



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA
TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE
ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON**

AUTORES: PAUCAR GUANOLUISA MARCO VINICIO

QUISHPI CONDO DANIEL

DIRECTOR: ING. RON EGAS MARIO BERNABE

SANGOLQUÍ

2016



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON**” realizado por los señores **PAUCAR GUANOLUISA MARCO VINICIO** y **QUISHPI CONDO DANIEL**, ha sido revisado en su totalidad y analizado por el Software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores **PAUCAR GUANOLUISA MARCO VINICIO** y **QUISHPI CONDO DANIEL** para que lo sustente públicamente.

Sangolquí, Febrero del 2016

Atentamente

Ing. Ron Egas Mario Bernabe

DIRECTOR



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, **PAUCAR GUANOLUISA MARCO VINICIO** con cédula de identidad N° 170770014-0 y **QUISHPI CONDO DANIEL** con cédula de identidad N°171338523-3, declaramos que este trabajo de titulación **“DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas. Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, febrero de 2016.

PAUCAR MARCO

C.C.170770014-0

QUISHPI DANIEL

C.C.171338523-3



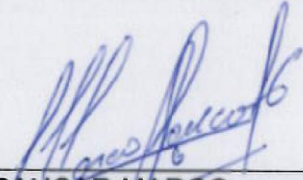
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotros, **PAUCAR GUANOLUISA MARCO VINICIO Y QUISHPI CONDO DANIEL** autorizamos a la UNIVERSIDAD DE LAS FUERZAS ARMADAS (ESPE) la publicación en la Biblioteca Virtual de la Institución, del trabajo titulado “**DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON**”, que es de nuestra propia autoridad y responsabilidad.

Sangolquí, Febrero 2016



PAUCAR MARCO
C.C.170770014-0



QUISHPI DANIEL
C.C.171338523-3

DEDICATORIA

El presente proyecto es el resultado de largas horas de investigación, esfuerzo y desarrollo.

Este proyecto está dedicado a mi familia y principalmente a mi hijo Jhordan Vinicio Paucar Balseca, quién en el transcurso de estos últimos años me enseñó que la vida te puede arrebatar todo menos la voluntad de seguir.

A mis padres, por apoyarme en cada momento y darme un buen ejemplo de superación.

A mi familia Paterna y amigos en general que de una u otra manera estuvieron en este proceso desde que ingrese a la Escuela Politécnica de Ejército y ahora Escuela politécnica de las Fuerzas Armadas.

PAUCAR GUANOLUISA MARCO VINICIO

DEDICATORIA

El presente trabajo dedico a mi Dios quién supo guiarme por el buen camino y ha estado conmigo a cada paso que doy, por darme fuerzas para seguir adelante y no desmayar en los problemas y adversidades de la vida.

A mis padres por ser los pilares más importantes en mi vida, brindándome sus consejos, apoyo, comprensión, amor, ayudándome en los momentos más difíciles de mi vida.

A mi esposa y mis hijos por compartir momentos significativos conmigo y por siempre estar dispuesta a escucharme y ayudarme, porque sin el apoyo y comprensión de ellos no hubiera logrado esta meta.

QUISHPI CONDO DANIEL

AGRADECIMIENTO

A mi Dios que supo guiar mi camino en el transcurso de este trayecto que en momentos con desmayos siempre me brindo una nueva esperanza.

A mi familia por ser un apoyo primordial que al pasar de los años demostró siempre su apoyo y me motivo para culminar la carrera de Ingeniería de Sistemas e Informática.

A mis padres por inculcarme valores los cuales en el transcurso de la vida me han sido muy útiles.

A mi Director de Tesis Ing. Mario Ron quién en las aulas compartió sus conocimientos y especialmente sus experiencias.

A la Dra. Milagritos Santamaría quién desde su escritorio y muchas horas de conversación me motivo a terminar la carrera.

A la empresa Tableros Édison quién confió en nosotros para desarrollar el Sistema SISPRO.

PAUCAR GUANOLUISA MARCO VINICIO

AGRADECIMIENTO

El presente trabajo de tesis quiero agradecer a mi familia por su apoyo incondicional en cada etapa de mi vida, a mi esposa y a mis hijos .A Dios por brindarme fuerza y sabiduría, a mis amigos y amigas con los cuales he compartido momentos buenos y malos convirtiéndose en una parte muy importante de mi vida.

Al ing. Mario Ron, quién con sus valiosas enseñanzas y consejos, han aportado en la realización de la presente tesis, lo cual me ha ayudado a mi crecimiento personal y profesional.

QUISHPI CONDO DANIEL

ÍNDICE DE CONTENIDO

CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
AGRADECIMIENTO.....	viii
ÍNDICE DE CONTENIDO.....	ix
ÍNDICE TABLAS.....	xiii
ÍNDICE FIGURA.....	xiii
RESUMEN.....	xv
ABSTRACT.....	xvi
CAPÍTULO I.....	1
ASPECTOS GENERALES.....	1
1.1. Planteamiento del Problema.....	1
1.2. Antecedentes.....	1
1.3. Justificación e importancia.....	3
1.4. Objetivos.....	4
1.4.1. Objetivo General.....	4
1.4.2. Objetivos Específicos.....	4
1.5. Alcance.....	4
1.6. Metodología.....	6
1.7. Factibilidad.....	6
1.7.1. Factibilidad Técnica.....	6
1.7.2. Factibilidad Operativa.....	7
1.7.3. Factibilidad Económica.....	7
1.8. Cronograma.....	9
CAPÍTULO II.....	10
MARCO TEÓRICO.....	10
2.1. Plataforma Java.....	10
2.1.1 Introducción.....	10

2.1.2 Versiones de Java	10
Las versiones de java se indica en la Tabla 2. (Oracle, s.f.).....	10
2.1.3 Tecnología Java.	11
2.1.4 Características.....	11
2.1.5 El lenguaje de programación.	11
2.1.6 Paradigma de la programación orientada a objetos	12
2.2. IDE Eclipse	14
2.2.1. Introducción.....	14
2.2.2. Arquitectura	15
2.2.3. Versiones de Eclipse.....	15
2.2.3. Características.....	16
2.3. Jboss.....	17
2.3.1 Características.....	17
2.3.2 Servidor de aplicaciones Jboss.....	17
2.4. PostgreSQL.....	18
2.4.1 Características.....	18
2.4.2 Componentes de PostgreSQL.	19
2.4.3 Alta Concurrencia.....	20
2.4.4 Limitaciones.	21
2.5. Lenguaje de Modelamiento Unificado (UML).....	22
2.5.1 Diagrama de clases.	22
2.5.2 Diagrama de Objetos.	23
2.5.3 Diagrama de Secuencia.	24
2.5.4 Diagrama de Comunicación.	25
2.5.5 Diagrama de Caso de Uso.	25
2.5.6 Diagrama de estado.	26
2.5.7 Diagrama de Actividades.	26
2.5.8 Diagrama de Componentes.	27
2.5.9 Diagrama de Despliegue.	28
2.6 Metodologías ágiles.	28
2.7 Metodología Scrum.....	29
2.7.1 Funcionalidad y proceso de Scrum.	29
2.7.2 Sprint.....	29

2.7.3 Proceso.....	30
2.7.4 Definición del proyecto (Product Backlog).	30
2.7.5 Definición del Sprint (Sprint Backlog).	30
2.7.6 Ejecución del Sprint.....	31
2.7.7 Entrega.....	31
2.7.8 Evolución del proyecto (Burn down):	31
2.7.9 Reuniones de trabajo en un contexto SCRUM.....	31
2.7.9.1 Planificación de sprint:.....	31
2.7.9.2 Reunión diaria:.....	32
2.7.10 Revisión de sprint:.....	32
2.7.11 Retrospectiva de sprint:.....	32
2.8 Metodología en Cascada.....	32
2.8.1 Análisis de requerimiento.	33
2.8.2 Diseño del Sistema.....	34
2.8.3 Codificación.....	34
2.8.4 Pruebas	34
2.8.5 Implantación	34
2.8.6 Mantenimiento	34
2.9 Comparación metodología ágiles y metodología Cascada.	34
CAPÍTULO III	36
ANÁLISIS Y DISEÑO DE LA APLICACIÓN.....	36
3.1. Captura de Requisitos	36
3.2. IDENTIFICACIÓN DE ROLES Y TAREAS.	37
3.3. Modelo de Casos de Uso.....	39
3.4. DIAGRAMA DE SECUENCIA	45
Diagrama de secuencia Usuario Administrador	45
Figura 22 Diagrama de secuencia Usuario Administrador.....	45
Diagrama de secuencia Usuario Subadministrador.....	46
Diagrama de secuencia Crear Usuario	47
Figura 24 Diagrama de secuencia Crear Usuario	47
Diagrama de secuencia Usuario Secretaria.	48
Diagrama de secuencia Crear Panel.....	49
Diagrama de secuencia Administrar Factura.	49

Figura 27 Diagrama de secuencia Administrar Factura	50
Diagrama de secuencia Administrar Seguimiento.....	51
Figura 28 Diagrama de secuencia Administrar	51
3.5. DIAGRAMA DE ACTIVIDADES.....	52
Diagrama de actividad Crear Usuario.....	52
Figura 29 Diagrama de actividad Crear Usuario.....	52
Diagrama de actividad Crear Panel.....	53
Figura 30 Diagrama de actividad Crear Panel.....	53
Diagrama de actividad Usuario Administrador.....	54
Figura 31 Diagrama de actividad Usuario Administrador.....	54
Diagrama de actividad Usuario Cliente.....	55
Figura 32 Diagrama de actividad Usuario Cliente.....	55
Diagrama de actividad Usuario Secretaria.....	56
Figura 33 Diagrama de actividad Usuario Secretaria.....	56
3.5.1. Requerimientos No Funcionales	57
3.6. DIAGRAMA FÍSICO DE LA BASE DE DATOS.....	58
Figura 34 : DIAGRAMA FÍSICO DE LA BASE DE DATOS.....	58
3.7. Diagrama de Clases	59
Figura 35 : Diagrama de Clases	59
3.8. Artefactos propios de la metodología	60
Product Backlog - Definición del Proyecto.....	60
Sprint Backlog- Definición del Sprint	63
BURNDOWN CHART	66
GRAFICO DE PROGRESO DEL PROYECTO	67
CAPÍTULO IV	68
DESARROLLO	68
4.1. Arquitectura a Implementarse.....	68
4.1.1. Arquitectura Web	68
4.1.2. Patrón de arquitectura estándar J2EE	69
4.1.3. EJB (Enterprise JavaBeans).....	71
4.2. Estándares de implementación	72
4.2.1. ESTÁNDARES DE CODIFICACIÓN	72
4.3. Desarrollo de los módulos del sistema	74

4.4. INTEGRACIÓN DEL CÓDIGO	75
4.5. PRUEBAS	75
CAPÍTULO V	76
CONCLUSIONES Y RECOMENDACIONES.....	76
5.1. Conclusiones	76
5.2. Recomendaciones.....	77
REFERENCIAS BIBLIOGRÁFICAS	78
Bibliografía	78

ÍNDICE TABLAS

Tabla 1	8
Tabla 2.....	10
Tabla 3.....	15
Tabla 4.....	20
Tabla 5.....	21
Tabla 6.....	35
Tabla 7.....	60
Tabla 8.....	63
Tabla 9.....	63
Tabla 10.....	64
Tabla 11.....	64
Tabla 12.....	65
Tabla 13.....	65
Tabla 14.....	66
Tabla 15.....	67

ÍNDICE FIGURA

Figura 1 Cronograma.....	9
Figura 2 (Oracle, 2014)	12
Figura 3 (Rodríguez, 2014).....	14
Figura 4 (PostgreSQL, s.f.).....	20
Figura 5 (Berzal, 2014).....	23
Figura 6 (Berzal, 2014).....	24
Figura 7 Diagrama de Secuencia (Berzal, 2014)	24
Figura 8 Diagrama de comunicación (Berzal, 2014)	25

Figura 9 Diagrama de Caso de Uso (Berzal, 2014).....	26
Figura 10 Diagrama de estado (Berzal, 2014)	26
Figura 11 Diagrama de Actividades (Berzal, 2014)	27
Figura 12 Diagrama de Componentes (Berzal, 2014).....	27
Figura 13 Diagrama de despliegue (Berzal, 2014)	28
Figura 14 Proceso	30
Figura 15 Metodología en Cascada (Braude, 2013)	33
Figura 16 Sistema de control de proceso para la producción y construcción de paneles.	39
Figura 17 Caso uso SISPRO	40
Figura 18 Caso de uso gestión de clientes.....	41
Figura 19 Caso de Uso pedidos	42
Figura 20 Caso uso de Producción	43
Figura 21 Caso de Uso de Costos	44
Figura 22 Diagrama de secuencia Usuario Administrador.....	45
Figura 23 Diagrama de secuencia Usuario Subadministrador	46
Figura 24 Diagrama de secuencia Crear Usuario	47
Figura 25 Diagrama de secuencia Usuario Secretaria	48
Figura 26 Diagrama de secuencia Crear Panel	49
Figura 27 Diagrama de secuencia Administrar Factura.	50
Figura 28 Diagrama de secuencia Administrar	51
Figura 29 Diagrama de actividad Crear Usuario.	52
Figura 30 Diagrama de actividad Crear Panel.....	53
Figura 31 Diagrama de actividad Usuario Administrador.....	54
Figura 32 Diagrama de actividad Usuario Cliente.....	55
Figura 33 Diagrama de actividad Usuario Secretaria.....	56
Figura 34 : DIAGRAMA FÍSICO DE LA BASE DE DATOS.....	58
Figura 35 : Diagrama de Clases	59
Figura 36 : Arquitectura Java EE (Jatun, s.f.).....	68
Figura 37 (Jatun, s.f.).....	69

RESUMEN

El presente proyecto “Desarrollo e implantación de un sistema de administración para la empresa TABLEROS ÉDISON” se implementó mediante la metodología SCRUM, debido a que permite obtener las características de los requerimientos del cliente, ejecutivos u otros miembros de la empresa para desarrollar el sistema SISPRO, además de su flexibilidad y agilidad. De esta manera se maximiza el tiempo de desarrollo puesto que hay la posibilidad de dividir en bloques cortos y definidos en periodos de tiempo. Con la implementación del sistema SISPRO, la empresa TABLEROS ÉDISON automatiza los procesos que se ejecutaban de manera manual, dando como resultado un servicio de calidad a clientes internos y externos de la empresa. La creación de un sistema en la web con Jboss tiene como objetivos registrar usuarios, insumos y productos para mostrar de manera virtual lo que ofrece, y, de esta forma, los clientes tengan la posibilidad de escoger y comprar lo que necesiten.

Palabras Clave:

SISPRO

Jdk 7

Eclipse

Postgresql

Jboss

Metodología SCRUM.

ABSTRACT

This project "Development and implementation of a management system for the company BOARDS ÉDISON" was implemented by SCRUM methodology because it allows for the characteristics of customer requirements, executives or other members of the company to develop the system SISPRO addition to its flexibility and agility. Thus it maximizing development time as there is the possibility of dividing in short periods of time defined in blocks. With the implementation of SISPRO system, the company BOARDS ÉDISON automates the processes that are executed manually, resulting in a quality service to internal and external customers of the company. The creation of a system on the web with Jboss aims to register users, inputs and outputs to display virtually what it offers, and, in this way, customers have the opportunity to choose and buy what they need.

Keywords:

SISPRO

Jdk 7

Eclipse

Postgresql

Jboss

SCRUM methodology.

CAPÍTULO I

ASPECTOS GENERALES

1.1. Planteamiento del Problema

En la actualidad, en todo tipo de organizaciones, resulta de gran importancia tener control y administración de la información, con la finalidad de brindar un servicio eficiente y tener un adecuado funcionamiento de los procesos que diariamente debe cumplir la empresa.

Con los avances de la tecnología tanto en Software como en Hardware, es conveniente que estas herramientas sean aprovechadas a través de diferentes procedimientos, para que lleguen a conformarse en aplicaciones, que servirán de gran ayuda para el desempeño de las organizaciones.

La empresa TABLEROS ÉDISON lleva la información en archivos Excel lo que dificulta que se realice un correcto manejo de la información porque esta se encuentra duplicada e inconsistente; y esto, a su vez, conlleva a que los clientes no tengan una información oportuna, precisa e idónea, dando como resultado tardanza en la entrega de información, productos e incluso, en algunos casos, pérdida de clientes.

1.2. Antecedentes

TABLEROS ÉDISON es una pequeña empresa que permanece en el mercado por más de treinta años y se dedica a la construcción de tableros metálicos de distribución eléctrica (contadores de energía) con protecciones.

Se encuentra registrada en la Empresa Eléctrica Quito, como proveedor oficial de estos tableros y distribuye también a otras ciudades como Nueva Loja, Esmeraldas, Ambato, Tulcán, etc.

Su planta industrial se encuentra ubicada en el sur de la ciudad de Quito, en el sector industrial de Guamaní. Cuenta con 8 empleados estables, equipos, maquinaria de metalmecánica y horno de pintura electrostática; cuenta, también, con proveedores de materiales eléctricos mayoristas, así como de materiales de metal mecánica y pintura.

En un mundo con plataformas virtuales, redes sociales y tecnología virtual es necesario modernizar a la empresa para que no se quede rezagada y capte clientes a través de las nuevas tecnologías que predominan el mundo actual.

El control administrativo y operativo los cumplen los dueños de la empresa, quiénes son familiares entre sí, de manera informal; los aspectos contables se realizan con outsourcing y se presentan de manera mensual; el control de producción se realiza de manera manual con hojas de trabajo en base de los contratos realizados con los clientes.

Tableros Eléctricos Édison surgió en 1976 por un accidente eléctrico cuando don Jorge Díaz era empleado de la empresa eléctrica quito en 1975.

El peligro constante al que se exponían los trabajadores al colocar los medidores en hogares y oficinas.

El problema era cuando se ponía más de cuatro medidores, una placa de cobre que alimentaba a los medidores se transformaba en una trampa que ocasionaba cortos circuitos y quemaduras a los trabajadores.

Don Jorge Días creó el tablero de tres cuerpos o armario de seguridad como un aporte al Ecuador y a sus compañeros, en su mente no pasó el de patentar su idea sino que sea un instrumento que evite accidentes a compañeros que en determinada situación podían ser mortales. Sin las herramientas adecuadas hizo el primero artesanalmente y logró venderlo inmediatamente. (Pérez, 2011)

Pionero: Esta idea se presentó en 1976 a la empresa Eléctrica Quito para la aprobación de los tendidos eléctricos en los hogares, diez años después se convirtió en una norma para la aprobación de los planos.

1.3. Justificación e importancia.

En los actuales momentos, la empresa no cuenta con un Sistema que cumpla con los requerimientos de su trabajo y por tanto se ha tenido un retraso en la entrega de reportes y en el manejo de los procesos.

Gran parte de los requerimientos de la empresa se fundamenta en reportes detallados, en concordancia con las necesidades de los usuarios que manejan el Sistema.

Se requiere una solución inmediata a todos los procesos y requerimientos de la empresa, para proporcionar datos de una manera confiable que garantice la adecuada administración.

La información para los clientes no se encuentra disponible por medios tecnológicos modernos, lo que dificulta el acceso a sus productos por parte de potenciales clientes.

Hoy en día el correcto manejo de la información es parte fundamental en el éxito de las PYMES. Para Tableros Édison como una organización destinada a proveer productos de tecnología, es de mucha importancia que cuente con un Sistema que brinde todas las facilidades en la entrega oportuna de la información que se requiera.

Por todas las características que ofrece la empresa es conveniente que se automatice sus procesos mediante un Sistema Administrativo, que permita

un correcto manejo de la información y que la empresa se desarrolle conforme se espera de las pequeñas y medianas industrias.

1.4. Objetivos.

1.4.1. Objetivo General

Realizar el desarrollo e implantación de un sistema de administración para la empresa TABLEROS ÉDISON.

1.4.2. Objetivos Específicos

- Especificar los requerimientos del sistema a desarrollar
- Realizar el análisis y diseño del Sistema.
- Desarrollar el sistema diseñado, con una herramienta segura y confiable.
- Realizar las pruebas del sistema.
- Implantar el sistema.

1.5. Alcance

El tema “DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON”, cubre las fases detalladas en los objetivos específicos, anteriormente indicados y los módulos detallados a continuación:

- Administración de Clientes y Proveedores: Comprende el control de toda la información que respecta a los socios:
 - Registro Datos personales.
 - Crear, actualizar, insertar y eliminar usuarios
 - Registro de seguimiento.

- Reportes individuales.
- Control de cuotas.
- Administración de Finanzas:
 - Presupuesto
 - Facturación
- Administración de Logística:
 - Adquisiciones
 - Inventarios
 - Mantenimiento
- Control de producción:
 - Contratación
 - Planificación de la producción
 - Control de bodega
- Seguridad de Acceso al Sistema: Control de los niveles de acceso de los usuarios al Sistema.
 - Usuarios Administradores.
 - Usuarios
- Información a clientes y proveedores vía WEB:
 - Comunicación y contacto con clientes
 - Publicidad de productos disponibles.
 - Noticias y servicios adicionales

El Sistema que se desarrollara llegara a su implementación en la empresa, para el correcto desenvolvimiento de las actividades que tiene que desempeñar y operar.

1.6. Metodología

El principal aspecto en que se basa la metodología de desarrollo es establecer un plan de trabajo, que servirá para mostrar la forma de desarrollar la aplicación para empresa "TABLEROS ÉDISON".

Es importante realizar el análisis de algunas metodologías para el buen desarrollo del sistema SISPRO, luego del análisis que se realizara a las diferentes metodologías se seleccionara la más apropiada para el desarrollo del sistema.

Se utilizara el lenguaje de modelamiento UML para visualizar, especificar, construir el sistema de Software.

1.7. Factibilidad

1.7.1. Factibilidad Técnica

La institución patrocinadora pondrá a disposición del proyecto, personal altamente calificado que labora en la Institución, para proporcionar ayuda técnica en lo referente a los requerimientos, que es donde básicamente va a estar centrado la investigación en la primera etapa.

- Las herramientas en la cual se va a desarrollar el sistema es el lenguaje Java que se adapta perfectamente al sistema que se tiene que desarrollar.
- Con lo referente a Base de Datos se utilizará una que sea robusta tanto en seguridad como en escalabilidad y tecnología (PostgreSQL).

- El Hardware mínimo con él se cuenta para la realización del proyecto es:
 - 2 laptop, cuyas características principales son:
 - Procesador: Intel Core i5
 - Velocidad: 2.4 GHz
 - Memoria RAM: 6 GB
 - Disco 500 GB.
 - Acceso al internet.

1.7.2.Factibilidad Operativa

Las instalaciones en las que se va llevar a cabo son de propiedad de la organización auspiciante; el seguimiento de este proyecto se realizará por parte de un miembro de la empresa que cuente con experiencia en aplicaciones de esta índole.

1.7.3.Factibilidad Económica

Para la realización del sistema SISPRO se utilizara Software libre por esta razón la empresa no incurrirá en gastos de licencias en las herramientas de desarrollo, las laptops son de propiedad de los desarrolladores (tesistas),tal como se indica en la Tabla 1.

PRESUPUESTO**Tabla 1****Presupuesto**

#	Descripción	Cantidad	V. Unitario	V. Total
1	HARDWARE			
	• Laptop	2	600	1200
2	SOFTWARE			
	• Licencia Eclipse	1	0	0
	• Licencia PostgreSQ	1	0	0
	• Jboss	1	0	0
	• Jdk	1	0	0
3	CONSULTORÍA			
	• Desarrolladores	2	1500	3000
	Total USD			4200

1.8. Cronograma

ACTIVIDADES	SEMANA	NOVIEMBRE				DICIEMBRE				ENERO				FEBRERO				MARZO			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Planificación del sistema					■	■															
Análisis de requerimiento					■	■	■	■													
Diseño del sistema						■	■	■	■												
Construcción del sistema								■	■	■	■	■	■	■							
Pruebas e implementación													■	■	■						
Documentación del proyecto				■	■	■	■	■	■	■	■	■	■	■	■						
Aprobación y defensa del proyecto																■					

Figura 1 Cronograma

CAPÍTULO II

MARCO TEÓRICO

2.1. Plataforma Java

2.1.1 Introducción.

- Java fue creado en 1991 por James Gosling, trabajador de la Sun Microsystem, la cual fue adquirida por la compañía Oracle.
- Se llamó inicialmente Oak, pero su nombre fue cambiado debido a que ya existía una marca comercial registrada para adaptadores de tarjetas gráficas. (Oracle, 2014)
- Java es un lenguaje de programación y una plataforma informática.
- Java es orientado a objetos.
- Java es multiplataforma.

2.1.2 Versiones de Java

Las versiones de java se indica en la Tabla 2. (Oracle, s.f.)

Tabla 2

Versiones de Java

Versión	Año		Versión	Año
JDK Beta	1995		J2SE 5.0	2004
JDK 1.0	1996		Java SE 6	2006
JDK 1.1	1997		Java SE 7	2011
J2SE 1.2	1998		Java SE 8	2014
J2SE 1.3	2000		Java SE 9	2017
J2SE 1.4	2002		Java SE 10	2018

2.1.3 Tecnología Java.

- Lenguaje de programación de alto nivel.
- Plataforma de desarrollo.
- Ambiente de desarrollo (javac, java, javadoc etc.)
- Ambiente y aplicación y despliegue.

2.1.4 Características.

- Orientada a objetos.
- Multitareas.
- Multiplataforma.
- Robusto.
- Alto rendimiento.

2.1.5 El lenguaje de programación.

- El código fuente es escrito en archivo de texto plano con extensión .java.
- Estos archivos son compilados por el compilador javac y dan como resultado el archivo con la extensión .class. que poseen los bytecodes.
- El archivo es ejecutado por el intérprete java, JVM (java Virtual Machine). (Oracle, 2014).
- A través de la máquina virtual un mismo programa puede correr en múltiples plataforma, tal como se indica en la Figura 2.

El lenguaje Java

- Compilación y ejecución en Java

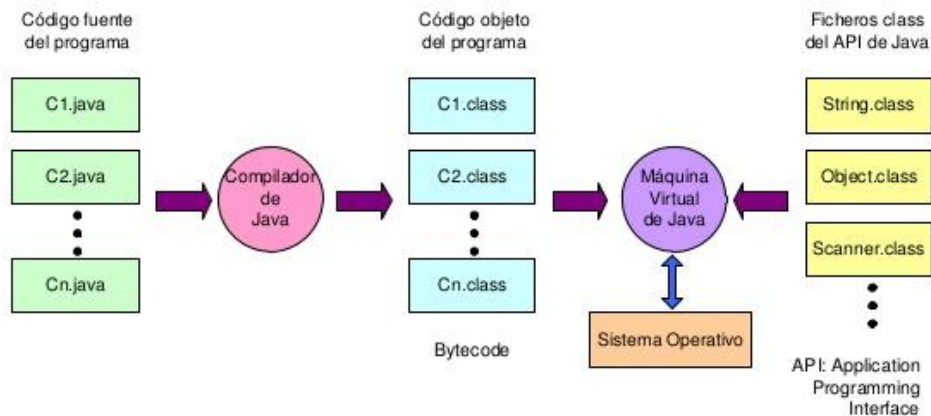


Figura 2 (Oracle, 2014)

2.1.6 Paradigma de la programación orientada a objetos

- **Clases**

Representa una plantilla de objetos, la cual define las variables y métodos comunes para todos los objetos de un cierto tipo.

- **Objeto**

Los objetos de Software sirven para representar los objetos del mundo real.

Los estados del objeto se encuentran representados por variables y su comportamiento por métodos.

Para crear un objeto se utiliza una plantilla (clases) a través de la llamada al constructor lo cual se lo conoce como el proceso de instanciación.

- **Encapsulamiento**

Es una técnica que permite proteger las características del objeto (variables).

Para acceder o modificar las variables se utilizan unos métodos con el nombre de getters y setters.

El encapsulamiento da tres ventajas al código flexibilidad, modularidad, y buen mantenimiento.

- **Herencia**

Es reutilizar las variables como métodos de la clase padre en la clase hija optimizando tiempo y recurso.

No existe herencia múltiple una clase hija puede heredar de un solo padre. (Rodríguez, 2014)

A la clase padre se le conoce como superclase y a la clase hija se le conoce como subclase, tal como se indica en la Figura 3.

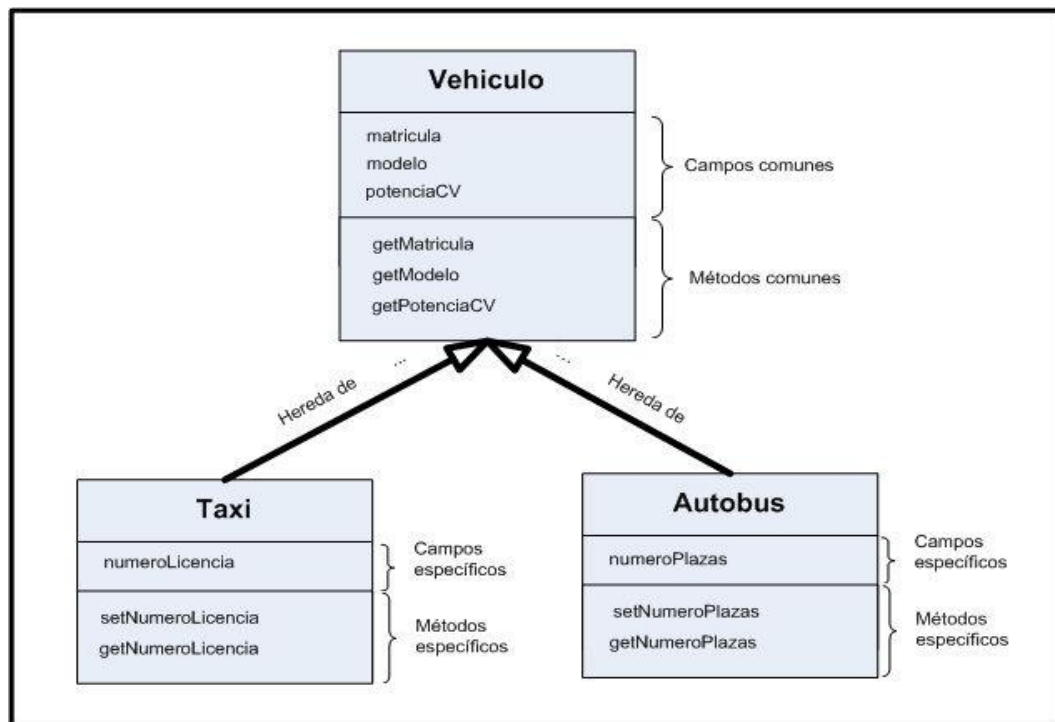


Figura 3 (Rodríguez, 2014)

- **Polimorfismo**

Permite que un solo nombre de clases o de métodos represente diferente código.

Cuando el polimorfismo se aplica a métodos se conoce con el nombre de sobrecargas.

Se puede tener más de un método con el mismo nombre pero con diferente firma (número y tipo de argumento).

2.2. IDE Eclipse

2.2.1. Introducción

- Eclipse es multiplataforma.
- Eclipse fue desarrollado originalmente por IBM en la actualidad es desarrollado por la fundación eclipse.

- Eclipse es un programa que para su ejecución no se debe instalar, se descomprime y es ejecutable, al instalar los plugins se puede comprimir y llevar a otro equipo donde se descomprimirá y funcionara.
- Eclipse se puede descargar para diferentes sistemas operativos y arquitectura de 32 o 64 bits este debe de ser de la misma arquitectura del JDK. (Eclipse, 2013)

2.2.2. Arquitectura

La plataforma de cliente enriquecido (del inglés Rich Client Platform RCP).

- Plataforma principal - inicio de Eclipse, ejecución de plugins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes, tal como se indica en la Tabla 3.

2.2.3. Versiones de Eclipse.

Tabla 3

Las Versiones de Eclipse se indican en la Tabla 3

11	Fecha de lanzamiento	Versión de plataforma	Proyectos
Neon	Junio de 2016 (planeado)	4.6	Neon projects
Mars	24 de junio de 2015	4.5	Mars projects
Luna	25 de junio de 2014	4.4	Luna projects

Kepler	26 de junio de 2013	4.3	Kepler projects
Juno	27 de junio de 2012	4.2	Juno projects
Indigo	22 de junio de 2011	3.7	Indigo projects
Helios	23 de junio de 2010	3.6	Helios projects
Galileo	24 de junio de 2009	3.5	Galileo projects
Ganymede	25 de junio de 2008	3.4	Ganymede projects
Europa	29 de junio de 2007	3.3	Europa projects
Callisto	30 de junio de 2006	3.2	Callisto projects
Eclipse 3.1	28 de junio de 2005	3.1	
Eclipse 3.0	28 de junio de 2004	3.0	

2.2.3. Características.

- Eclipse permite trabajar en un determinado entorno de trabajo de forma óptima, lo que se refiere a las perspectivas que es una pre-configuración de ventanas.
- Gestión de proyectos ,es un conjunto de recursos que se relacionan entre sí, por ejemplo el código fuente, documentación, ficheros de configuración, etc.,
- Depurador de código, el cual viene incluido en el IDE.
- El reconocimiento de las palabras reservadas se colorean.

- Permite completar el código automáticamente y se llama code completion.

2.3. Jboss

Jboss es un servidor de código abierto creado en Java que funciona en cualquier sistema operativo que esté instalado la máquina virtual de java.

La empresa JBoss Inc. Fue fundada por Marc Fleury ,Red Hat adquirió en abril del 2006. Jboss es un servidor de aplicación certificado J2EE. (Chávez, 2011)

2.3.1 Características.

Es una plataforma de alojamiento de servicios y aplicaciones java Enterprise Edition (EE) integrada.

Jboss se puede ejecutar en máquinas virtuales en diferentes sistemas operativos Red Hat ,Enterprise Linux ,otras distribuciones de Linux, Unix y Windows.

Jboss es compatible con base de datos con JDBC , Mysql server, Mysql, Oracle, PostgreSQL,Sybase.

2.3.2 Servidor de aplicaciones Jboss.

EJB (Enterprise JavaBeans) proporciona un modelo de componente distribuido estándar del lado del servidor, permitiendo al programador centrarse en el desarrollo del negocio del sistema, la concurrencia, las transacciones, la persistencia, la seguridad, etc., lo realiza EJB.

Jboss AOP (programación orientada a aspectos) permite a las transacciones, la seguridad, la persistencia a clases simples de JAVA.

Hibernate es una herramienta de mapeo objeto- relacional, permite la manipulación de datos en la base de datos operando sobre objetos, incluye herencia, polimorfismo, colecciones.

2.4. PostgreSQL

Inicia en 1982 con el proyecto Ingres, en 1985 se retoma llamándole post-ingress o simplemente POSTGRESS.

En 1996 cambiaron el nombre de Postgres 95 a PostgreSQL ,cuya primera versión de código abierto fue lanzada en agosto de 1996, en 1997 se libera la primera versión formal de PostgreSQL. (Postgresql, 2015)

2.4.1 Características.

Las características que más se han tomado en el desarrollo del PostgreSQL son estabilidad, potencia, robustes, copias de seguridad en caliente (online –hot backup).

Es respaldable la información sin que se corte el servicio a los usuarios.

Unicode.- es el estándar de codificación de caracteres, disponible para Linux y Unix en todas sus variables, Windows para la 32 y 64 bits., facilidad de la administración y la implementación.

Posee En 1996 cambiaron el nombre de Postgres 95 a PostgreSQL ,cuya primera versión de código abierto fue lanzada en agosto de 1996, en 1997 se libera la primera versión formal de PostgreSQL.

2.4.2 Componentes de PostgreSQL.

Los componentes de PostgreSQL son:

- Aplicación cliente.
- Demonio postmaster
- Ficheros de configuración:
- Procesos hijos postgres:
- PostgreSQL share buffer cache:
- PostgreSQL para almacenar datos en caché.
- Write-Ahead Log (WAL):
- Kernel disk buffer cache:
- Disco:

El diagrama de componentes de PostgreSQL se indica en la Figura 4

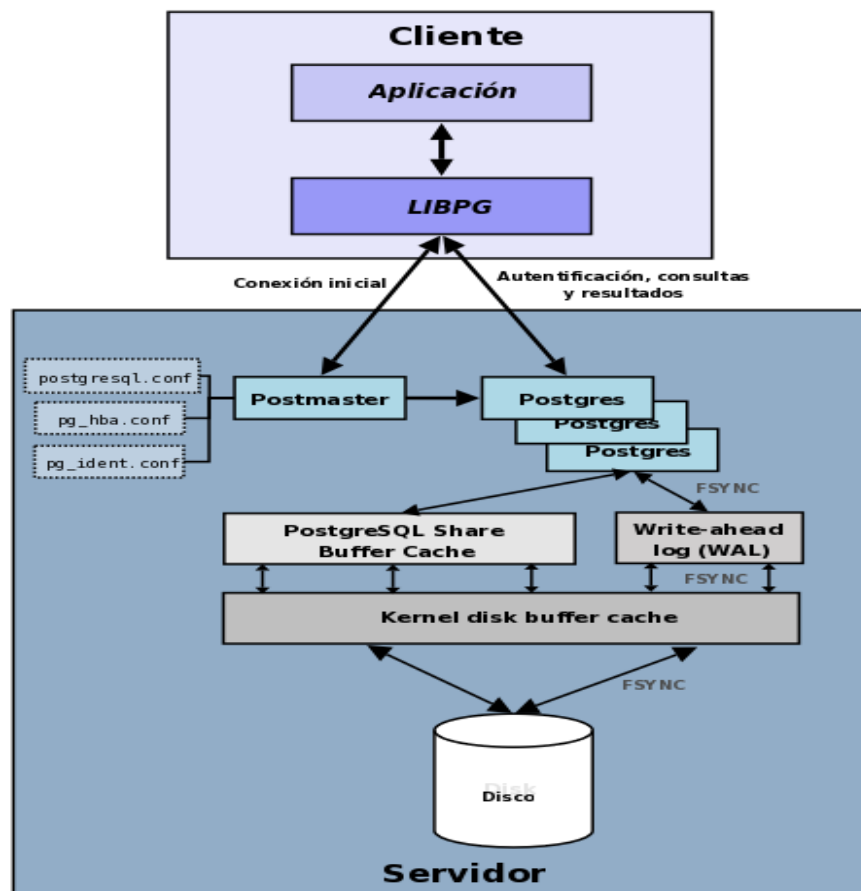


Figura 4 (PostgreSQL, s.f.)

2.4.3 Alta Concurrencia.

Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos, tal como se indica en la Figura 4.

Tabla 4

Alta Concurrencia (Postgresql, 2015)

	ACID	Integridad referencial	Transacciones	Unicode
Adaptive Server Enterprise	✓ Sí	✓ Sí	✓ Sí	✓ Sí
ANTs Data Server	✓ Sí	✓ Sí	✓ Sí	✓ Sí
DB2	✓ Sí	✓ Sí	✓ Sí	✓ Sí
Firebird	✓ Sí	✓ Sí	✓ Sí	✓ Sí
HSQLDB	✓ Sí	✓ Sí	✓ Sí	✓ Sí
Informix	✓ Sí	✓ Sí	✓ Sí	✓ Sí
Ingres	✓ Sí	✓ Sí	✓ Sí	✓ Sí
InterBase	✓ Sí	✓ Sí	✓ Sí	✓ Sí
SapDB	✓ Sí	✓ Sí	✓ Sí	✓ Sí
MaxDB	✓ Sí	✓ Sí	✓ Sí	✓ Sí
Microsoft SQL Server	✓ Sí	✓ Sí	✓ Sí	✓ Sí
MySQL	Depende ¹	Depende ¹	Depende ¹	✓ Sí
Oracle	✓ Sí	✓ Sí	✓ Sí	✓ Sí
PostgreSQL	✓ Sí	✓ Sí	✓ Sí	✓ Sí
SQLite	✓ Sí	✗ No ²	Básico ²	✓ Sí
	ACID	Integridad referencial	Transacciones	Unicode

2.4.4 Limitaciones.

Las principales limitaciones de PostgreSQL, tal como se indica en la Tabla 5.

Tabla 5

Limitaciones

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

2.5. Lenguaje de Modelamiento Unificado (UML)

El lenguaje UML es un estándar OMG (Object Management group) es una organización sin fines de lucro que da las guías y especificaciones para uso de la tecnología orientado a objetos por lo cuanto permite modelar, construir y documentar Software orientado a objetos. (Berzal, 2014)

El modelo es una abstracción de la realidad simplificada, indica el comportamiento que se desea del sistema, indica la estructura de un sistema complejo se comprende mejor lo que se quiere construir, se puede tener la simplificación y reutilización.

El modelo proporciona los planos de un sistema.

Los sistemas se pueden describir tanto en modelo estructural (organización del sistema) y modelo comportamental.UML estandariza nueve tipos de diagrama que representa visualmente un sistema desde diferentes puntos de vistas.

2.5.1 Diagrama de clases.

Es un diagrama que describe la estructura del sistema detallando las clases sus atributos y métodos, tal como se indica en la Figura 5.

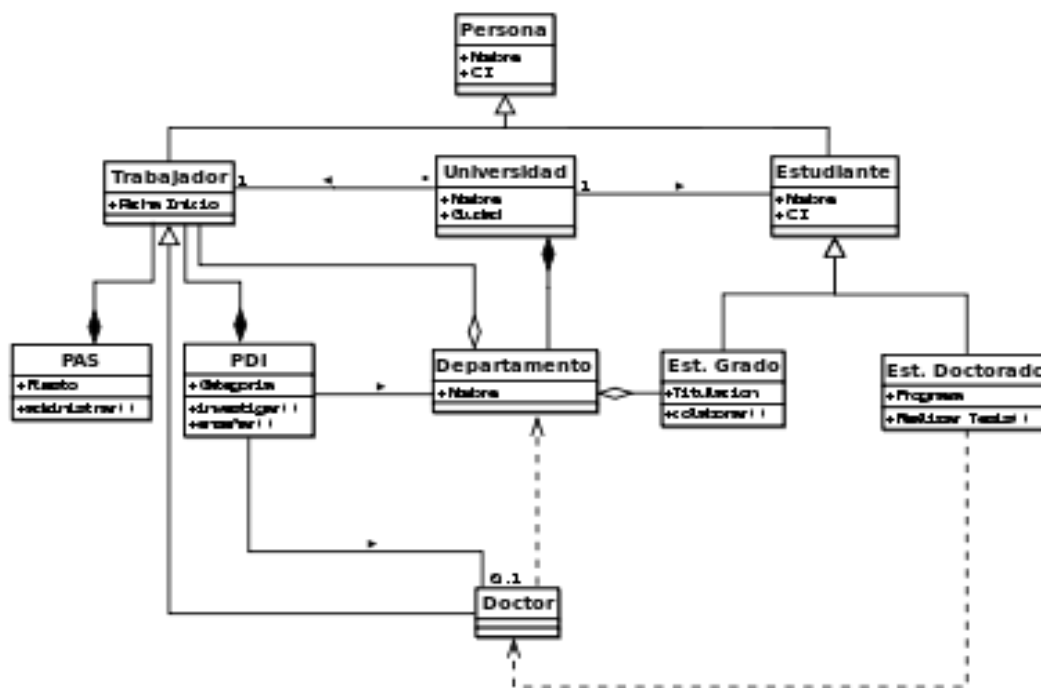


Figura 5 (Berzal, 2014)

2.5.2 Diagrama de Objetos.

Es un gráfico de instancias incluyendo objetos y datos, los diagrama de objetos están ligados a los diagramas de clases que comparten los mismos símbolos para la notación, tal como se indica en la Figura 6.

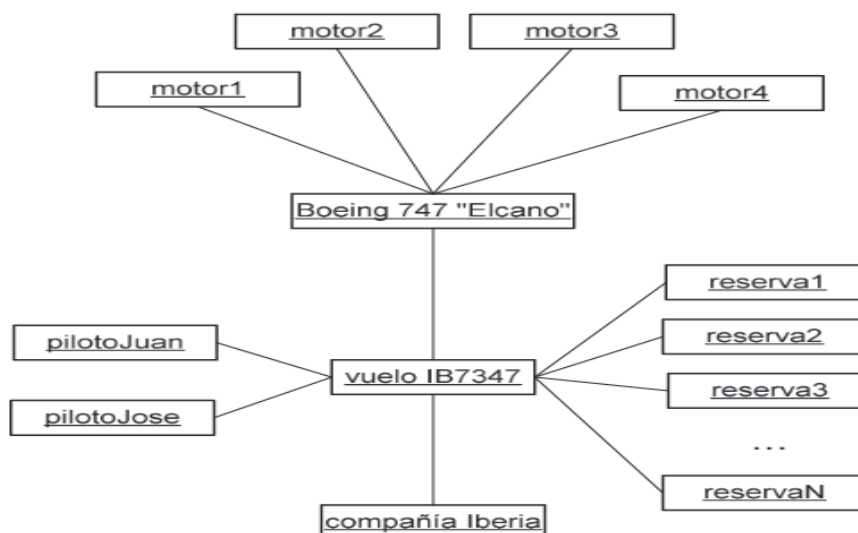
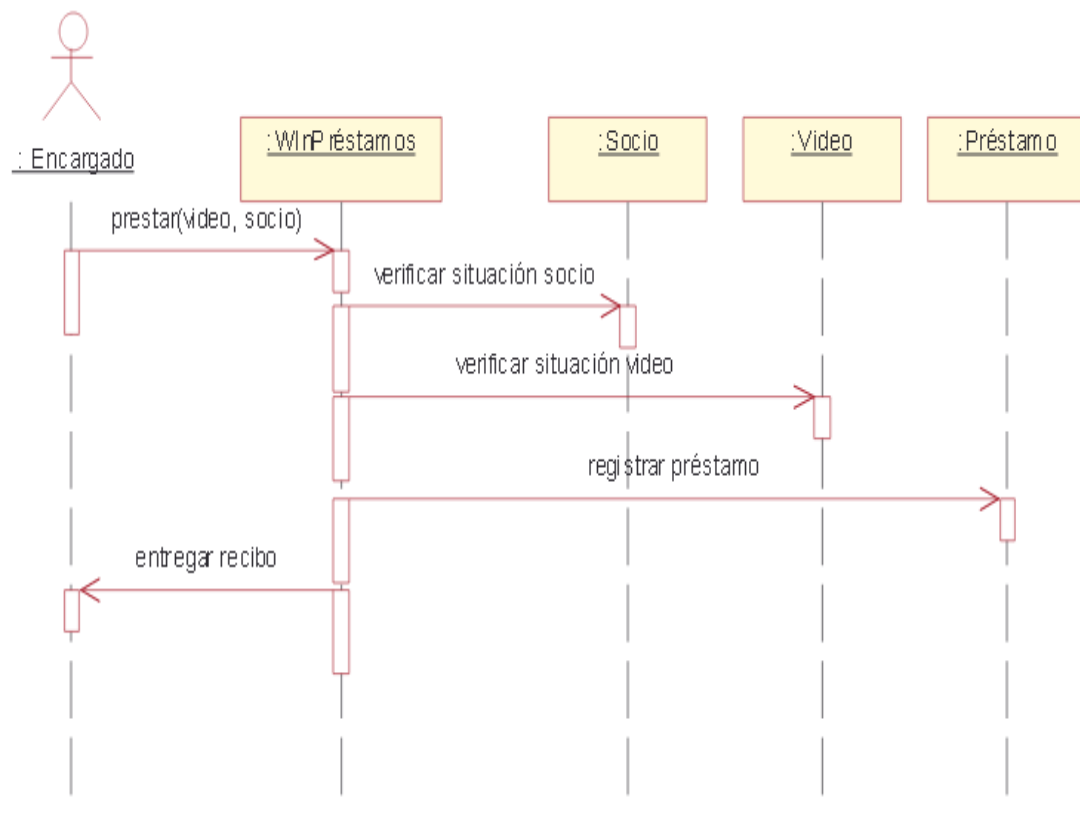


Figura 6 (Berzal, 2014)

2.5.3 Diagrama de Secuencia.

Muestra la comunicación de un conjunto de objetos a través del tiempo, tal como se indica en la Figura 7.

**Figura 7 Diagrama de Secuencia (Berzal, 2014)**

2.5.4 Diagrama de Comunicación.

Un diagrama de comunicación modela la interacción entre objetos o partes en términos de mensajes en secuencias, tal como se indica en la figura 8.

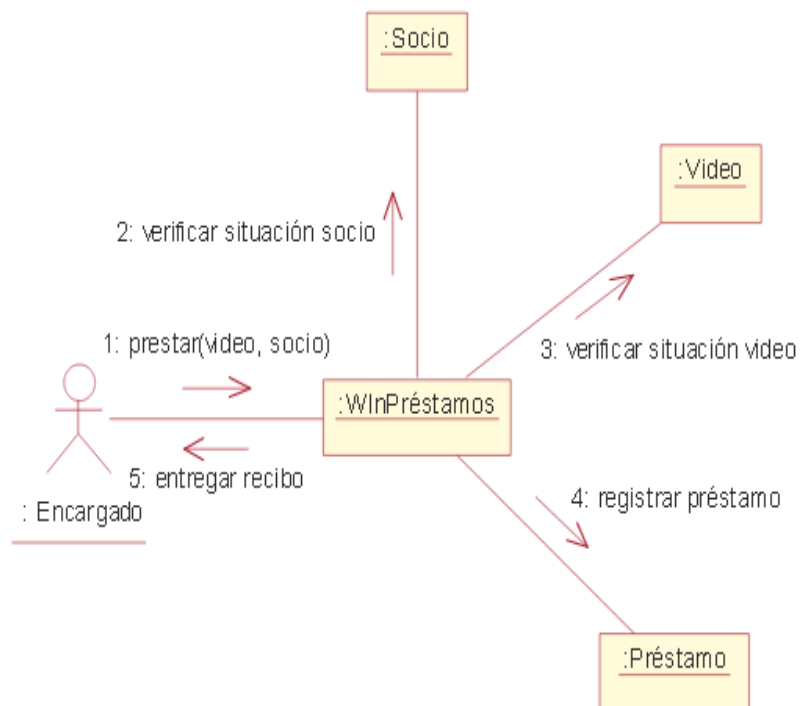


Figura 8 Diagrama de comunicación (Berzal, 2014)

2.5.5 Diagrama de Caso de Uso.

Se utiliza para modelar desde el punto de vista de los usuarios para representar las acciones que hacen cada usuario, tal como se indica en la figura 9.

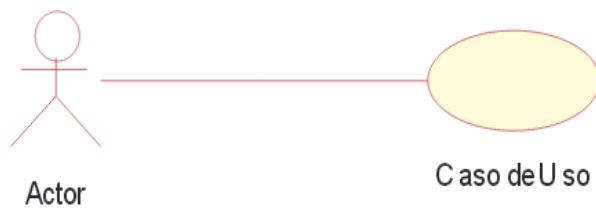


Figura 9 Diagrama de Caso de Uso (Berzal, 2014)

2.5.6 Diagrama de estado.

Permite describir el comportamiento de un sistema reactivo y el comportamiento está dirigido por eventos, tal como se indica en la figura 10.

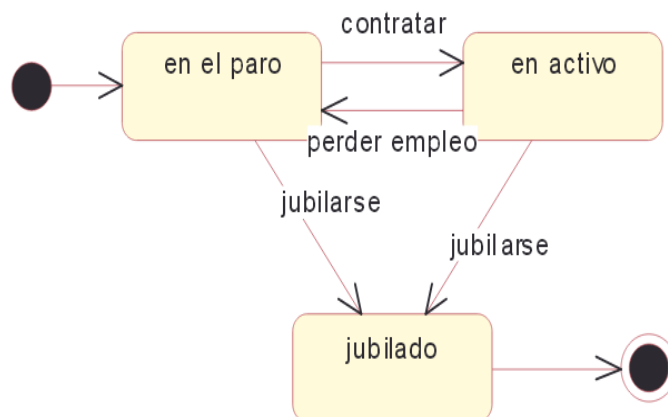


Figura 10 Diagrama de estado (Berzal, 2014)

2.5.7 Diagrama de Actividades.

Muestra el orden en que se va realizar las tareas dentro del sistema, tal como se indica en la Figura 11.

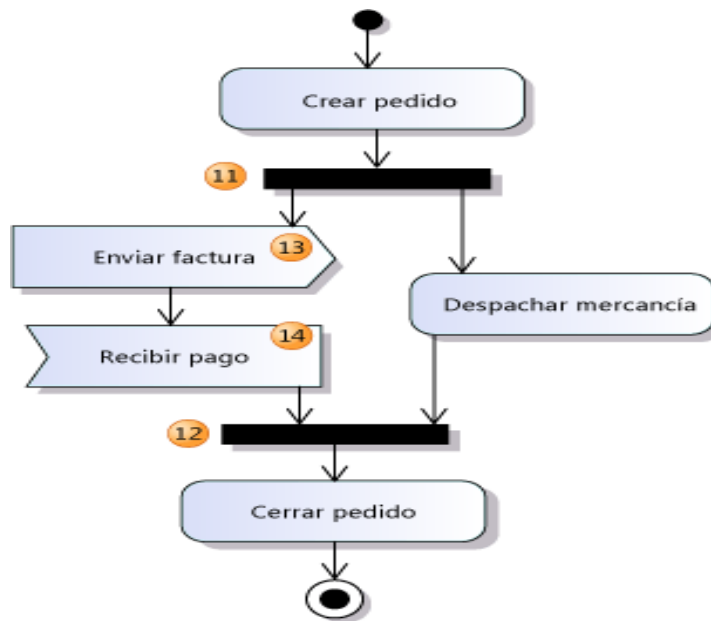


Figura 11 Diagrama de Actividades (Berzal, 2014)

2.5.8 Diagrama de Componentes.

Indica como en un sistema de Software se divide en componentes e indica las dependencias entre componentes, tal como se indica en la Figura 12.

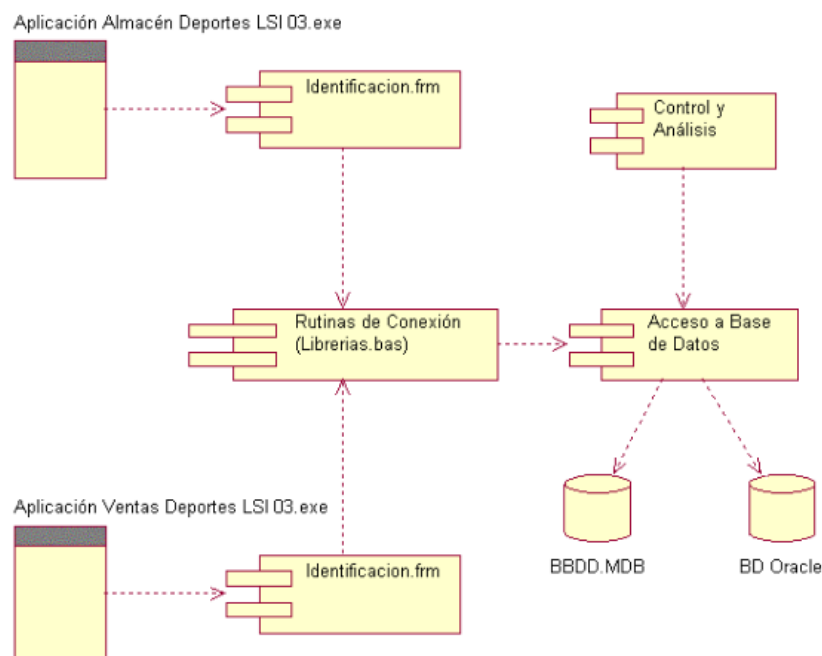


Figura 12 Diagrama de Componentes (Berzal, 2014)

2.5.9 Diagrama de Despliegue.

Modela la arquitectura en tiempo de ejecución, muestra la configuración de los elementos de Hardware (nodos), tal como se indica en la Figura 13.

- Diagramas de despliegue

(nodos de procesamiento y componentes)

Configuración del sistema en tiempo de ejecución

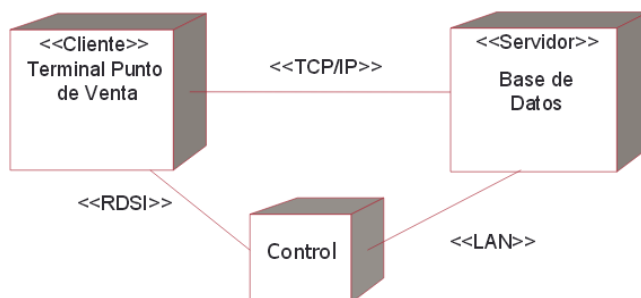


Figura 13 Diagrama de despliegue (Berzal, 2014)

2.6 Metodologías ágiles.

Las metodologías ágiles están basados en el desarrollo interactivos e incremental, donde existe la colaboración de grupos auto organizados y multi disciplinados.

Se enfatizan en la comunicación directa (cara a cara) en lugar de la documentación.

Entre las metodologías ágiles se tienen Scrum, Crystal Clean, programación extrema (XP eXtreme Programming).

Todas las metodologías ágiles cumplen con una serie de principios:

- 1) La prioridad es satisfacer al cliente mediante entregas de Software tempranas y continuas.
- 2) Los cambios en los requerimientos son aceptados.

- 3) Software que funcione se entrega frecuentemente, con el menor intervalo posible entre entregas.
- 4) El cliente y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- 5) El proyecto se construye en base a individuos motivados.
- 6) El dialogo cara a cara es el método más eficiente y efectivo para comunicar información dentro del equipo.
- 7) El Software que funcione es la medida principal del progreso.
- 8) Los procesos ágiles promueven el desarrollo sostenido.
- 9) La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- 10) La simplicidad es esencial.
- 11) Las mejores arquitecturas, requerimientos y diseños surgen de equipos auto-organizados.
- 12) El equipo reflexiona en cómo ser más efectivos, y ajusta su comportamiento en consecuencia.

2.7 Metodología Scrum.

2.7.1 Funcionalidad y proceso de Scrum.

Scrum se basa en ejecutar se en forma de bloques cortos y definidos en periodos de tiempos establecidos por el jefe de proyecto y los desarrolladores.

2.7.2 Sprint.

Es el proceso definido en el cual se encuentran los módulos, procesos que serán revisados por el cliente, tal como se indica en la Figura 14. (Schwaber Ken, 2013)

2.7.3 Proceso.

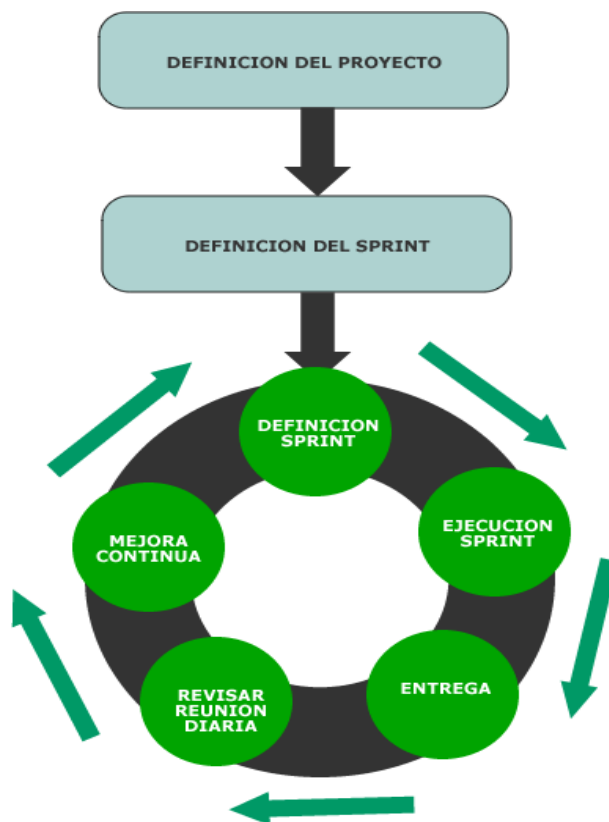


Figura 14 Proceso

2.7.4 Definición del proyecto (Product Backlog).

Consiste en un documento que recoge el conjunto de requerimientos que se asocian al proyecto. Es responsabilidad del Product Owner realizar esta definición y establecer las prioridades de cada requerimiento. Es un documento de alto nivel, que contiene descripciones genéricas (no detalladas), y que está sujeto a modificaciones a lo largo del desarrollo.

2.7.5 Definición del Sprint (Sprint Backlog).

Un sprint debe entenderse como un subconjunto de requerimientos, extraídas del product backlog, para ser ejecutadas durante un periodo entre 1 y 4 semanas de trabajo. El sprint backlog

sería el documento que describa las tareas que son necesario realizar para abordar los dichos subconjuntos de requerimientos.

2.7.6 Ejecución del Sprint.

Sería el periodo de entre 1 y 4 semanas (periodo definido previamente en base a las tareas recogidas en el sprint backlog) durante el cual el equipo de trabajo abordaría las tareas de desarrollo correspondientes. Una vez iniciada la ejecución de un sprint definido, este no podrá ser modificado, y en caso de ser necesario introducir cambios estos se harán una vez concluido el periodo a través de la definición de otro sprint backlog. (Scrum Manger, 2014).

2.7.7 Entrega.

Una vez concluida la ejecución del sprint, se dispondrá de una porción de la aplicación potencialmente definitiva.

2.7.8 Evolución del proyecto (Burn down):

Es un documento que refleja el estado del proyecto, indicando el volumen de requerimientos que en ese momento se encuentran pendientes de ser abordados (en el product backlog), los requerimientos que en ese momento se están desarrollando (sprint backlog) y los requerimientos cuyo desarrollo ya se ha completado en su totalidad.

2.7.9 Reuniones de trabajo en un contexto SCRUM

2.7.9.1 Planificación de sprint:

Se realiza al principio de cada ciclo de sprint, y está encaminada a seleccionar el conjunto de requerimientos del product backlog que serán abordados, el equipo de trabajo que será necesario y el tiempo que se estima (entre 1 y 4 semanas) para su desarrollo.

2.7.9.2 Reunión diaria:

Conocida como daily scrum, se realiza al comienzo de cada día en que ese esté ejecutando un sprint. Es una reunión corta (no más de 30 minutos) en la que los integrantes del equipo responden las siguientes preguntas:

- ¿Qué has hecho desde la última reunión?
- ¿Qué problemas has encontrado para realizar el trabajo previsto?
- ¿Qué planeas hacer antes de la próxima reunión?

2.7.10 Revisión de sprint:

Una vez concluido el ciclo de sprint se mantiene una reunión en la que se define qué parte del trabajo previsto se ha completado y qué parte permanece pendiente. En cuanto al trabajo completado se realiza una revisión (demo) del mismo al product owner y otros usuarios que pudiesen estar involucrados.

2.7.11 Retrospectiva de sprint:

Es una reunión en la que todos los miembros del equipo realizan una valoración del trabajo realizado en el último sprint, identificando puntos de mejora de cara a los siguientes a realizar. El objetivo principal es introducir un componente de mejora continua en el proceso.

2.8 Metodología en Cascada.

También conocida como modelo lineal secuencial, modelo tradicional, modelo clásico, es un método de desarrollo rígido.

Es una secuencia de seis etapas importantes, para iniciar una etapa se debe de haber finalizado la etapa anterior, tal como se indica en la Figura 15.

- Análisis de requisitos
- Diseño del Sistema
- Diseño del Programa
- Codificación
- Pruebas
- Implantación
- Mantenimiento

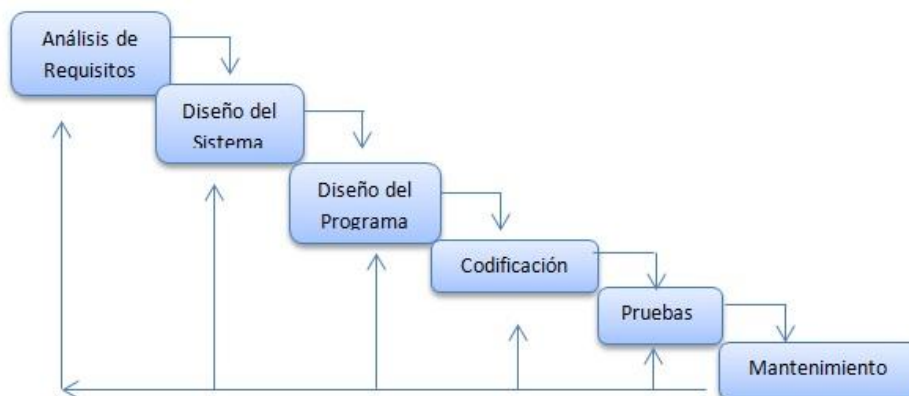


Figura 15 Metodología en Cascada (Braude, 2013)

2.8.1 Análisis de requerimiento.

En esta etapa se analiza las necesidades importantes de los usuarios finales de Software para determinar los requerimientos que se debe cubrir. En esta etapa surge un documento de especificaciones de requisitos denominada SRD, la que contiene las especificaciones completa sin entrar en detalle lo que debe hacer el sistema.

Esta etapa es primordial para las siguientes etapas ya que en la elaboración del Software no se podrá agregar más requerimientos.

2.8.2 Diseño del Sistema.

Descompone y organiza el sistema por módulos los cuales se pueden elaborar por separados dando como resultado el documento de diseño de Software (SDD) que contiene la descripción de la estructura global del sistema y la especificación de las funciones de cada módulo y describiendo la comunicación entre los módulos.

2.8.3 Codificación

También conocida como etapa de programación-implementación, en donde se implementa el código fuente, realizando prototipos, pruebas y ensayos para detectar y corregir errores.

Dependiendo del lenguaje de programación se crean bibliotecas y componentes para poder ser reutilizados.

2.8.4 Pruebas

Los módulos programados se ensamblan y se comprueba el correcto funcionamiento.

2.8.5 Implantación

Es poner en marcha el sistema y se conoce como producción.

2.8.6 Mantenimiento

Se da soporte al sistema y el mantenimiento respectivo.

2.9 Comparación metodología ágiles y metodología Cascada.

La comparación de la metodología se indica en la Tabla 6.

Para el presente proyecto se utilizará la metodología Scrum.

Tabla 6

Aspecto	Cascada	Scrum
Estilo de desarrollo	Anticipativo	Adaptativo
Requisitos	Conocidos, estables, claramente definidos y documentados	Desconocidos a priori, definidos durante el proyecto.
Arquitectura	Pesada y sobredimensionada para los actuales y futuros requisitos	Filosofía YAGNI no se debe nunca agregar funcionalidad excepto que sea necesario
Gestión	Centrada en procesos: mando y control	Centrada en la gente: liderazgo y colaboración.
Documentación	Detallada, conocimiento explícito	Ligera, conocimiento tácito.
Metas	Previsibilidad y optimización.	Exploración y adaptación.
Cambios	Aversión al cambio	Acepta el cambio
Organización del equipo	Equipos pree estructurados.	Equipos auto organizados.
Involucración del cliente	Pasivo, poca involucración.	Activo, un miembro más del equipo.
Cultura organizativa	Jerarquía con mando y control establecido.	Menos jerárquica con liderazgo y colaboraciones
Desarrollo del Software.	Acercamiento universal con solución predecible y altamente segura.	Acercamiento flexible adaptado a las necesidades particulares del proyecto.
Medida del éxito	Conforme a un plan	Evaluar globalmente el negocio.

CAPÍTULO III

ANÁLISIS Y DISEÑO DE LA APLICACIÓN

3.1. Captura de Requisitos

La presente especificación de requerimientos pertenece al desarrollo del Caso de Estudio " DESARROLLO E IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN PARA LA EMPRESA TABLEROS ÉDISON".

Introducción

Propósito

La intención del presente documento es definir los requerimientos que debe de tener la aplicación Web para el seguimiento de la elaboración de los tableros eléctricos de la empresa TABLEROS ÉDISON. Con las especificaciones levantadas se formaliza la funcionalidad y el aplicativo con el representa de la empresa. Tomando en cuenta que se va utilizar la metodología Scrum se podrá hacer modificaciones luego de cada Sprint.

Definiciones, Acrónimos y abreviaturas

SISPRO.- Abreviatura para definir el Sistema de seguimiento y control de Producción.

Java.- Lenguaje de programación utilizado para desarrollar del sistema SISPRO.

PostgreSQL.- Base de datos utilizada para el desarrollo e implementación del sistema SISPRO.

Eclipse.- IDE utilizado para el desarrollo de la aplicación.

3.2. IDENTIFICACIÓN DE ROLES Y TAREAS.

1. Roles

Administrador

Es el usuario que asigna los roles, los alias y las contraseñas a los demás usuarios.

Bodeguero

Es el usuario que gestiona el ingreso de los materiales que van a ser utilizados para la elaboración de los tableros eléctricos industriales.

Cliente:

Usuario que puede realizar el seguimiento de la producción de los productos contratados.

Secretaria:

Usuaría encargada de tomar el pedido de los clientes y luego elaborar la factura correspondiente.

Técnico operador:

Usuario que registra el status del avance de los productos pedidos por los clientes.

2. Tareas

Administrador

- Crear Usuarios.
- Asignar permisos a Usuarios.

Bodeguero

- Codificar los materiales
- Cuantificar los materiales.
- Ingresar materiales al sistema.

Cliente.

- Cotizar pedidos de productos.
- Realizar pedido de productos.
- Cancelar la factura.
- Realizar seguimiento de los productos.
- Retirar el producto.

Secretaria

- Recibir pedido del cliente.
- Elaborar proforma.
- Elabora el pedido del producto.
- Elaborar factura.

Técnico operador.

- Informe del proceso.
- Cambiar el status del producto.

3.3. Modelo de Casos de Uso

Sistema de control de proceso para la producción y construcción de paneles, tal como se indica en la Figura 16.

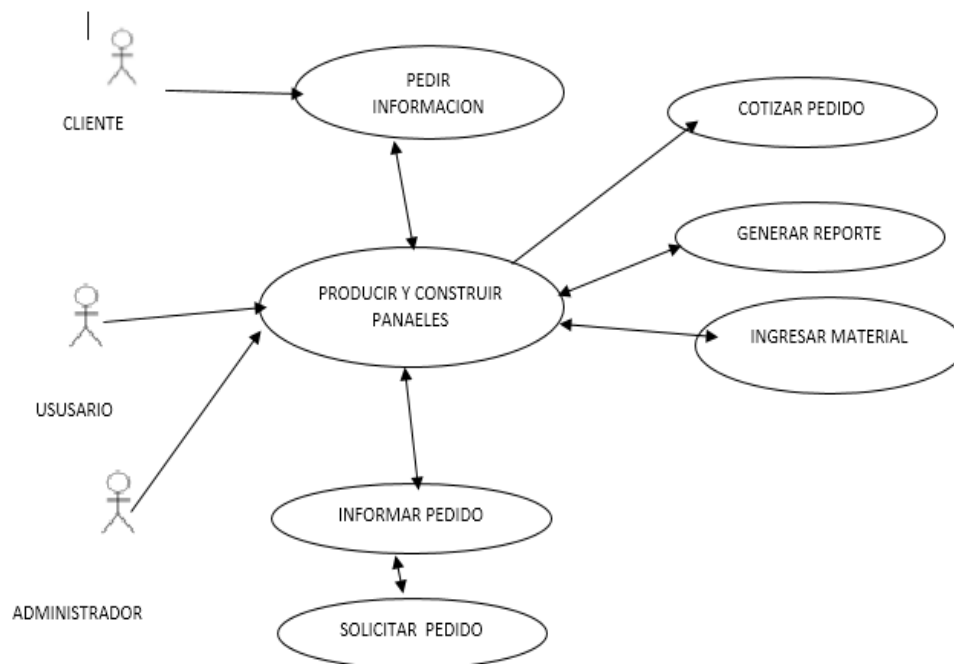


Figura 16 Sistema de control de proceso para la producción y construcción de paneles.

Caso de uso SISPRO

Como se indica en la Figura 17

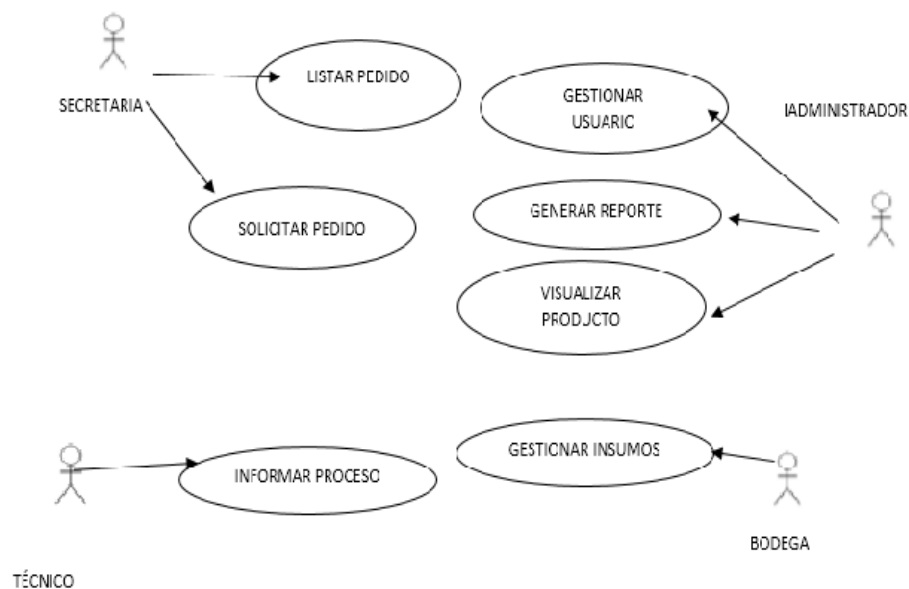


Figura 17 Caso uso SISPRO

Descripción del caso de uso SISPRO:

El administrador creará, eliminará, actualizará y consultará los usuarios del sistema.

El administrador podrá crear los reportes para la Gerencia así como también podrá cambiar el estado del producto.

Bodega realiza la gestión de los insumos.

El técnico está encargado de realizar el informe del proceso, el cambio de las etapas de la construcción y el estado del producto.

La secretaria listara los pedidos y solicitará los pedidos y podrá manejar la información.

Caso de uso gestión de clientes.

Como se indica en la Figura 18.

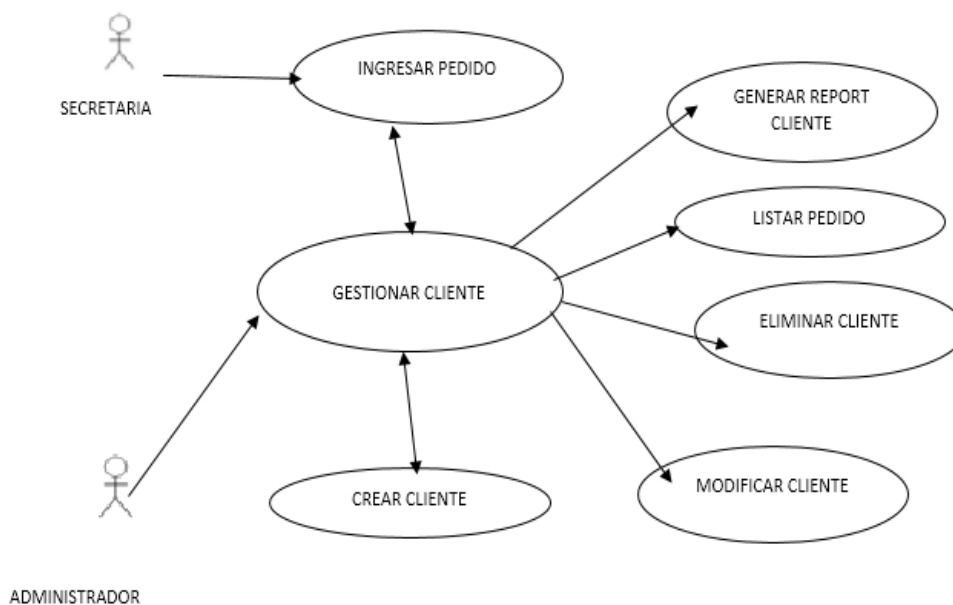


Figura 18 Caso de uso gestión de clientes.

El administrador como la secretaria tendrán acceso para gestionar la cuenta del cliente que consta de crear, modificar, eliminar y actualizar los datos de los clientes.

Podrá generar los pedidos de los clientes para lo cual requerirá la información exacta de qué tipo de tablero desea realizar el contrato.

Para generar la proforma se tomará en cuenta las tres etapas que consiste para la creación de un tablero eléctrico.

La primera etapa es la construcción del armario, donde tendrá que revisar todos los materiales que incluirán en esta etapa para poder proseguir a la siguiente.

La etapa de pintura es donde se lijará y pintará el mueble, para lo cual se tiene que determinar el presupuesto de los materiales empleados.

Por último la etapa de cableado se contabilizará todo el material eléctrico que va ser empleado tomando en cuenta que para cada región (sierra-costa) hay un material específico y por tal motivo una diferencia en el costo.

Caso de uso pedidos.

Como se indica en la Figura 19.

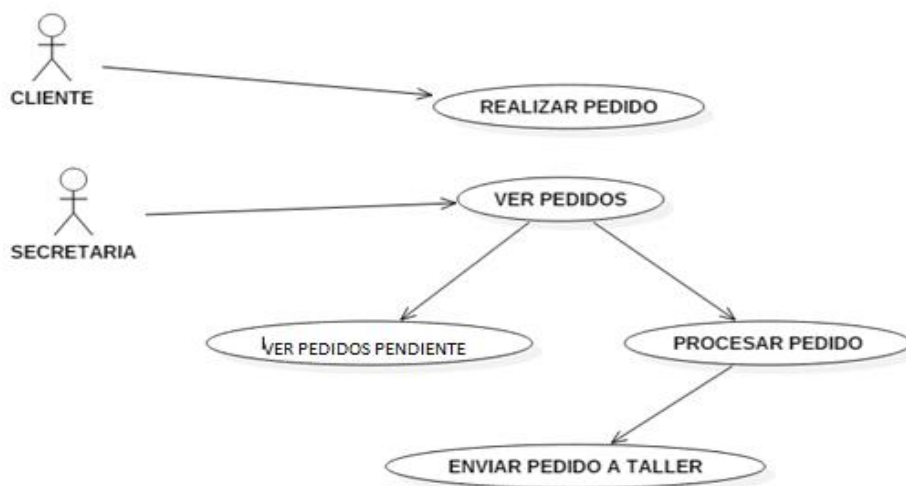


Figura 19 Caso de Uso pedidos

El cliente realiza el pedido de acuerdo a sus necesidades, la secretaria revisará los pedidos que los usuarios han realizado si por algún motivo

existiera pedidos pendientes gestionará para que estos trabajos se procesen.

Los pedidos que estén listos para enviar al taller tendrán que haber realizado un abono o anticipo en las condiciones que se fijaron el contrato.

Si se cumplieron los requisitos anteriores se enviara el pedido al taller para su construcción o producción.

Casos de uso Producción

Como se indica en la Figura 20.

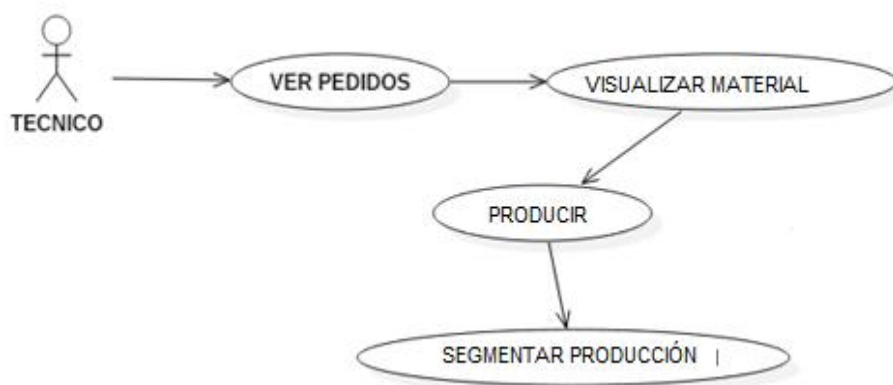


Figura 20 Caso uso de Producción

El técnico realiza los pedidos dependiendo los trabajos de lista de materiales para la construcción de la caja de distribución, cuando el producto sea elaborado lista los materiales para la etapa de lijado y pintura, terminado esto la última etapa es la construcción parte eléctrica.

Se debe de considerar los materiales empleado en la elaboración del producto tiene que estar acorde a la ubicación de la construcción que

se va a emplear ya que la humedad tiene un factor importante en mantenimiento del tablero.

Caso de uso Costos

Como se indica en la Figura 21.

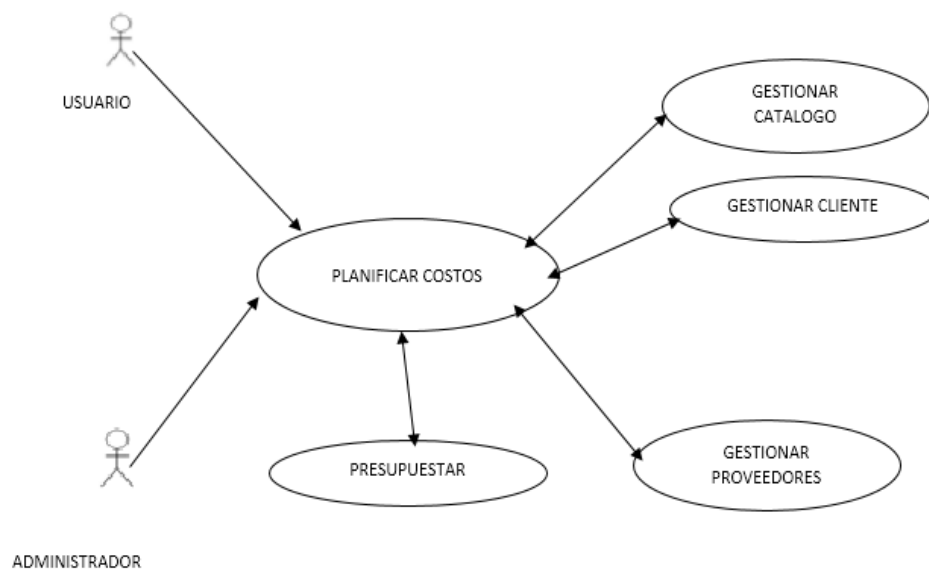


Figura 21 Caso de Uso de Costos

3.4. DIAGRAMA DE SECUENCIA

Como se indica en la Figura 22.

Diagrama de secuencia Usuario Administrador

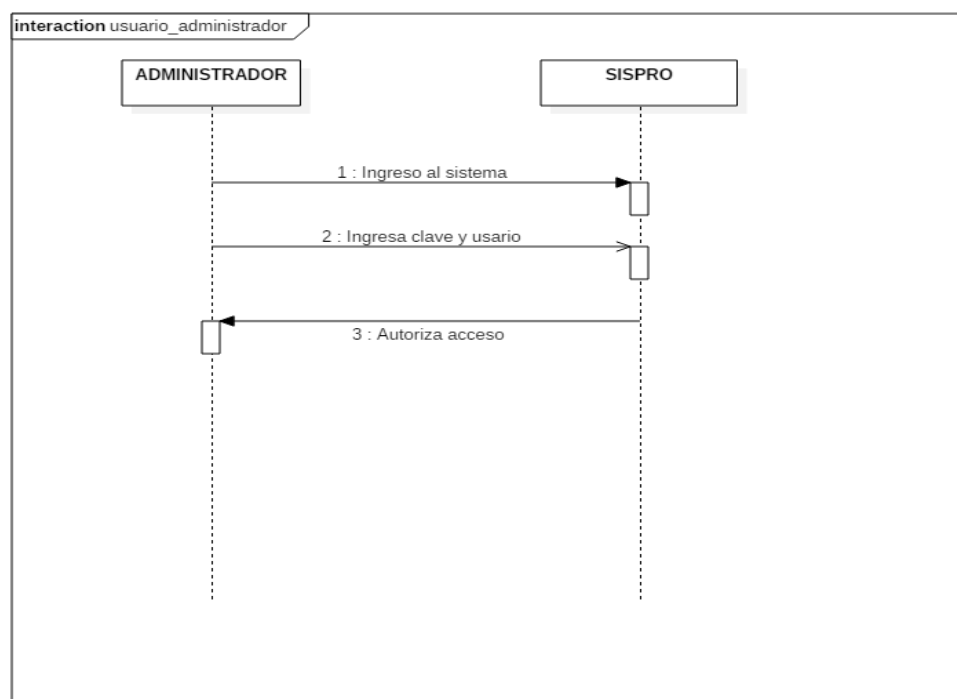


Figura 22 Diagrama de secuencia Usuario Administrador

El administrador ingresará al sistema SISPRO, para ingresar clave y usuario de los clientes internos y externos de la empresa, el sistema SISPRO validará las credenciales.

Si las credenciales son correctas permitirá el acceso al sistema, si las credenciales son erróneas dará un mensaje de error.

Diagrama de secuencia Usuario Subadministrador

Como se indica en la Figura 23.

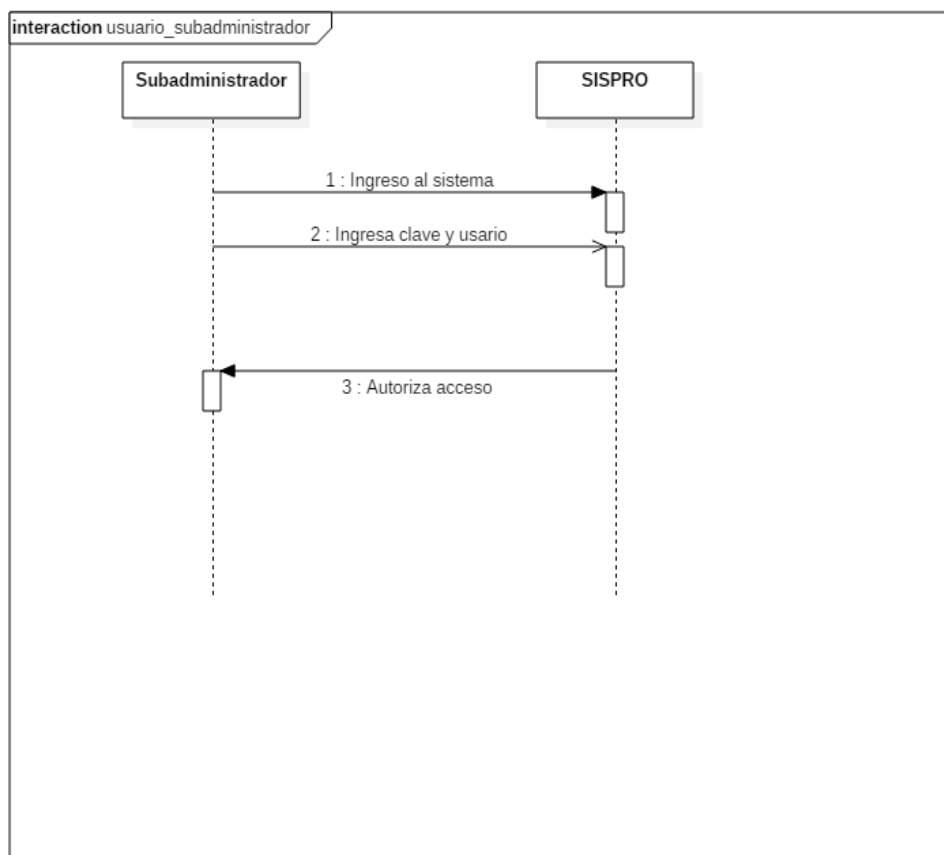


Figura 23 Diagrama de secuencia Usuario Subadministrador

El Subadministrador ingresará al sistema SISPRO, para ingresar clave y usuario de los clientes internos y externos de la empresa, el sistema SISPRO validará las credenciales.

Si las credenciales son correctas permitirá el acceso al sistema, si las credenciales son erróneas dará un mensaje de error.

Diagrama de secuencia Crear Usuario

Como se indica en la Figura 24.

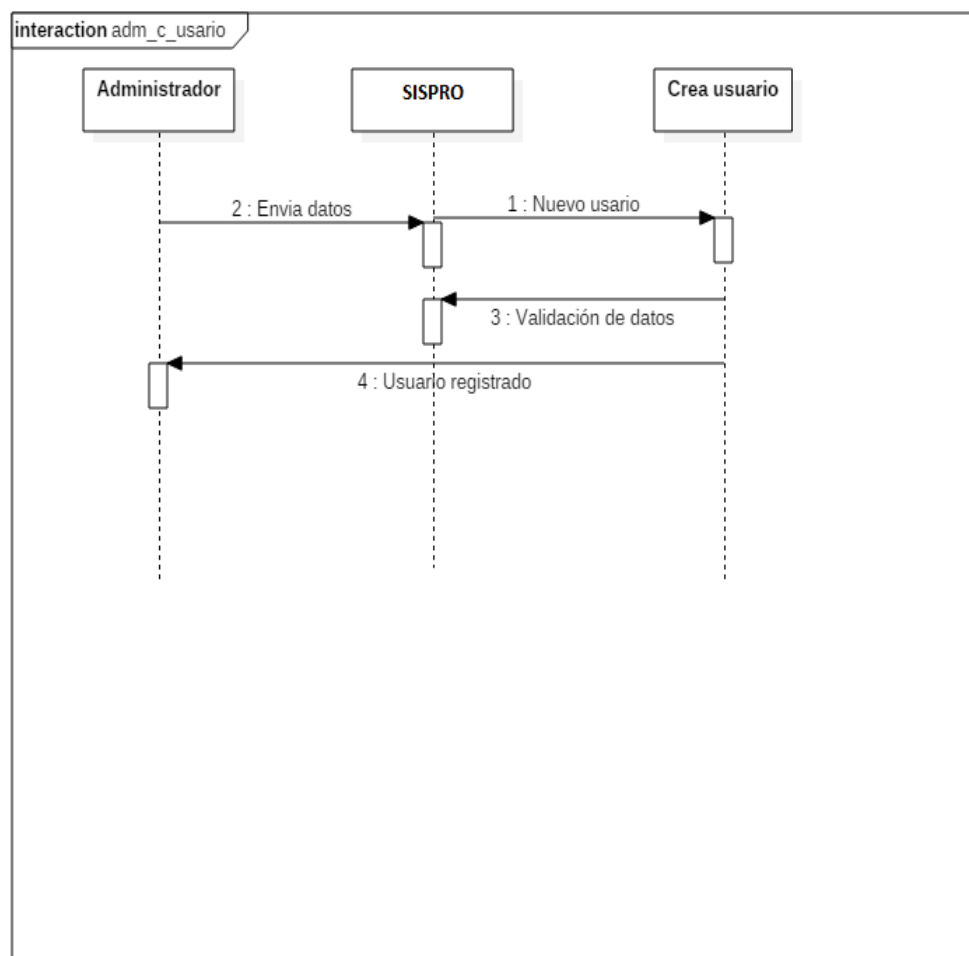


Figura 24 Diagrama de secuencia Crear Usuario

El administrador enviara datos al sistema SISPRO, para crear un nuevo usuario y clave, el sistema valida los datos y envía un mensaje al administrador que los datos fueron generados exitosamente, y el usuario queda registrado.

Diagrama de secuencia Usuario Secretaria.

Como se indica en la Figura 25.

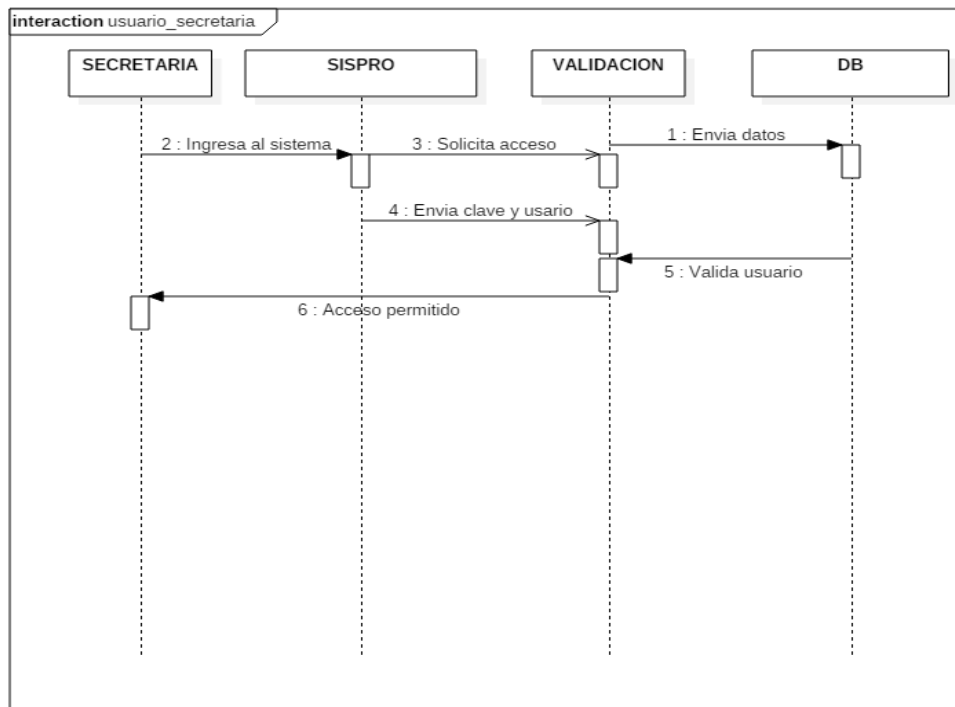


Figura 25 Diagrama de secuencia Usuario Secretaria

Diagrama de secuencia Crear Panel.

Como se indica en la Figura 26.

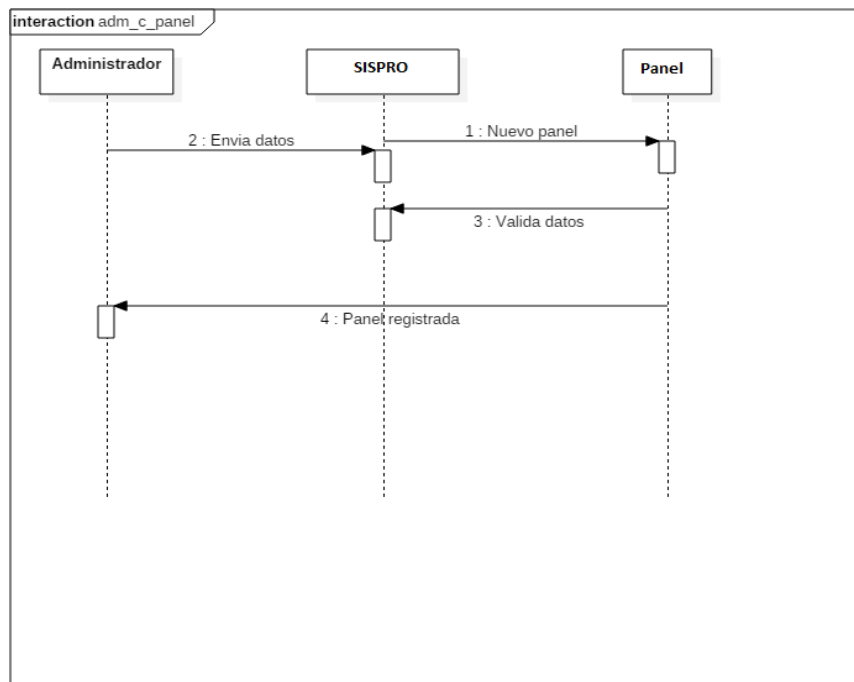


Figura 26 Diagrama de secuencia Crear Panel

Diagrama de secuencia Administrar Factura.

Como se indica en la Figura 27.

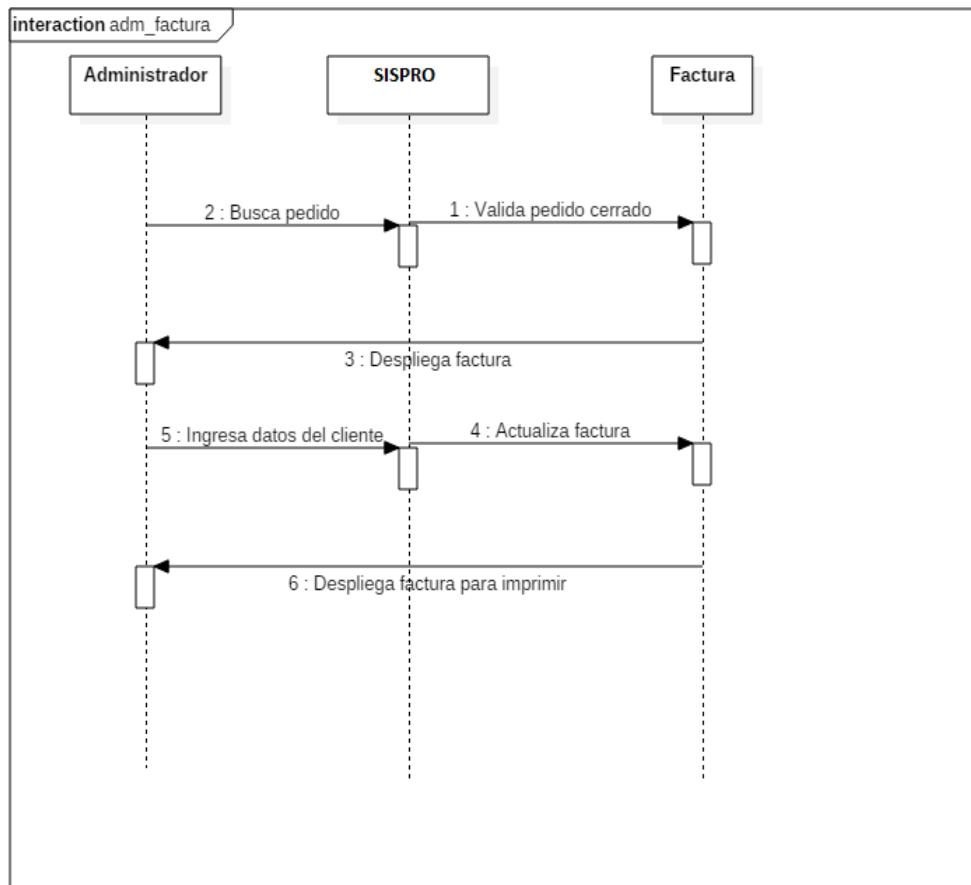
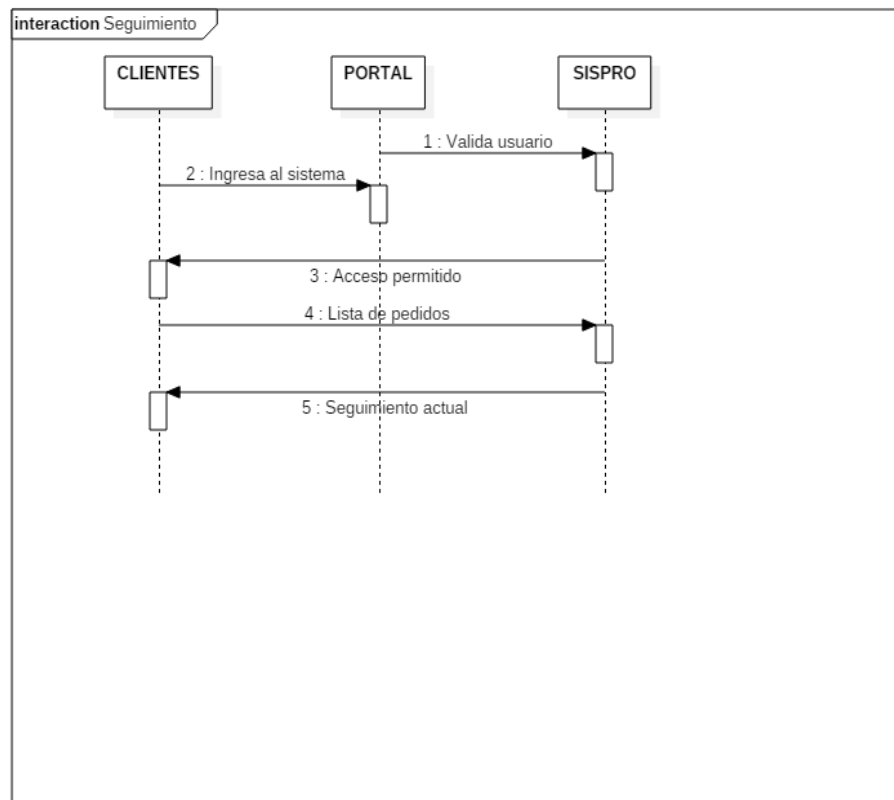


Figura 27 Diagrama de secuencia Administrar Factura.

Diagrama de secuencia Administrar Seguimiento.

Como se indica en la Figura 28.

Figura 28 Diagrama de secuencia Administrar



3.5. DIAGRAMA DE ACTIVIDADES.

Diagrama de actividad Crear Usuario.

Como se indica en la Figura 29

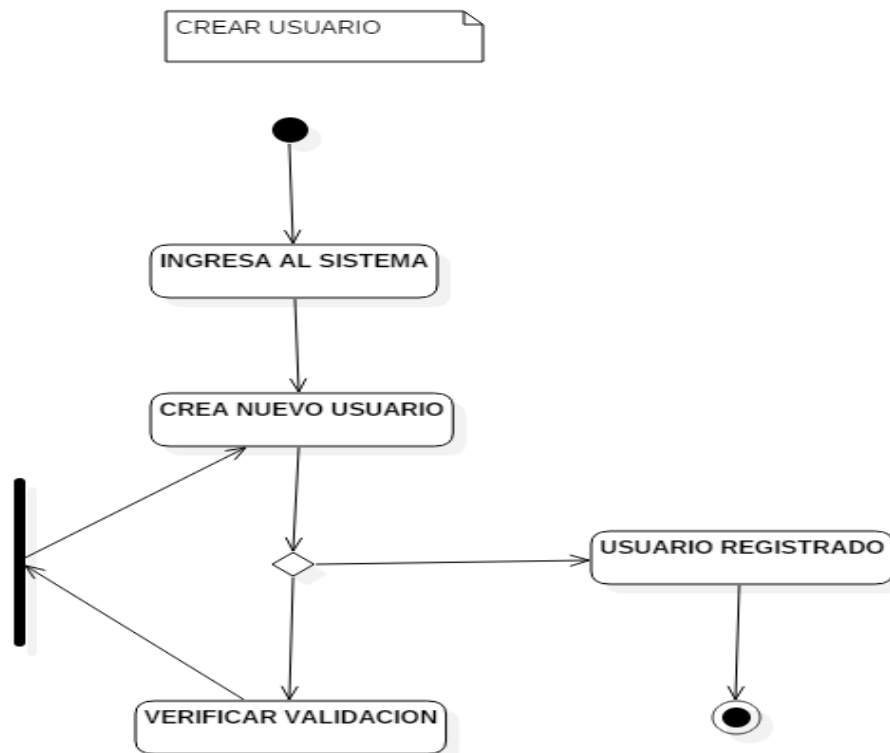


Figura 29 Diagrama de actividad Crear Usuario.

Diagrama de actividad Crear Panel.

Como se indica en la Figura 30.

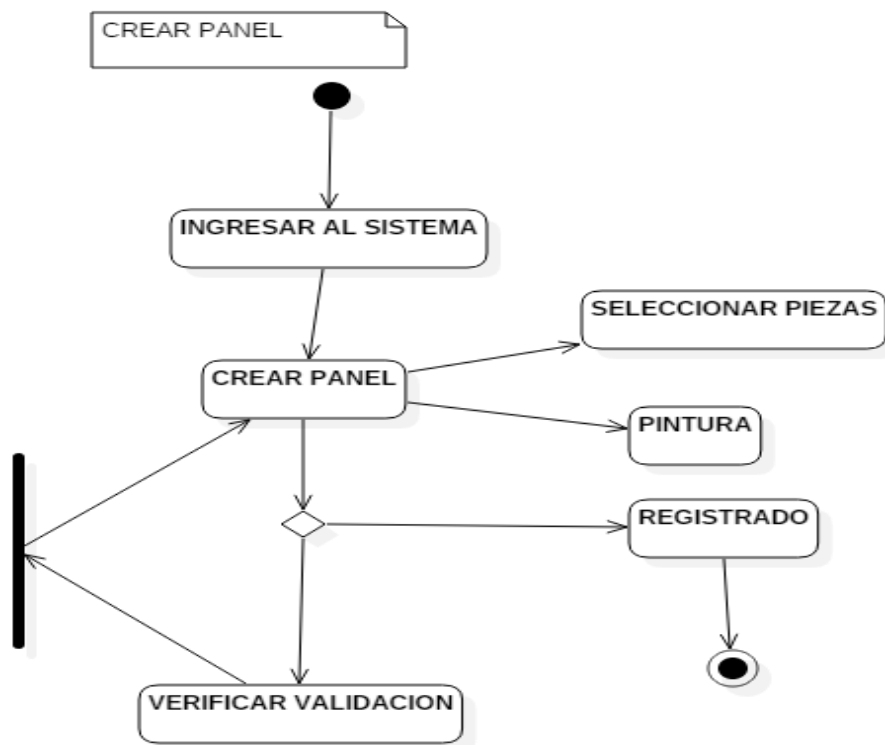


Figura 30 Diagrama de actividad Crear Panel.

Diagrama de actividad Usuario Administrador.

Como se indica en la Figura 31.

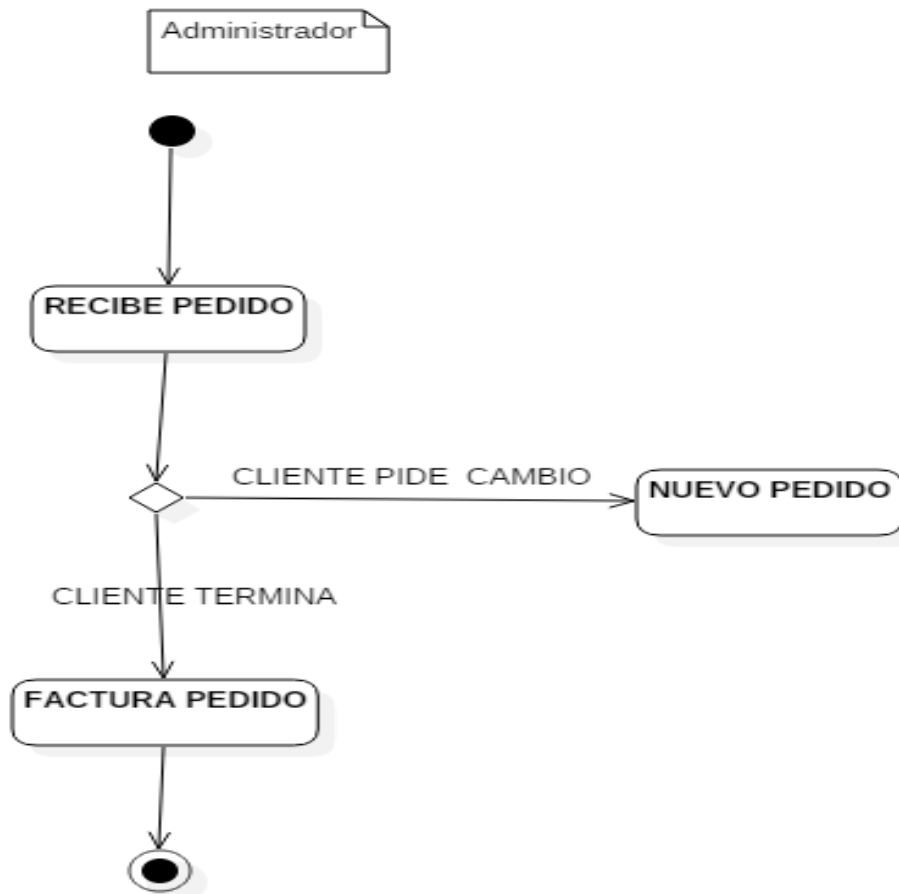


Figura 31 Diagrama de actividad Usuario Administrador.

Diagrama de actividad Usuario Cliente.

Como se indica en la Figura 32.

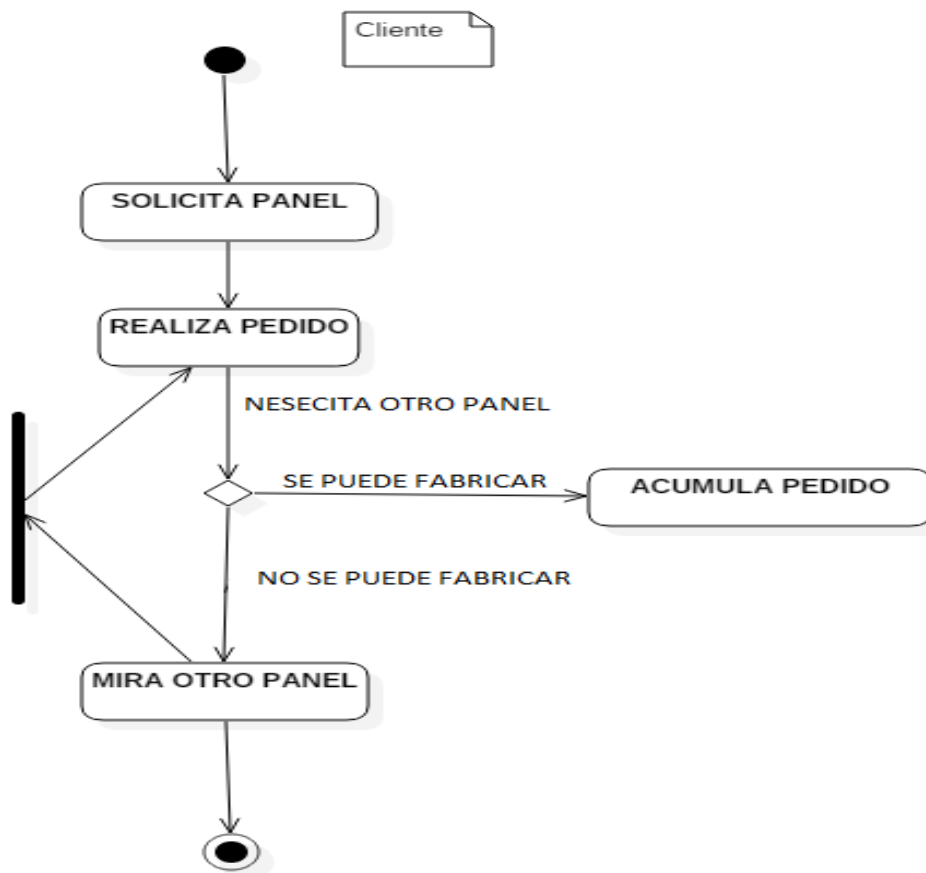


Figura 32 Diagrama de actividad Usuario Cliente.

Diagrama de actividad Usuario Secretaria.

Como se indica en la Figura 33.

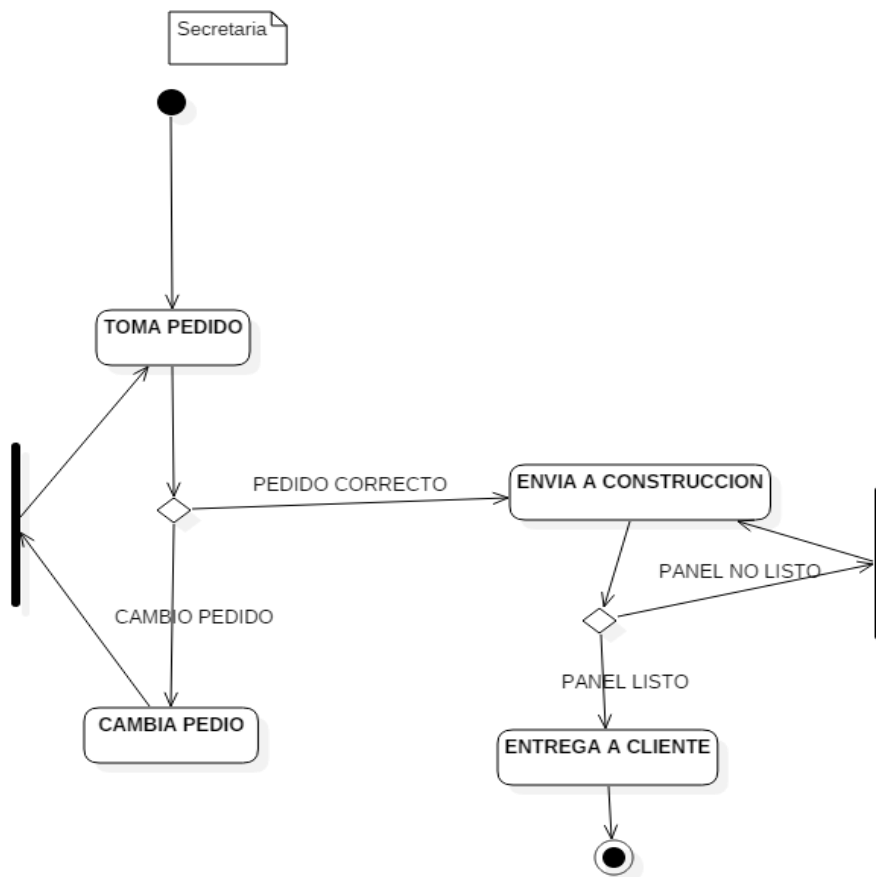


Figura 33 Diagrama de actividad Usuario Secretaria.

La secretaria toma el pedido para la fabricación del tablero, si el cliente necesita hacer un cambio en el pedido lo puede realizar, cuando el pedido se encuentra correcto y el cliente acepta es enviado al área de construcción para su elaboración, cuando el panel eléctrico está listo se lo entrega al cliente.

3.5.1.Requerimientos No Funcionales

- El sistema SISPRO poseerá usuarios y contraseñas para cada persona que acceda al sistema, individuo que no disponga usuario o contraseña no podrá ingresar. Usuario que ingrese al sistema solo podrá acceder a las funciones asignadas por el administrador.
- El sistema estará instalado en la intranet de la empresa teniendo la posibilidad de instalarlo en un hosting si la empresa lo requiera.

3.6. DIAGRAMA FÍSICO DE LA BASE DE DATOS.

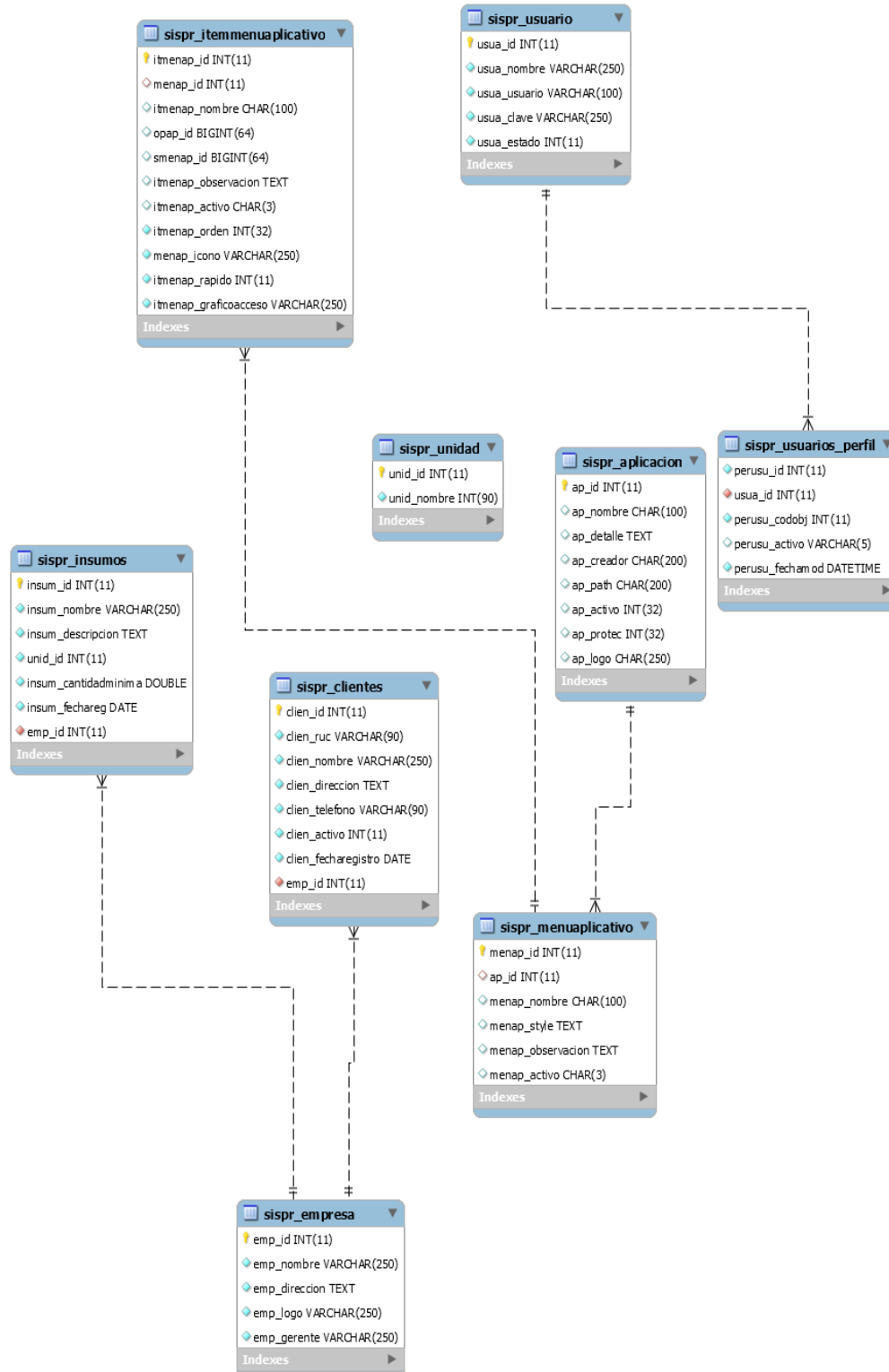


Figura 34 : DIAGRAMA FÍSICO DE LA BASE DE DATOS.

3.7. Diagrama de Clases

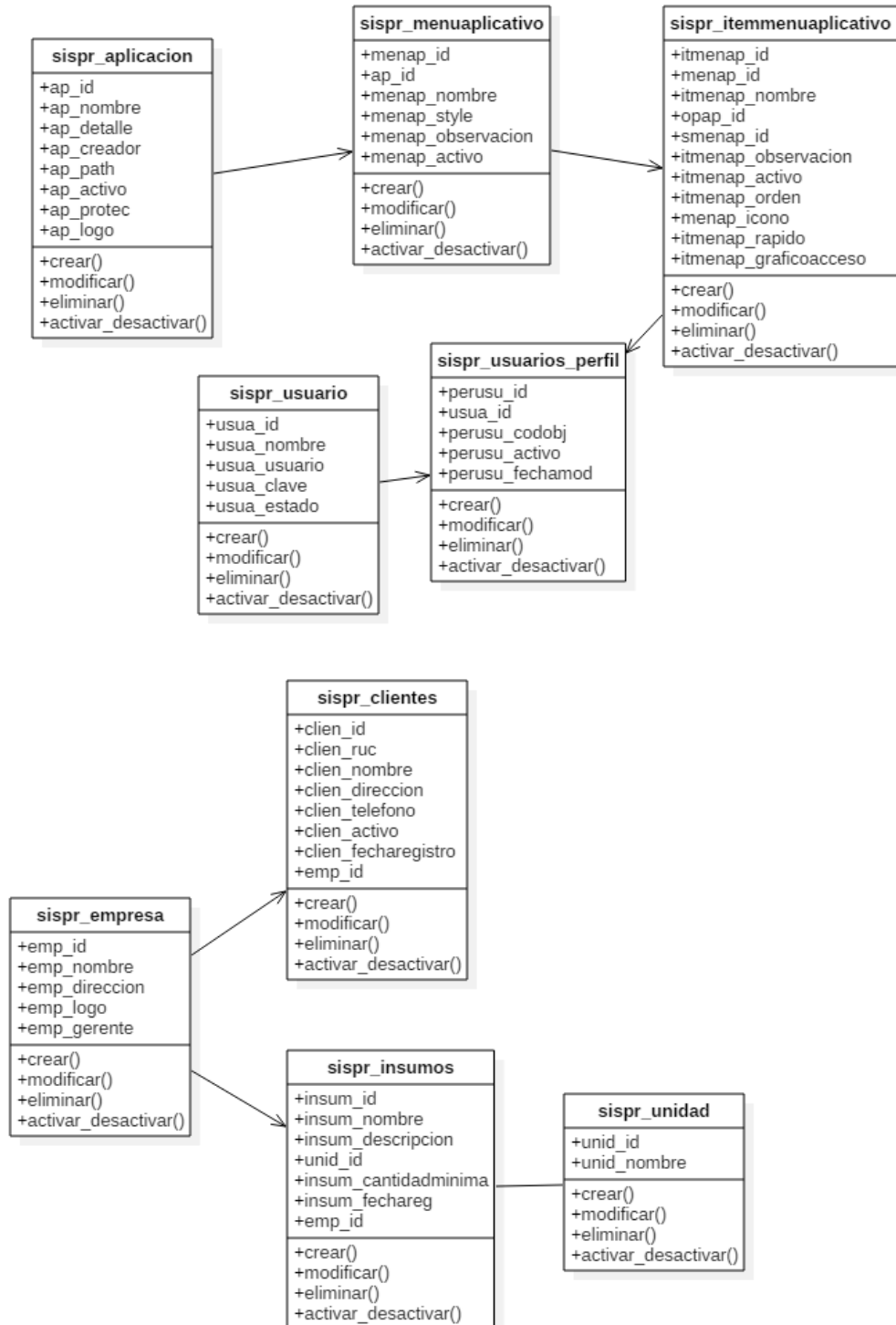


Figura 35 : Diagrama de Clases

3.8. Artefactos propios de la metodología

Tabla 7

Product Backlog - Definición del Proyecto

Información del Proyecto		
Nombre del Proyecto:	SISTEMA PARA EL MANEJO DE PRODUCCIÓN DE TABLEROS	
Gerente del Proyecto:	Daniel Quishpi	
Tipo de Proyecto de Software:	Nuevo desarrollo	Mantenimiento
	X	
Características		
Duración (meses) :	3	
Presupuesto Estimado:	4200	
Descripción del Proyecto :	Se requiere realizar un desarrollo que permita llevar la administración del proceso de producción de tableros en la empresa Tablero Edison.	
Justificación del Proyecto :	Al momento el procedimiento de compra de insumos, construcción de tableros se lo lleva de forma manual lo cual provoca que se desperdicie material y los tiempos de entrega no sean óptimos, por este motivo se requiere automatizar este proceso.	
Objetivos específicos:	<ul style="list-style-type: none"> - Optimizar el uso de insumos. - Optimizar el proceso de producción. 	

	<ul style="list-style-type: none"> - Mejorar los tiempos de entrega. - Mantener reportes de entregas y producción. 	
Hitos Entregables Mayores		
Nombre	Comentario /fecha de entrega	
SISPRO		
<ul style="list-style-type: none"> • Fase de Inicio 		
<ul style="list-style-type: none"> • Fase de Realización de Trabajo 		
<ul style="list-style-type: none"> • Fase de Desarrollo 		
<ul style="list-style-type: none"> • Fase de Cierre Formal de Proyecto 		
Dentro del Alcance		
Fuera del Alcance		
Actualizaciones a la Base de Datos y/o tablas, campos nuevos		
Creación de tipos de usuarios (Usuarios, Usuarios Externos)		
Recurso Humano (Inicial)		
Recurso (rol)	Tareas Asignadas	Cantidad
Daniel Quishpi	Administrador de proyecto asignado	1
Daniel Quishpi	Gerente de proyecto	1
Marco Paucar	Líder Técnico (Desarrollador)	1
Marco Paucar	Responsable-Sponsor (Desarrollador)	1
Observaciones:		

Riesgos	Probabilidad de Ocurrencia	Impacto
Tiempo asignado para cumplir con ejecución del proyecto	Baja	1
Falta de recursos	Media	2
Suspension del proyecto	Baja	3
<p>Probabilidad de Ocurrencia:</p> <p>Nivel 3 = Alta La amenaza está altamente motivada y suficientemente capaz de llevarse a cabo.</p> <p>Nivel 2 = Media La amenaza es posible y puede medianamente llevarse a cabo.</p> <p>Nivel 1= Baja La amenaza no posee la suficiente motivación y es mínima de que se lleve a cabo</p>	<p>Impacto:</p> <p>Nivel 3 = Catastrófico Afectación directa a la ejecución del proyecto</p> <p>Nivel 2 = Moderado Tiene afectación , pero controlable</p> <p>Nivel 1= Menor La amenaza está presente pero no afecta al proyecto</p>	

Sprint Backlog- Definición del Sprint

Definiciones de los Sprints, como se indica en las Tablas del 8 al13

Tabla 8

Sprint 1

Numero:	Sprint 1
Tareas:	Recolección de requerimientos Análisis de requerimientos
Tiempo:	2 semanas
Recurso:	Tesistas
Porcentaje a realizarse:	100%

Tabla 9

Sprint 2

Numero:	Sprint 2
Tareas:	Casos de usos del proyecto Desarrollar la base de datos
Tiempo:	2 semanas
Recurso:	Tesista
Porcentaje a realizarse:	100%

Tabla 10

Sprint 3

Numero:	Sprint 3
Tareas:	Formulario de creación de usuarios Autenticación de usuarios en el sistema
Tiempo:	2 semanas
Recurso:	Tesista
Porcentaje a realizarse:	100%

Tabla 11

Sprint 4

Numero:	Sprint 4
Tareas:	Formulario de asignación de permisos Formulario de ingreso de productos
Tiempo:	2 semanas
Recurso:	Tesista
Porcentaje a realizarse:	100%

Tabla 12

Sprint 5

Numero:	Sprint 5
Tareas:	Formulario de ingreso de producto elaborado Formulario Factura.
Tiempo:	2 semanas
Recurso:	Tesista
Porcentaje a realizarse:	100%

Tabla 13

Sprint 6

Numero:	Sprint 6
Tareas:	Formulario de avance de producción. Elaboración de reportes
Tiempo:	2 semanas
Recurso:	Tesista
Porcentaje a realizarse:	100%

Tabla 14

Burndown Chart

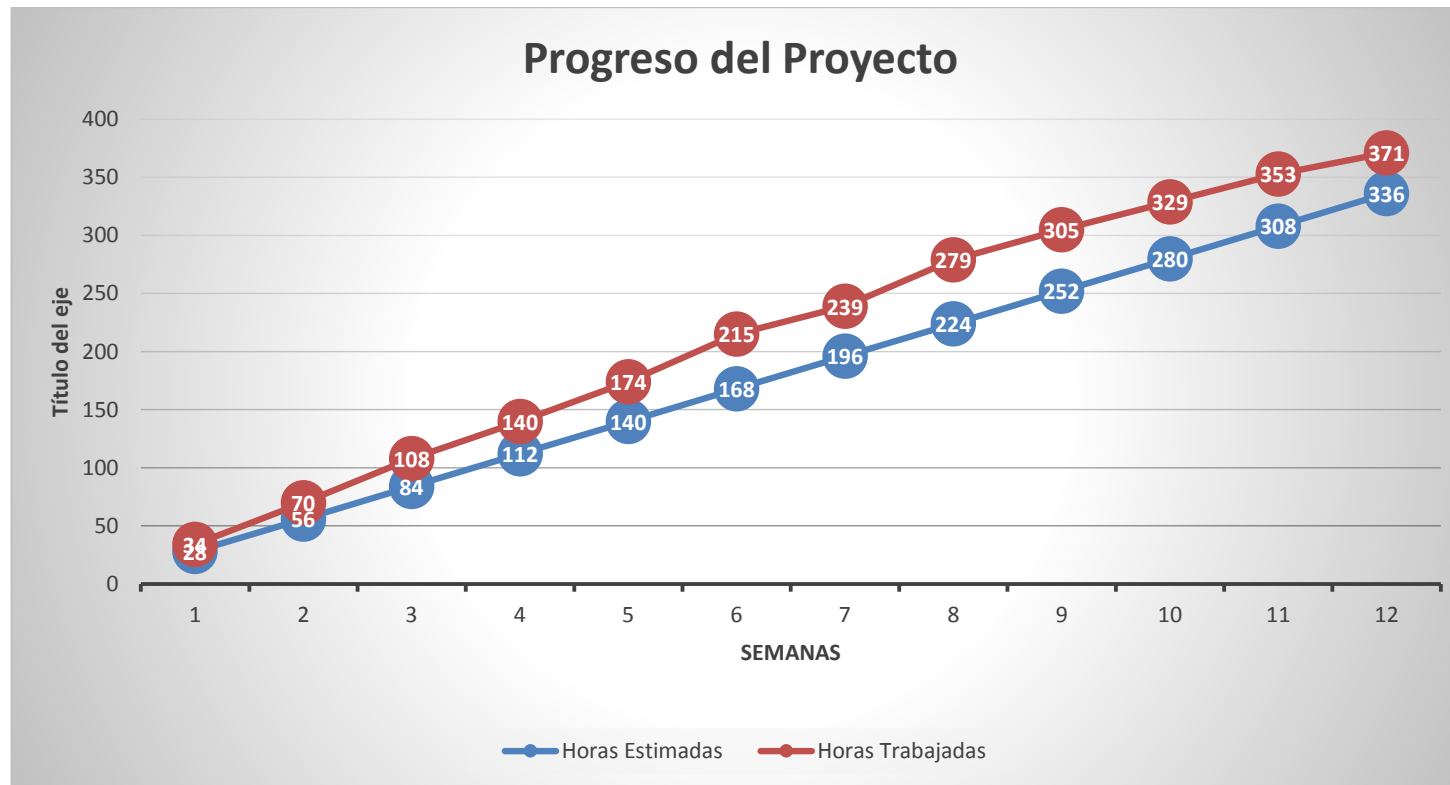
BURNDOWN CHART

Sprint		Sprint 1 - Semana		Sprint 2 - Semana		Sprint 3 - Semana		Sprint 4 - Semana		Sprint 5 - Semana		Sprint 6 - Semana	
Tarea	Horas x Tare	1	2	5	6	9	10	13	14	13	14	13	14
Recolección de requerimientos	36	30	6	0	0	0	0	0	0	0	0	0	0
Análisis de requerimientos	34	4	30	0	0	0	0	0	0	0	0	0	0
Casos de usos del proyecto	30	0	0	30	0	0	0	0	0	0	0	0	0
Desarrollar la base de datos	32	0	0	0	32	0	0	0	0	0	0	0	0
Formulario de creación de usuarios	28	0	0	0	0	26	2	0	0	0	0	0	0
Autenticación de usuarios en el sistema	30	0	0	0	0	0	30	0	0	0	0	0	0
Formulario de asignación de permisos	30	0	0	0	0	0	0	20	10	0	0	0	0
Formulario de ingreso de productos	34	0	0	0	0	0	0	4	30	0	0	0	0
Formulario de ingreso de producto elaborado	26	0	0	0	0	0	0	0	0	26	0	0	0
Formulario Factura.	24	0	0	0	0	0	0	0	0	0	24	0	0
Formulario de avance de producción.	22	0	0	0	0	0	0	0	0	0	0	20	2
Elaboración de reportes	20	0	0	0	0	0	0	0	0	0	0	4	16
Horas estimadas	336	28	28	28	28	28	28	28	28	28	28	28	28
Horas Realizadas	371	34	36	38	32	34	41	24	40	26	24	24	18

GRAFICO DE PROGRESO DEL PROYECTO

Tabla 15

Progreso del Proyecto



CAPÍTULO IV

DESARROLLO

4.1. Arquitectura a Implementarse

4.1.1. Arquitectura Web

Para el presente proyecto utiliza Jboss como servidor de aplicaciones que es de código abierto dispone de una plataforma de alto rendimiento para e-business.

La arquitectura de un servidor web está constituido por tres componentes principales que son un servidor Web, Conexión de red, uno o más Clientes.

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó, tal como se indica en la Figura 36

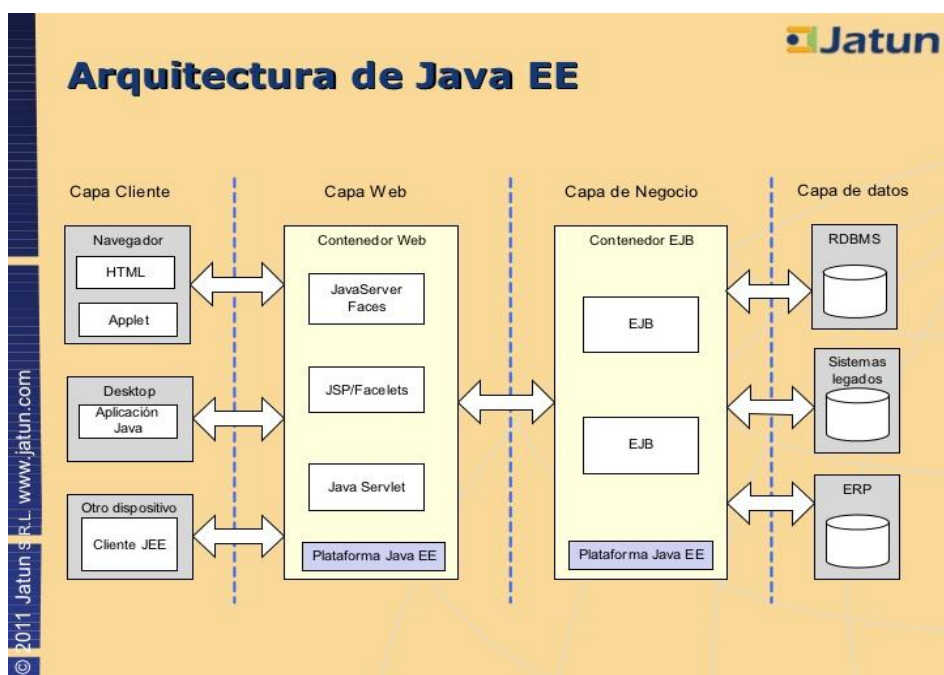


Figura 36 : Arquitectura Java EE (Jatun, s.f.)

4.1.2. Patrón de arquitectura estándar J2EE

Los componentes de la arquitectura estándar del J2EE se indican en la Figura 37

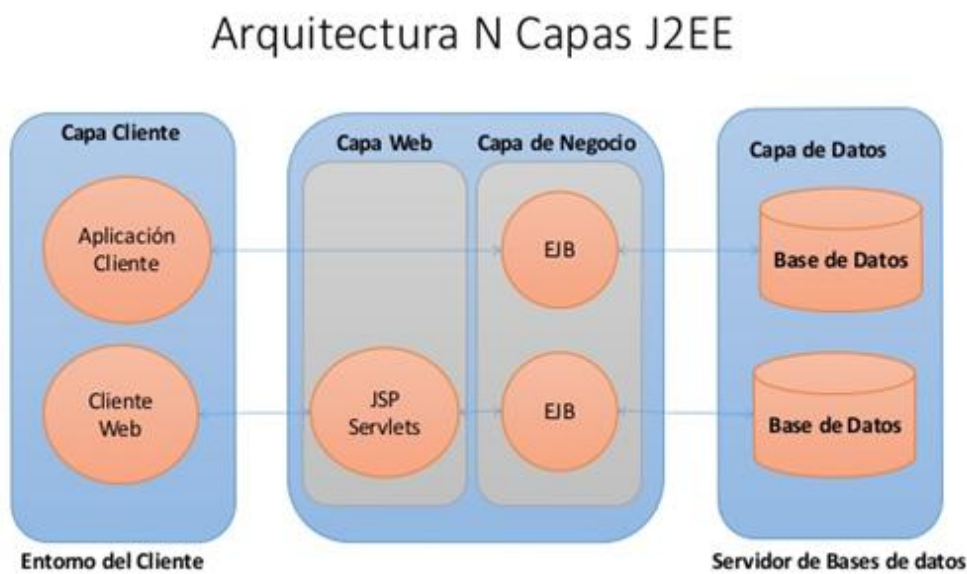


Figura 37 (Jatun, s.f.)

J2EE son las siglas de Java 2 Enterprise Edition. Comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales. Debido a que J2EE no deja de ser un estándar, permitiendo a la aplicación ser escalable, estable.

J2EE se basa en una arquitectura multicapa y utiliza un patrón arquitectónico MVC (Model – View – Controller). Con este modelo se permite una fácil separación de la interfaz gráfica y del modelo de negocios, gracias a un controlador que los mantiene desacoplados. Con esto podríamos tener configuraciones en las que el cliente tan sólo disponga de interfaz gráfica accede a un servidor donde se implementa el modelo. De este modo, cambios en el modelo sólo afectarían al servidor.

Tanto Servlets como JSP son soluciones para el desarrollo de aplicaciones empresariales.

Difieren en su modelo de programación. Una página JSP es esencialmente un documento que especifica contenido dinámico, mientras que un servlet se puede comparar más a un programa que produce este tipo de contenido. Aunque ambas tecnologías pueden presentarse como soluciones independiente para el desarrollo de aplicaciones, son realmente potentes en un contexto MVC trabajando conjuntamente.

El modelo–vista–controlador (MVC) es un patrón de arquitectura de Software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de Software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Modelo.- Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'

Controlador.- Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la

información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'

Vista.- Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

En contexto un ambiente de servicios de aplicaciones distribuidas se compone de:

- Runtime hosting Applications(Servidor de aplicaciones.)
- Java Database Connectivity (jdbc)
- Remote Method Invocation(RMI)
- Enterprise Java Beans (EJB)
- Java Server Pages (JSP)
- Java Message Service(JMS)
- Java Naming Directory Interface(JNDI)
- Java Transaction API
- Java Mail
- Estándar Java API

4.1.3.EJB (Enterprise JavaBeans)

EJB (Enterprise JavaBeans) es un modelo de programación que nos permite construir aplicaciones Java mediante objetos ligeros (como POJO's). Cuando construimos una aplicación, son muchas las responsabilidades que

se deben tener en cuenta, como la seguridad, transaccionalidad, concurrencia, etc. El estandar EJB nos permite centrarnos en el código de la lógica de negocio del problema que deseamos solucionar y deja el resto de responsabilidades al contenedor de aplicaciones donde se ejecutará la aplicación.

4.2. Estándares de implementación

4.2.1. ESTÁNDARES DE CODIFICACIÓN

Para el presente proyecto se utiliza los estándares internacionales, buenas prácticas, etc.

NOMENCLATURA –Paquetes

El paquete consiste en nombre de la empresa como estaría en la página web. En el orden invertido sin poner el www. Para el caso del presente proyecto es com.tédison teniendo este como raíz, se irán agregando más nombres de acuerdo a las necesidades separadas por un punto.

- Físicamente se crearán carpetas por cada punto que este en el paquete.
- Los nombres que contenga el paquete serán en.

NOMENCLATURA –Interfaces

- Serán creados en el IDE eclipse como Interface.
- El nombre comenzara con mayúscula y con formato CamelCase.

NOMENCLATURA –Clases

- Para los nombres se utiliza el formato CamelCase
- La primera letra en Mayúscula
- Los nombres son simples y descriptivos.
- Sustantivo.

NOMENCLATURA –Métodos

- Los métodos son verbos en infinitivo.
- Están en formato CamelCase.
- La primera letra en minúscula
- No contienen caracteres especiales.
- Los nombres son suficientemente descriptivos.

NOMENCLATURA –Variables

- Comienzan con minúscula y formato CamelCase.
- Variables globales Todo en mayuscula
- No contienen caracteres especiales

ESTILO DE CODIFICACIÓN –Comentarios

- No se hace un uso abusivo de ellos
- /** Para documentar
- /* Comentario varias líneas
- // una sola línea
- Se evita el uso de caracteres especiales.

ESTILO DE CODIFICACIÓN –Declaraciones

- Se evita la duplicidad de los nombres de variables en diferentes niveles dentro de la misma clase.

ESTILO DE CODIFICACIÓN –Sentencias

- Para bloque se utiliza { }.

- Dentro de una iteración no se modifica la variable que este como contador.

BUENAS PRÁCTICAS –Propiedades

- Para ingresar a los datos de una clase se utiliza los getters y los setters, con esto se precautela la información de cada clase y se cumple el paradigma del encapsulamiento.

BUENAS PRÁCTICAS –Métodos

- No se accede a un método estático desde una instancia de una clase.

4.3. Desarrollo de los módulos del sistema

Módulo de Administración

Es el encargado de administrar los (Usuario y clave)

- Crear Usuarios.
- Asignar permisos a Usuarios (de acuerdo al perfil).

Módulo de Reportes

- Elabora los reportes
- Elabora las proformas
- Elabora las facturas.

Módulo Productos

Es el encargado de los productos que se utilizan en la fabricación de un tablero:

- Ingresar productos al sistema
- Modificar productos del sistema

- Eliminar productos del sistema
- Actualizar productos del sistema
- Ingresar el costo de los productos.

Con estos valores se puede realizar el cálculo de los tableros que se venderán.

Módulo Seguimiento de Productos

- Verificar el estado del producto

4.4. INTEGRACIÓN DEL CÓDIGO

La integración del código en este proyecto se realiza en la puesta a producción del sistema de gestión de bodegas.

- Elaboración de archivos entregables
- Elaboración de la base de datos
- Generar el archivo WAR del SISPRO.
- Elaboración de manuales del usuario y manual técnico.

4.5. PRUEBAS

Se realizan en pre-producción en servidor local.

- Se revisa con la representante de empresa Tableros Édison que se ejecute lo especificado en el levantamiento de requerimientos.
- Se revisa el ingreso de perfiles.
- Se verifica que los reportes estén correctos.
- Se revisa que la información ingresada este en la base de datos correctamente.
- Los errores encontrados son solucionados.
- Se firma la carta de aceptación por parte de la empresa.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Para realizar la especificación de requerimientos del Sistema SISPRO se utilizó el Lenguaje de Modelamiento Unificado (UML), porque indica el comportamiento del sistema.
- Para el análisis y diseño del Sistema SISPRO se utilizó la metodología SCRUM, dando como resultado entregas periódicas del sistema.
- En el desarrollo del Sistema SISPRO se utilizó Software estables y robustos como base datos PostgreSQL, IDE de desarrollo Eclipse y servidor de aplicaciones Jboss, dando como resultado una interfaz amigable y sencilla que proporciona a la empresa confianza para utilizarlo.
- Para la revisión de pruebas del Sistema SISPRO, se lo realizó en las instalaciones de la empresa con el personal asignado por el gerente dando como resultado lo esperado por la empresa.
- La implantación se lo realizo en un servidor de la empresa en la actualidad se encuentra en funcionamiento.

5.2. Recomendaciones

- Para desarrollar el Sistema SISPRO, se debe considerar la metodología más adecuada, analizando ventajas y desventajas que se puedan presentar en un proyecto, es importante tomar las características propias del negocio, para la correcta elección de la metodología.
- Recomendamos el uso de una metodología ágil como Scrum que se acopla fácilmente con la gestión de riesgo, propone una constante revisión del proyecto, y permite detectar y solucionar inconvenientes encontrados en el desarrollo de un sistema, sin afectar el alcance o el tamaño del proyecto.
- En el desarrollo de un sistema se tiene que considerar los estándares de codificación para poder trabajar en equipo.

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

- Berzal, F. (2014). *UML El Lenguaje Unificado de Modelado*. Obtenido de <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>
- Braude. (2013). *CASCADA*. Obtenido de <http://metodologiaencascada.blogspot.com/>
- Chávez, C. (2011). *servidor de aplicaciones Jboss*. Obtenido de <http://asteriscus.com/presentacion/presentacion%20A71922.pdf>
- Eclipse. (2013). *Eclipse documentation - Current Release*. Obtenido de <http://help.eclipse.org/mars/index.jsp>
- IBM. (s.f.). *Iniciándose en la plataforma Eclipse*. Obtenido de <https://www.ibm.com/developerworks/ssa/library/os-ecov/>
- Jatun. (s.f.). *J2EE*. Obtenido de <http://www.jatun.com/web>
- Manuel, T. (s.f.). *GESTION DE PROYECTOS*. Obtenido de <http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf>
- Oracle. (2014). *Centro de ayuda de Java*. Obtenido de <https://www.java.com/es/download/help/>
- Oracle. (s.f.). *Información técnica acerca de java*. Obtenido de <https://www.java.com/es/download/faq/techinfo.xml>
- Pérez, M. (26 de 9 de 2011). Emprendimiento familiar. *Semanario Económico y negocios LÍDERES*, pág. 10.
- Postgresql. (5 de 8 de 2015). *Características, limitaciones y ventajas de PostgerdSQL*. Obtenido de <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>
- PostgreSQL. (s.f.). *Sobre PostgreSQL*. Obtenido de http://www.postgresql.org.es/sobre_postgresql
- Rodríguez, A. (2014). *CONCEPTO O DEFINICIÓN DE HERENCIA POO*. Obtenido de http://aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=592
- Schwaber Ken, S. J. (Julio de 2013). *La guía de Scrum*. Obtenido de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Scrum Manger. (2014). *Gestión de proyecto Scrum Manager*. Obtenido de http://www.scrummanager.net/files/sm_proyecto.pdf

X, A. (Febreo de 2015). *¿Proyectos complejos? ¿Necesitas resultados? Conoce Scrum*. Obtenido de www://proyectosagiles.org