/CAR%20Accenture%20-%20SOA%5B1%5D.pdf
11. Microsoft Corporation. (2006). La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Obtenida el 29 de diciembre del 2015, de http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf
12. Instituto Nacional de Estadística e Informática. (2015). Peru: Estructura Empresarial, 2014. Obtenida el 5 de diciembre del 2015, de

https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1262/index.html

13. Rational. (2001). Rational Unified Process Best Practices for Software Development Teams. Obtenida el 10 de diciembre del 2015, de https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

14. IBM Rational Software. (2006). Using a Single Business Pattern with the Rational Unified Process. Obtenida el 29 de diciembre del 2015, de <http://www.redbooks.ibm.com/redpapers/pdfs/redp3877.pdf>

15. Universidad de Cádiz de España (2007). Gestión de Procesos en la UCA, guía para identificación y análisis de procesos, Obtenida el 4 de diciembre del 2015, de http://servicio.uca.es/personal/guia_procesos