

**UNIVERSIDAD DEL BÍO - BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE
INFORMACIÓN**



**“Sistema de gestión para alquiler y control del
equipamiento de montaña en bodega, para la
Pyme Dream’s Ski”**

Víctor Andrés Quintana Aedo

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INFORMÁTICO

Chillán, junio 2015

UNIVERSIDAD DEL BÍO - BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y TECNOLOGÍAS DE
INFORMACIÓN

**Sistema de gestión para alquiler y control
del equipamiento de montaña en bodega,
para la Pyme Dream's Ski**

Víctor Andrés Quintana Aedo

Profesor Guía : Miguel Romero Vázquez

Profesor Informante : Gilberto Gutiérrez Retamal

Nota Final :

Memoria para optar al título de Ingeniero Civil en Informática

Chillán, junio 2015

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática. El proyecto titulado “Sistema de gestión para alquiler y control del equipamiento de montaña en bodega, para la Pyme Dream’s Ski”.

El crecimiento de las tecnologías ha permitido resolver la necesidad de obtener información de manera eficaz para tomar distintas decisiones dentro de una organización. Sin embargo, esta realidad no se evidencia en todas las entidades o corporaciones debido a la falta de recursos, entre otras causas. He ahí la principal motivación por la cual se ha llevado a cabo este proyecto.

Este informe presenta la construcción de una aplicación web para la PyMe Dream’s Ski la cual le permitirá de mejor forma la gestión de su proceso de negocio. La metodología de desarrollo utilizada fue la iterativa incremental con el enfoque de orientación a objetos, utilizando la arquitectura MVC (Modelo Vista Controlador), UML para el modelado del problema y Java para su implementación. Entre las tecnologías que se incluyeron, destacan Struts 2 y JQuery, siendo esta última la responsable de manejar tanto información como aspectos gráficos en el sistema.

Entre los beneficios más destacados del sistema se cuenta la capacidad para obtener información rápida y accesible en todo momento y lugar por parte del personal, así como la ayuda a la toma de decisiones para la empresa.

El sistema desarrollado consta de una aplicación web que realiza la automatización de tareas específicas, como gestionar tanto a clientes y equipos como los alquileres que realiza la empresa para así poder controlarlos y a su vez obtener y almacenar información de manera eficaz y obtenerla a priori eficientemente. Además para la seguridad de la información contara con perfiles de usuario para facilitar el manejo a los usuarios y además proteger la información del personal que no la requiere.

El proyecto trajo como resultado un módulo que cumple las necesidades y requerimientos de los usuarios.

Contenido

introducción.....	12
capítulo 1	13
descripción general del proyecto	13
1.1 antecedentes generales de la empresa	14
1.2 organigrama institucional.....	15
1.3 descripción de la situación actual	15
1.4 descripción de la problemática	19
1.5 problemas detectados	19
capítulo 2.....	21
marco teórico	21
2.1 ambiente de ingeniería de software	22
2.2 enfoque de desarrollo de software: el modelo incremental.....	23
2.2.1 conceptos fundamentales de la programación orientado a objetos [1].	24
2.2.2 estructura de un objeto	26
2.3 arquitectura.....	27
2.4 lenguajes y notación de modelado	28
2.4.1 uml (unified modeling language)	28
2.4.2 bpmn (business process modeling notation)	32
2.5 tecnologías utilizadas	33
2.5.1 j2ee.....	33
2.5.2 mysql.....	34
2.5.3 struts 2.....	34
2.5.4 javaserver pages (jsp).....	36

2.5.5	javascript	36
2.5.6	jquery	37
2.5.7	hibernate.....	37
2.5.8	ajax.....	39
2.5.9	apache tomcat	39
2.5.10	ireport.....	40
2.5.11	jasper reports	40
2.6	herramientas	41
2.6.1	myeclipse.....	41
2.6.2	yed graph editor.....	41
2.6.3	ireport.....	42
2.6.4	mysql-front	42
2.7	definiciones, siglas y abreviaciones.....	43
capítulo 3 identificación de requerimientos		44
3.1	alcances	45
3.2	objetivo del software.....	46
3.2.1	objetivo general:	46
3.2.2	objetivos específicos:	46
3.3	descripción global del producto.....	47
3.3.1	interfaz de hardware.....	47
3.4	requerimientos específicos	47
3.4.1	breve descripción.....	47
3.4.2	requisitos no funcionales del sistema.....	48
3.5	requisitos funcionales del sistema	49

3.5.1	interfaces externas de entrada.....	51
3.5.2	interfaces externas de salida	52
3.5.3	atributos del producto	53
capítulo 4	55
	solución propuesta y análisis de factibilidad	55
4	solución propuesta	56
4.1	análisis de factibilidad.....	59
4.2	factibilidad técnica.....	59
4.3	factibilidad operacional.....	60
4.3.1	complejidad del nuevo sistema.....	61
4.3.2	resistencia al cambio	61
4.3.3	mantención y administración del sistema	62
4.4	factibilidad política	62
4.5	estudio económico	63
4.5.1	costo de desarrollo	63
4.5.2	costo de implementación.....	64
4.5.3	costo de operación	64
4.5.4	beneficios tangibles.....	66
4.5.5	cálculo de flujos de caja neto	69
4.5.6	cálculo del van.....	70
4.5.7	conclusión factibilidad	71
capítulo 5	análisis de requerimientos y diseño de la solución.....	72
5.1	determinación de requerimientos.....	73
5.2	usuarios	73

5.2.1	administrador:.....	73
5.2.2	bodeguero:.....	75
5.3	casos de uso	76
5.4	descripción de casos de uso, funcionalidades generales	77
5.5	diagrama de arquitectura del sistema	95
5.6	diagramas de secuencia del sistema	99
5.7	diagrama de colaboración del sistema	102
5.6	ilustración modelo entidad-relación.....	106
5.7	modelo de base de datos.....	107
5.8	requerimientos técnicos para el desarrollo de la aplicación.....	108
5.9	salidas.....	109
5.10	interfaz de la aplicación	113
capítulo 6	pruebas	122
6.1	pruebas de caja negra	123
6.1.1	prueba cu1, autenticar usuario	124
6.1.2	prueba cu 7, ingresar cliente.....	125
6.1.3	prueba cu 15, registrar alquiler	126
6.1.4	prueba cu 17, devolver	127
capítulo 7	conclusiones y referencias.....	128
7.1	128
	conclusiones	129
7.2	trabajos futuros	130
7.3	bibliografía	131

Índice de tablas

Tabla 1 - variedad de equipos de Ski	16
Tabla 2 - variedad equipos de snowboard	17
Tabla 3 - variedad equipos y accesorios en general.....	17
Tabla 4 - empleados fijos por temporada y como aumentan cuando existe sobredemanda.	17
Tabla 5 - tipos y cantidad de clientes	17
Tabla 6 - periodo de mayor demanda.....	18
Tabla 7 - Requisito navegador web	48
Tabla 8 - Requisito interfaz de software	48
Tabla 9 - Requisito tiempo de respuesta	49
Tabla 10 - Requisitos Funcionales.....	51
Tabla 11 - Interfaces Externas de Entrada	52
Tabla 12: Interfaces Externas de Salida	53
Tabla 13 - compatibilidad de portabilidad	53
Tabla 14 - costos desarrollador.....	65
Tabla 15 - costos de implementación.....	65
Tabla 16 - costos de operación	66
Tabla 17 - cantidad de alquileres realizados en periodo 2013.....	66
Tabla 18 - cantidad de alquileres realizados en periodo 2014.....	67
Tabla 19 - información general de logística de la empresa	67
Tabla 20 Flujo de Caja	70
Tabla 21: Actor administrador y sus privilegios del sistema	74
Tabla 22: Actor Bodeguero y sus privilegios del sistema.....	75
Tabla 27: Descripción Caso de Uso - Iniciar Sesión.....	77

Tabla 28: Descripción Caso de Uso - Cerrar Sesión.....	78
Tabla 29: Descripción Caso de Uso - Agregar Equipo.....	79
Tabla 30: Descripción Caso de Uso – Dar de baja Equipo.....	80
Tabla 31: Descripción Caso de Uso - Editar Equipo	81
Tabla 32: Descripción Caso de Uso - Listar Equipos.....	82
Tabla 33 - Descripción Caso de Uso - Agregar Cliente.....	83
Tabla 34 - Descripción Caso de Uso – Dar de baja Clientes	84
Tabla 35 - Descripción Caso de Uso - Editar Clientes	85
Tabla 36 - Descripción Caso de Uso - Listar Clientes.....	86
Tabla 37 - Descripción Caso de Uso - Agregar Usuario.....	87
Tabla 38 - Descripción Caso de Uso – Eliminar Usuarios.....	88
Tabla 39 - Descripción Caso de Uso - Editar Usuarios	89
Tabla 40 - Descripción Caso de Uso - Listar Usuarios.....	90
Tabla 41 - Descripción Caso de Uso - Agregar alquiler	91
Tabla 42 - Descripción Caso de Uso – Devolver	92
Tabla 43 - Descripción Caso de Uso – eliminar alquiler	93
Tabla 44 - Descripción Caso de Uso – Listar Alquiler	94
Tabla 23 - Autenticar Usuario.	124
Tabla 24 - Ingresar Cliente.....	125
Tabla 25 - registrar alquiler	126
Tabla 26 - Devolver	127

Índice de Ilustraciones

Ilustración 1- Organigrama Empresarial	15
Ilustración 2 - Modelo Iterativo-Incremental del proyecto	24
Ilustración 3- Estructura de un Objeto.	26
Ilustración 4 - Jerarquía de los diagramas UML 2.0.....	29
Ilustración 5 - MVC Strut.....	35
Ilustración 6 - Bpm solución	56
Ilustración 7 - Bpm solución detalle	57
Ilustración 8 - Bpm devolver equipo	58
Ilustración 9 - Diagrama Caso de uso	76
Ilustración 10 - Diagrama de Arquitectura del Sistema	96
Ilustración 11 - detalle de la lógica del negocio del diagrama de arquitectura	97
Ilustración 12 - detalle de la Persistencia del diagrama de arquitectura del sistema	98
Ilustración 13- diagrama de secuencia Buscar equipo	99
Ilustración 14 - diagrama de secuencia ingresar cliente.....	100
Ilustración 15 - diagrama de secuencia registrar Alquiler	100
Ilustración 16 - diagrama de secuencia logear	101
Ilustración 17 - Diagrama de Colaboración Buscar Equipo	103
Ilustración 18 - Diagrama de Colaboración Edita Equipo	103
Ilustración 19 - Diagrama de Colaboración listar Equipo.....	104
Ilustración 20 - Diagrama de Colaboración Elimina Equipo	104
Ilustración 21 - Diagrama de Colaboración Ingresa Equipo	105
Ilustración 22 - modelo entidad relación.....	106
Ilustración 23 - modelo base de datos.....	107

Ilustración 24 - reporte de equipos	109
Ilustración 25 - reporte clientes	110
Ilustración 26 - reporte alquileres realizados por un cliente	111
Ilustración 27 - salida de contrato.....	112
Ilustración 28 - diseño interfaz	113
Ilustración 29 - pantalla de inicio para ingresar	114
Ilustración 30 - pantalla de bienvenida para administrador	114
Ilustración 31 - pantalla de bienvenida para bodeguero	115
Ilustración 32 - pantalla listado de equipo	115
Ilustración 33 - pantalla ingresar equipo	116
Ilustración 34 - pantalla ingreso de equipo con código de barras.....	116
Ilustración 35 - pantalla listado de clientes	117
Ilustración 36 - pantalla agregar cliente	117
Ilustración 37 - pantalla listar alquileres	118
Ilustración 38 - pantalla generar alquiler, seleccionar el cliente	118
Ilustración 39 - pantalla generar alquiler, seleccionar el equipo	119
Ilustración 40 - pantalla generar alquiler, completar datos de alquiler.....	119
Ilustración 41 - pantalla usuarios bodegueros del sistema	120
Ilustración 42 - pantalla usuarios administradores del sistema	120
Ilustración 43 - pantalla lista de equipos de un cliente	121
Ilustración 44 - pantalla seleccionar reportes	121

INTRODUCCIÓN

Dream's Ski, es una micro-empresa emergente ubicada en el Km. 72 camino a las termas de chillan, las trancas. Se presenta como una nueva alternativa a las ya establecidas en el rubro en esta localidad, presentando como premisa la calidad del servicio y atención.

Hoy en día los informes, reportes y documentos en general son todos llevados en papel, escritos a mano, lo que en temporada alta le significa tiempo al tener que llenarlos y dinero al tener que comprar muchos insumos.

Ningún proceso, se encuentra automatizado, si bien en la empresa se cuenta con computadores y personal capacitado en un nivel suficiente para el uso de tecnologías, la realidad de la empresa en cuanto al uso de estas es completamente diferente, debido a que solo organizan sus grandes cantidades de documentos en carpetas digitales y también otros en archivadores que ocupan espacio físico dentro de la empresa, los cuales contienen, dentro de otras cosas, datos históricos de la institución.

Teniendo en cuenta que la realidad organizacional podría ser mejorada considerablemente, el objetivo es dar solución a aquellas tareas de mayor frecuencia y que soportan una gran cantidad de datos a ser controlados, de esta manera agilizar los procesos dentro de la empresa. Es por ello que se ha considerado un diseño modular y en base a una metodología iterativa-incremental para implementar la automatización de las tareas del control y gestión de los alquileres realizados por la empresa, los equipos y los clientes necesarias para el arriendo, pensando en que a futuro este sistema pueda continuar creciendo hasta abarcar el número de tareas automatizadas que la empresa estime pertinente, y lograr así una mayor efectividad en sus procesos.

A continuación en los siguientes capítulos de esta memoria se dará a conocer la historia, la problemática y la solución propuesta y desarrollada para la empresa Dream's Ski.



UNIVERSIDAD DEL BÍO-BÍO

CAPÍTULO 1

DESCRIPCIÓN GENERAL

DEL PROYECTO

1.1 Antecedentes generales de la Empresa

- Nombre : Dream's Ski.
- RUT : 15876876-3
- Dirección : las trancas Km 72 S/n.
- Teléfono : +56 9 8579 5547
- Rubro : deportes Extremos.
- Servicios : préstamos y arriendos de equipamiento deportivo.

Misión

Somos una empresa de prestación de servicios en el área de la entretención y deportes de montaña con fuertes cimientos en confiabilidad, calidad de servicio tanto en la calidad de nuestros productos y nuestra gente, orientada a satisfacer las necesidades y aspiraciones de nuestros clientes, estableciendo relaciones de largo plazo. Siendo así un aporte positivo para el turismo local y a su vez aportando al cuidado de la naturaleza y medio ambiente de nuestra zona.

Visión

Ser líder en la prestación de servicios en el sector del entrenamiento deportivo, satisfaciendo las necesidades de nuestros clientes. Comprometiéndonos con la calidad del servicio prestado y obligando valores de resguardo de la flora y fauna local.

1.2 Organigrama Institucional

En la ilustración 1 se presenta el organigrama empresarial el cual se formuló en conjunto con los integrantes de la empresa ya que no existía y el cual nos muestra cómo están ordenados jerárquicamente los cargos más importantes dentro de Dream's Ski.

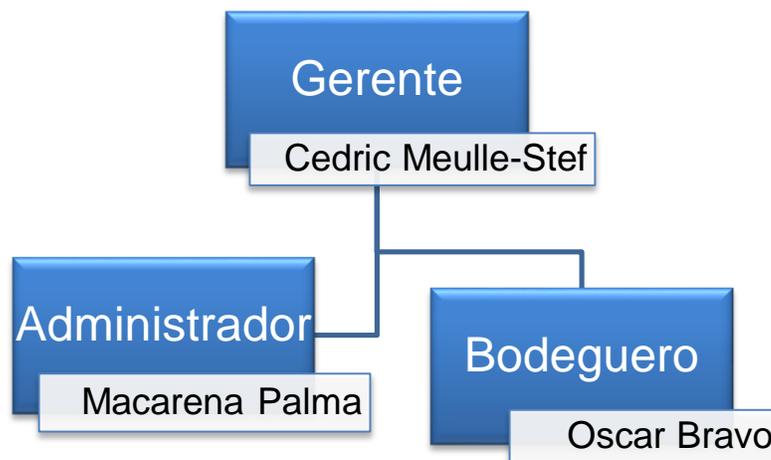


Ilustración 1- Organigrama Empresarial

1.3 Descripción de la Situación Actual

La administración del centro de alquiler de equipamiento deportivo Dream's Ski, toma las decisiones de los alquileres a realizar según el cliente que lo solicite (P.E. se reserva derechos de admisión, según información histórica de clientes y los infractores de alguna cláusula del contrato) y el stock en bodega que quede disponible para el alquiler, con estos datos se logra maximizar los tiempos de

operación de los encargados, repercutiendo en una mayor fluidez del servicio prestado.

El enfoque de este proyecto tiene su base en el área operacional de la empresa, ya que lo que se busca como objetivo es optimizar tanto el tiempo de servicio y controlar los arriendos, velando por el cumplimiento del contrato pactado. Además se enfoca en controlar de mejor manera el stock del equipamiento en bodega y también las entradas y salidas de estos cuando son alquilados.

En esta área podemos identificar y dar roles. Primero está el bodeguero, quien se encarga de registrar todos los movimientos del equipamiento, luego está el administrador del local y el gerente quien por poseer más responsabilidad en la empresa, se le otorgara un rol de administrador del sistema, quien tendrá acceso a todo lo anterior, además de ver la documentación de la empresa para la toma de decisiones, debido a esto el sistema le entregará los beneficios de los reportes para la ayuda en sus labores y así tener una visión más amplia del tipo de servicio que se está prestando y las estadísticas y estimaciones que la empresa provee.

A continuación en las siguientes tablas se detalla en general las estadísticas en relación a los clientes equipos, empleados de la empresa.

TIPO	VALOR ALQUILER	STOCK
Equipo Completo Ski Standard	\$ 21.000	14
Equipo Completo Ski Hi - Performance	\$ 24.000	11
Sólo Ski Standard	\$ 16.000	21
Sólo Ski Hi - Performance	\$ 19.000	14
Sólo Botas Ski	\$ 8.000	23
Sólo Bastones Ski	\$ 6.000	50
Equipo Completo Ski Niño	\$ 14.000	8
Sólo Ski Niño	\$ 9.000	17
Sólo Botas Ski Niño	\$ 7.000	19

Tabla 1 - variedad de equipos de Ski

TIPO	VALOR ALQUILER	STOCK
Equipo Completo Snowboard	\$ 21.000	17
Sólo Snowboard	\$ 16.000	23
Sólo Botas Snowboard	\$ 8.000	47
Equipo Completo Snowboard Niño	\$ 14.000	22
Sólo Snowboard Niño	\$ 9.000	16
Botas Snowboard Niño	\$ 7.000	21

Tabla 2 - variedad equipos de snowboard

TIPO	VALOR ALQUILER	STOCK
Traje de Nieve	\$ 14.000	19
Pantalones	\$ 7.000	34
Parka (Chaqueta)	\$ 7.000	31
Traje de Nieve Niño	\$ 14.000	15
Pantalones Niño	\$ 6.000	17
Parka Niño (Chaqueta)	\$ 6.000	19
Guantes	\$ 5.000	49
Antiparras	\$ 5.000	38
Casco	\$ 5.000	62

Tabla 3 - variedad equipos y accesorios en general

empleados	cantidad	sobre demanda
administrador	1	2
bodegero	2	3

Tabla 4 - empleados fijos por temporada normal y cuando existe sobredemanda

cliente	porcentaje promedio	cantidad promedio
hombre	47%	184
mujer	32%	125
niño(a)	21%	82

Tabla 5 - tipos y cantidad de clientes

En relación a la tabla 5, podemos observar que la cantidad y variedad de los clientes que realizan alquileres son los que muestran en la columna de la cantidad de clientes promedio, al realizar una operación con la tabla 17 del capítulo 5 la cual entrega un promedio de 390 clientes por la temporada 2014.

A continuación se presenta la tabla 6, la cual entrega en porcentaje la cantidad de alquileres de cada mes en relación a la cantidad total de alquileres realizados que se muestra en la tabla 18 del apartado 4.5.4, lo que denota los periodos de mayor demanda en la empresa.

Periodo 2014	porcentaje cantidad alquileres
junio	18%
julio	23%
agosto	30%
septiembre	29%

Tabla 6 - periodos de mayor demanda

Podemos observar en la tabla 6 los periodo de mayor demanda, los cuales son a fines de junio y la quincena de agosto debido a la cantidad de nieve registrada que cae en las canchas de ski donde Dream's Ski provee a sus clientes de los equipos necesarios. Cabe destacar que estos periodos son derechamente los fines de semana y festivos.

1.4 Descripción de la problemática

La problemática que presenta la empresa es que al momento de realizar el alquiler de equipos, resulta en momentos complejo organizarse por la aglomeración que se produce en el local en las horas punta. También se produce confusiones con los documentos que pide la empresa para realizar el alquiler como son los diferentes contratos, y comprobantes que hay sobre el mostrador, generados y administrados por los encargados del local.

Un problema adicional y similar se produce en la hora de devolución de los equipos cuando todos los clientes los llegan a devolver al mismo tiempo, donde se debe verificar, en los registros manuales que todo lo que se devuelve sea lo real y no falte ningún equipo. Este proceso es igual de engorroso que el antes mencionado, cuando hay mucha afluencia de clientes.

Dentro de los documentos mencionados se encuentra el contrato de arriendo el cual lleva un código numérico y la lista de equipos arrendados.

1.5 Problemas detectados

Uno de los problemas principales que actualmente presenta la empresa Dream's Ski, es el control de cada alquiler que se realiza, la administración de la información del equipamiento que se necesita arrendar, ya que no cuenta con ningún sistema que ayude a agilizar y asegurar sus operaciones, teniendo así un sistema de trabajo lento e inseguro en base a una gran cantidad de archivadores con una gran cantidad de datos.

De esta forma, luego de realizar un estudio de la situación actual de la empresa, se tienen en cuenta las diferentes carencias que posee en cuanto a sistemas de información, los cuales se listan a continuación.

1. Escaso control de cada uno de los alquileres que se están realizando por la empresa, teniendo en cuenta que cada alquiler lleva asociadas boletas emitidas, personal asociado a las labores de la empresa. Y el equipamiento arrendado.
2. Lentitud en la recuperación de la información en Dream's Ski, para saber que Equipamiento se está utilizando, o se necesitará en el corto plazo, para llevar a cabo un arriendo.
3. Dificultad para acceder a los datos relevantes para la toma de decisiones de la empresa como lo son, el equipamiento en uso y el disponible, los alquileres realizados y los clientes de la empresa.
4. Dificultad para organizar y acceder a la información de cada uno de los equipamientos en bodega.

La actividad general de la empresa es tratar de dar al cliente lo que ellos requieren y ojala mantener un stock adecuado de equipamiento disponible en la temporada. Así como también mantener un orden y control de los equipos disponibles.

CAPÍTULO 2

MARCO TEÓRICO

2.1 AMBIENTE DE INGENIERÍA DE SOFTWARE

Para dar inicio al proyecto, este se despliega en base a la metodología incremental de desarrollo de software, en conjunto con el paradigma de programación orientada a objetos. En cuanto al modelado de los datos, se utiliza el lenguaje BPMN para modelar el negocio y el lenguaje UML para llevar a cabo el modelado del sistema. Se utiliza el modelo iterativo–incremental para lograr obtener productos tangibles en cada iteración y así poder ir mostrando además el grado de avance desarrollado al cliente.

El diseño de la aplicación está basado en la arquitectura llamada Modelo Vista Controlador, la cual se divide en tres áreas funcionales:

- Vista (Interfaz)
- Modelo (Lógica del Negocio)
- Controlador (Lógica de la aplicación)

Considerando el hecho que la base de datos es relacional, se utiliza para el modelado de datos “Modelo Entidad Relación” para finalmente, pasar a un modelo lógico basado en tuplas.

Para finalizar cabe mencionar que se trabaja con tecnología JAVA, con el motor de base de datos MYSQL, y para la programación se utiliza el entorno de desarrollo MyEclipse.

2.2 ENFOQUE DE DESARROLLO DE SOFTWARE: EL MODELO INCREMENTAL

El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software. Se eligió esta metodología ya que aquí se entrega el software en partes pequeñas, pero utilizables, llamadas (incrementos), las cuales van siendo entregadas al cliente, y este va acotando sus observaciones inmediatamente. En general, cada incremento se construye sobre aquél que ya ha sido entregado [25].

El primer incremento a menudo es un producto esencial. Es decir se afrontan requisitos básicos, pero muchas funciones suplementarias quedan sin extraer. El cliente utiliza el producto central como un resultado de utilización y de evaluación donde se desarrolla un plan para el incremento siguiente. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones y características adicionales. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo [25].

Los primeros incrementos son versiones «incompletas» del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación. Con cada incremento se agrega nueva funcionalidad, se cubren nuevos requisitos o bien se mejora la versión previamente implementada del producto software [25]. Las ventajas que se tienen con respecto a otros modelos:

- Se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.
- Provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.

- Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.
- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

En la ilustración 3 se presenta el modelo incremental, mostrando los diferentes incrementos y detallando sus etapas.

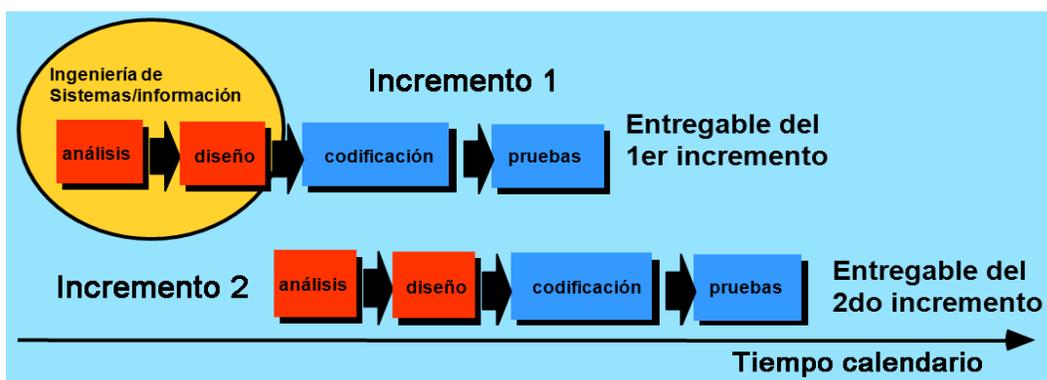


Ilustración 2 - Modelo Iterativo-Incremental del proyecto
(Larman, 2003)

2.2.1 CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN ORIENTADO A OBJETOS [1].

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a los problemas que ocasionaba la programación estructurada. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas [29].

Herencia: (por ejemplo, herencia de la clase C a la clase D) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos

atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de OOP [29].

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase [29].

Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema [29].

Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera [29].

Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó [29].

Propiedad o atributo: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método [29].

Estado interno: es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase [29].

Componentes de un objeto: atributos, identidad, relaciones y métodos [29].

Identificación de un objeto: un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes. [29].

2.2.2 ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes: relaciones, atributos y métodos [29], tal como se presenta en la ilustración 3.

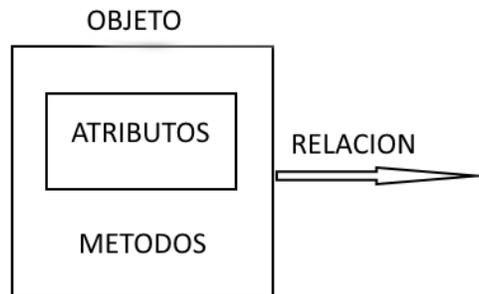


Ilustración 3- Estructura de un Objeto.

2.3 ARQUITECTURA

2.3.1 MVC (MODELO VISTA CONTROLADOR)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Este patrón, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista [20].. Describiendo el patrón, tenemos:

Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado [20].

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario [20].

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista [20].

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la programación por capas representa la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero sí pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, es por esto que hemos escogido este patrón ya que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí [20].

2.3.2 PATRONES DE DISEÑO

Los patrones de diseño “son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan” [14].

A continuación se detallan los patrones de diseño a utilizar en el desarrollo de este proyecto.

2.3.2.1 Data Access Object

Data Access Object (DAO) es un patrón de diseño que permite abstraer y encapsular todos los accesos a una base de datos en un objeto especial que utiliza las conexiones para ejecutar las consultas necesarias para obtener y almacenar datos.

Un DAO implementa los mecanismos de acceso requeridos para trabajar con la base de datos, El DAO debe esconder completamente los detalles de la implementación a sus clientes. Así, cualquier cambio en la base de datos no requiere cambio en los clientes si no sólo en el DAO.

Un objeto DAO sólo debe contener operaciones de lectura y modificación de datos. No debe contener operaciones con lógica de negocio. [8].

2.4 LENGUAJES Y NOTACIÓN DE MODELADO

2.4.1 UML (UNIFIED MODELING LANGUAGE)

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

estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

“UML ofrece un estándar para describir el modelo del sistema, incluyendo aspectos conceptuales, tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.” (LARMAN, 2003)

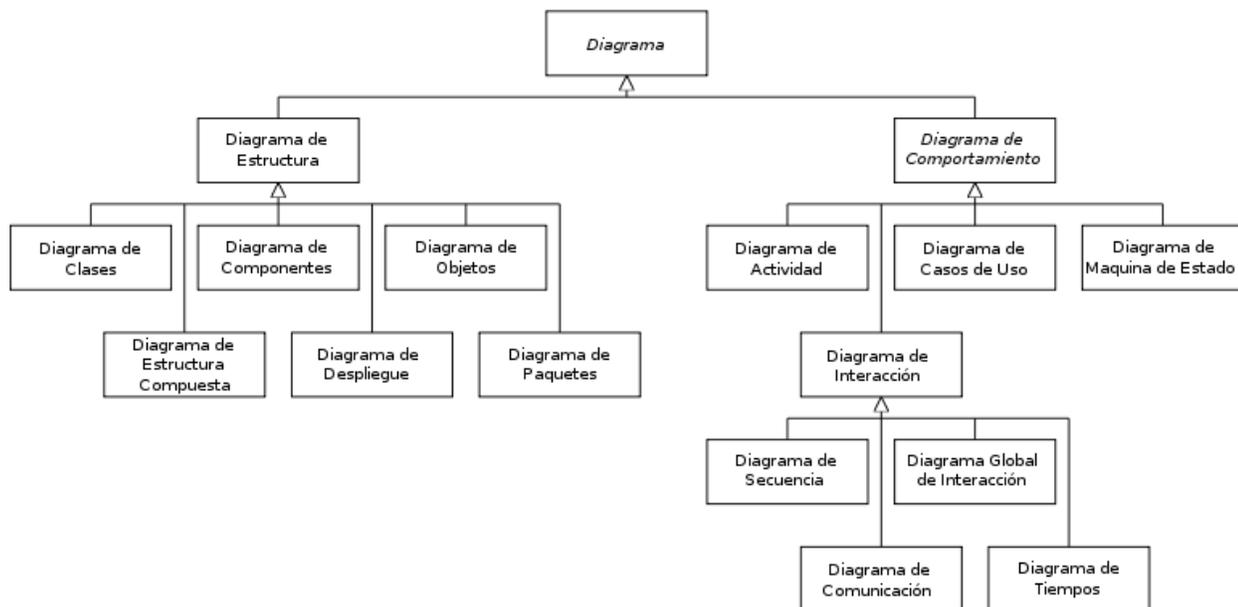


Ilustración 4 - Jerarquía de los diagramas UML 2.0

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la Ilustración 4.

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

Diagrama de clases: Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro [24].

Diagrama de componentes: Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema [24].

Diagrama de objetos: Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase. Una diferencia con los diagramas de clase es que el compartimiento de arriba va en la forma Nombre de objeto: Nombre de clase. Por ejemplo, Miguel: Persona [24].

Diagrama de estructura compuesta (UML 2.0): muestra la estructura interna de una clase y las colaboraciones que esta estructura hace posibles. Esto puede incluir partes internas, puertas mediante las cuales, las partes interactúan con cada una de las otras o mediante las cuales, instancias de la clase interactúan con las partes y con el mundo exterior, y conectores entre partes o puertas. Una estructura compuesta es un conjunto de elementos interconectados que colaboran en tiempo de ejecución para lograr algún propósito. Cada elemento tiene algún rol definido en la colaboración [24].

Diagrama de despliegue: se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes [24].

Diagrama de paquetes: muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema [24].

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

Diagrama de actividades: Representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general [24].

Diagrama de casos de uso: UML define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso [24].

Diagrama de estados: es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

Diagrama de secuencia: Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de

secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos [24].

Diagrama de comunicación: es una versión simplificada del diagrama de colaboración de la versión de UML 1.x.

Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

Diagrama de tiempos (UML 2.0): es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás [24].

Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0): Los modelos de interacción pueden llegar a ser muy grandes para sistemas complejos. Si el número de líneas de vida participantes y el número de mensajes intercambiados exceden una cierta medida, se impone “modularizar” las interacciones y dividir en partes pequeñas, más manejables, de acuerdo a principios universales del diseño de sistemas, que también pueden ser visualizadas con la ayuda de un clásico diagrama de secuencias. La visión de conjunto de toda la interacción, de manera que la Big Picture o bien el cuadro global, puede entonces ser representada con la ayuda del diagrama global de las interacciones, provisto para eso [24].

2.4.2 BPMN (BUSINESS PROCESS MODELING NOTATION)

En español (**Notación para el Modelado de Procesos de Negocio**) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow). BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG

(Object Management Group), después de la fusión de las dos organizaciones en el año 2005. Su versión actual, a abril de 2011, es la 2.0.

El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos) [24].

El por qué elegimos esta notación es porque BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Para una persona que no es experta en programación o informática se le hace mucho más fácil comprender uno de estos diagramas en vez de mostrarle un diagrama de clases por ejemplo.

2.5 TECNOLOGÍAS UTILIZADAS

2.5.1 J2EE

J2EE son las siglas de Java 2 Enterprise Edition que es la edición empresarial del paquete Java creada y distribuida por Oracle Corporation. Comprenden un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.

Es una plataforma que habilita soluciones para desarrollo, uso efectivo y manejo de multicapas en aplicaciones centralizadas en el servidor.

J2EE utiliza la plataforma Java 2 SE, para tender una completa, estable, segura, y rápida plataforma Java en el ámbito de la empresa. Habilita una plataforma que reduce de manera significativa los costos y la complejidad de desarrollo de soluciones multicapas, resultando en servicios que pueden ser desarrollados rápidamente y ampliados fácilmente. [10].

2.5.2 MYSQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo: C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux) y FreeBASIC.

Cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP. [11].

2.5.3 STRUTS 2

Struts 2 es la nueva versión del popular framework de desarrollo web en Java Apache Struts. Sin embargo, el código de ambos tiene poco que ver, dado que Struts 2 no se basa en el código de Struts 1.x, sino en el de otro framework de desarrollo web en Java llamado WebWork, un framework que el creador de Struts consideraba superior a Struts 1.x en varios aspectos.

Struts 2 está basado en el patrón MVC (Modelo-Vista-Controlador), una arquitectura que busca reducir el acoplamiento dividiendo las responsabilidades en 3 capas claramente diferenciadas:

El modelo, que hace referencia a los datos que maneja la aplicación y las reglas de negocio que operan sobre ellos y que se traducen en Struts 2 en las acciones.

La vista, encargada de generar la interfaz con la que la aplicación interacciona con el usuario. En Struts 2 equivale a los resultados.

El controlador, que comunica la vista y el modelo respondiendo a eventos generados por el usuario en la vista, invocando cambios en el modelo, y devolviendo a la vista la

información del modelo necesaria para que pueda generar la respuesta adecuada para el usuario. El controlador se implementa en Struts 2 mediante el filtro FilterDispatcher.

Esto se resume en la siguiente figura:

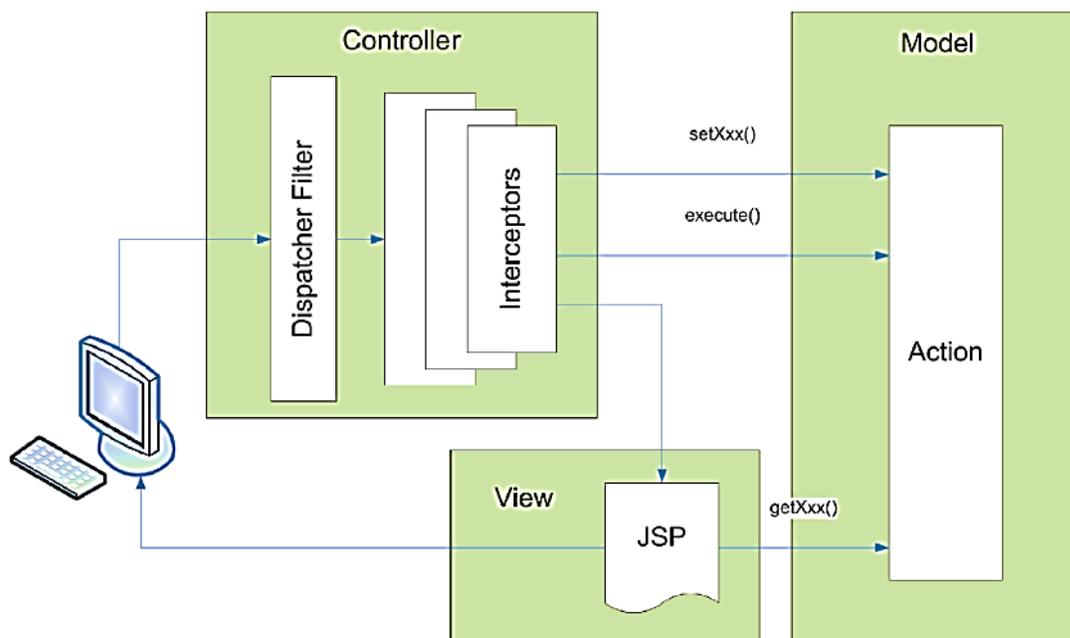


Ilustración 5 - MVC Struts

La ventaja de utilizar el framework Struts 2 es que éste se puede encargar de obtener y transformar los parámetros de las peticiones del cliente, validar los datos, buscar qué acción se debe realizar en el modelo según la petición, buscar la página a mostrar al usuario según la respuesta del modelo, proporcionar los datos del modelo a la vista, ocuparse de la internacionalización, etc.

Cabe mencionar, que un componente fundamental de Struts 2 que es utilizado en este proyecto son los interceptors, los cuales son responsables de la mayoría de los procesos hechos por el framework tales como: validación de datos, transferencia de datos manejo de excepciones. Los interceptors mejoran inmensamente la separación de lo conceptual de la aplicación, removiendo tareas que son transversales a toda la aplicación como las mencionadas anteriormente. Por ejemplo, podemos configurar a un interceptor para que

compruebe si el usuario está autenticado y si no lo está, redireccionar a la página de Login. Esto es posible debido a que los interceptors se ejecutan antes de la acción en curso. [5].

2.5.4 JavaServer Pages (JSP)

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. [3].

2.5.5 JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. [2].

2.5.6 JQUERY

JQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. [6].

2.5.7 HIBERNATE

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones y un gran número de tipos de datos. De una manera muy rápida y optimizada se pueden generar BD en cualquiera de los entornos soportados: Oracle, DB2, MySQL, Postgres, etc. y además, es open source, lo que supone, entre otras cosas, que no se debe pagar nada por adquirirlo. [29].

Uno de los posibles procesos de desarrollo consiste en, una vez que se tiene el diseño de datos realizado, es proceder a mapear las tablas del diseño de datos a archivos de extensión XML siguiendo la DTD de mapeo que posee Hibernate, desde éstos podremos generar el código de nuestros objetos persistentes en clases Java y también crear base de datos (BD) independientemente del entorno escogido.

Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Una posible configuración básica de Hibernate se muestra en la figura.

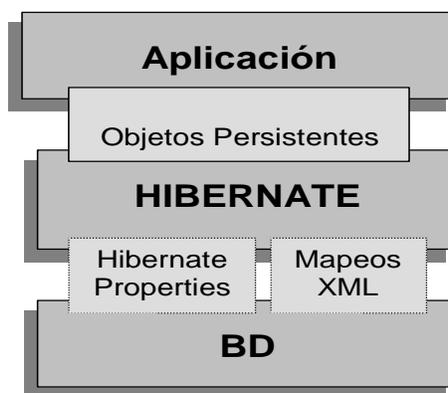


Figura 1: Configuración básica de Hibernate

Hibernate utiliza la base de datos y la configuración de los datos para proporcionar servicios y objetos persistentes a la aplicación que se encuentre justo por arriba de él.

Algunos beneficios de Hibernate que se pueden mencionar son:

Productividad: El código relacionado con la persistencia resulta tedioso para las aplicaciones de Java. Hibernate elimina la mayor parte de éste y permite concentrarse principalmente en el problema, sin importar la estrategia de programación que se este utilizando top-down (empezando del modelo global del sistema) o bottom-up (empezando con un esquema de base de datos existente).

Capacidad de mantenimiento: Menor número de LOC (Líneas de código) hacen un sistema más comprensible debido a que se da énfasis en la lógica del programa, haciendo más fácil su mantenimiento. Automatizando la persistencia de los objetos se reduce sustancialmente las LOC y de esta manera se reduce la complejidad de la aplicación independientemente de aislar el modelo relacional del orientado a objetos facilitando el empleo más elegante a la programación en Java.

Rendimiento (Performance): La persistencia implementada a mano puede ser tan rápida y realizar las mismas tareas como la automatizada, pero existen grandes diferencias entre éstas cuando se consideran algunas restricciones como tiempo y dinero. Cuando se realiza una tarea y es necesario manejar datos persistentes es posible realizar varias optimizaciones, a pesar de ser fácil su implementación con código escrito a mano (por

ejemplo usando SQL/JDBC), la mayoría de las veces es más fácil de realizarlas con un ORM (Object/Relational Mapping). Cuando en un proyecto se está restringido por el tiempo, la persistencia codificada a mano solamente permite realizar algunas optimizaciones y únicamente bajo ciertas condiciones; Hibernate nos provee de muchas más optimizaciones que son usadas todo el tiempo. Además, la persistencia automatizada aumenta la productividad del desarrollador de manera que se cuenta con más tiempo para optimizar unos cuantos problemas restantes.

- Independencia de proveedor: Una gran ventaja de los ORM es que cuentan con herramientas de soporte para la mayoría de los manejadores de base de datos y esto añade un alto nivel de portabilidad a la aplicación desarrollada, independientemente de facilitar su desarrollo.

2.5.8 AJAX

“Asynchronous JavaScript And XML” (JavaScript asíncrono y XML). Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. [7].

2.5.9 APACHE TOMCAT

Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Oracle Corporation. [28].

2.5.10 IREPORT

iReport es un diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc.. iReport está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos, incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, entre otros. [16].

2.5.11 JASPER REPORTS

JasperReports es una herramienta de creación de informes Java que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a archivos PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. Ésta herramienta es utilizada comúnmente con iReport. [17].

2.6 HERRAMIENTAS

2.6.1 MYECLIPSE

MyEclipse es un entorno de desarrollo que está disponible en el mercado de Java EE y AJAX IDE, creado y mantenido por la empresa Genuitec, miembro fundador de la Fundación Eclipse.

MyEclipse se basa en la plataforma Eclipse e integra soluciones de código abierto y propietario en el entorno de desarrollo.

MyEclipse tiene dos versiones principales (aparte de las "Blue Edition y "MyEclipse Edición Primavera"): una profesional y una edición estándar. La edición estándar incluye herramientas de base de datos, un diseñador de páginas web visuales, herramientas de persistencia, herramientas de Struts y JSF, y un número de otras características para el perfil básico de Java Developer Eclipse. Compite con el Web Tools Project (Proyecto de Herramientas Web), que es una parte de Eclipse en sí, pero MyEclipse es un proyecto independiente por completo y ofrece un conjunto de características diferentes.

Más recientemente, MyEclipse ha puesto a disposición una herramienta de aprovisionamiento que mantiene los perfiles de software Eclipse, incluyendo los que utilizan MyEclipse. Además, MyEclipse está ofreciendo una versión personalizada de los productos de IBM, "MyEclipse Blue Edition", que añade soporte específico para Rational Software y el desarrollo de WebSphere. En la actualidad, MyEclipse Blue Edition está disponible para Windows y Linux, aunque para Mac no es compatible. [21].

2.6.2 YED GRAPH EDITOR

Editor gráfico de una gran potencia, que entrega la posibilidad de crear de manera rápida esquemas o mapas conceptuales.

2.6.3 IREPORT

iReport es un diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos, incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, entre otros. [16].

2.6.4 MYSQL-FRONT

Mysql-front es un “front-end” de software para el servidor de base de datos MySQL. La estructura de base de datos y los datos pueden ser manejados a través de cuadros de diálogo y comandos de SQL. Importar o exportar en varios formatos es compatible. Además es posible el acceso al servidor MySQL directamente a través de HTTP tunneling. [22].

2.7 DEFINICIONES, SIGLAS Y ABREVIACIONES

Este ítem se incluye la definición de las siglas, abreviaciones, conceptos técnicos o de negocio que son necesarios para el buen entendimiento de este documento.

Alquiler o renta: Hace referencia al rubro de la empresa, y no es nada más que la acción de prestarle al cliente el equipamiento deportivo de montaña a cambio de una suma de dinero correspondiente en un plazo determinado y mediante el cual se firma un contrato de consentimiento y aceptación de las condiciones que además enmarca las responsabilidades que posee cada parte, el cliente y la empresa.

Hosting: El alojamiento web (del inglés web Hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web. Es una analogía de "hospedaje o alojamiento en habitaciones" donde uno ocupa un lugar específico, en este caso la analogía alojamiento web o alojamiento de páginas web, se refiere al lugar que ocupa una página web, sitio web, sistema, correo electrónico, archivos etc. en internet o más específicamente en un servidor que por lo general hospeda varias aplicaciones o páginas web.



CAPÍTULO 3

IDENTIFICACIÓN DE REQUERIMIENTOS

3.1 Alcances

El sistema busca controlar y gestionar las operaciones que realiza la empresa Dream's Ski. Entre las operaciones que se gestionan, se contempla el control del equipamiento, alquileres, tanto así como la información de clientes que se relacionan con la empresa. De esta forma el sistema permite un acceso rápido, confiable y sencillo a los datos a través de una interfaz intuitiva, realizando gran parte de las tareas que actualmente se desarrollan en la empresa, de manera automatizada.

Entre las operaciones que el sistema permite realizar se encuentran:

- Inserción
- Dar de baja
- Edición y actualización
- Búsqueda
- Generación de reportes

Para cada una de las entidades del sistema, relacionándose así con los equipamientos de la empresa, los clientes y los alquileres que se realizan, entregando así toda la información relevante de una forma más rápida y de fácil acceso para el personal de la organización.

Entre las limitaciones que posee el software están los siguientes:

- No tendrá un módulo correspondiente a las finanzas y contabilidad de la empresa
- La aplicación web solo se centrará en el control y gestión de equipamiento en bodega y los que han sido alquilados, además de los alquileres realizados a clientes y la información relacionada con los clientes.

3.2 Objetivo del software

3.2.1 Objetivo General:

Desarrollar una aplicación web para apoyar los procesos del negocio relacionados con el alquiler de equipamiento y los clientes para así lograr mantener la información correcta, actualizada y segura con un acceso eficiente.

3.2.2 Objetivos Específicos:

- Implementar un módulo de gestión y control de los equipamientos, para así ser accedido desde cualquier lugar y en todo momento.
- Implementar un módulo de gestión y control de los clientes, para así ser accedido desde cualquier lugar y en todo momento.
- Implementar un módulo de gestión y control de los alquileres de equipos realizados a los clientes, para así ser accedido desde cualquier lugar y en todo momento.
- Controlar de forma eficiente y segura los datos de las transacciones con la implementación del protocolo AAA (del inglés, Authentication, Authorization y Accounting), para así mantener la información confidencial.
- Realizar las operaciones de ingreso, modificación, eliminación y consultas con respecto a los equipos, alquileres y clientes, para así tener la información correcta y actualizada de cada uno de estos.
- Implementar un módulo de reportes, el cual entregue información de manera rápida y óptima, de los equipos, clientes y alquileres realizados.

3.3 Descripción Global del Producto

3.3.1 Interfaz De Hardware

La aplicación web puede interactuar con un lector láser de códigos de barra, el cual ayuda a la reducción de entradas por teclado por parte del usuario, y además elimina el tiempo dedicado a la inserción manual del código, aumentando así la efectividad del sistema. Debido a esto se configura con opción a esta característica en cuanto al control de los Equipos de la empresa. A continuación se detalla el posible periférico de hardware acordado:

- Lector Láser de Código de barras, USB:

Lector Láser de código de barras, de mano y con conexión mediante el puerto USB, con diseño ergonómico, que facilite la lectura de códigos de barras lineales con precisión, agilizando el flujo de Equipos de la empresa. Al conectarse mediante USB no necesita configuración, debido a que el reconocimiento es automático por parte del sistema operativo, cumpliendo la función del teclado en las entradas de los códigos de barras, agilizando sus flujos y evitando errores.

El lector debe contemplar al menos las codificaciones más utilizadas como lo son: EAN-8 y EAN-13.

3.4 Requerimientos Específicos

3.4.1 Breve descripción

“El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo”. (Larman, 2003)

3.4.2 Requisitos No Funcionales del Sistema

Los Requisitos no funcionales del sistema nos indican aspectos visibles pero que no están relacionados directamente con el comportamiento funcional de éste. En la tabla 1, 2 y 3 se detallan los requisitos no funcionales del sistema.

Ref.# RNFA01	Prioridad: Alta
Requisito: Navegador web.	
Descripción: Permitir la compatibilidad del software con los navegadores más utilizados por parte de la empresa: Mozilla Firefox y Google Chrome.	
Necesidad: Esencial.	
Verificación: Comprobar las funcionalidades del sistema en los dos navegadores mencionados anteriormente.	

Tabla 7 - Requisito navegador web

Ref.# RNFA02	Prioridad: Alta
Requisito: Interfaz de Software.	
Descripción: Se consideran 2 interfaces, una para cada perfil del sistema. El posicionamiento de los elementos en la pantalla es el mismo para todas, sin embargo el contenido cambia de acuerdo a cada perfil.	
Necesidad: Esencial.	
Verificación: Se verifica que al ingresar con cada perfil, se genere una vista distinta.	

Tabla 8 - Requisito interfaz de software

Ref.# RNFB01	Prioridad: Alta
Requisito: Tiempo de respuesta	
Descripción: El sistema no debe tardar más de 30 segundos en responder a cualquier consulta o evento que realice el usuario.	
Necesidad: Esencial.	
Verificación: Se verifica cronometrando el tiempo de respuesta a varios eventos y consultas.	

Tabla 9 - Requisito tiempo de respuesta

3.5 Requisitos Funcionales del Sistema

Los Requisitos funcionales nos indican que debe hacer específicamente el sistema, indicando qué y cómo hacerlas. Estos Requisitos se obtienen directamente del usuario y sus necesidades

En cuanto a los tipos de categorías (Larman, 2003), estas se clasifican en:

Evidente: Debe realizarse, y el usuario debería saber que se ha realizado.

Ocultas: Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a muchos servicios técnicos subyacentes, como guardar información en un mecanismo persistente de almacenamiento. Las funciones ocultas a menudo se omiten (erróneamente) durante el proceso de obtención de los Requisitos.

- **Superfluas:** Opcionales; su inclusión no repercute significativamente en el costo ni en otras funciones.

A continuación se desarrollan los Requisitos funcionales durante la etapa de captura de requisitos. En esta etapa se desarrollaron dos entrevistas, la primera para recopilación de información y la segunda para validar los requisitos obtenidos con el usuario.

La siguiente tabla muestra la lista de los requisitos validados por el usuario.

Ref.#	Función	Categoría
RF01	-El sistema debe tener sesiones para los diferentes usuarios (administrador, bodeguero), dándole distintas vistas y privilegios a cada uno, siempre y cuando la persona esté registrada en la base de datos.	Oculta
RF02	-El sistema debe llevar el registro de los distintos equipamientos que han sido registrados y estén almacenados. Indicando las fechas y las cantidades.	Oculta
RF03	-El sistema permite agregar, modificar, dar de baja y listar toda la información sobre los equipamientos, siempre y cuando esta acción sea realizada por un usuario que tenga los permisos respectivos.	Evidente
RF04	-El sistema permite agregar, modificar, dar de baja y listar a todos los clientes que presenta la empresa, siempre y cuando esta acción sea realizada por un usuario que tenga los permisos respectivos.	Evidente
RF05	-El sistema debe llevar el control de todos los alquileres que se realicen, tomando los tiempos de utilización de los equipamientos en cada uno de ellos. Será registrado mediante fechas, en las cuales se mantendrá ocupado a cada ítem respectivamente, indicando así fecha de inicio de uso y final de uso de cada equipamiento. Además, de poder contar con las funcionalidades de agregar, modificar y listar alquileres.	Evidente
RF06	-El sistema permite agregar, modificar, eliminar, y listar usuarios, siempre y cuando esta acción sea realizada por un usuario que tenga los permisos respectivos (administrador).	Evidente
RF07	-El sistema debe exportar el reporte de equipamiento solicitado por el usuario respectivo (administrador maestro), en un formato PDF, teniendo como atributos: nombre, código, cantidad, categoría, descripción.	Evidente

RF08	-El sistema debe exportar el reporte de clientes solicitado por el usuario respectivo (administrador maestro), en un formato PDF, teniendo como atributos: nombre, Rut, teléfono, dirección, descripción, estado.	Evidente
RF09	-El sistema debe exportar el reporte de contrato del cliente que se necesita para hacer el alquiler	Evidente
RF10	-El sistema debe exportar el reporte de la lista de quipos que serán alquilados por la empresa al cliente.	Evidente

Tabla 10 - Requisitos Funcionales

3.5.1 Interfaces Externas de Entrada

En la tabla 5 se muestran todas las interfaces externas de entrada presentes en el sistema.

ID	NOMBRE DEL ÍTEM	DETALLE DE DATOS CONTENIDOS EN ÍTEM
De_01	Datos del cliente	id_cliente, rut, nombre, ap_paterno, ap_materno, dirección, ciudad, correo, teléfono, descripción.
De_02	Datos del equipamiento	id, nombre, descripción, categoría, sub_categoria, cantidad,

De_03	Datos del Usuario	id_usuario, username, password, genero, permiso, descripcion_usuario.
De_04	Datos de alquiler	id, id_cliente, fecha_inicio, fecha_fin, fecha_fin_real, descripción, estado_alquiler.

Tabla 11 - Interfaces Externas de Entrada

3.5.2 Interfaces Externas de Salida

En la tabla 6 se muestran todas las interfaces externas de salida presentes en el sistema.

ID	NOMBRE DEL ÍTEM	DETALLE DE DATOS CONTENIDOS EN ÍTEM	MEDIO SALIDA
Is_01	Reporte de equipamiento	Nombre, cantidad, categoría, subcategoría, descripción.	archivo PDF impresora pantalla
Is_02	Contrato	Id, cliente, fecha inicio, fecha fin, fecha fin real, descripción, estado.	archivo PDF impresora pantalla

Is_03	Lista de equipos	Id, cliente, fecha inicio, fecha fin, fecha fin real, descripción, Nombre, cantidad, categoría, subcategoría.	archivo PDF impresora pantalla
Is_04	Reporte de clientes	Rut, nombre, dirección, correo, teléfono, descripción.	archivo PDF impresora pantalla

Tabla 12: Interfaces Externas de Salida

3.5.3 Atributos del producto

Para la definición de los atributos del producto este proyecto se basa en el modelo de la ISO/IEC 9126. Los atributos más relevantes son los que se detallan a continuación:

Portabilidad: Al ser una aplicación web el sistema puede ser ejecutado en cualquier navegador y en cualquier equipo con acceso a internet independiente de la plataforma utilizada. A continuación se muestra una tabla de compatibilidad de navegadores de prueba bajo dos sistemas.

TABLA COMPATIBILIDAD		
navegador	windows	linux
chrome	aprobado	aprobado
safari	aprobado	aprobado
mozilla	aprobado	aprobado
opera	aprobado	aprobado

Tabla 13 - compatibilidad de portabilidad

Mantenibilidad: La aplicación web permite ser actualizada de forma rápida, sin la necesidad que los usuarios se encarguen de esa tarea.

Seguridad: La aplicación web posee mecanismos de seguridad para la restricción de acceso a esta, permitiendo el uso de sus funcionalidades solo a los usuarios registrados.

CAPÍTULO 4

SOLUCIÓN PROPUESTA

Y ANÁLISIS DE

FACTIBILIDAD

Solución Propuesta

La solución propuesta, se refleja en los siguientes modelos de negocio.

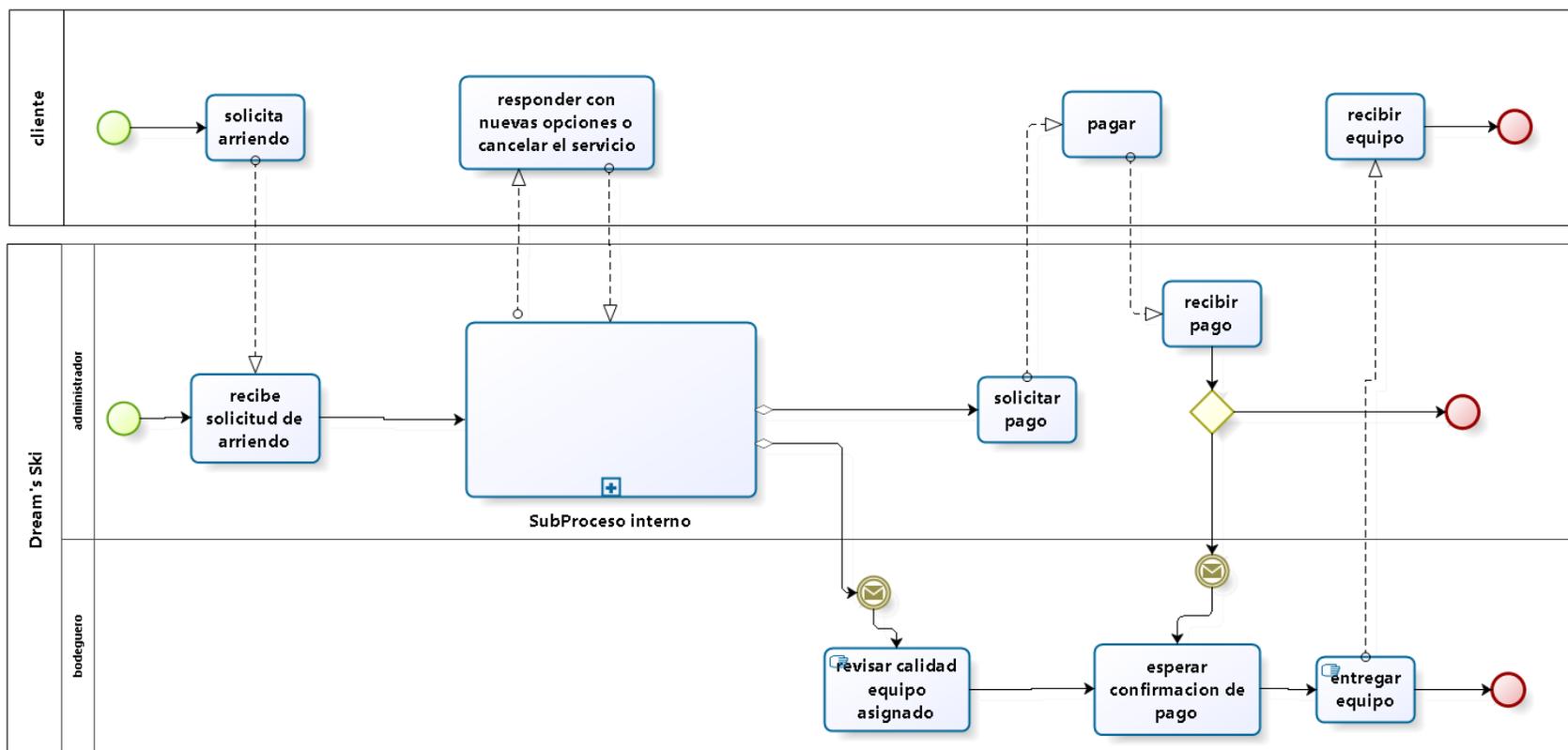


Ilustración 6 - Bpm solución

En la Ilustración 7 se detalla el subproceso interno del modelo de proceso de negocio para el administrador.

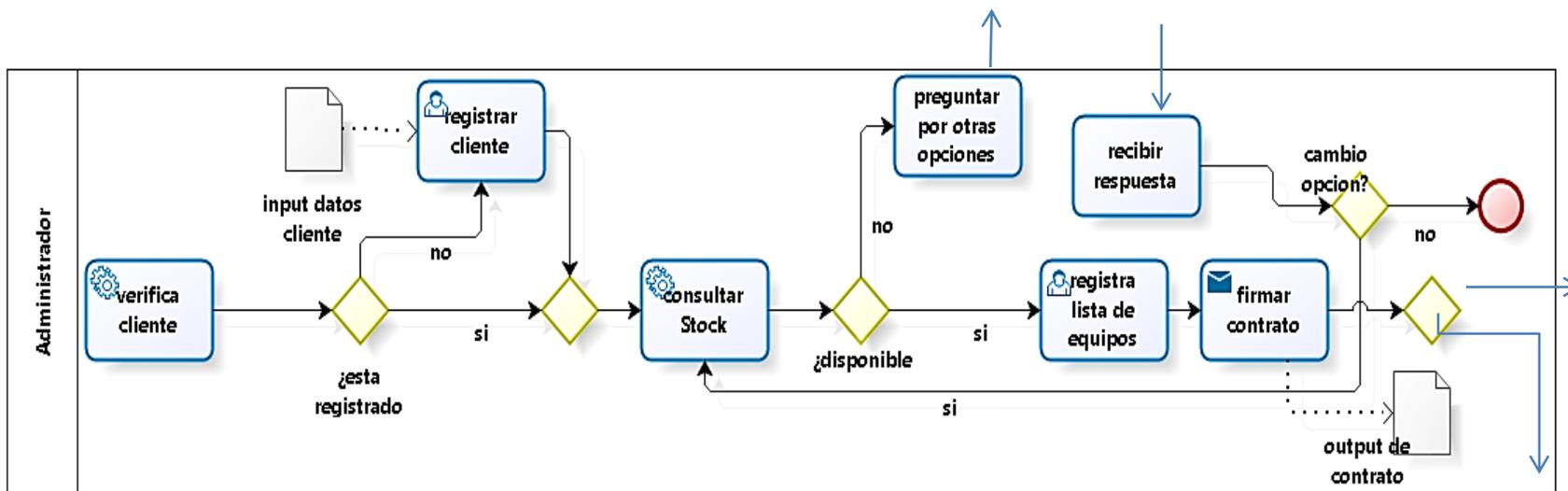


Ilustración 7 - Bpm solución detalle

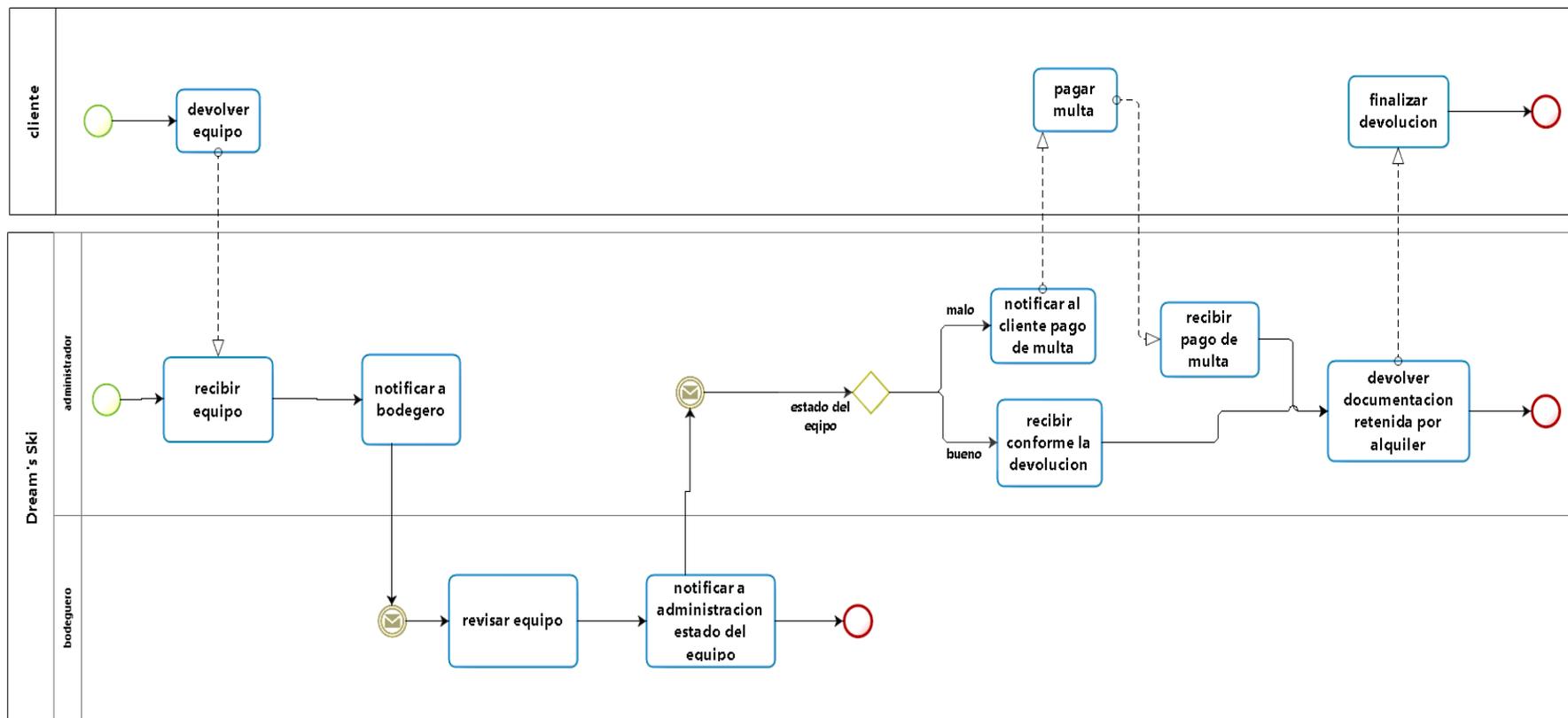


Ilustración 8 - Bpm devolver equipo

4.1 Análisis de Factibilidad

Antes de iniciar cualquier proyecto, se debe realizar un análisis de factibilidad que permita determinar si este es viable o no, es decir, si conviene llevarlo a cabo o no. Un estudio de factibilidad incluye tres aspectos básicos:

- Estudio operacional,
- Estudio técnico,
- Estudio económico.

Estos tres aspectos entregarán información que permitirá determinar la realización o no del proyecto.

En este capítulo se presenta el estudio de Factibilidad del Proyecto. Se definen costos y beneficios, para llevarlos a cifras tangibles, y así determinar finalmente la rentabilidad de la implementación del sistema.

4.2 Factibilidad Técnica

El estudio de factibilidad técnica está centrado en la realización de un análisis del equipamiento computacional e instalaciones que posee la empresa, determinando si éstos cumplen con los requisitos para su utilización o si, de lo contrario, es necesario complementarlos con nuevos recursos ya sea de software o hardware.

Una vez realizado este análisis, se pueden enumerar los siguientes elementos:

1. Las instalaciones cuentan con 4 ambientes:
 - a. La sala principal donde está el mostrador que separa al administrador de los clientes que se atienden, junto a la sala de espera.
 - b. Una bodega de iguales proporciones donde alberga todo el equipamiento disponible.

- c. Una pequeña oficina de gerencia.
 - d. Un baño.
2. Centrándonos en los primeros 3 puntos, en la sala principal se cuenta con un equipo computacional que posee suficientes características para realizar todas las tareas que diariamente demanda la empresa, y además ser utilizados para el acceso y uso del nuevo sistema informático y así quedaría en el mismo lugar a disposición de la administración y del nivel de usuario “Administrador”.
 3. La bodega posee 1 equipo, completamente apto para la realización de cada una de sus labores diarias en la empresa de inventariado de equipo en Excel, lo que lo hace compatible de usar con el nuevo sistema.
 4. El gerente posee su propio equipo computacional portátil, el que cumple con los requisitos para poder acceder al nuevo sistema.

Para esta situación se hace necesario el arriendo del servicio de Hosting u hospedaje, el cual cumpla con los siguientes Requisitos:

- Soporte para aplicaciones web basadas en JAVA EE.
- Soporte para bases de datos MySQL.
- Espacio de al menos 1 Gb.

4.3 Factibilidad Operacional

En cuanto al estudio de factibilidad operacional, ésta se centra en determinar la probabilidad existente de que los usuarios finales utilicen el nuevo sistema informático, contemplando las dificultades que puedan presentarse al implementar el sistema. Para llevar a cabo este análisis, se describirán tres aspectos fundamentales para determinar si es factible o no operacionalmente, estos aspectos son: La “complejidad del nuevo sistema”, la “resistencia al cambio”, y también la “mantención y administración del sistema”.

4.3.1 Complejidad del nuevo sistema

Es importante comenzar indicando que el nuevo sistema está pensado de tal forma que no presente mayor complejidad para los usuarios finales de éste, contemplando que los usuarios poseen conocimientos en cuanto al manejo de computadores de a lo menos un nivel medio. Se ha tenido presente la incorporación de elementos que facilitan su utilización. La idea contempla que los usuarios del sistema puedan navegar fácilmente a través de la aplicación web, considerando que éstos tienen las capacidades suficientes para aprender y comprender el funcionamiento del sistema.

4.3.2 Resistencia al cambio

En lo que respecta a este punto, se establece que pudiera existir algún nivel de resistencia al cambio, puesto que la empresa no posee ningún otro sistema informático que controle y automatice sus procesos. Pero además se estableció, que el hecho de que estén acostumbrados o tengan que realizar todo de forma manual, es más bien una problemática presente desde siempre en la empresa, debido a que estos procesos suelen ser largos y tediosos. Además, cabe destacar que la empresa considera como una necesidad la automatización de sus procesos mediante un sistema informático, y específica que sus usuarios finales están de acuerdo con cambiar la modalidad de realizar las tareas respectivas, con tal de lograr la automatización de procesos, debido a que estos les demandan demasiado tiempo de ejecución y papeleo. Por lo tanto, no debería existir mayor problema, debido a que además el personal de la empresa usuario del sistema posee conocimientos suficientes de manejo de computadores y experiencia utilizando otros sistemas y/o programas correspondientes a sus competencias, porque se encuentran en total disposición para la implantación del nuevo sistema informático propuesto.

4.3.3 Mantención y administración del sistema

En esta alternativa, la responsabilidad de la administración y mantención del sistema recae sobre la empresa que presta el servicio de Hosting, la cual debe contar con personal idóneo para realizar dichas actividades, las cuales son fundamentales para el buen funcionamiento de la aplicación. Para esta alternativa se determinó que es operacionalmente factible, ya que existen empresas que presta estos servicios y cumplen como mínimo con todo lo solicitado.

4.4 Factibilidad Política

En el estudio de factibilidad política, se determinará si es posible la realización y puesta en marcha del proyecto, considerando si éste se ajusta a las políticas y normas internas de la empresa.

Debido a esto, es importante destacar que el proyecto cuenta con la autorización del gerente de la empresa, quienes desean aprovechar la tecnología actual en ayuda de la optimización de los recursos para realizar la gestión y control de sus Equipo y alquileres.

Por lo tanto, se considera que la alternativa solución propuesta no presenta amenazas para el funcionamiento normal, y que además su puesta en marcha no va en contra de las políticas internas existentes en la empresa, por lo que es además políticamente factible.

4.5 Estudio Económico

En este estudio se determinan los recursos necesarios para desarrollar el proyecto y los costos en los que se debe incurrir para su implementación, se realiza una comparación entre los costos en hardware, software y mano de obra con los beneficios que se obtendrán cuando el sistema esté en su fase de explotación.

A continuación se detallan los costos considerados en el desarrollo del proyecto, los cuales corresponden a valores de mercado, obtenidos consultando la página Web www.mifuturo.cl

4.5.1 Costo de desarrollo

- **Hardware y Software de desarrollo:** Tiene un costo total de \$0, ya que el hardware no se requiere adquirir y es propiedad del desarrollador.

- **Encargado del desarrollo:** para llevar a cabo el proyecto se requiere de un ingeniero civil informático:
 - Costo de mercado de un analista desarrollador para lenguaje java hora/hombre es de \$9.888 aprox.
 - El trabajo se estima en un período de 4 meses y se trabajará 40 hrs. semanales, lo que se traduce en un total de 720 hrs. de elaboración del proyecto.
 - El costo total del ingeniero es de \$7.119.360

El costo total calculado anteriormente, no es considerado, debido a que el desarrollador, es un alumno que se encuentra realizando su proyecto de título.

4.5.2 Costo de implementación

Costos Software: En cuanto al software, se tiene contemplado utilizar software libre en cuanto a todas las herramientas de desarrollo y base de datos, los cuales se listan a continuación:

- ❖ IDE de programación: Eclipse
- ❖ Gestor de Base de Datos: MySQL Server 5.0
- ❖ Servidor Web: Apache HTTP Server 2.4.3
- ❖ Contenedor de JSP y Servlets: Apache Tomcat 6.
- ❖ Java Development Kit (JDK) 1.7

Costos de adquisición de máquina lectora de códigos de barras: Para la implementación del sistema también está estipulado el uso de una máquina lectora de códigos de barras, la cual agilizaría los procesos de ingreso de información.

- ❖ Costo lector laser códigos de barras USB = **\$37.990** pesos chilenos.

4.5.3 Costo de operación

Costos en el personal de soporte: El encargado de la mantención y soporte necesario al sistema será un profesional del área informática por contratación externa a la empresa.

- ❖ El costo de hora/hombre estimado de es de \$4.500 pesos.
- ❖ Tiempo dedicado al soporte 4 horas semanales, realizado 2 ves al mes.
- ❖ Costos del tiempo de soporte = \$36.000 mensuales.

Costo total anual = **\$432.000** pesos chilenos.

Arriendo Servicio Hosting: El costo anual por arrendar el servicio Hosting que posee los requerimientos pedidos es de **\$220.000** pesos chilenos y corresponde a la empresa Hosting Pro con el Plan Hosting Tomcat Profesional (consultado el 22 de junio de 2015), que posee entre otras características las siguientes:

- 500 MB de Espacio en Disco Duro para hospedar sitio web
- Tomcat 6 (Servidor Compartido) para Servlets Java WAR JSP
- Acceso Base Datos MySQL 5 ó PostgreSQL con 1 usuario administrador
- Tamaño Base Datos 80 MB adicionales a sitio web
- Soporte para Frameworks Struts o JSF, entre otros.

Las siguientes tablas que se muestran a continuación resumen todos los costos analizados, como forma de apreciar la cantidad total de inversión en el desarrollo, implementación y operación del proyecto.

Costos de Desarrollo	
Costos Personal Desarrollador	\$7.119.360
Total Desarrollo	\$7.119.360

Tabla 14 - costos desarrollador

Costos de Implementación	
Costos software	\$ 0
Costo Lector Laser Códigos de Barras USB	\$37.990
Total Implementación	\$37.990

Tabla 15 - costos de implementación

Costos de Operación	
Costos en el personal de soporte	\$432.000
Costos Arriendo Servicio Hosting	\$220.000
Total Operación	\$652.000

Tabla 16 - costos de operación

4.5.4 Beneficios tangibles

Al momento de evaluar beneficios desde el punto de vista económico, podemos encontrar ciertos aspectos difíciles de determinar, debido a que existen ventajas que son complejas de cuantificar, pero que representan grandes beneficios y mejoran la calidad de trabajo en la empresa, provocando incluso una mejora en las relaciones del personal, por consecuencia de las disminuciones de cargas laborales y automatizaciones de procesos.

Para poner el foco de este análisis, previamente se mostrará información recopilada por la empresa como estadísticas de los clientes que asisten al local y los alquileres que ellos realizan en los periodos en los cuales se centra su empresa.

A continuación en las tablas siguientes se detalla dicha información:

Periodo 2013	Promedio cantidad alquileres
junio	70
julio	85
agosto	130
septiembre	100
total periodo	385

Tabla 17 - cantidad de alquileres realizados en periodo 2013

Periodo 2014	Promedio cantidad alquileres
junio	75
julio	90
agosto	110
septiembre	115
total periodo	390

Tabla 18 - cantidad de alquileres realizados en periodo 2014

datos	cantidad
capacidad del local	8 clientes
capacidad atención	4 clientes por hora
tiempo de espera del cliente efectivo	30 minutos
clientes cotizando	75%
horas punta	de 9 a 13 horas

Tabla 19 - información general de logística de la empresa

Hay que tomar cuenta que el promedio de las cantidades de las tablas 11 y 12 son tomadas solo de los meses que se trabajan que son cuatro, donde solo se trabaja el fin de semana lo que lleva a contemplar solo 35 días como un supuesto máximo.

Dado la información entregada en las tablas previas se puede cuantificar los tiempos de atención y la cantidad de clientes promedio que atienden en el día esto se puede llevar a monetizar para manejarlo como utilidad.

Además sabemos que el tiempo que toma verificar el estado de un equipamiento antes de alquilar, o verificar el stock de estos es relativamente predeterminado, porque hay que buscarlo, y revisarlo cada vez que un cliente lo solicita, pero ahora se puede tomar un día o un tiempo muerto para poder llevar esa información de una vez a la aplicación y basta solo con revisarla en el momento para saber en qué estado se encuentra determinado equipo, esto llevaría consigo un ahorro de tiempo en el momento del alquiler y además un beneficio futuro ya que ese tiempo se puede utilizar para atender más rápido y así aumentar la capacidad de atención y evitar la deserción del local por parte de los clientes que esperan demasiado.

Enfocaremos el análisis en los beneficios, desde el punto de vista de la optimización del tiempo, que genera el uso del software al momento de revisar el equipamiento al momento de pedirlo un cliente para su alquiler.

Tomando en cuenta el tiempo que se demora en revisar el stock y el estado cada vez que se requería realizar un alquiler este toma un tiempo promedio de 3 minutos, y dado que ahora solo basta revisar el sistema con un tiempo promedio de 1 minuto, se puede deducir que se ha ganado dos tercios del tiempo para poder agilizar nuevos alquileres. Lo que podemos monetizar de 2 formas:

Primero, con el tiempo del personal: con un sueldo de \$450.000 que antes les permitía hacer una tarea, ahora ya han ganado $2/3$ del tiempo lo que lleva a un Beneficio de \$300.000 por cada trabajador ya que puede hacer 2 tareas más en el mismo tiempo.

Segundo, con la agilización de los tiempos de espera de los clientes, donde si estaban ocupados los trabajadores, estos solo esperan 30 minutos, lo que significa que con la capacidad de 8 clientes del local por hora y la capacidad de atención de 4 clientes por hora, aplicando Teoría de líneas de espera de Investigación de operaciones, se lleva a que solo atiende 12 clientes por día (cada hora que pasa pierden un cliente por la demora, obviando la frecuencia con la que llegan) Luego, como el margen de $2/3$ obtenidos como beneficio del sistema, se puede inferir que ahora la capacidad de atención aumentara de 4 a 12, o sea 8 nuevos clientes por hora, pero debido a la premisa entregada anteriormente, solo el 25% de los clientes se atiende efectivamente los demás solo cotizan, por lo que la capacidad real de atención aumenta de 4 a 6 clientes por hora lo que aumenta en un 50 % más la capacidad de atención teóricamente. Todo está sujeto a la frecuencia de ingreso de los clientes al local y eso no se puede inferir sin un estudio.

En un día común el precio promedio de un alquiler es de \$27.000 pesos por alquiler y que en un día normal se realizan doce alquileres. Con esta nueva modalidad se estima que se podría atender a 8 nuevos clientes al día, que significa un ingreso de \$ 216.000 pesos adicional diarios. También tomaremos en cuenta que los meses que se trabajan son cuatro, donde solo se trabaja el fin de semana lo que lleva a contemplar solo 35 días de un supuesto máximo que les permita obtener las utilidades estimadas que ascienden a \$7.560.000 pesos por temporada anual.

4.5.5 Cálculo de Flujos de Caja Neto

Para poder calcular la factibilidad económica de este proyecto, se usará como indicador el Valor Actual Neto (VAN).

Consideraciones del análisis:

- Se estimará un tiempo de 4 años de vida útil del proyecto
- No se incluirá el valor del desarrollador por ser proyecto de título.
- El proyecto se someterá a una evaluación con una tasa de descuento del 8%, de exigencia.
- La tasa de descuento es la medida de rentabilidad mínima exigida por el proyecto y que permite recuperar la Inversión inicial, cubrir los costos efectivos de producción y obtener beneficios. La tasa de descuento representa la tasa de interés a la cual los valores futuros se actualizan al presente. Se tomará esta tasa de descuento de un 8% ya que hay que evaluar el coste de oportunidad y alternativas más viables, actualmente la tasa promedio de un fondo de ahorro sin riesgo es de un 4,13% por lo tanto hay que agregarle a esta tasa otra prima de riesgo, además debido que como es un proyecto de título para la universidad del Bío-Bío, se tomara en definitiva la taza que está le exige a sus proyectos, que es de un 8%.
- El flujo de caja se realizará en base a las utilidades que se podrían efectuar por el uso del software sobre el manual, optimizando el tiempo de los trabajadores como lo vimos en el apartado anterior y lo llamaremos utilidad por uso de software

	Año 0	Año 1	Año 2	Año 3	Año 4
(-)Costo de desarrollo	0	0	0	0	0
(-)Costo de implementación	37.990	0	0	0	0
(-)Costo de operación	652.000	652.000	652.000	652.000	652.000
(+) utilidad por uso de software	0	7.560.000	7.560.000	7.560.000	7.560.000
Total bruto	689.990	6.908.000	6.908.000	6.908.000	6.908.000
Impuesto (20%)		1.381.600	1.381.600	1.381.600	1.381.600
Total neto	689.990	5.526.400	5.526.400	5.526.400	5.526.400
	(iva inc.)				

Tabla 20 Flujo de Caja

4.5.6 Cálculo del VAN

La factibilidad económica del presente proyecto será calculada en base al indicador VAN, el cual permite saber si los beneficios superan a los costos en un umbral de 5 años que se consideran como vida útil del proyecto.

Este cálculo es realizado mediante la siguiente fórmula:

$$\sum_{i=1}^n \frac{FC_i}{(1+k)^i} - I_0$$

Dónde:

n = número de años de vida útil del proyecto.

i = representa el año actual en la fórmula.

FC_i = representa a cada uno de los flujo de caja neto.

k = representa la tasa de interés o de descuento.

I_0 = representa la inversión inicial.

$$VAN(8\%) = \frac{5.526.400}{(1 + 0.08)^1} + \frac{5.526.400}{(1 + 0.08)^2} + \frac{5.526.400}{(1 + 0.08)^3} + \frac{5.526.400}{(1 + 0.08)^4} + \frac{5.526.400}{(1 + 0.08)^5} - 689.990$$

$$VAN(8\%) = 21.415.610$$

4.5.7 Conclusión Factibilidad

Como se ha presentado este análisis de factibilidad, tanto en el aspecto de costos, esfuerzo y tecnología el proyecto es totalmente realizable ya que el cálculo del VAN dió positivo por lo que el proyecto es viable económicamente. Además, genera un valor agregado no calculable en términos monetarios simples.

Se puede inferir por cierto que la satisfacción del cliente aumentará ya que los tiempos de espera se disminuirán en la atención, y la percepción de que los procesos se hacen automatizados a través de un software le dan un valor agregado tanto para el cliente como para el empleado.

Por otra parte mencionar la exigencia que le demandamos al proyecto con la tasa de descuento de un 8%, dado que puede ser más viable poner el dinero en otra inversión menos riesgosa, con una exigencia mínima como puede ser una tasa de fondos mutuos, pero se decidió invertir en este proyecto que requiere más esfuerzo y es más riesgoso por ende le exigimos más.

Si bien es cierto se presentan una serie de cifras para los cálculos de los montos, estos no son estrictamente representativos, se usaron estimaciones de tiempo de los procesos, ya que depende de una serie de factores externos complejos de cuantificar.



UNIVERSIDAD DEL BÍO-BÍO

CAPÍTULO 5

ANÁLISIS DE

REQUERIMIENTOS Y

DISEÑO DE LA

SOLUCIÓN

5.1 Determinación de Requerimientos.

“Los requerimientos son capacidades y condiciones con las cuales debe ser conformado el sistema y más ampliamente, el proyecto. La especificación de requerimientos es encontrar, comunicar y recordar lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo” (Larman, 2003).

La información para determinar los requerimientos del sistema a desarrollar se obtuvo a través de reuniones con los trabajadores de la empresa quienes constituyen los principales usuarios del sistema, los cuales fueron presentados en el CAPÍTULO 3 IDENTIFICACIÓN DE REQUERIMIENTOS.

5.2 Usuarios

En la empresa Dream's Ski los actores fueron detectados por su directa relación con la necesidad de utilizar el Sistema, para cumplir con sus necesidades dentro de la empresa, estos actores son los siguientes:

5.2.1 Administrador:

Persona autorizada para la utilización del sistema, su rol dentro de la empresa es la de controlar todos los movimientos en la empresa, siendo el principal encargado en la toma de decisiones sobre los alquileres realizados, los niveles que este necesita para utilizar el software son mínimos ya que el sistema está implementado para una rápida adaptación y uso. Este es el usuario con más privilegios dentro del sistema ya que tiene más funcionalidades y accesos. Esta persona principalmente revisa los reportes viendo toda la información correspondiente para la ayuda en la toma de decisiones. No obstante, tiene además acceso a 100% de las funcionalidades del sistema, lo que quiere decir

que puede realizar tanto sus funcionalidades exclusivas como Administrador, más cada una de las funcionalidades del actor bodeguero.

En la tabla 7 se muestra al actor administrador y sus funcionalidades y privilegios en el sistema.

Actor	Privilegios del sistema
<p>Administrador</p>	<p>Generales</p>
	<ul style="list-style-type: none"> -Iniciar Sesión -Cerrar Sesión -Agregar Equipo -Eliminar Equipo -Editar Equipo -Listar Equipo
	<p>Exclusivas</p>
<ul style="list-style-type: none"> -Agregar Usuario -Modificar Datos de Usuario -Eliminar Usuario -Listar Usuario -Agregar Cliente -Editar Cliente -Eliminar Cliente -Listar Cliente -Agregar Alquiler -Eliminar Alquiler -devolver equipos (Alquiler) -Listar Alquiler 	

Tabla 21: Actor administrador y sus privilegios del sistema

5.2.2 Bodeguero:

Persona autorizada para la utilización del sistema web, su rol dentro de la empresa es la de cumplir con recolectar e ingresar toda la información de los movimientos que esta tiene de los equipos disponibles en la empresa, los niveles de conocimiento que necesita para utilizar el software son mínimos ya que el sistema está implementado para una rápida adaptación y uso. Esta persona es el responsable de que los datos de las diferentes entidades que se presentan en el proyecto, se encuentren actualizados. El nivel de acceso que tiene en la aplicación es limitado ya que solamente se encarga de ingresar datos de los Equipos.

En la Tabla 8 se muestra al actor Bodeguero y sus funcionalidades y privilegios en el sistema.

Actor	Privilegios en el sistema
Bodeguero	<p>Generales</p> <ul style="list-style-type: none"> -Iniciar Sesión -Cerrar Sesión -Agregar Equipo -Eliminar Equipo -Editar Equipo -Listar Equipo

Tabla 22: Actor Bodeguero y sus privilegios del sistema

5.3 Casos de Uso

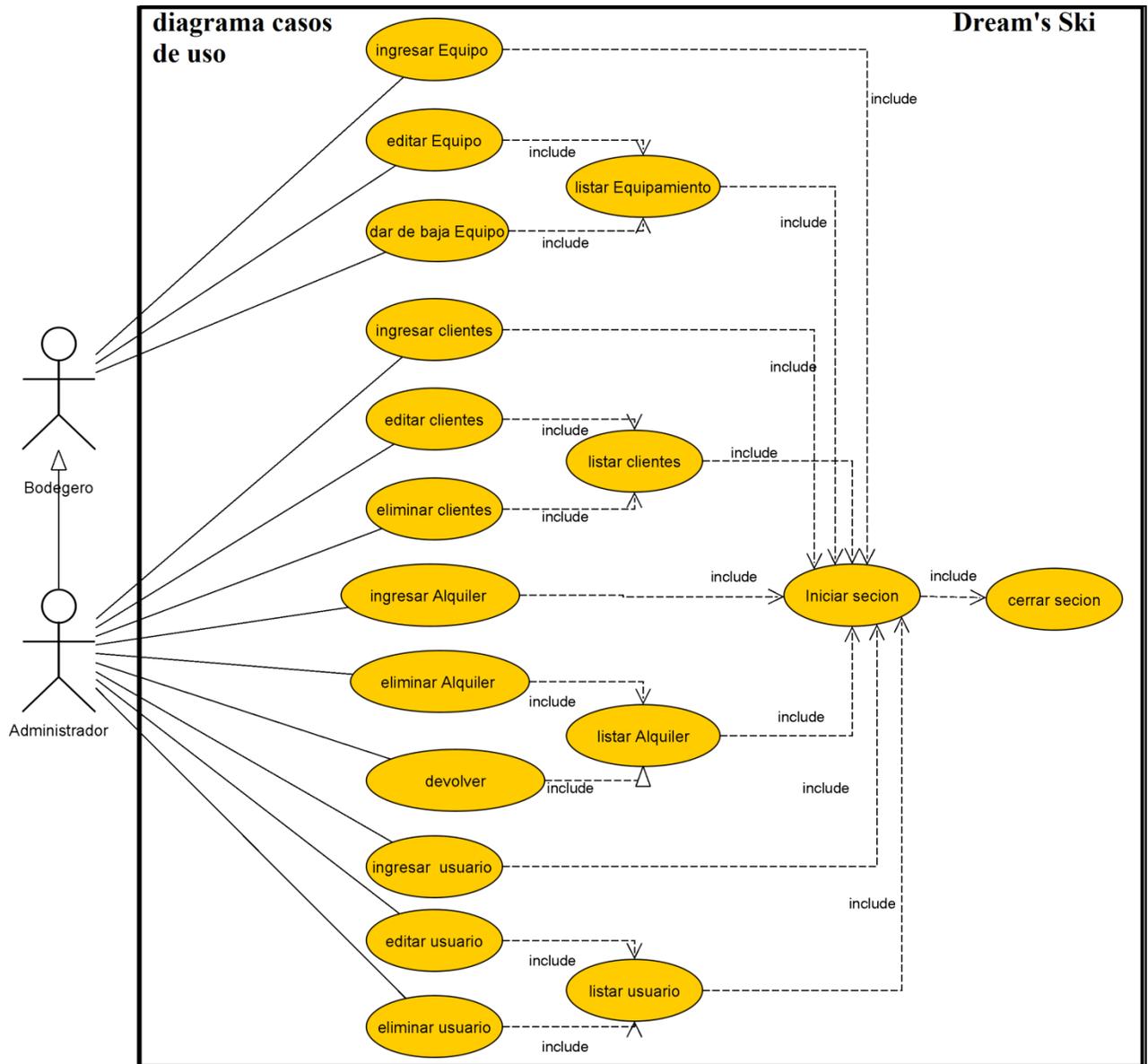


Ilustración 9 - Diagrama Caso de uso

5.4 Descripción de casos de uso, funcionalidades generales

Se presenta desde la tabla 27 a la 44, la descripción de cada uno de los casos de usos correspondientes al primer incremento, en los cuales se da a conocer cada una de sus propiedades, así como su completa definición.

ID	CU01	
Caso de Uso	Iniciar Sesión	
Referencias	RF01	
Actores	Administrador, Bodeguero	
Descripción	Este Caso de Uso es el encargado de permitir el ingreso al administrador o bodeguero con sus respectivos permisos, para que utilice el sistema.	
Precondiciones	- Que el usuario exista en el sistema.	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
	1.- El actor selecciona la opción "Iniciar Sesión".	2.- El sistema solicita que el actor ingrese sus datos: username password
	3.- El actor ingresa los datos solicitados por el Sistema.	4.- El Sistema valida los datos ingresados por el actor.
		5.- El Sistema muestra el menú respectivo.
	6.- El actor recibe la información pedida anteriormente.	
	7.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
4a		
		1.- El sistema no encuentra el usuario en la base de datos: username o password mal escrito Username o password inexistente o erróneo.
		2.- Se solicita al actor que verifique y reingrese los datos al Sistema.
	3.- Se vuelve al paso 3	
Postcondiciones	<ul style="list-style-type: none"> - Se obtienen los menús respectivos para el actor correspondiente. - El actor está dentro del sistema. 	

Tabla 23: Descripción Caso de Uso - Iniciar Sesión

ID	CU02	
Caso de Uso	Cerrar Sesión	
Referencias	RF01	
Actores	Administrador, Bodeguero	
Descripción	Este Caso de Uso es el encargado de cerrar la sesión del administrador y Bodeguero, guardando todos los cambios pertinentes que este realizó, en la base de datos.	
Precondiciones	- Que el usuario exista en el sistema.	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
	1.- El actor selecciona la opción "Cerrar Sesión".	2.- El sistema guarda todos los cambios realizados hasta el momento por el actor.
		3.- El Sistema direcciona a la página de login para que el usuario deba acceder nuevamente al sistema.
	4.- El actor se encuentra en la página de login, pudiendo iniciar nuevamente su sesión.	
	5.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
Postcondiciones	<ul style="list-style-type: none"> - Se direcciona al actor a la página de login del usuario, para que inicie su sesión nuevamente. - El actor está fuera del sistema. 	

Tabla 24: Descripción Caso de Uso - Cerrar Sesión

ID		CU03	
Caso de Uso	Agregar Equipo		
Referencias	RF03		
Actores	Administrador, Bodeguero		
Descripción	Este Caso de Uso permite ingresar información al administrador y Bodeguero, con respecto a los Equipos que se utilizarán en la empresa.		
Precondiciones	- Que el usuario exista en el sistema.		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "Agregar".		2.- El sistema solicita que el actor ingrese los datos del Equipo: nombre_Equipo, descripción, categoría, cantidad y otros atributos.	
3.- El actor ingresa los datos solicitados por el Sistema.		4.- El Sistema recibe los datos ingresados por el actor.	
		5.- El Sistema guarda en la base de datos los datos respectivos del Equipo.	
6.- El actor verifica la información ingresada anteriormente.			
7.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
3a			
1.- El actor ingresa los datos erróneamente.		2.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
3.- Se vuelve al paso 3			
Postcondiciones	- Quedan ingresados en la base de datos los datos del nuevo Equipo.		

Tabla 25: Descripción Caso de Uso - Agregar Equipo

ID	CU04	
Caso de Uso	Dar de baja Equipo	
Referencias	RF04	
Actores	Administrador, Bodeguero	
Descripción	Este Caso de Uso permite al administrador y Bodeguero Dar de baja los distintos Equipos, los cuales ya no serán utilizados por la Empresa.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Equipo, para así poder dar de baja alguno. - Que se muestre en pantalla un listado con todos los Equipos que están en la base de datos de la empresa. 	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
		1.- El sistema muestra un listado con todos los Equipos que se encuentran en la base de datos.
	2.- El actor selecciona el Equipo que desea eliminar, haciendo clic en la casilla “dar de baja”.	3.- El sistema da de baja el Equipo en la base de datos cambiando su atributo “Estado” de “disponible” a “No disponible”
		4.- El sistema direcciona a la página principal de Equipos, al usuario.
	5.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
Postcondiciones	<ul style="list-style-type: none"> - El equipo seleccionado cambia su atributo de Estado de disponible a no disponible, por lo que seguirá en la base de datos, pero no estará a disposición para su uso. 	

Tabla 26: Descripción Caso de Uso – Dar de baja Equipo

ID		CU05	
Caso de Uso	Editar Equipo		
Referencias	RF03		
Actores	Administrador, Bodeguero		
Descripción	Este Caso de Uso permite al administrador y Bodeguero modificar los distintos datos del Equipo.		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Equipo, para así poder editar alguno. - Que se muestre en pantalla un listado con todos los Equipos que están en la base de datos de la empresa. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
		1.- El sistema muestra un listado con todos los Equipos que se encuentran en la base de datos.	
2.- El actor selecciona el Equipo que desea modificar, haciendo clic en la casilla "Editar".		3.- El sistema le pide que ingrese los nuevos datos del Equipo: nombre_Equipo, descripción, categoría, cantidad y otros atributos	
4.- El actor ingresa los nuevos datos del Equipo.		5.- El Sistema recibe los datos ingresados por el actor.	
		6.- El sistema modifica los datos del Equipo en la base de datos.	
		7.- El Sistema muestra el listado de todos los Equipos.	
8.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
4a			
1.- El actor ingresa los datos con errores.		2.- El sistema encuentra errores en los datos ingresados del Equipo.	
		3.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
4.- Se vuelve al paso 4			
Postcondiciones	<ul style="list-style-type: none"> - Quedan modificados los datos del Equipo en la base de datos. 		

Tabla 27: Descripción Caso de Uso - Editar Equipo

ID	CU06	
Caso de Uso	Listar Equipos	
Referencias	RF03	
Actores	Administrador, Bodeguero	
Descripción	Este Caso de Uso es el encargado de entregar la información de todos los Equipos que se encuentran en la empresa.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Equipo, para así poder listar alguno. 	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
	1.- El actor selecciona la opción "Listar Equipos".	2.- El sistema muestra una tabla con todos los Equipos que se encuentran en la base de datos, mostrando su: nombre_Equipo, descripción, cantidad y demás atributos.
	3.- El actor recibe la información pedida anteriormente.	
	4.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
Postcondiciones	- Se obtiene un listado de todos los Equipos por pantalla.	

Tabla 28: Descripción Caso de Uso - Listar Equipos

ID		CU07	
Caso de Uso	Agregar Cliente		
Referencias	RF04		
Actores	Administrador		
Descripción	Este Caso de Uso permite ingresar información al administrador, con respecto a los clientes que se registraran en el sistema de la empresa.		
Precondiciones	- Que el usuario exista en el sistema.		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "Agregar".		2.- El sistema solicita que el actor ingrese los datos del Cliente.	
3.- El actor ingresa los datos solicitados por el Sistema.		4.- El Sistema recibe los datos ingresados por el actor.	
		5.- El Sistema guarda en la base de datos los datos respectivos del Cliente.	
6.- El actor verifica la información ingresada anteriormente.			
7.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
3a			
1.- El actor ingresa los datos erróneamente.		2.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
3.- Se vuelve al paso 3			
Postcondiciones	- Quedan ingresados en la base de datos los datos del nuevo Cliente.		

Tabla 29 - Descripción Caso de Uso - Agregar Cliente

ID	CU08	
Caso de Uso	Dar de baja Cliente	
Referencias	RF04	
Actores	Administrador	
Descripción	Este Caso de Uso permite al administrador Dar de baja A los clientes que ya no sean necesarios administrar por la empresa, solo para registros históricos se mantendrán en la base de datos.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Cliente, para así poder dar de baja alguno. - Que se muestre en pantalla un listado con todos los Clientes que están activos en la base de datos de la empresa. 	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
		1.- El sistema muestra un listado con todos los Clientes que se encuentran en la base de datos.
	2.- El actor selecciona el Cliente que desea dar de baja, haciendo clic en la casilla "borrar".	3.- El sistema da de baja al Cliente en la base de datos cambiando su atributo "Estado" de "disponible" a "No disponible"
		4.- El sistema direcciona a la página principal de Clientes al usuario.
	5.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
Postcondiciones	<ul style="list-style-type: none"> - El Cliente seleccionado cambia su atributo de Estado de disponible a no disponible, por lo que seguirá en la base de datos, pero no estará a disposición para su uso. 	

Tabla 30 - Descripción Caso de Uso – Dar de baja Clientes

ID	CU09	
Caso de Uso	Editar Cliente	
Referencias	RF04	
Actores	Administrador	
Descripción	Este Caso de Uso permite al administrador modificar los distintos datos de los Clientes.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Cliente, para así poder editar alguno. - Que se muestre en pantalla un listado con todos los Clientes que están en la base de datos de la empresa. 	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
		1.- El sistema muestra un listado con todos los Clientes que se encuentran en la base de datos.
	2.- El actor selecciona el Cliente que desea modificar, haciendo clic en la casilla "Editar".	3.- El sistema le pide que ingrese los nuevos datos del Cliente
	4.- El actor ingresa los nuevos datos del Cliente.	5.- El Sistema recibe los datos ingresados por el actor.
		6.- El sistema modifica los datos del Cliente en la base de datos.
		7.- El Sistema muestra el listado de todos los Clientes.
	8.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
4a		
	1.- El actor ingresa los datos con errores.	2.- El sistema encuentra errores en los datos ingresados del Cliente.
		3.- Se solicita al actor que verifique y reingrese los datos al Sistema.
	4.- Se vuelve al paso 4	
Postcondiciones	- Quedan modificados los datos del Cliente en la base de datos.	

Tabla 31 - Descripción Caso de Uso - Editar Clientes

ID		CU10	
Caso de Uso	Listar Cliente		
Referencias	RF04		
Actores	Administrador		
Descripción	Este Caso de Uso es el encargado de entregar la información de todos los Clientes que se encuentran en la empresa.		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Cliente, para así poder listar alguno. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "Listar Clientes".		2.- El sistema muestra una tabla con todos los Clientes que se encuentran en la base de datos, mostrando sus Atributos.	
3.- El actor recibe la información pedida anteriormente.			
4.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
Postcondiciones	- Se obtiene un listado de todos los Clientes por pantalla.		

Tabla 32 - Descripción Caso de Uso - Listar Clientes

ID		CU11	
Caso de Uso	Agregar Usuario		
Referencias	RF06		
Actores	Administrador		
Descripción	Este Caso de Uso permite ingresar información al administrador, con respecto a los Usuarios que se registraran en el sistema de la empresa.		
Precondiciones	- Que el usuario exista en el sistema.		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "Agregar".		2.- El sistema solicita que el actor ingrese los datos del Usuario.	
3.- El actor ingresa los datos solicitados por el Sistema.		4.- El Sistema recibe los datos ingresados por el actor.	
		5.- El Sistema guarda en la base de datos los datos respectivos del Usuario.	
6.- El actor verifica la información ingresada anteriormente.			
7.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
3a			
1.- El actor ingresa los datos erróneamente.		2.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
3.- Se vuelve al paso 3			
Postcondiciones	- Quedan ingresados en la base de datos los datos del nuevo Usuario.		

Tabla 33 - Descripción Caso de Uso - Agregar Usuario

ID		CU12	
Caso de Uso	Eliminar Usuario		
Referencias	RF06		
Actores	Administrador		
Descripción	Este Caso de Uso permite al administrador Eliminar A los Usuarios que ya no sean necesarios administrar por la empresa,		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Usuario, para así poder Eliminar alguno. - Que se muestre en pantalla un listado con todos los Usuarios que están activos en la base de datos de la empresa. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
		1.- El sistema muestra un listado con todos los Usuarios que se encuentran en la base de datos.	
2.- El actor selecciona el Usuario que desea dar de baja, haciendo clic en la casilla "borrar".		3.- El sistema Elimina al Usuario en la base de datos.	
		4.- El sistema direcciona a la página principal de Usuarios al usuario.	
5.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
Postcondiciones			

Tabla 34 - Descripción Caso de Uso – Eliminar Usuarios

ID	CU13	
Caso de Uso	Editar Usuario	
Referencias	RF06	
Actores	Administrador	
Descripción	Este Caso de Uso permite al administrador modificar los distintos datos de los Usuarios.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Usuario, para así poder editar alguno. - Que se muestre en pantalla un listado con todos los Usuarios que están en la base de datos de la empresa. 	
FLUJO NORMAL DE EVENTOS		
Actor	Sistema	
	1.- El sistema muestra un listado con todos los Usuarios que se encuentran en la base de datos.	
2.- El actor selecciona el Usuario que desea modificar, haciendo clic en la casilla "Editar".	3.- El sistema le pide que ingrese los nuevos datos del Usuario	
4.- El actor ingresa los nuevos datos del Usuario.	5.- El Sistema recibe los datos ingresados por el actor.	
	6.- El sistema modifica los datos del Usuario en la base de datos.	
	7.- El Sistema muestra el listado de todos los Usuarios.	
8.- El actor termina la operación.		
FLUJO DE EVENTOS ALTERNATIVOS		
4a		
1.- El actor ingresa los datos con errores.	2.- El sistema encuentra errores en los datos ingresados del Usuario.	
	3.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
4.- Se vuelve al paso 4		
Postcondiciones	- Quedan modificados los datos del Usuario en la base de datos.	

Tabla 35 - Descripción Caso de Uso - Editar Usuarios

ID	CU14	
Caso de Uso	Listar Usuario	
Referencias	RF06	
Actores	Administrador	
Descripción	Este Caso de Uso es el encargado de entregar la información de todos los Usuarios que se encuentran en la empresa.	
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 Usuario, para así poder listar alguno. 	
FLUJO NORMAL DE EVENTOS		
	Actor	Sistema
	1.- El actor selecciona la opción "Listar Usuarios".	2.- El sistema muestra una tabla con todos los Usuarios que se encuentran en la base de datos, mostrando sus Atributos.
	3.- El actor recibe la información pedida anteriormente.	
	4.- El actor termina la operación.	
FLUJO DE EVENTOS ALTERNATIVOS		
Postcondiciones	- Se obtiene un listado de todos los Usuarios por pantalla.	

Tabla 36 - Descripción Caso de Uso - Listar Usuarios

ID		CU15	
Caso de Uso	Agregar alquiler		
Referencias	RF05		
Actores	Administrador		
Descripción	Este Caso de Uso permite ingresar información al administrador, con respecto a los alquileres que se realizan en la empresa.		
Precondiciones	- Que el usuario exista en el sistema. El cliente esté debidamente registrado, y los quipos que solicite estén disponibles.		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "registrar alquiler".		2.- El sistema solicita que el actor ingrese los datos del Equipo y del cliente, posteriormente los propios del alquiler	
3.- El actor ingresa los datos solicitados por el Sistema.		4.- El Sistema recibe los datos ingresados por el actor.	
		5.- El Sistema guarda en la base de datos los datos respectivos al alquiler	
6.- El actor verifica la información ingresada anteriormente.			
7.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
3a			
1.- El actor ingresa los datos erróneamente.		2.- Se solicita al actor que verifique y reingrese los datos al Sistema.	
3.- Se vuelve al paso 3			
Postcondiciones	- Quedan ingresados en la base de datos los datos del nuevo Equipo.		

Tabla 37 - Descripción Caso de Uso - Agregar alquiler

ID		CU16	
Caso de Uso	Devolver		
Referencias	RF05		
Actores	Administrador		
Descripción	Este Caso de Uso permite al administrador devolver a sus bodegas los equipos que fueron alquilados por determinados clientes.		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1alquiler, para así poder devolver - Que se muestre en pantalla un listado con todos los alquileres que están en la base de datos de la empresa. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
		1.- El sistema muestra un listado con todos los alquileres que se encuentran en la base de datos.	
2.- El actor selecciona el alquiler que desea devolver a sus bodegas, haciendo clic en la casilla “devolver”.		3.- El sistema muestra otra ventana el permita al usuario ingresar a bodega la cantidad de equipos que fueron alquilados.	
4.- el cliente ingresa la cantidad de quipos devueltos con la fecha de devolución real,		5.- el sistema agrega a la lista de equipos la cantidad de estos que estaba alquilado y asi liberarlo para el futuro	
		6.- El sistema direcciona a la página principal de alquiler, al usuario.	
7.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
Postcondiciones	<ul style="list-style-type: none"> - El equipo seleccionado cambia su atributo de Estado de disponible a no disponible, por lo que seguirá en la base de datos, pero no estará a disposición para su uso. 		

Tabla 38 - Descripción Caso de Uso – Devolver

ID		CU17	
Caso de Uso	Eliminar alquiler		
Referencias	RF05		
Actores	Administrador,		
Descripción	Este Caso de Uso permite al administrador eliminar los alquileres.		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 alquiler realizado. - Que se muestre en pantalla un listado con todos los alquileres que están en la base de datos de la empresa. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
		1.- El sistema muestra un listado con todos los alquileres que se encuentran en la base de datos.	
2.- El actor selecciona el alquiler que desea eliminar, haciendo clic en la casilla "aliminar".		3.- El sistema lo elimina en la base de datos"	
		4.- El sistema direcciona a la página principal de Equipos, al usuario.	
5.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
Postcondiciones	<ul style="list-style-type: none"> - Los equipos que estaban alquilados vuelven a estar disponibles. 		

Tabla 39 - Descripción Caso de Uso – eliminar alquiler

ID		CU18	
Caso de Uso	Listar alquiler		
Referencias	RF03		
Actores	Administrador,		
Descripción	Este Caso de Uso es el encargado de entregar la información de todos los alquileres que se realizaron en la empresa.		
Precondiciones	<ul style="list-style-type: none"> - Que el usuario exista en el sistema. - Que haya en la base de datos al menos 1 alquiler, para así poder listar alguno. 		
FLUJO NORMAL DE EVENTOS			
Actor		Sistema	
1.- El actor selecciona la opción "registrar alquiler".		2.- El sistema muestra una tabla con todos los Equipos que se encuentran en la base de datos, mostrando su: nombre, descripción, cantidad y demás atributos.	
3.- El actor recibe la información pedida anteriormente.			
4.- El actor termina la operación.			
FLUJO DE EVENTOS ALTERNATIVOS			
Postcondiciones	<ul style="list-style-type: none"> - Se obtiene un listado de todos los Equipos por pantalla, donde además da la opción de registrar un nuevo alquiler y obtener una lista de los equipos del último alquiler realizado. 		

Tabla 40 - Descripción Caso de Uso – Listar Alquiler

5.5 Diagrama de arquitectura del sistema

El diagrama de arquitectura del sistema es un diseño donde se modelan de forma gráfica y simplificada lo que se quiere construir.

El objetivo principal de este diagrama es ofrecer una visión simplificada del sistema, de forma que con solo mirar el diagrama se puede entender lo que se quiere conseguir. Esto resulta de gran utilidad ante la llegada de nuevos miembros al proyecto y así explicar el funcionamiento general del sistema.

A continuación en la Ilustración 10 se presentara el diagrama de arquitectura del sistema.

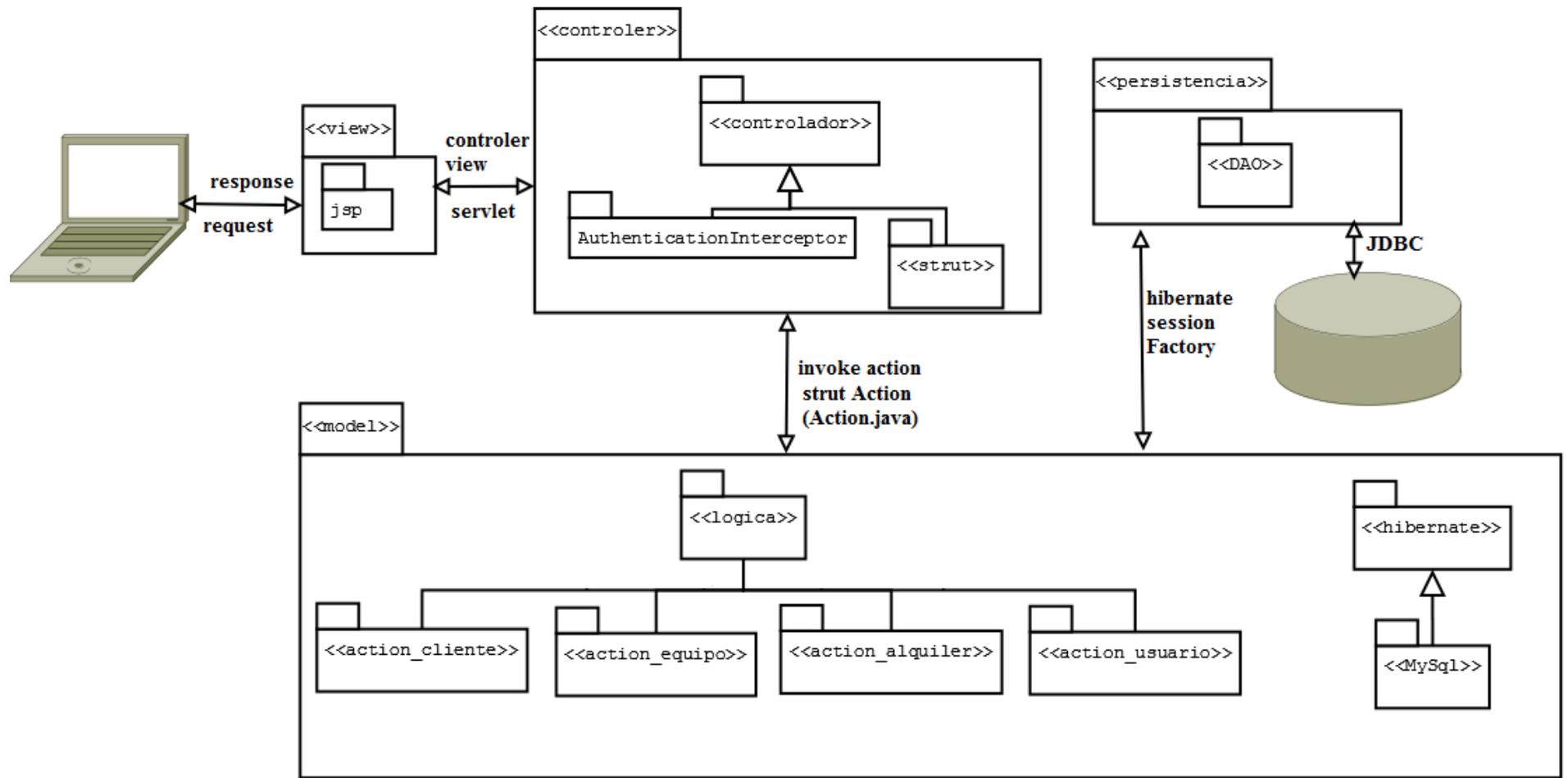


Ilustración 10 - Diagrama de Arquitectura del Sistema

A continuación se presentara el detalle de la lógica del modelo en Ilustración 11 y las clases DAO en la Ilustración 12 para su mayor claridad, las otras se omitirán por ser las que comúnmente corresponden al modelo vista controlador con el Framework Strut.

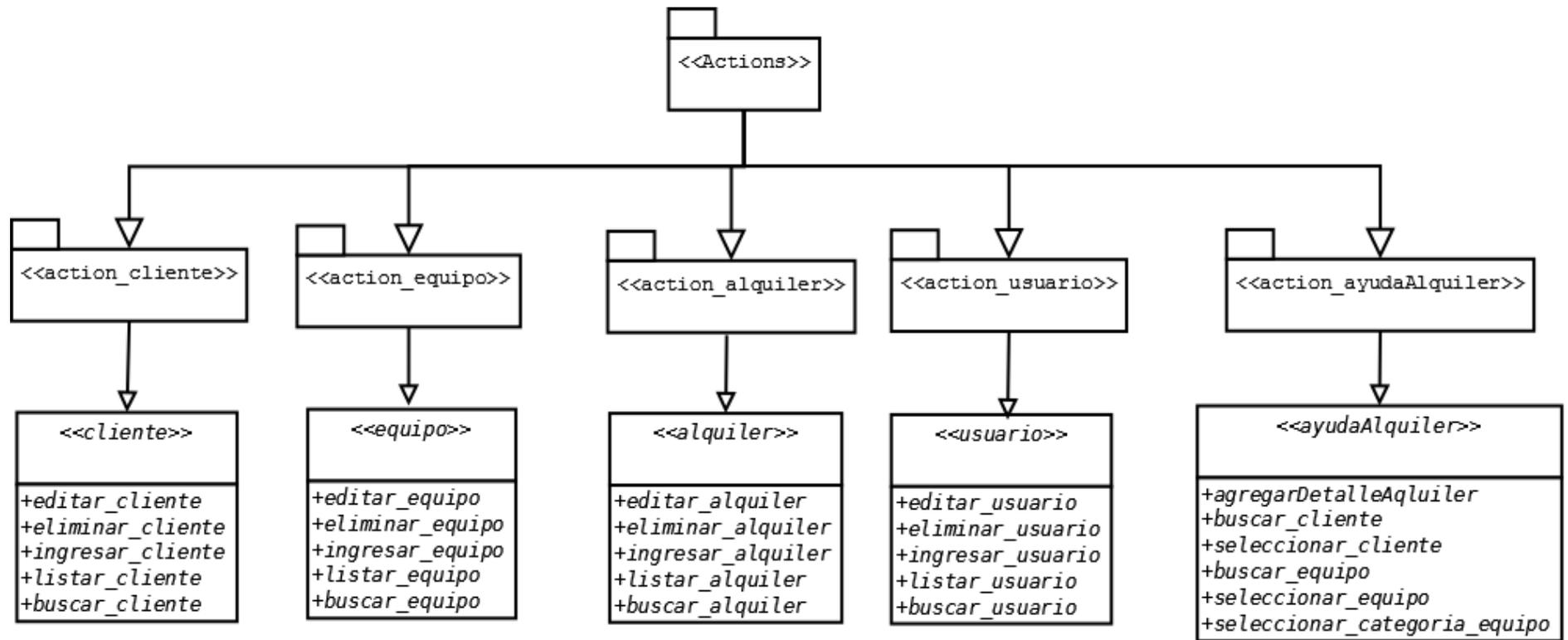


Ilustración 11 - detalle de la lógica del negocio del diagrama de arquitectura

Como podemos apreciar la lógica del negocio que se presenta en la figura anterior está basada en el CRUD para todas sus clases a excepción del Action ayuda alquiler el cual su lógica se basa en buscar y seleccionar un cliente al cual alquilarle un equipo, luego se busca y selecciona el equipo a alquilar, y finalmente se agregan detalles propios del alquiler como son las fechas correspondientes y cantidades.

A continuación en la Ilustración 12 se presenta el detalle completo de la persistencia del sistema, las clases DAO.

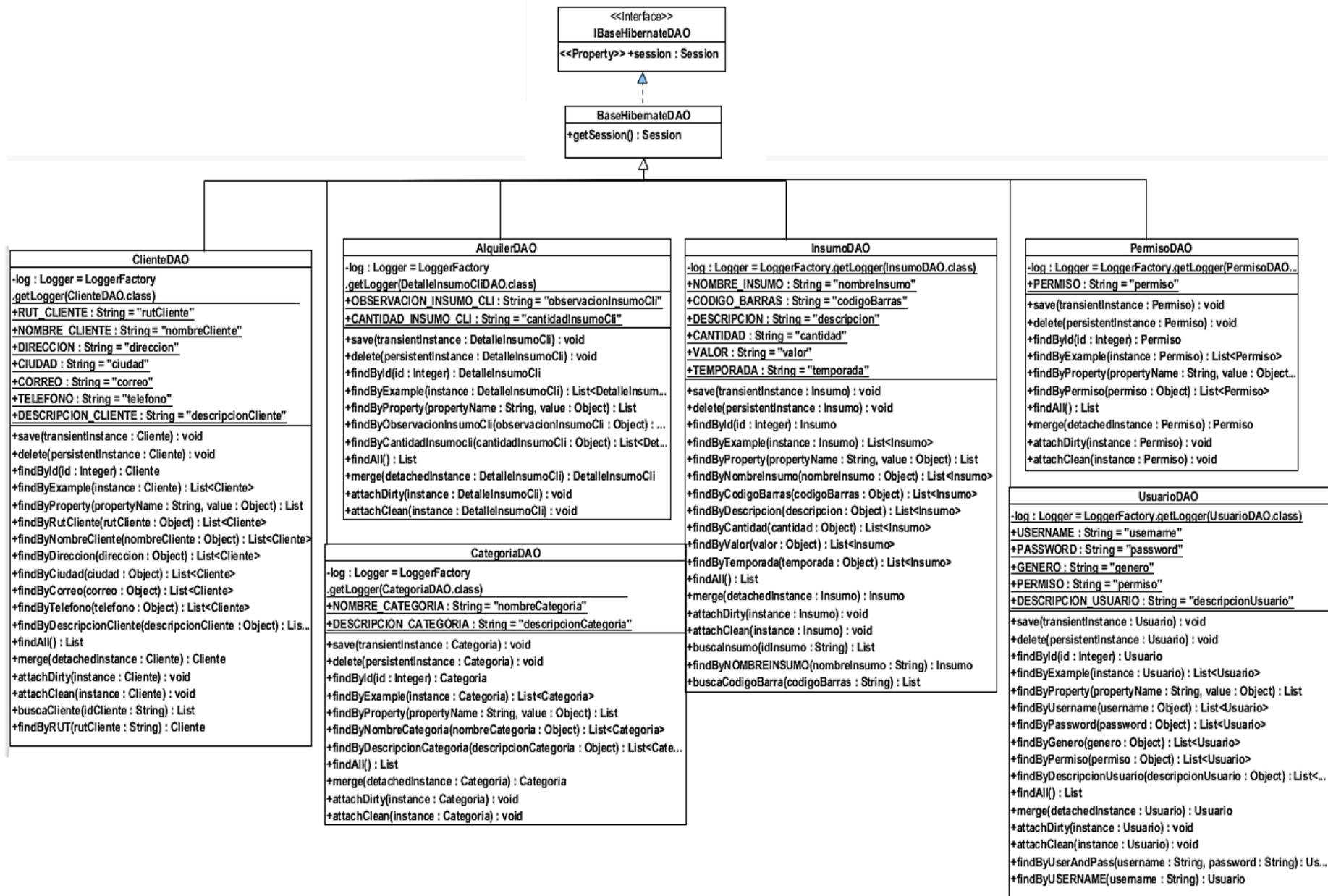


Ilustración 12 - detalle de la Persistencia del diagrama de arquitectura del sistema

5.6 Diagramas de Secuencia del Sistema

Los diagramas de secuencia de sistema son una representación que muestra el comportamiento del sistema en un determinado caso de uso, en este caso el comportamiento del sistema, se entenderá como una descripción de lo que hace el sistema sin explicar cómo lo hace. Los diagramas muestran los eventos generados por los actores externos, su orden y los eventos del sistema tratados como una caja negra. En el diagrama los eventos están ordenados cronológicamente hacia abajo lo que permite seguir los eventos que realizan los actores y la respuesta del sistema a esos eventos (De la Cruz García, 2006).

A continuación se muestra una descripción detallada de 4 diagramas de secuencia más significativos para el entendimiento de la solución.

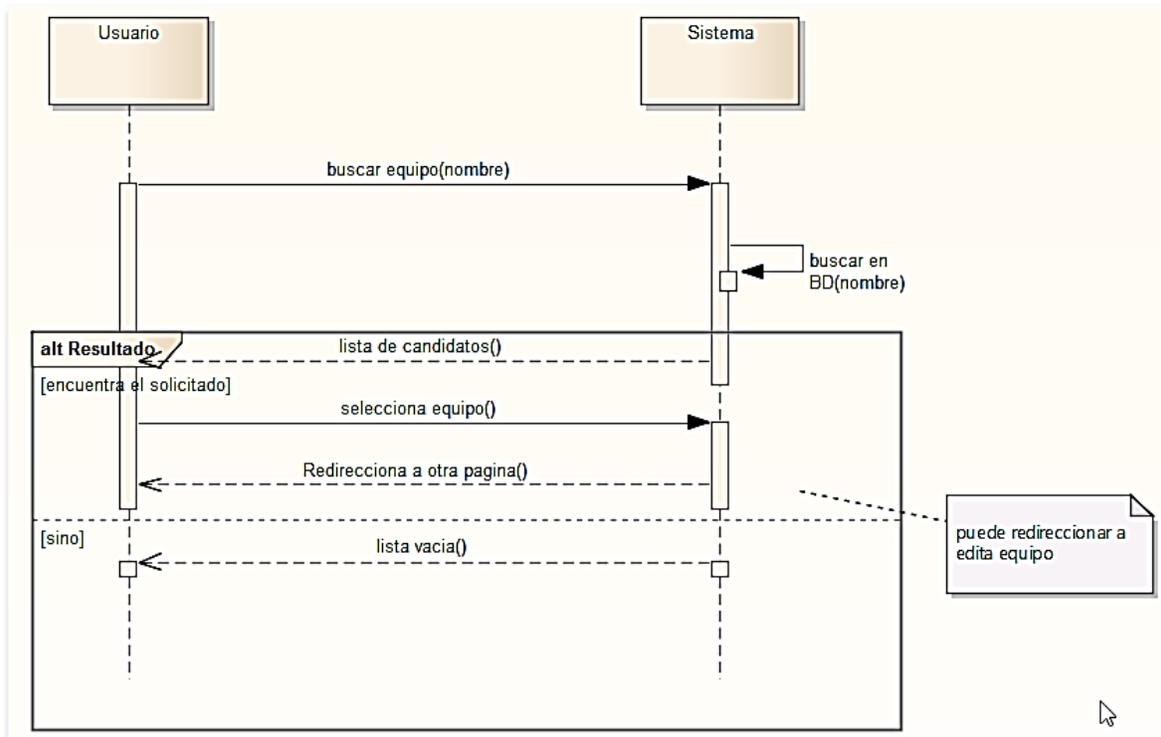


Ilustración 13- diagrama de secuencia Buscar equipo

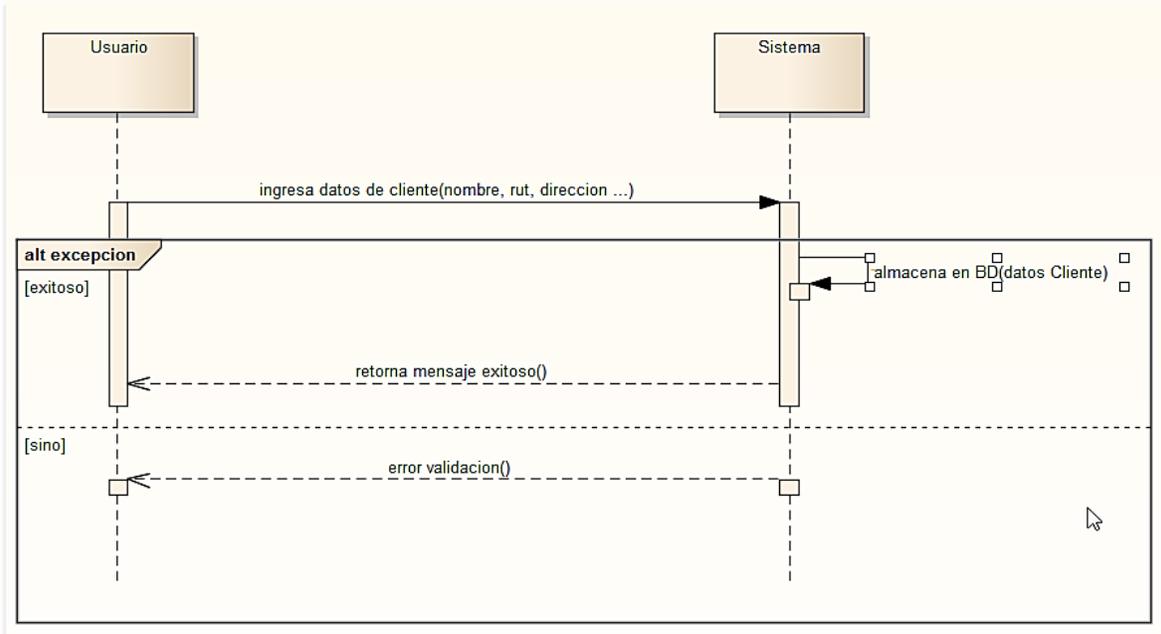


Ilustración 14 - diagrama de secuencia ingresar cliente

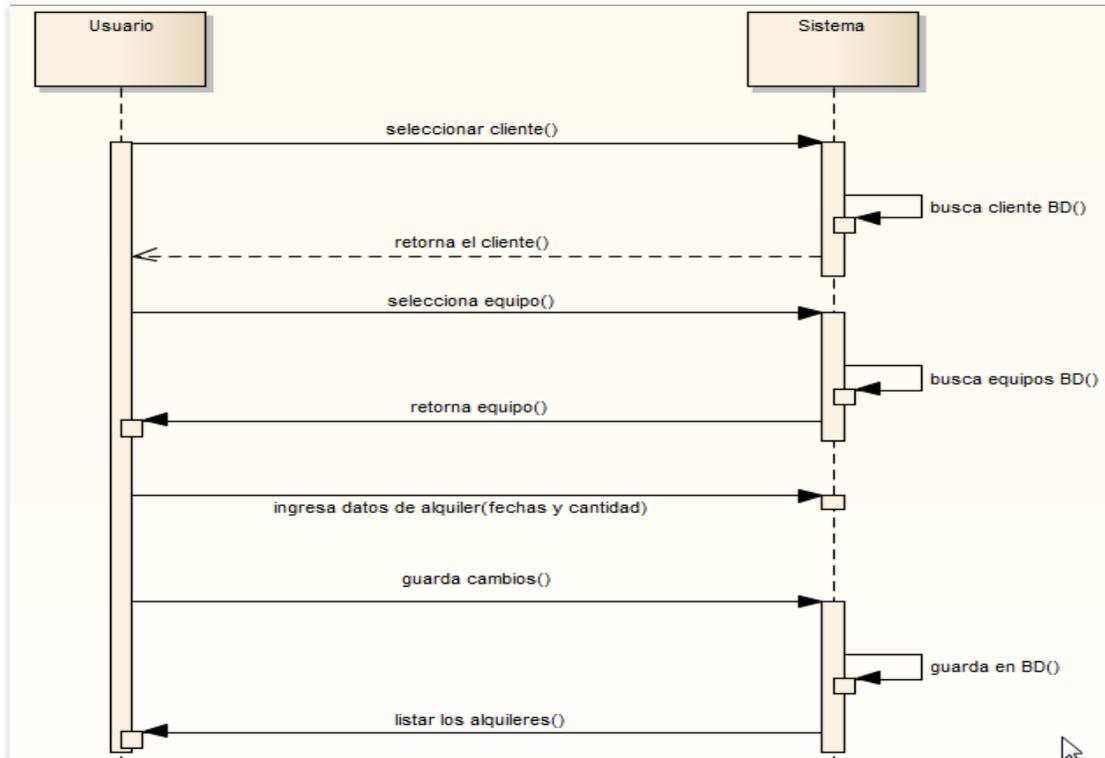


Ilustración 15 - diagrama de secuencia registrar Alquiler

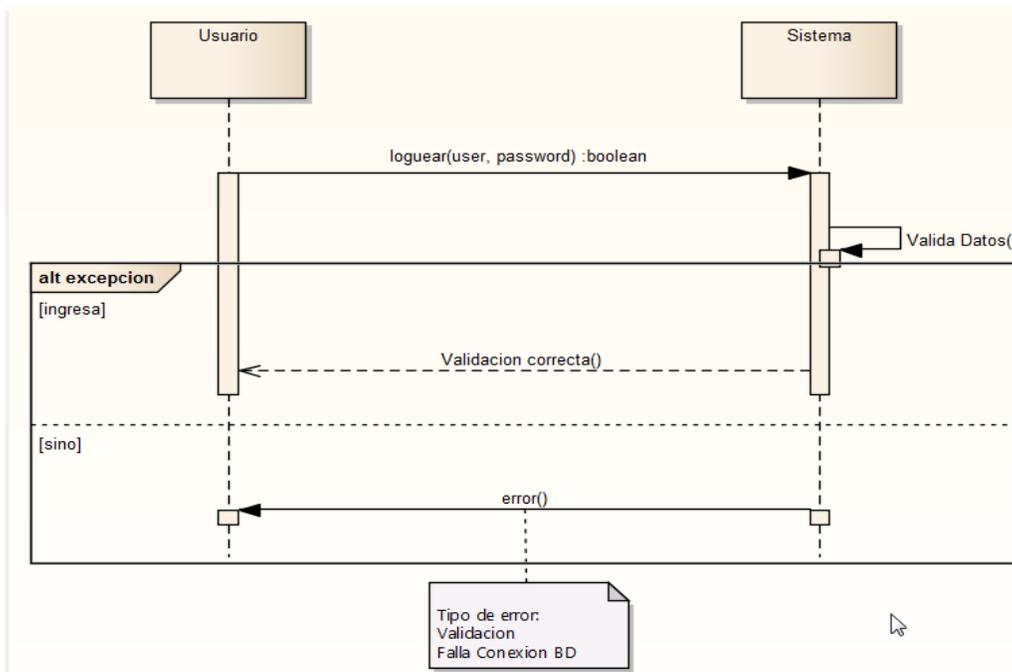


Ilustración 16 - diagrama de secuencia logear

5.7 Diagrama de colaboración del Sistema

Los diagramas de colaboración por su parte muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

En el sistema hay 4 clases, equipo, cliente, alquiler y usuario los cuales, cada uno realizan operaciones CRUD, y solo la clase equipo es compartida por ambos actores del sistema, tanto el bodeguero como administrador, y por ende es esta la que usaremos como ejemplo en los diagramas de colaboración para estos usuarios.

A continuación se presentan 5 diagramas de colaboración que hacen referencia a los CRUD (ingresar, listar, modificar, eliminar y buscar) que presenta el programa, los demás se omiten por su obvia semejanza.

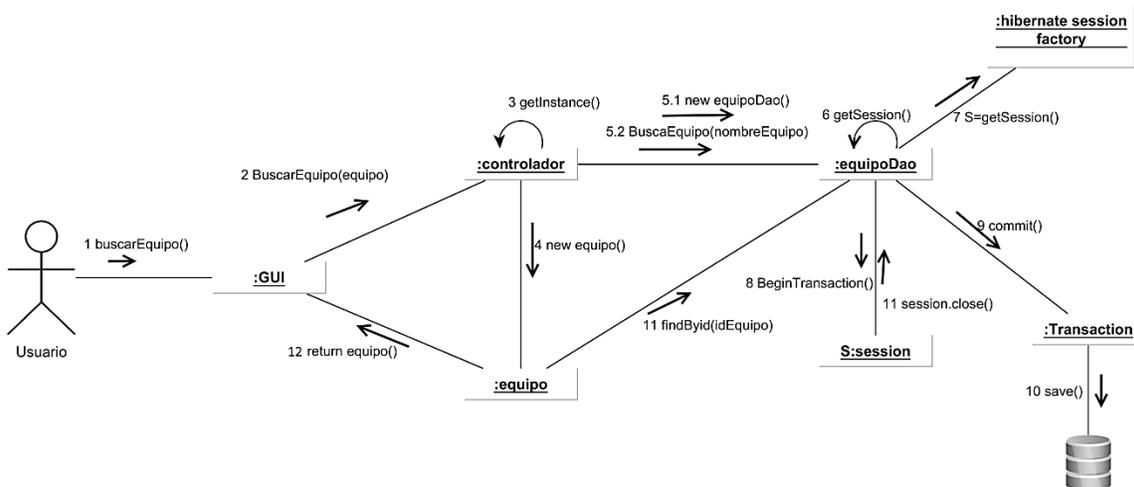


Ilustración 17 - Diagrama de Colaboración Buscar Equipo

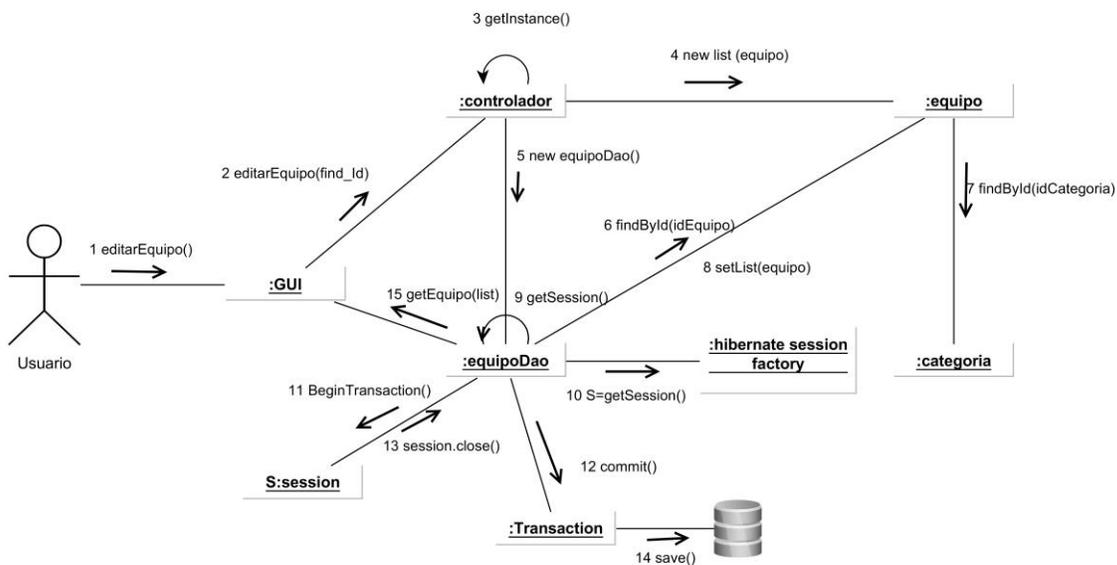


Ilustración 18 - Diagrama de Colaboración Edita Equipo

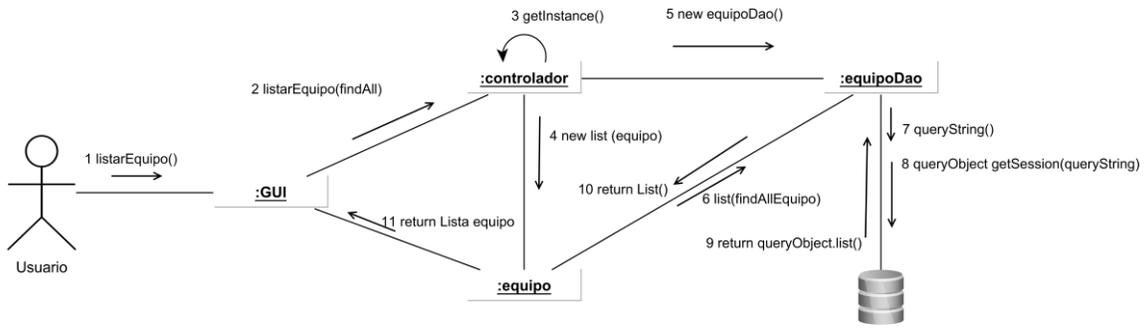


Ilustración 19 - Diagrama de Colaboración listar Equipo

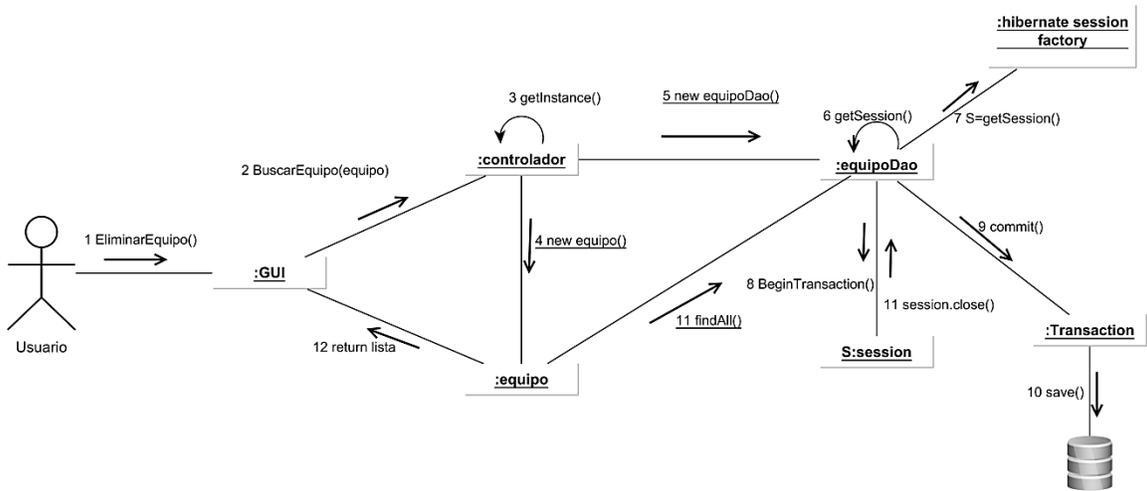


Ilustración 20 - Diagrama de Colaboración Elimina Equipo

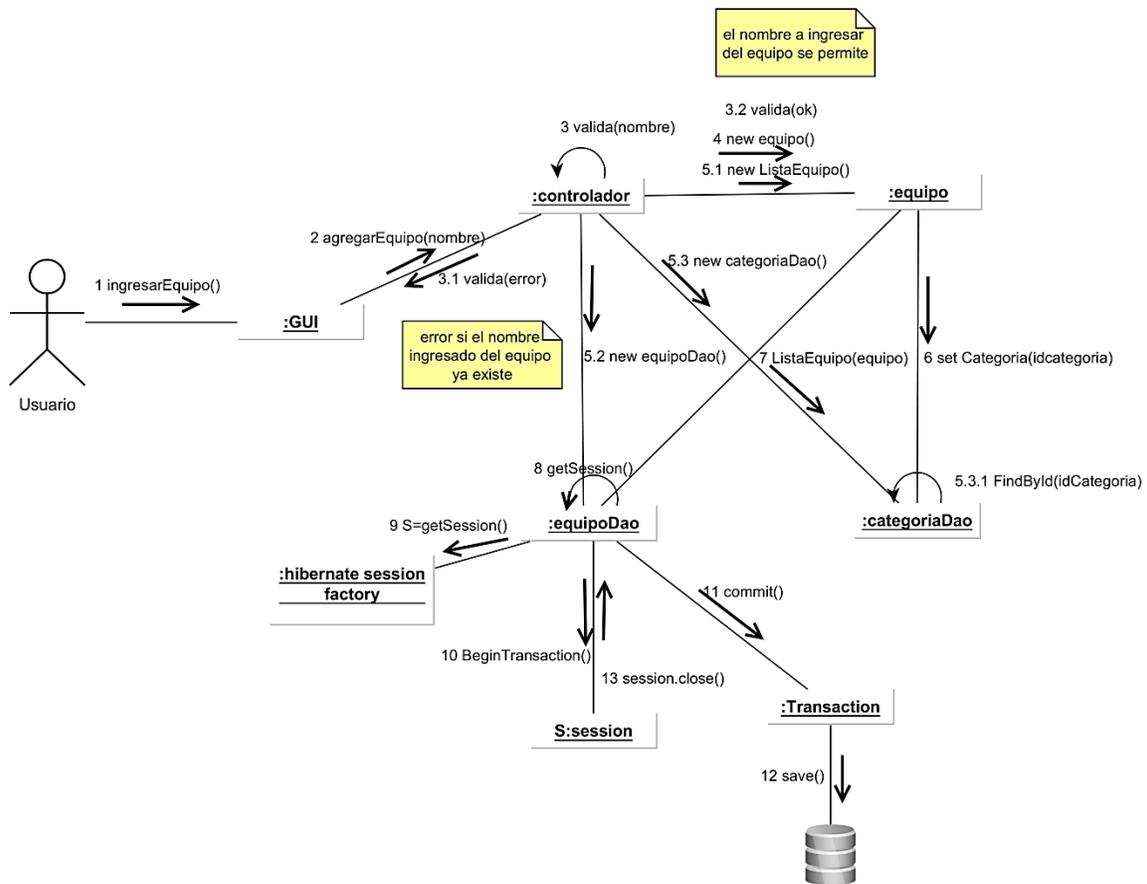


Ilustración 21 - Diagrama de Colaboración Ingresar Equipo

5.6 Ilustración Modelo Entidad-Relación

El Modelo Entidad Relación, fue introducido por Peter Chen en 1976 y se define como una herramienta de modelamiento de datos que describe las asociaciones que existen entre las diferentes categorías de datos dentro de un sistema de empresa o de información. (Elmasri, 2002)

A continuación se muestra el modelo entidad-relación.

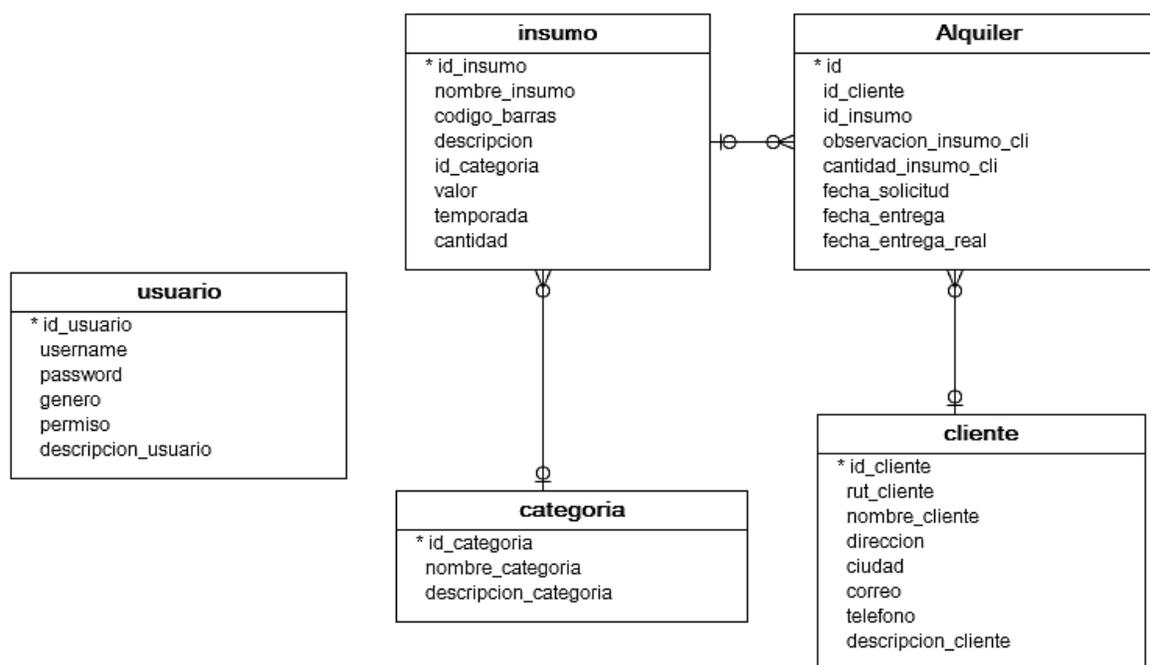


Ilustración 22 - modelo entidad relación

5.7 Modelo de Base de Datos

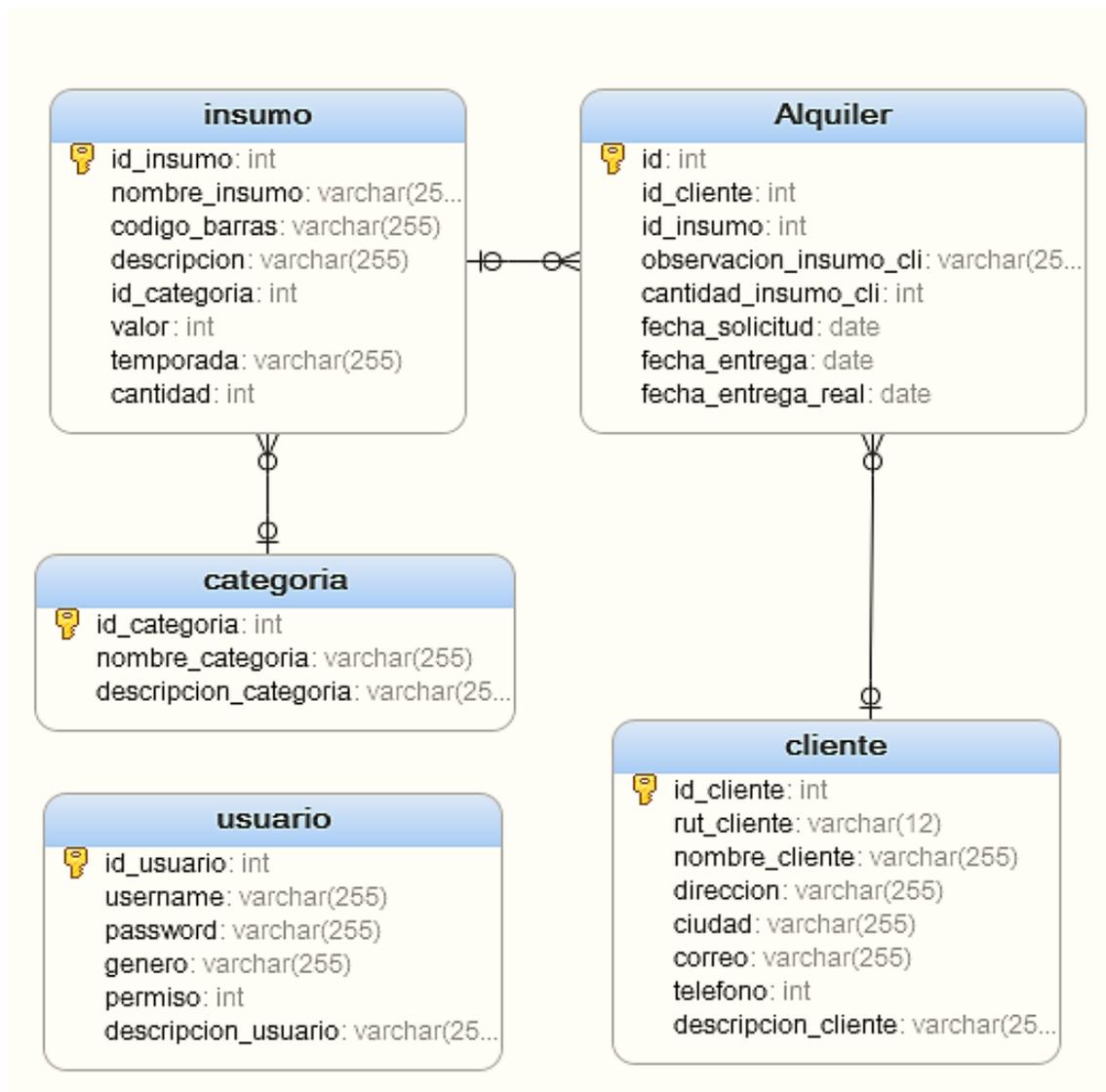


Ilustración 23 - modelo base de datos

5.8 Requerimientos técnicos para el desarrollo de la aplicación

El equipo que se utilizará para el desarrollo del sistema será un notebook HP 425 con procesador AMD Athlon II Dual-Core P320 cuya velocidad del procesador es de 2.1 GHz, una memoria RAM de 2 GB DDR3, memoria Caché 1MB L2, un disco duro de 320 Gb, tamaño pantalla 14 pulgadas, una tarjeta de red de 10/100 Mbps, y Wi-Fi 802.11 b/g/n.

5.9 Salidas

A continuación se mostrara las salidas que genera el software como reportes.

Detalle de Equipos



05/06/2015

RESUMEN DETALLE EQUIPOS

Nombre	Código Barras	Descripción	valor	temporada	Cantidad	Categoría
tablas sky columbia	432125534534		12000	temporada 1	9	tablas
casco	2345		5000	temporada 1	8	cascos snow
guantes	321	calentitos	7000	temporada 1	7	accesorios
pantalón termico	98765	calentitos	8000	temporada 1	3	accesorios

Página1 De 1

Ilustración 24 - reporte de equipos

Detalle De Clientes



05/06/2015

RESUMEN DETALLE CLIENTE

Rut	Nombre	Dirección	Correo	Teléfono	Descripción
16.784.781-1	victor	pinto	felpilla@gmail.com	62969064	buen cliente
182146641	andrea	trancas	andy@gmail.com	56789597	buen cliente
167847811	nana	pinto	k@g.com	2345678	buen cliente
16784781-1	andre	pintos	felpilla@gmail.com	62969064	buen cliente
15.876.876-3	maca	into	maca@gmailcl	54545454	buen cliente
19.946.745-k	oscar	pinto	kakumenes@gmail.com	55555555	buen cliente

Lista detalle Alquiler



05/06/2015

rut: 16.784.781-1

RESUMEN DETALLE DE OBRAS POR ESTADO

equipo	descripcion	categoria	cantidad	valor c/u	total	detalle	fecha	fecha devolucion
tablas sky		tablas	6	12000	72000		04-jun-15	05-jun-2015
casco		cascos snow	11	5000	55000		04-jun-15	05-jun-2015

total: 127000

Página1 De 1

Ilustración 26 - reporte alquileres realizados por un cliente

Contrato de Arriendo



REUNIDOS:

DE UNA PARTE, Don(a) Macarena Aedo, con R.U.T 15.876.876-3 mayor de edad, con domicilio en la Calle Patricio Lynch S/N interior Pinto, administradora de Dream's Sky, denominado(a) desde ahora ARRENDADOR

DE OTRA PARTE, Don(a): oscar asi mismo, mayor de edad, con R.U.T: 19.946.745-k quien tiene su domicilio en pinto Los Angeles sera denominado des de ahora como ARRENDATARIO.

El ARRENDADOR y el ARRENDATARIO, en adelante, podrán ser denominadas, individualmente, "la Parte" y, conjuntamente, "las Partes", reconociéndose mutuamente capacidad jurídica y de obrar suficiente para la celebración del presente Contrato.

Se reconocen los reunidos con la capacidad civil necesaria para contratar y obligarse y, en especial, para otorgar el presente documento; y a tal efecto, actuando ambos en su propio nombre y derecho, de su concorde voluntad, dicen y pactan cuanto a continuación se relaciona el contrato.

EXPONEN

PRIMERO: Que el ARRENDATARIO está interesado en el arrendamiento de determinados equipos de montaña.

SEGUNDO: Que el ARRENDADOR es una empresa dedicada en la prestación de servicios de arrendamiento de equipos de montaña.

TERCERO: Que las Partes están interesadas en celebrar un contrato de arriendo de equipos de montaña en virtud del cual el ARRENDADOR arriende al ARRENDATARIO los equipos que se relacionan en este contrato. En el anexo "lista de equipamiento".

5.10 Interfaz de la Aplicación

En la ilustración 28 se presenta el diseño de la interfaz del sistema, indicando cada uno de los contenedores utilizados y su respectiva descripción.

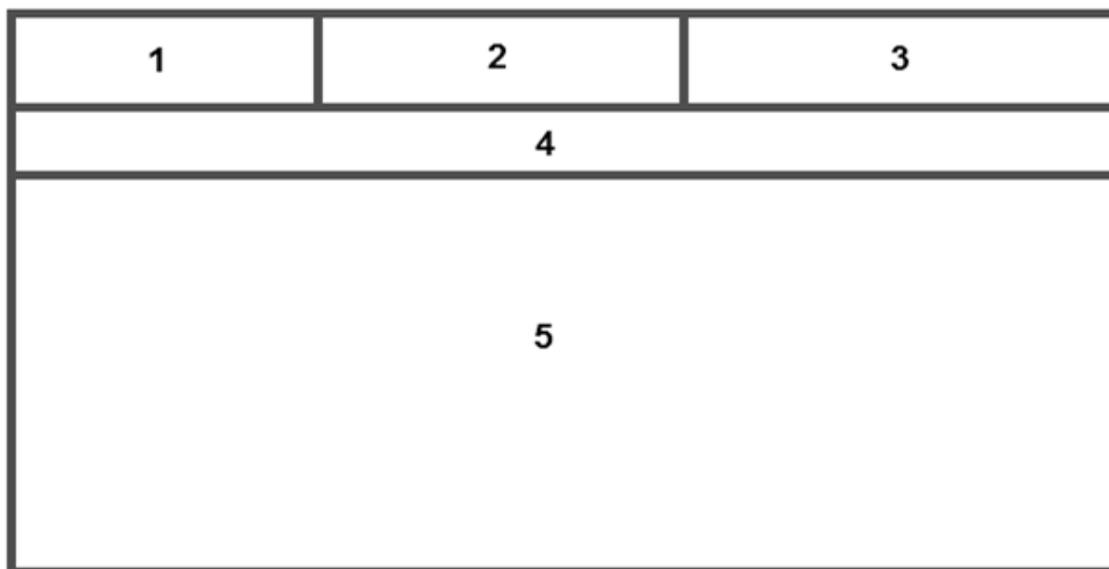


Ilustración 28 - diseño interfaz

Contenedores:

1. Área reservada para el logo o imagen corporativa de la empresa.
2. Espacio sin ocupar, por diseño de la página.
3. Menú propio de sesión. Incluye información como: Nombre de usuario y perfil. Además, agrega la opción de cerrar sesión.
4. Menú de navegación principal. Incluye los elementos propios de navegación del sistema, siendo estos elementos variables según el perfil y las funciones de los usuarios.
5. Despliegue de formularios e información requerida.

A continuación se mostrara algunas capturas Para ilustrar el aspecto que tendrá el sistema, en cuanto a los colores y las imágenes de fondo a utilizar, además de especificar aún más el diseño de la interfaz y las funciones que se puedan visualizar.

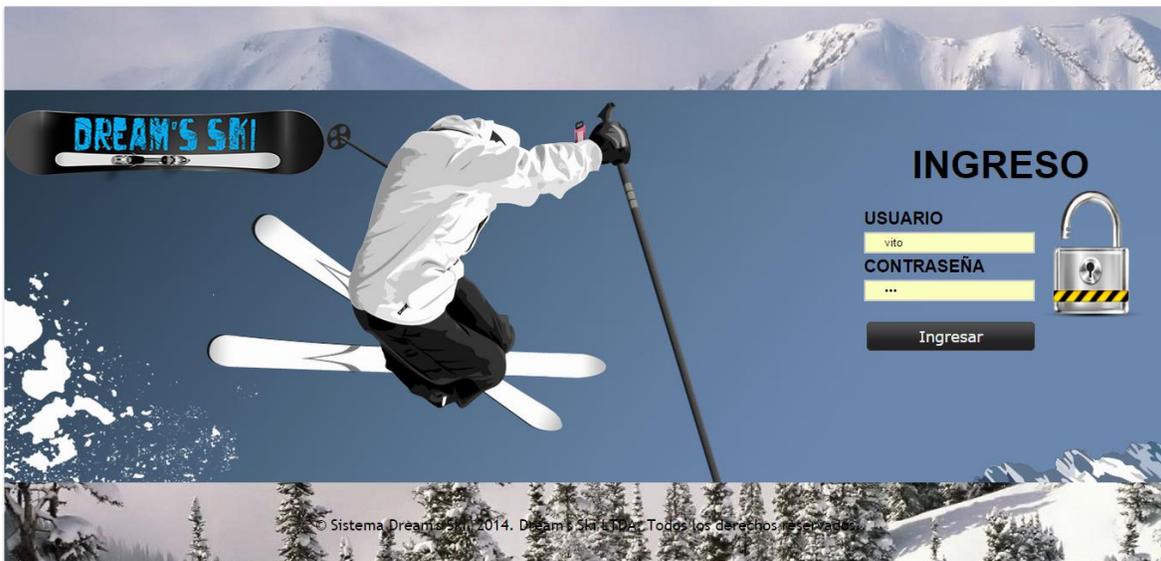


Ilustración 29 - pantalla de inicio para ingresar



Ilustración 30 - pantalla de bienvenida para administrador

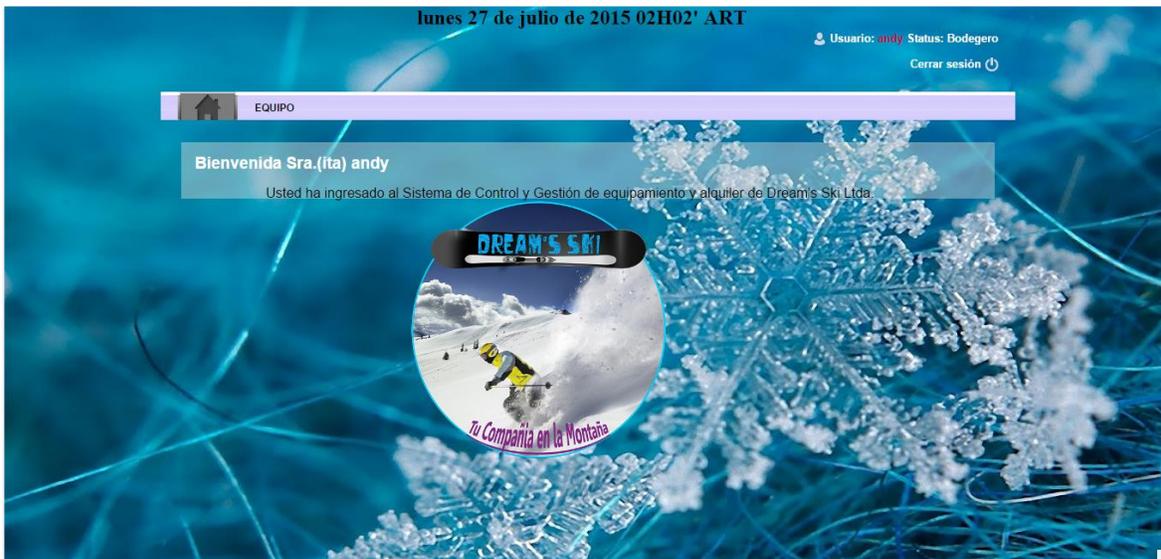


Ilustración 31 - pantalla de bienvenida para bodeguero

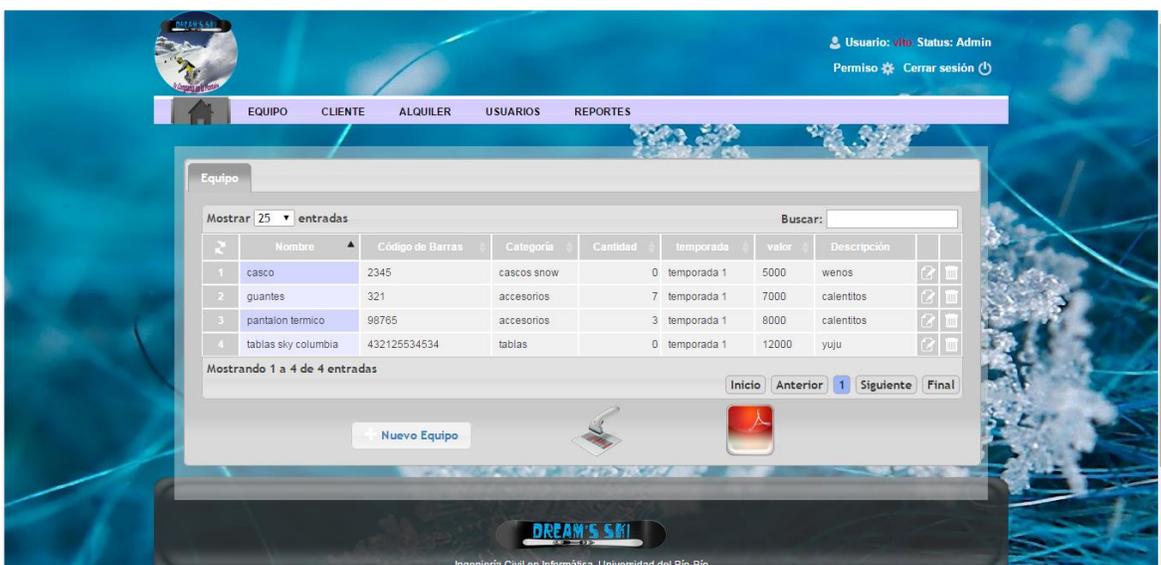


Ilustración 32 - pantalla listado de equipo

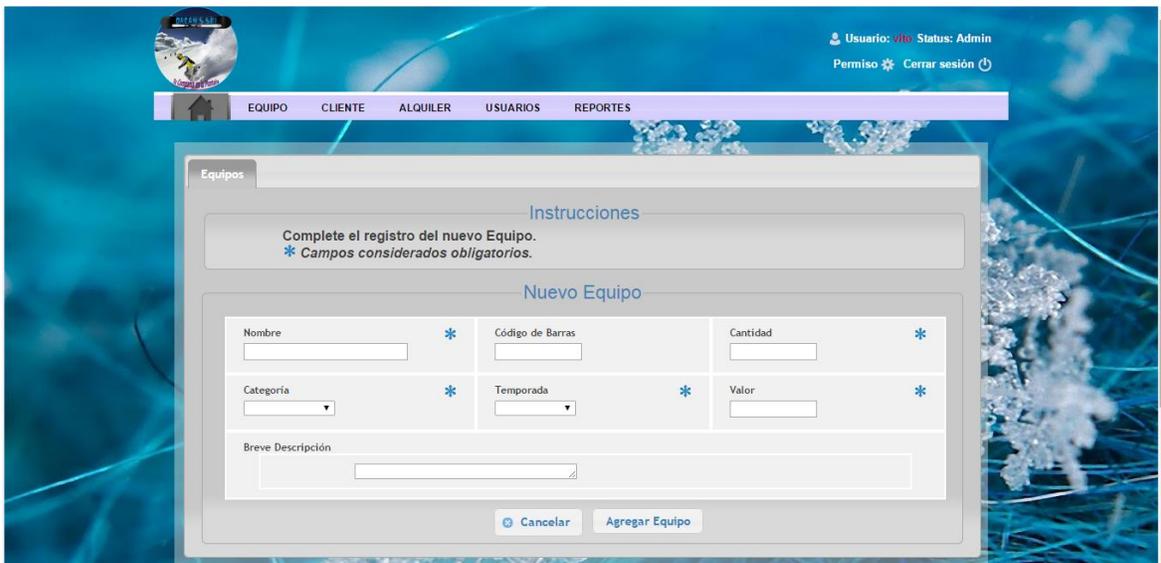


Ilustración 33 - pantalla ingresar equipo

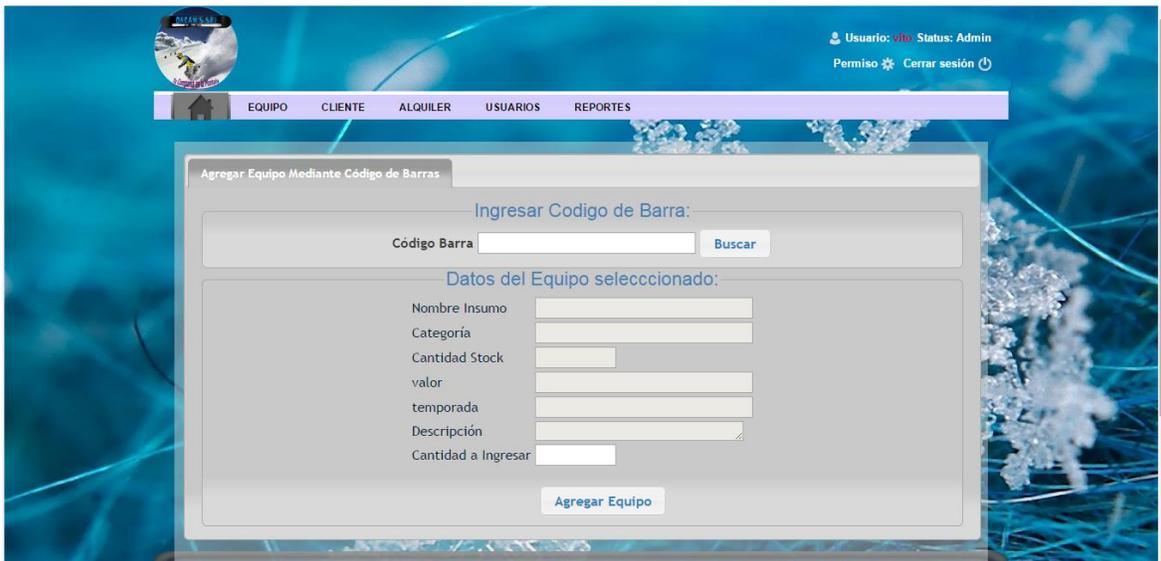


Ilustración 34 - pantalla ingreso de equipo con código de barras

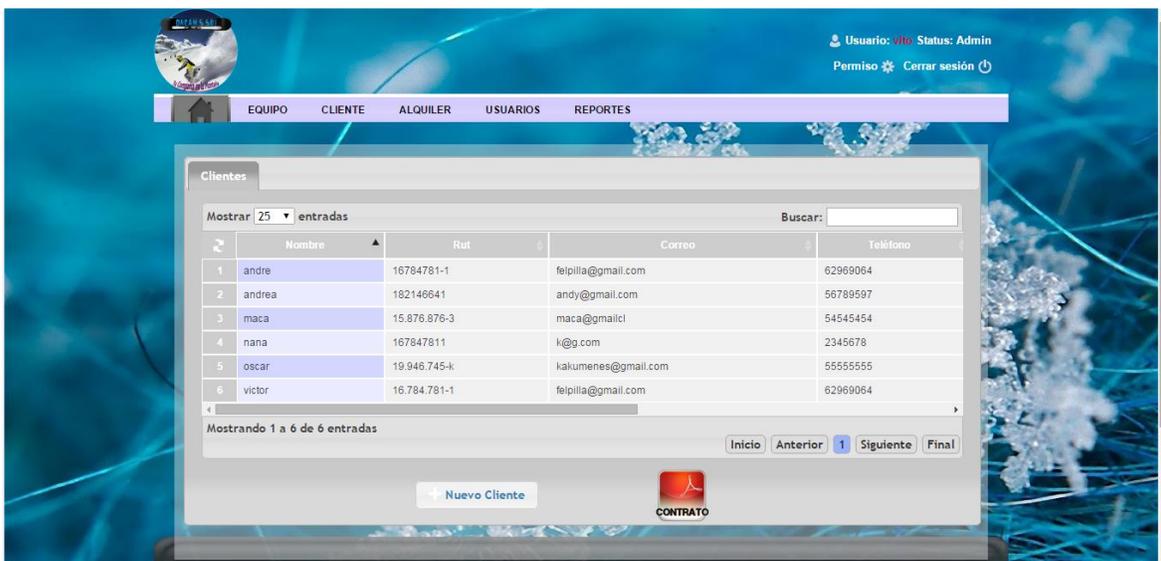


Ilustración 35 - pantalla listado de clientes

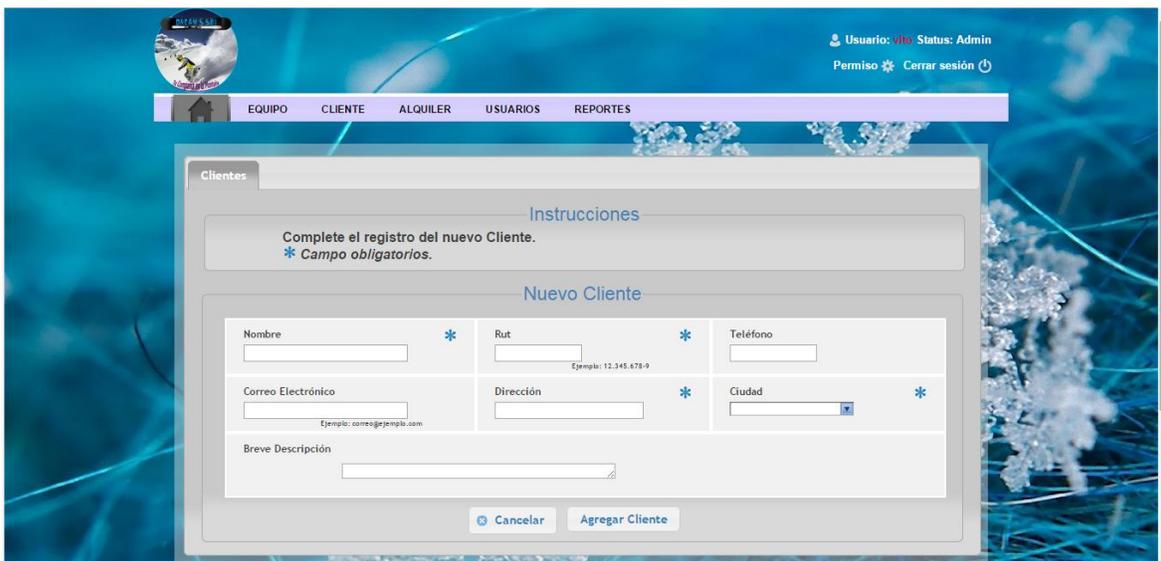


Ilustración 36 - pantalla agregar cliente

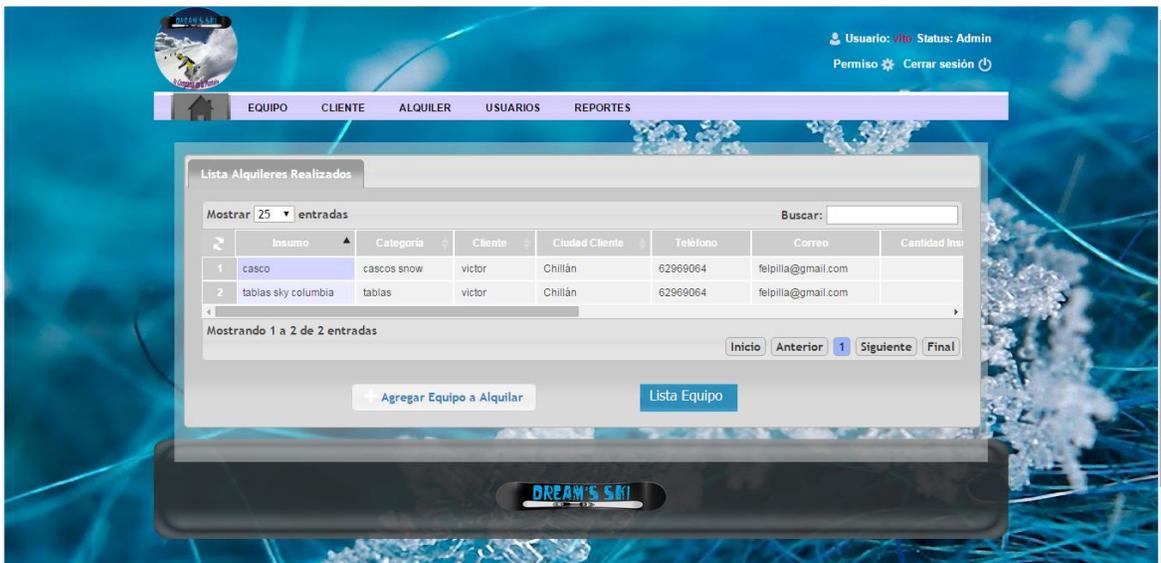


Ilustración 37 - pantalla listar alquileres

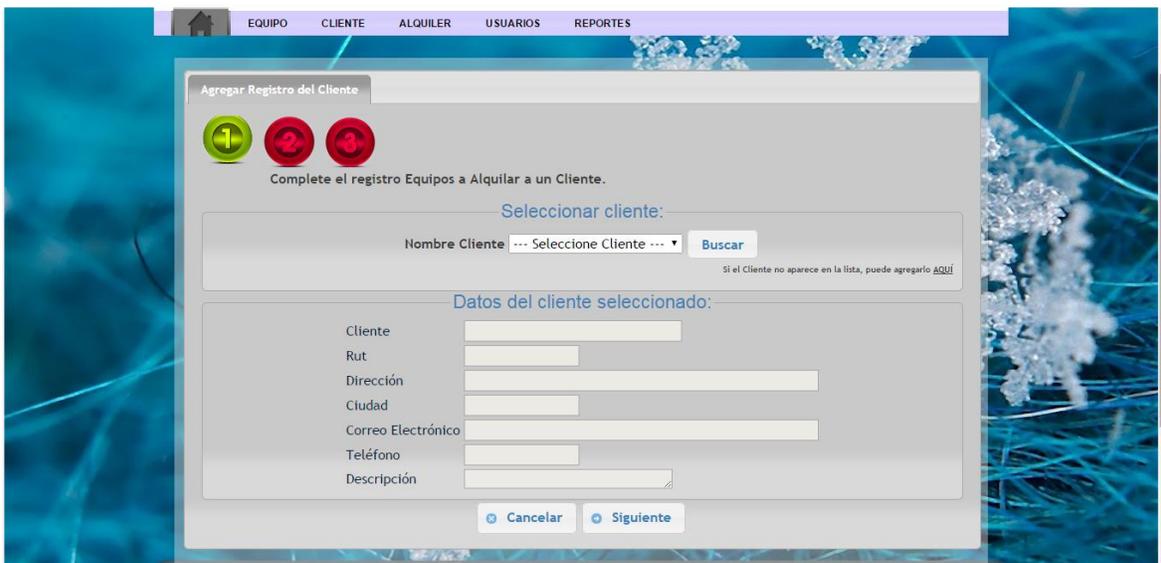


Ilustración 38 - pantalla generar alquiler, seleccionar el cliente

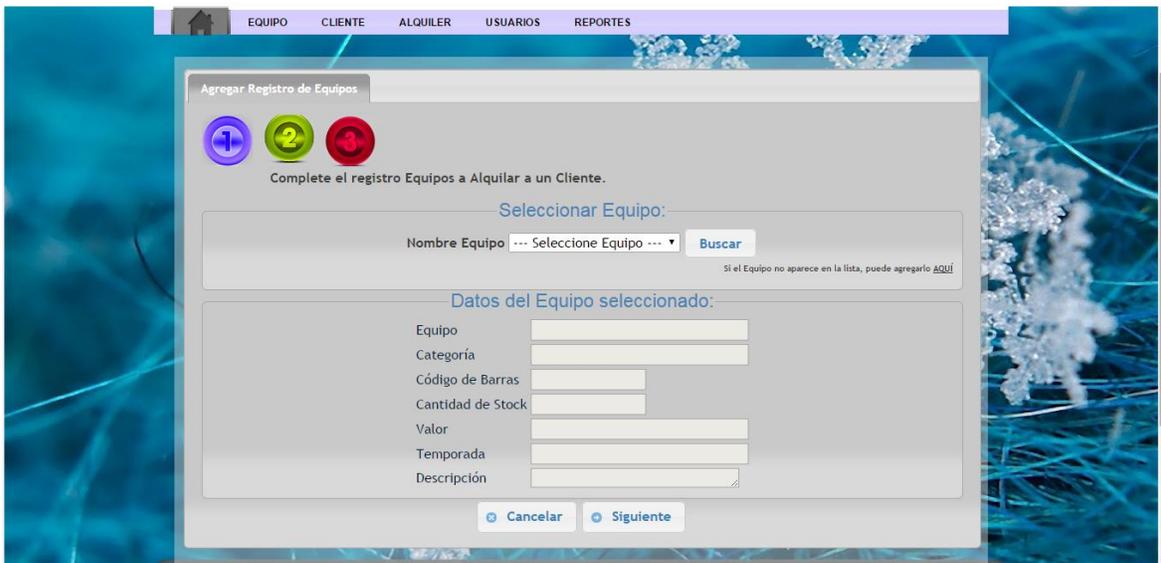


Ilustración 39 - pantalla generar alquiler, seleccionar el equipo



Ilustración 40 - pantalla generar alquiler, completar datos de alquiler

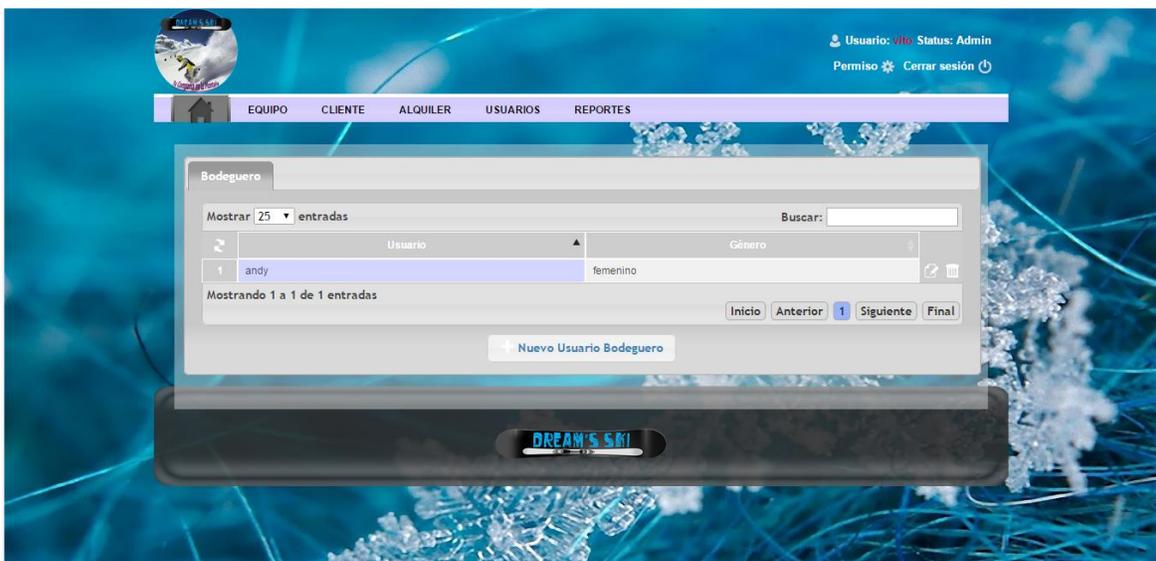


Ilustración 41 - pantalla usuarios bodegueros del sistema

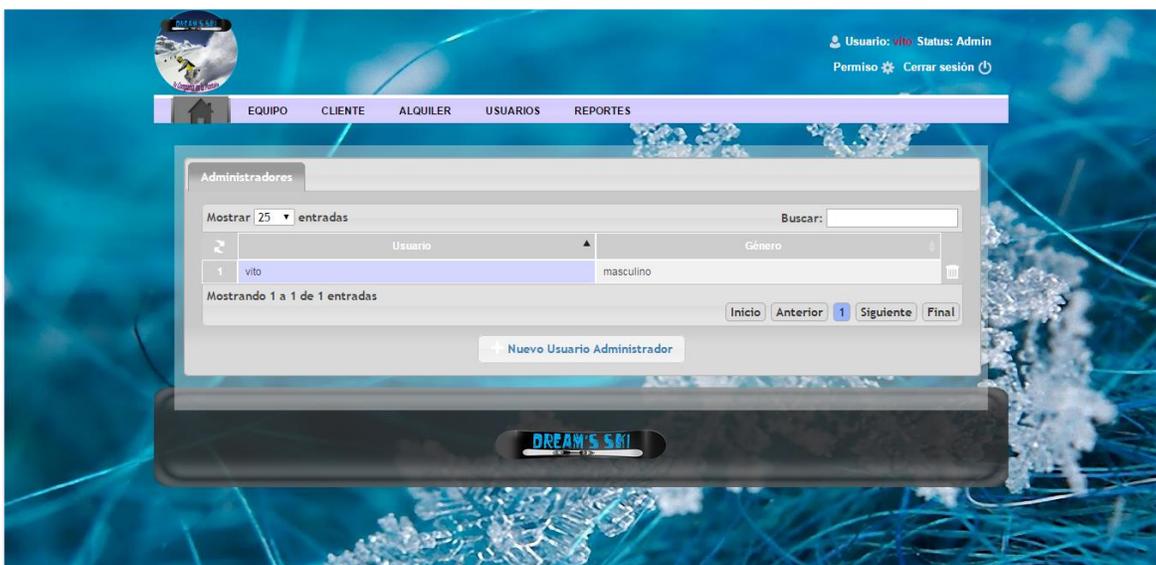


Ilustración 42 - pantalla usuarios administradores del sistema

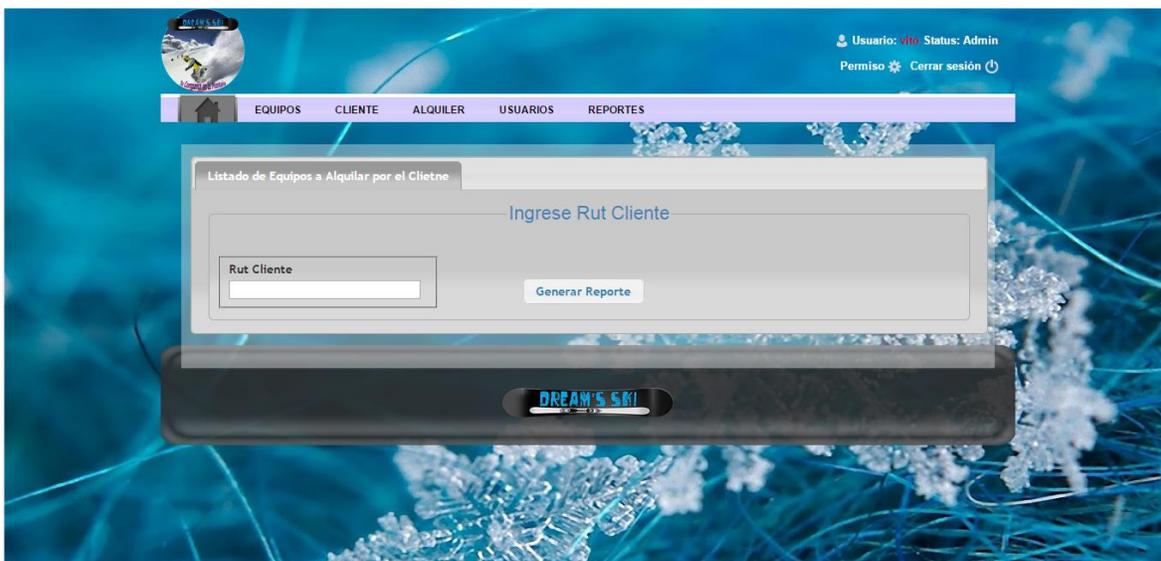


Ilustración 43 - pantalla lista de equipos de un cliente



Ilustración 44 - pantalla seleccionar reportes



CAPÍTULO 6

PRUEBAS

6.1 Pruebas de Caja Negra

Este tipo de prueba consiste en ingresar datos de prueba correctos, erróneos e inexistentes con el objetivo de evaluar el comportamiento del sistema frente a estos ambientes, que puede arrojar resultados correctos o erróneos.

Para realizar estas pruebas, se toma cada uno de los requisitos del sistema (mediante sus Casos de Uso) y se asignan valores de prueba.

Para llevar a cabo este tipo de pruebas se utilizó la siguiente metodología:

- Definir el propósito de la prueba.
- Definir los prerequisites para poder acceder a la instancia que se probará.
- Definir claramente los datos con los cuales se llevará a cabo la prueba.
- Definir los pasos de cómo se llevó a cabo la prueba.
- Definir los resultados que se esperan previo a la realización de la prueba.
- Determinar cuáles fueron los resultados obtenidos con el desarrollo.
- Finalmente, evaluar la prueba describiendo si se detectaron errores y las medidas a adoptar para la corrección.

Las Pruebas de Caja Negra se realizaron a todos los casos de uso, los que muestran a continuación son los más relevantes ya que los demás son similares:

- Autenticar Usuario.
- Ingresar equipo
- Ingresar alquiler
- Devolver alquiler

6.1.1 Prueba CU1, Autenticar Usuario

Propósito	Probar el ingreso de un usuario registrado al sistema.
Prerrequisitos	Ninguno.
Datos de prueba	Rut = { 1678587811, 3.333.333.333; (vacío) } Clave = { calve; 123; 1234567890 ; (vacío) }
Pasos	1. Ingresar Rut y clave. 2. Hacer clic en Ingresar.
Resultados esperados	-Si todos los datos son ingresados correctamente, se iniciará el sistema mostrando las opciones disponibles al perfil del usuario identificado. - Si los datos ingresados son incorrectos o en blanco, el sistema debería mostrar el siguiente mensaje: “Los datos ingresados no son correctos”.
Resultados obtenidos	-Cuando los datos fueron correctos, el sistema se inició mostrando el menú disponible al usuario encontrado. -Cuando se hizo clic en ingresar y los datos ingresados eran incorrectos o en blanco, el sistema mostró el siguiente mensaje “Los datos ingresados no son correctos”.
Evaluación de la prueba	No se encontraron errores durante esta prueba.

Tabla 41 - Autenticar Usuario.

6.1.2 Prueba CU 7, Ingresar Cliente

Propósito	Ver que se ingrese correctamente los datos y se reflejen en la BD.
Prerrequisitos	El usuario debe estar autenticado como Administrador.
Datos de prueba	La información personal de personas ficticias alterando algunos parámetros.
Pasos	<ol style="list-style-type: none"> 1. Seleccionar en el menú equipo --> agregar nuevo cliente 2. Completar los datos requeridos. 3. Guardar los cambios 4. Verificar en la lista de equipo.
Resultados esperados	<p>-si los datos validos son ingresados correctamente, y respetando los campos obligatorios, el ingreso de un nuevo equipo se realizara con éxito.</p> <p>- en el caso de que no lo logre verificar los mensajes de error que muestran, ya que todos los datos están validados para que se ingresen con la forma y tipo que corresponden.</p>
Resultados obtenidos	<p>-cuando se ingresó un equipo respetando todos obligatorios y validaciones correspondientes, la acción fue exitosa.</p> <p>-cuando no ingreso envía mensajes por cada error de validación de dato ingresado, por ejemplo, un Rut no correspondiente, o una cantidad que contenga una letra.</p>
Evaluación de la prueba	Los errores encontrados han sido corregidos y probados satisfactoriamente.

Tabla 42 - Ingresar Cliente

6.1.3 Prueba CU 15, registrar alquiler

Propósito	Que el alquiler sea registrado y que los equipos alquilados se descuenten de la bodega.
Prerrequisitos	El usuario debe estar autenticado como Administrador.
Datos de prueba	Datos de clientes y equipos registrados en el sistema previamente.
Pasos	<ol style="list-style-type: none"> 1. Seleccionar Menú → registrar alquiler 2. buscar equipo a alquilar → seleccionar el equipo 3. buscar cliente que desea alquilar → seleccionar el cliente 4. ingresar las fechas de solicitud y devolución con la cantidad de equipos a alquilar y observaciones. 5. registrar los datos
Resultados esperados	<ul style="list-style-type: none"> - Si el alquiler se registra, se debe verificar en la lista de equipos el campo cantidad disponible y debería reflejarse una disminución en relación a la cantidad alquilada de ese equipo
Resultados obtenidos	-resultados satisfactorios
Evaluación de la prueba	No se encontraron errores durante esta prueba.

Tabla 43 - registrar alquiler

6.1.4 Prueba CU 17, Devolver

Propósito	Que se refleje en la lista de equipo que se restauró la cantidad a la bodega de los quipos devueltos
Prerrequisitos	El usuario debe estar autenticado como Administrados
Datos de prueba	Los alquileres previamente registrados en el sistema
Pasos	<ol style="list-style-type: none"> 1. Seleccionar menú registro de alquiler → ir a la lista de alquileres 2. seleccionar la casilla devolver. 3. cuando se abra la ventana de devolución ingresar la cantidad a devolver y la fecha real de devolución. 4. guardar los cambios.
Resultados esperados	<ul style="list-style-type: none"> -que en la lista de quipos, el equipo devuelto vuelva a tener el stock en bodega. - si no se agrega la cantidad devuelta, revisar la lógica de la función devolución.
Resultados obtenidos	-al momento de la devolución el sistema agrego la cantidad de equipos devueltos al stock de bodega
Evaluación de la prueba	No se encontraron errores durante esta prueba.

Tabla 44 - Devolver

CAPÍTULO 7

CONCLUSIONES Y

REFERENCIAS

7.1

CONCLUSIONES

En el presente informe se detallaron los procesos de planificación, diseño y desarrollo del proyecto el cual tiene por objetivo principal la automatización de las labores diarias de los usuarios, controlando y gestionando las operaciones primordiales de la empresa.

Al concluir el proceso de desarrollo de la solución planteada, se puede destacar el cambio que ocurre con la implantación de un sistema hecho a la medida de las necesidades de una organización, que resuelve y mejora en gran medida los procesos en que se ve inserto. Se puede ver que con este sistema se tiene un completo registro del proceso de alquiler, además de obtener estadísticas fiables y oportunas de las operaciones, para poder tomar decisiones con un respaldo concreto.

Durante el proceso de gestación de la solución se debió tratar algunos cambios que gracias a la metodología iterativa incremental se pueden sostener y ser un aporte al perfeccionamiento de la solución.

Con respecto al lenguaje de programación utilizado, se puede rescatar la facilidad en la revisión del código fuente que presenta JAVA, lo cual permite observar el sistema en tiempo de ejecución, con lo cual se hizo más fácil la tarea de detección de errores. La gran comunidad de desarrolladores son un gran apoyo al momento de resolver dudas, y la documentación existente es abundante y clara.

7.2 TRABAJOS FUTUROS

Como trabajos futuros se puede implementar un módulo de finanzas y contabilidad en la empresa, que interactúe directamente con el sistema, para llevar el control de todas las cuentas con respecto a sus ingresos, arriendos, bienes, entre otros, que maneja.

7.3 Bibliografía

1. *La Caja de Pandora*. (14 de Diciembre de 2007). Recuperado el 8 de 06 de 2015, de <http://jhodrickgg.wordpress.com/2007/12/14/programacion-orientada-a-objetos/>
2. Álvarez, M. Á. (2001). *Introducción a Javascript*. Recuperado el 11 de 6 de 2015, de <http://www.desarrolloweb.com/articulos/490.php>
3. Álvarez, M. Á. (2002). *Qué es JSP*. Recuperado el 12 de 5 de 2015, de <http://www.desarrolloweb.com/articulos/831.php>
4. Álvarez, M. Á. (2010). *desarrolloweb.com*. Recuperado el 2 de 6 de 2015, de <http://www.desarrolloweb.com/articulos/497.php>
5. Anónimo. (2009). *Struts 2*. Recuperado el 2 de 4 de 2015, de <http://mundogeek.net/archivos/2009/02/08/struts-2/>
6. Anónimo. (2010). *Actualidad JQuery*. Recuperado el 15 de 5 de 2015, de <http://www.actualidadjquery.es/acerca-de/>
7. Anónimo. (2010). *Ajax: Asynchronous JavaScript And XML*. Recuperado el 5 de 5 de 2015, de <http://tipsgeeks.blogspot.com/2009/06/ajax-asynchronous-javascript-and-xml.html>
8. Bravo. (16 de 10 de 2008). *Universidad de Salamanca – Departamento de Informática y Automática*. Recuperado el 12 de 6 de 2015, de <http://ocw.usal.es/enseñanzas-tecnicas/ingenieria-del-software/contenidos/Tema6-DOO-1pp.pdf>
9. Cornejo, J. E. (12 de 06 de 2011). *DocIRS*. Recuperado el 15 de 10 de 2012, de <http://www.docirs.cl/uml.htm>
10. De la Cruz García, M. T. (2006). *Proyecto Fin de Carrera: Sistema de Gestión con Replicación de Datos*. Recuperado el 25 de 6 de 2015, de <http://www.iit.upcomillas.es/pfc/resumenes/450a66346db72.pdf>
11. Domotica.Net. (2009). *domotica.us*. Recuperado el 18 de 6 de 2015, de <http://www.domotica.us/MySQL>
12. Elmasri, R. (2002). *Fundamentos de sistemas de bases de datos*. Madrid: Addison-Wesley.
13. Fajardo, J. U. (2010). Recuperado el 15 de 05 de 2015, de <http://www.slideshare.net/gugarte/bpmn-estandar-para-modelamiento-de-procesos-presentation>
14. García, J. (27 de 05 de 2005). *IngenieroSoftware*. Recuperado el 12 de 06 de 2015, de <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
15. Herrera, C. (29 de 04 de 2005). *Adictos Al Trabajo*. Recuperado el 15 de 6 de 2015, de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>

16. Herrera, C. (29 de 04 de 2005). *Java con IReport*. Recuperado el 12 de 4 de 2015, de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>
17. ITEM. (2010). *ITEM: Innovación Tecnológica Empresarial*. Recuperado el 28 de 5 de 2015, de <http://www.item.com.ve/technology.php>
18. Larman. (2003). *UML y Patrones. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado*.
19. Larman, C. (2003). *Uml y Patrones*. Pearson.
20. Microsoft. (2012). *msdn*. Recuperado el 20 de 06 de 2015, de <http://msdn.microsoft.com/en-us/library/ff649643.aspx>
21. MyEclipse. (2012). *My Eclipse IDE*. Recuperado el 14 de 6 de 2015, de <http://www.myeclipseide.com/>
22. MySQL-Front. (2012). *Mysql-front*. Recuperado el 10 de 3 de 2015, de <http://www.mysqlfront.de/>
23. Oracle. (2012). *The Java Tutorials*. Recuperado el 20 de 5 de 2015, de <http://docs.oracle.com/javase/tutorial/java/javaOO/>
24. osc (2015). Análisis y diseño de sistemas – UML. Recuperado el 25 de 6 de 2015, de <http://osc.co.cr/analisis-y-diseno-de-sistemas-uml/>
25. Pressman, R. (2005). *Ingeniería de Software*. McGrawHill.
26. PROYECT-IS.WIKISPACES.COM 2015. Recuperado el 23 de 06 de 2015, de <http://proyect-is.wikispaces.com/2.4.-+Modelo+Incremental>
27. Sommerville, I. (2005). *Ingeniería de Software*. Pearson.
28. The Apache Software Foundation. (2010). *Apache Tomcat*. Recuperado el 19 de 6 de 2015, de <http://tomcat.apache.org/>
29. Vázquez Rodríguez, J. A. (16 de 5 de 2006). *Hibernate*. Recuperado el 1 de 5 de 2015, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/vazquez_r_ja/capitulo2.pdf
30. wikipedia.org 2014. Recuperado el 12 de 05 de 2015, de https://es.wikipedia.org/wiki/Programacion_orientada_a_objetos
31. WORDPRESS.COM. (2010). *MI CURSO JAVA & CO*. Recuperado el 24 de 05 de 2015, de <http://micursojava.files.wordpress.com/2010/07/uml.pdf>