

CAPÍTULO I

EL PROYECTO



I.1 PRESENTACIÓN DEL PROYECTO.

I.1.1 Título.

MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC.

I.1.2 Área del Proyecto.

Área administrativa.

I.1.3 Responsable del Proyecto

Carrera de Ingeniería Informática – Taller III – Grupo 5

I.1.4 Entidades Asociadas.

Productos Lácteos Tarija “PROLAC”.

I.1.5 Compromiso del Director del Proyecto.

Yo, Leonardo Farfán Arroyo Director del grupo de desarrollo, acepto las bases y condiciones del concurso, asimismo asumo la responsabilidad de cumplir los compromisos de ejecución del proyecto “MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC” en caso de aprobarse.	
Nombre del Director	Firma del Director

Tabla N° 1: Compromiso del Director del Proyecto.

I.1.6 Grupo Responsable del Proyecto.

Univ. Leonardo Farfán Arroyo.

I.1.7 Talleres Asociados.

Taller III Encargado de la ejecución del proyecto.

I.1.8 Duración (Meses).

La duración del proyecto tiene un periodo de 8 meses aproximadamente.

I.1.9 Director Responsable del Proyecto.

NOMBRE	TALLER / GRUPO	TALLERES PROGRAMADOS
Leonardo Farfán Arroyo C.I. 5792591 Tja.	Taller III / Grupo 5	Taller III
E MAIL	TELEFONO	
leonardo_f19@hotmail.com leo_farroyo@hotmail.com	66-47593 70225626	

Tabla N° 2: Director Responsable del Proyecto

I.1.10 Equipo de trabajo de: Empresas/Instituciones/Organizaciones participantes/Cooperantes

Nombre: Ing. Víctor Cardozo			
Dirección: B/ San Jorge		Teléf. Oficina:	
Nombre y Apellidos	Cargo	C.I.	Firma
Víctor Cardozo	Gerente General		

Tabla N° 3: Empresas/Instituciones/Organizaciones participantes/Cooperantes

I.1.11 Actividades previstas para los integrantes del equipo de investigación

Responsable *	Actividades
Director Leonardo Farfán Arroyo	Dirigir y coordinar las actividades para lograr la finalización del Proyecto, podrá tomar decisiones y realizar algunos ajustes dentro del desarrollo del Proyecto.
Investigador 1 Leonardo Farfán Arroyo	Se encarga del análisis y diseño de los sistemas, interactuando con los involucrados para el intercambio de información y así establecer los requerimientos para cada sistema
Investigador 2 Leonardo Farfán Arroyo	Sera el encargado del diseño y programación del sistema.
Capacitador Leonardo Farfán Arroyo	Será el encargado de capacitar al personal de PROLAC sobre el manejo del sistema informático.
Asesor Docentes de la materia	Encargados de cumplir con las normas y expectativas

Taller III	planificadas para el desarrollo del Proyecto correcto.

Tabla N°4: Actividades previstas para los integrantes del equipo de investigación

I.2. Descripción del Proyecto

I.2.1. Resumen Ejecutivo del Proyecto

Productos Lácteos de Tarija “PROLAC” es una empresa privada de la Ciudad de Tarija que está encargada de:

- Elaborar productos lácteos para la ciudad de Tarija como también para otras ciudades y localidades de Bolivia.

La Empresa realiza varias tareas administrativas las cuales principalmente son:

- Control de asistencia del personal.
- Registro de personal.
- Generar planillas de Sueldos y Salarios.
- Generar planillas de Aguinaldos.
- Generar reportes de la asistencia del personal.

La principal falencia de esta empresa es la de no contar con un registro y control apropiado del personal, el proyecto plantea solucionar esta falencia haciendo una recolección de datos.

Por otra parte plantea el desarrollo e implementación de un Sistema Informático en las dependencias de PROLAC el cual entre sus funciones realizará el registro y control del personal, llevará un control minucioso de la asistencia del personal (asistencia, retrasos, faltas, permisos, etc.), además de generar todo tipo de reportes útiles para la empresa el sistema tendrá como base de información la recolección de datos realizadas en las entrevistas.

Por el impacto que ha tenido el desarrollo de proyectos informáticos con el fin de automatizar la información para facilitar el trabajo, reducir el tiempo de acceso a la información, aumentar la eficiencia, etc. a tenido mucho éxito en los lugares donde fueron desarrollados y a los problemas identificados como perdida de tiempo, acceso a la información, se vio por conveniente desarrollar un proyecto informático para el fortalecimiento de la Gestión de Información. El cual va a contar con dos componentes:

- I. El componente 1: Sistema Informático para el control de Asistencia del personal de PROLAC, se encargara de registrar los datos del personal, registros de asistencia, además de generar reportes de acuerdo al requerimiento del usuario.

- II. El componente 2: Taller de capacitación sobre el uso de las TIC, orientado al personal de PROLAC, específicamente va dirigida a la gerencia, sobre la importancia de la implementación de los recursos TICs en el contexto y el manejo correspondiente del sistema informático para la gestión de información, a los beneficios con este, para que conozcan, manejen, manipulen las herramientas tecnológicas que van a facilitar su labor y desempeño en el cargo. Además se otorgara certificados de participación a cada uno de los asistentes. Para llevar a cabo la capacitación, se utilizo toda la tecnología posible necesaria para llegar con gran facilidad, agrado y detalle de comprensión a los usuarios finales.

El proyecto garantiza el mejoramiento del sistema de administración de PROLAC en el control del personal, que favorecerá a la empresa y miembros de la empresa con la ejecución de los componentes.

I.2.2. Descripción y Fundamentación del Proyecto (qué y por qué)

La realización del proyecto **“MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC”** tiene como objetivo principal coadyuvar a brindar un mejor control y servicio del personal de la empresa, basándonos en el Plan de Desarrollo de esta empresa los objetivos de la misma plantea solucionar las falencias de esta empresa en el área administrativa realizando entre otros componentes un Sistema Informático que se encargue de:

- **Llevar un mejor control de los datos del personal.**

La falta de datos actualizados del personal no permite llevar un buen control de los mismos, produce un retardo en las diferentes actividades a cargo de esta empresa, el registro que maneja actualmente esta empresa no es el apropiado por que no permite generar datos estadísticos, el acceso a estos es muy moroso y poca fiabilidad, la solución que brindará el sistema es la de realizar un kárDEX completo del personal de la empresa que estará registrado en una Base de Datos y así evitar pérdida de información.

- **Control de asistencia.**

Una vez teniendo un registro actualizado y automatizado de datos del personal el sistema podrá tener un control de la asistencia del personal seguro y fiable para la empresa como para el personal.

- **Emitir Reportes.**

El sistema brindará la posibilidad de generar reportes de todo tipo como ser registro de asistencia, reincidencias de faltas, retrasos, planillas de sueldos y salarios, etc.

El proyecto contempla acreditar a todos los integrantes de la empresa con la finalidad de identificarlos por áreas de trabajo para así facilitar el uso del sistema anteriormente descrito, otro factor es el de dar una buena presentación y brindar confiabilidad a los clientes.

Una vez realizado el sistema se procederá a dar una adecuada capacitación del mismo a las personas encargadas de manipular los procesos que realizará el sistema.

a) Árbol de Problemas

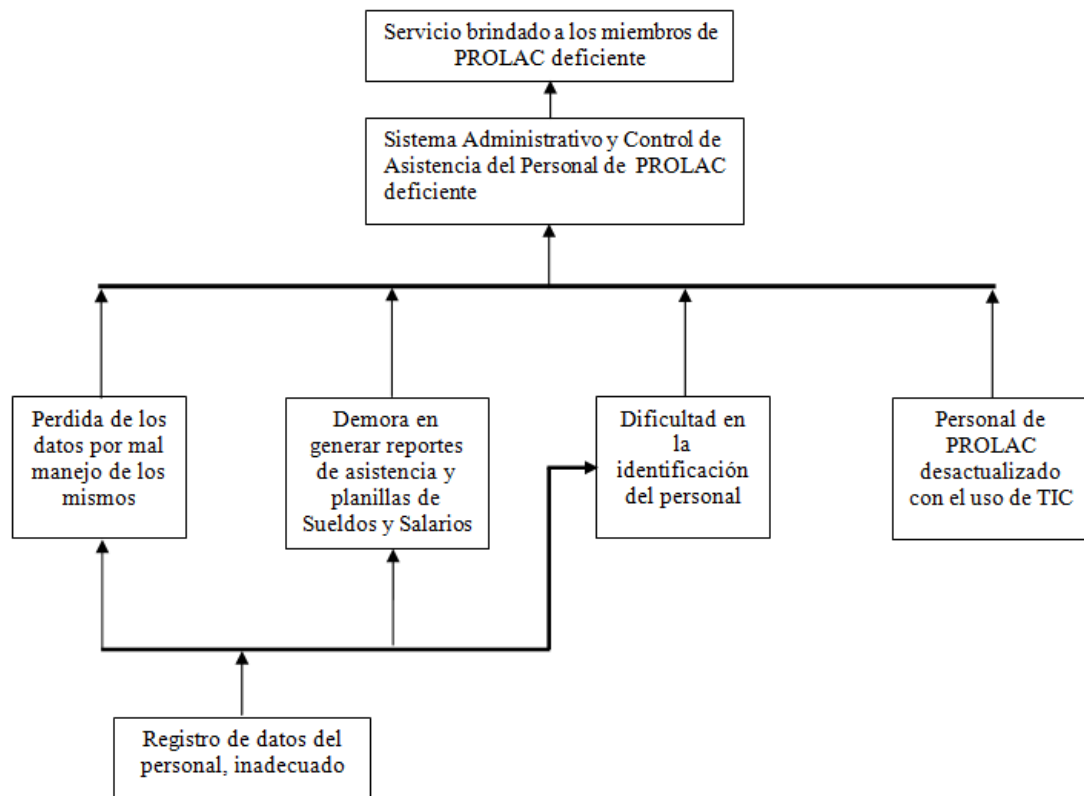


Figura N° 1: Árbol de Problemas

b) Árbol de Objetivos y Problemas

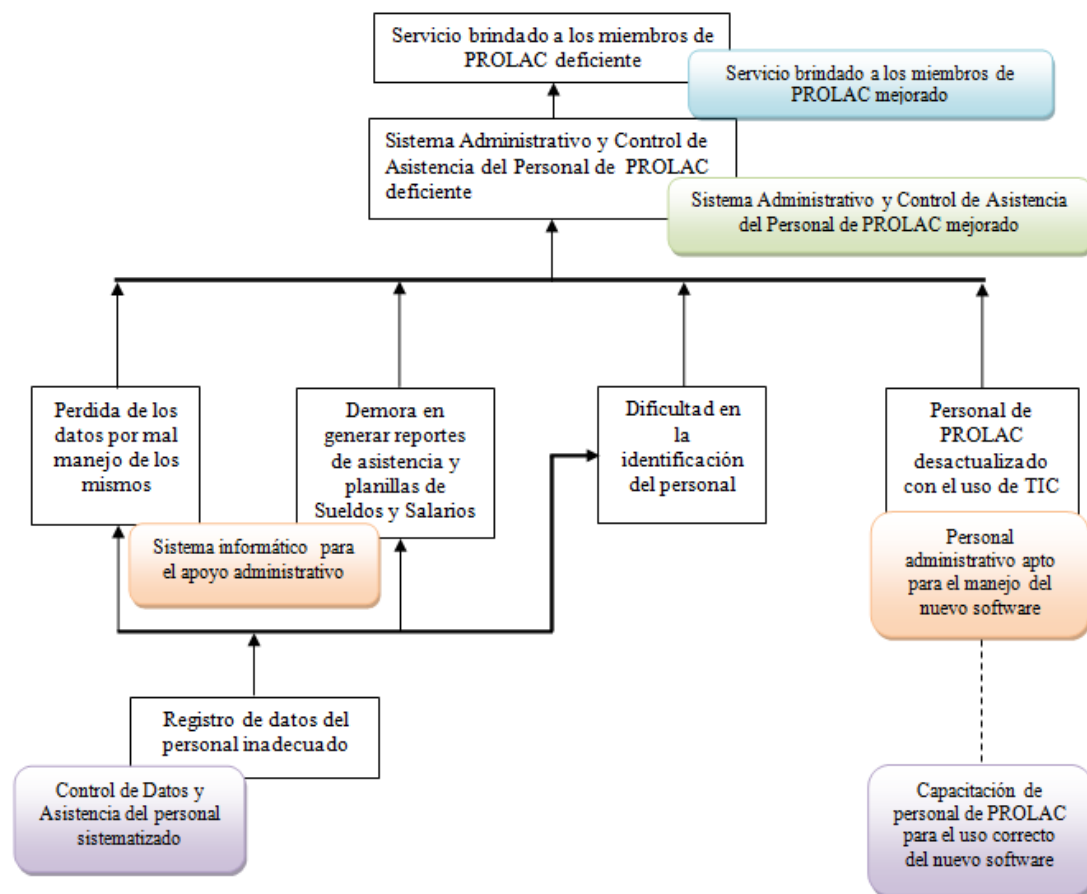


Figura N° 2: Árbol de Problemas

c) Situación planteada Con y Sin Proyecto

Situación sin proyecto	Situación con proyecto
• Registro de datos del personal deficiente. • Demora en generar reportes sobre la asistencia del personal. • Personal de PROLAC desactualizado con el uso de TIC. • Búsqueda de datos del personal inadecuado. • Dificultad en la identificación del personal.	• Registro de datos del personal eficiente. • Rapidez en generar reportes sobre la asistencia del personal. • Personal de PROLAC actualizado con el uso de TIC. • Búsqueda de datos del personal adecuado. • Eficiencia en la identificación del personal.

Tabla N°5: Situación planteada Con y Sin Proyecto

I.2.3. Objetivos

I.2.3.1 Objetivo General

MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC.

I.2.3.2 Objetivos Específicos

- Sistema Informático para el apoyo administrativo de la empresa PROLAC de Tarija.
- Realizar un Taller de Capacitación y Socialización sobre el uso de las TIC.

I.2.4 Matriz de Marco Lógico del Proyecto

RESUMEN NARRATIVO DEL PROYECTO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
<u>Fin</u> Contribuir a mejorar la calidad de trabajo de los miembros de la empresa PROLAC.	Al final del 1º año de ejecución del proyecto, la gestión de información y registro de asistencia del personal, se incrementa en un 40% en comparación al año base (2011).	Informe de gestión de PROLAC a un año de ejecutado el Sistema Informático.	El apoyo del Gerente General de PROLAC se mantiene brindando la suficiente información para la ejecución del Proyecto.
<u>Objetivo General</u> (<u>Propósito</u>) Sistema Administrativo y Control de Asistencia del	Al finalizar el proyecto la gestión de	Certificado de conformidad en el que se	Existe apoyo del Gerente General de

Personal de PROLAC mejorado.	información y registro de asistencia del personal se ve fortalecida en un 60% con un entorno tecnológico.	constata la mejora de la administración y control de asistencia del personal en un 60 % firmado por el Ing. Víctor Cardozo Gerente General de PROLAC.	brindar la información confiable y oportuna.
		• Se cuenta con copias de reportes generados por el Sistema Informático avalando la mejora y rapidez en un 75 % en la obtención de la información de la asistencia del personal de PROLAC. • El Gerente General considera importante la implementación de un sistema informático en sus dependencias para brindar un mejor servicio y control del personal.	

<u>Objetivos Específicos</u> <u>(Componentes)</u>			
1. Sistema Informático para el apoyo administrativo desarrollado.	1.1 El Sistema Informático para la administración de PROLAC se realizó en un 100 % desde el 10 de Mayo hasta el 22 de Noviembre del 2011. 1.2 Análisis y Diseño del sistema se realizó en un 100 % desde el 10 de Mayo hasta 11 de Noviembre del 2011 cuyo número de analistas fue un miembro.	• Certificado de conformidad de PROLAC del Sistema Informático firmado por el Ing. Víctor Cardozo Gerente General de PROLAC. • Documento de Análisis y Diseño del Sistema Informático revisado por los docentes de la materia Taller III.	• Administrativos de PROLAC brindaron información correcta y oportuna para la realización del sistema. • Disponibilidad de recursos y herramientas de software y hardware para el desarrollo del Sistema Informático.

2. Miembros de la administración de PROLAC capacitados sobre el manejo del Sistema Informático	1.3 La fase de desarrollo del sistema se realizó en un 100 % desde el 1 de Junio hasta el 16 de Noviembre del 2011 cuyo número de programadores fue un miembro.	Manual de usuario.	Predisposición del personal de administración de PROLAC en participar en la capacitación.
2.1 El curso de capacitación para el manejo del Sistema Informático, se realizó del 24 al 25 de Noviembre del 2011.	Certificado de conformidad de la PROLAC de la capacitación sobre el manejo del Sistema Informático, firmado por el Ing. Víctor Cardozo Gerente General de PROLAC.		

		• Guías de capacitación (Manual de usuario). • Planilla con firmas de las personas capacitadas. • Certificado de conformidad de PROLAC de la entrega de certificados de participación en la capacitación firmado por el Ing. Víctor Cardozo Gerente General de PROLAC.	
<u>Actividades</u> 1.1 Fase de Análisis y	1. Sistema Informático	1.1 Documento de análisis y	• Disponibilidad de

Diseño del Sistema. 1.2 Fase de Desarrollo del Sistema.	➤ El Análisis y Diseño se realizó desde el 10 de Mayo hasta 11 de Noviembre del 2011, cuyo número de analistas es de un miembro. 1.2 La fase de desarrollo del sistema se realizó en un 100 % desde el 1 de Junio hasta el 16 de Noviembre del 2011 cuyo número de programadores fue de un miembro.	diseño revisado por los docentes de Taller III. 1.2 Presentación del sistema cumpliendo todos los requerimientos establecidos en la fase de análisis.	herramientas para realizar el desarrollo del sistema. • Contar con equipos de computación para la implementación del sistema.

1.3 Fase de Pruebas del Sistema.	1.3 La fase de pruebas se realizo en un 100 % desde el 17 al 22 de Noviembre del 2011.		
2.1 Cursos de Capacitación	2. Capacitación y Socialización. 2.1 Cursos de capacitación se realizo del 24 al 25 de Noviembre del 2011.	2.1 Planilla del personal capacitado con firmas de asistencia.	• Predisposición del personal de PROLAC en participar en la capacitación. • Disponibilidad de un ambiente y equipos de computación para la capacitación.
2.2 Entrega de Certificados de	2.2 La entrega de certificados a las	2.2 Certificado de conformidad sobre la	

Capacitación.	personas capacitadas se realizo el 28 de Noviembre del 2011.	capacitación emitido por PROLAC.	
	Presupuesto por partidas (ver punto 3 Presupuesto/Justificación)		

Tabla N°6: Marco Lógico del Proyecto

I.2.5 Metodología de Trabajo

- Utilizar la metodología RUP (Proceso Unificado Racional) para el desarrollo del Software.
- Desarrollar la herramienta UML (Lenguaje Unificado de Modelado) para el análisis y diseño del software.
- Utilizar para la base de datos la herramienta PostgreSQL.
- La programación se desarrollará en lenguaje JAVA.
- Se desarrollará diferentes pruebas para la usabilidad.
- El Software a desarrollar será amigable y de fácil manejo para el usuario.

a) Descripción y Relación de las Estrategias con los Objetivos

Estrategias	Objetivos Específicos
1.1 El Sistema Informático para la administración de PROLAC se realizó en un 100 % desde el 10 de Mayo hasta el 22 de Noviembre del 2011. 1.2 Análisis y Diseño del sistema se realizó en un 100 % desde el 10 de Mayo hasta el 11 de Noviembre del 2011 cuyo número de analistas fue	<u>Componentes</u> 1. Se dispone de un Sistema Informático para el apoyo administrativo.

un miembro. 1.3 La fase de desarrollo del sistema se realizo en un 100 % desde el 01 de Junio hasta el 16 de Noviembre del 2011 cuyo número de programadores fue un miembro. 1.4 La fase de pruebas se realizo en un 100 % desde el 17 al 22 de Noviembre del 2011. 2.1 El programa de capacitación y socialización llegó a dos personas: 1 Secretaria y el Gerente General. Se realizo del 24 al 25 de Noviembre del 2011.	2. Capacitación y Socialización al personal de la administración de PROLAC sobre el Sistema Informático.

Tabla N° 7: Descripción y Relación de las Estrategias con los Objetivos

b) Cronograma de Actividades

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos	ma
1	Mejoramiento del Sistema Administrativo y Control de Asistencia del Personal de PROLAC	148 días?	mar 10/05/11	lun 28/11/11		2000 \$us	
2	Sistema Informático	144 días?	mar 10/05/11	mar 22/11/11		1850 \$us	
3	Fase de Analisis y Diseño	137 días?	mar 10/05/11	vie 11/11/11		700 \$us	
4	Fase de Desarrollo	124 días?	mié 01/06/11	mié 16/11/11		900 \$us	
5	Fase de Pruebas	4 días?	jue 17/11/11	mar 22/11/11	4	250 \$us	
6	Capacitación y Socialización	3 días?	jue 24/11/11	lun 28/11/11		150 \$us	
7	Cursos de Capacitación	2 días?	jue 24/11/11	vie 25/11/11		130 \$us	
8	Entrega de Certificados de Capacitación	1 día?	lun 28/11/11	lun 28/11/11	7	20 \$us	

Proyecto: Diag Gantt Fecha: sáb 18/02/12	Tarea División Progreso Hito Resumen	Resumen del proyecto Tareas externas Hito exte Tarea División

Tabla N° 8: Diagrama de Gantt

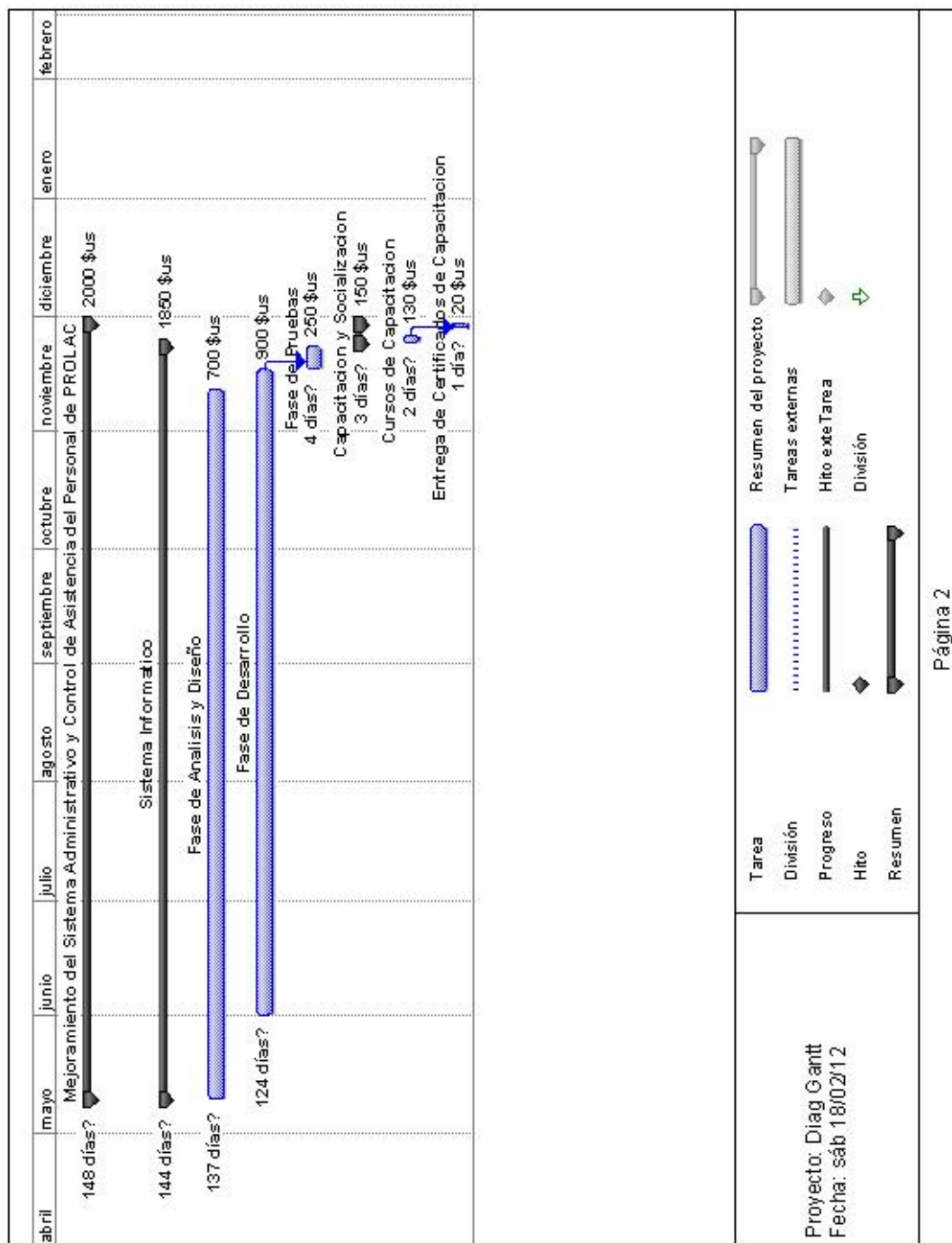


Figura N° 3: Diagrama de Gantt

I.2.6 Resultados Esperados

- Sistema realizará un control minucioso de la asistencia del personal, además de generar todo tipo de reportes útiles para la empresa (asistencia, retrasos, faltas, permisos), el sistema tendrá como base de información la recolección de datos realizadas en las entrevistas. Acreditación de los miembros de la empresa, los datos necesarios para esta actividad serán extraídos del sistema informático.
- Miembros encargados del Control capacitados en el uso apropiado del software implementado y el uso de Tecnologías Informáticas.

I.2.7 Transferencia de Resultados

a) Medios y Estrategias para la Transferencia de Resultados.

- Una vez firmado el acuerdo con PROLAC para el Tratamiento de información, se estableció un plazo para la finalización y posterior puesta en marcha de los resultados esperados por ambas partes.
- Se realizó una reunión culminante para presentar los resultados ante el equipo conformado por PROLAC.

b) Grupo de beneficiarios de los resultados

- Gerente General PROLAC.
- Secretaria.

I.3. Presupuesto / Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
10000	SERVICIOS PERSONALES			
	12000 Empleados no Permanentes			
	Sub total rubro			
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos			320
	22000. Servicios de transporte			540
	23000. Alquileres			240
	24000. Mantenimiento y reparación			-
	25000. Servicios Profesionales y Comerciales			4684
	Sub total rubro			5974
30000	MATERIALES Y SUMINISTROS			-
	31000. Alimentos y Productos Forestales			2460
	32000. Productos de Papel, Cartón e Impresos			920
	33000. Textiles y Vestuario.			-
	34000. Productos Químicos, Combustibles y Lubricantes			-
	39000. Productos Varios.			60
	Sub total rubro			3440
40000	ACTIVOS REALES			
	43000. Maquinaria y Equipo.			-

	46000. Descripción de estudios y proyectos para inversión			-
	49000. Otros Activos			-
	Sub total rubro			0
	TOTAL			9414
	TOTAL + 40% Incentivo			19600

1) GRUPO 10000. SERVICIOS PERSONALES

a) SUB GRUPO 12000. Empleados no Permanentes

Partida	Personal	Remuneración	Tiempo/meses	Total
12100	Personal Eventual			-
	Total			-

* Se refiere a gastos para remunerar a personas sujetas a contrato dependientes según la necesidad de cada entidad

2) GRUPO 20000. SERVICIOS NO PERSONALES

b) SUB GRUPO 21000. Descripción de los gastos de servicios básicos

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación			-
21200	Energía Eléctrica	40	8	320
21300	Agua			-
21400	Servicios Telefónicos			-
				-
	Total			320

* Se refiere principalmente a los gastos por servicios; como: servicio de correo, radiogramas, servicio telefónico, fax, Internet.

c) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

Partida	Personal	Lugar	Nº de viajes	Costo unitario*	Costo total
22100	Pasajes		270	2	540
					-
	Total				540

* En el caso de pasajes debe indicarse el costo de ida y vuelta (costo unitario), indicando el número de viajes.

Partida	Personal	Lugar	Duración (días)	Costo unitario*	Costo total
22200	Viáticos				-
22300	Fletes y Almacenamientos				-
22600	Transporte de Personal				-
Total					-
Total sub grupo 22000					-

* En el caso de los viáticos, debe considerarse la escala establecida por la UAJMS.

d) SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios			-
23200	Alquiler de Equipos y Maquinaria	6	2	240
23300	Alquiler de Tierras y Terrenos	-	-	-
Total				-

* Se refiere principalmente a los gastos por el uso de edificios y equipos y maquinaria en general

e) SUB GRUPO 24000. Descripción mantenimiento y reparación

Partida	Mantenimiento y reparación de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
24100	Mantenimiento y Reparación de Edificios y Equipos	-	-	-
24300	Otros Gastos por Mantenimiento y Reparación	-	-	-
Total				-

* Se refiere principalmente a los gastos por el mantenimiento y reparación de edificios y equipos y maquinaria en general

f) SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25200	Estudios e Investigaciones				-
25500	Publicidad				-
25600	Imprenta	26	105		700
25700	Capacitación de Personal	1	700	1	700
25800	Estudios e Investigaciones Para Proyectos de Inversión				-
Total					700

* Se refiere a gastos por servicios profesionales de asesoramiento especializado, se incluyen, estudios, investigaciones, publicidad, imprenta, fotocopias, capacitación de personal y otros ejecutados por terceros.

3) GRUPO 30000. MATERIALES Y SUMINISTROS

g) SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
31110	Refrigerios y Gastos Administrativos	820	3	2460
31200	Alimento para Animales			-
31300	Productos Agroforestales y Pecuarios			-
Total				105

* Se refiere a la adquisición de materiales y bienes como: alimentos y productos agroforestales, alimentos y bebidas para personas (indicar el total de refrigerios), alimentos para animales, productos pecuarios.

h) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	30	35	1050
32200	Productos de Artes Graficas, Papel y Cartón	10	45	450
32300	Libros y Revistas	10	90	470
32400	Textos de Enseñanza			-
32500	Periódicos			-
Total				1970

* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases, impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

i) SUB GRUPO 33000. Descripción del gasto en textiles y vestuario

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
33100	Hilados y Telas			-
33200	Confecciones Textiles			-
33300	Prendas de vestir			-
33400	Calzados			-
Total				-

* Se refiere principalmente a los gastos por vestuario uniformes, ropa de trabajo

j) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total
34110	Combustibles y Lubricantes para Consumo			-
34200	Productos químicos y Farmacéuticos			-
34400	Productos de Cuero y Caucho			-
34500	Productos de Minerales no Metálicos y Plásticos			-
34600	Productos Metálicos			-
34700	Minerales			-
34800	Herramientas Menores			-
Total				-

* Se refiere a gastos de combustibles, químicos, productos farmacéuticos, llantas etc.

k) SUB GRUPO 39000. Descripción del gasto en productos varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39100	Material de Limpieza			-
39400	Instrumental Menor Medico - Quirúrgico			-
39500	Útiles de Escritorio y de Oficina			-
39700	Útiles y Materiales Eléctricos			-
39800	Otros Repuestos y Accesorios			-
Total				-

*Se refiere principalmente a los gastos por productos de limpieza, todo lo referente a la funcionamiento de la oficina en material de escritorio.

4) GRUPO 40000. ACTIVOS REALES

l) SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo

Partida	Tipos de productos	Cantidad	Costo/Unitario	Total
43100	Equipo de Oficina y Muebles			-
43200	Maquinaria y Equipo de Producción			-
43300	Equipos de Transporte, Tracción y Elevación			-
43400	Equipo Médico y de Laboratorio			-
43600	Equipo Educativo y Recreativo			-
43700	Otra Maquinaria y Equipo			-
Total				-

* Se refiere principalmente a los gastos por muebles y enseres, equipo de oficina, comunicación, equipamiento.

m) SUB GRUPO 46000. Descripción de estudios y proyectos para inversión

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
46100	Para Construcción de Bienes de Dominio Privado			-
Total				-

* Se refiere principalmente a los gastos por servicios de terceros para la realización de investigaciones y otras actividades técnico – Profesionales necesarias para la construcción y mejoramiento de bienes.

n) SUB GRUPO 49000. Descripción del gasto de Otros Activos

Partida	Tipos de productos *	Cantidad	Costo/Unitario	Total
49100	Activos Intangibles			-
49200	Compra de Bienes Muebles Existentes (Usados)			-
49300	Semovientes y otros Animales			-
49900	Otros Activos			-
Total				-

* Se refiere a los gastos en la compra de software, licencias.

I.4. Curriculum Vitae (del Director y Equipo de Trabajo: llenar una ficha para cada uno de ellos)

I.4.1 Antecedentes personales

Farfán Apellido Paterno	Arroyo Apellido Materno	Leonardo Nombre	5792591 Tja. C.I.
17/01/1985 Fecha de nacimiento	M..X... F..... Sexo	B/ Luis de Fuentes C/ Carlos Zenteno N° 543 Dirección	
Tarija Ciudad	66-47593 Teléfono Domicilio	70225626 Celular	leonardo_f19@hotmail.com Correo electrónico

I.4.2 Antecedentes académicos

Título obtenido	Universidad	País	Año

I.4.3 Participación en proyectos de investigación

Título proyecto	Institución	Cargo	Año

I.4.4 Publicaciones realizadas (libros, revistas, compendios y otros)

Autor	Tipo de publicación, Año, título, volumen, páginas, editorial

* Indique las publicaciones realizadas durante los últimos 5 años

I.4.5 Presentaciones realizadas

Título	Nombre del evento, lugar, fecha, año

* Incluir hasta 5 presentaciones de los últimos 5 años

I.4.6 Antecedentes en docencia

a) Experiencia Docente

Experiencia Docente		
Institución	Dedicación	Años

b) Jerarquía Docente (en la presente gestión, y de gestiones anteriores)

Categoría docente	Carrera/Departamento	Tiempo dedicación	Gestión

CAPÍTULO II

COMPONENTES



II.1 COMPONENTE 1: SISTEMA INFORMATICO PARA EL APOYO ADMINISTRATIVO

II.1.1 INTRODUCCION



Plan de Desarrollo del Software

II.1.1 Introducción

Este plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido ofertado por “Leonardo Farfán Arroyo” basado en la metodología de Rational Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque desarrollado propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

El Software desarrollado para el proyecto **“MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC”** es un Sistema de gestión que utiliza un Modelo de Software “Incremental”. Ya que en su etapa de construcción el Sistema se divide en Módulos que fueron desarrollados jerárquicamente.

Este modelo nos ofrece las ventajas de poder extraer y generar datos de los primeros módulos para desarrollar los posteriores:

ESPECIFICACIÓN DE MÓDULOS CON LOS QUE CONTARÁ EL SISTEMA

MÓDULO 1: SISTEMA

MÓDULO 2: PERSONAL

MÓDULO 3: CONTROL ASISTENCIA

MÓDULO 4: REPORTES

II.1.1.2 Objetivos

II.1.1.2.1 Objetivo General

La realización del Sistema Informático para el proyecto **“MEJORAMIENTO DEL SISTEMA ADMINISTRATIVO Y CONTROL DE ASISTENCIA DEL PERSONAL DE PROLAC”** tiene como objetivo principal solucionar las falencias de esta institución en el área administrativa, facilitando el acceso a datos de una manera más rápida, eficiente y confiable, además de gestionar las actividades mas comunes que requieren un registro de datos y control de asistencia del personal, basándonos en el plan de desarrollo de esta institución y los objetivos que la misma plantea.

II.1.1.2.2 Objetivos Específicos

- **Llevar un mejor Control de Asistencia de Personal**

La falta de datos actualizados del personal no permite llevar un buen control asistencia de los mismos lo que produce retrasos en la generación de reportes, el registro que maneja actualmente esta institución no es el apropiado por que no permite generar datos seguros, el acceso a estos es muy moroso y de poca fiabilidad, la solución que brindara el Sistema es la de realizar un registro completo de las horas trabajadas del personal que estará registrado en una base de datos, así evitar perdida de información y tener un rápido y fácil acceso a estos.

- **Emitir Descuentos.**

La gerencia de PROLAC recibirá informes del control de la asistencia del personal, para posteriormente emitir los descuentos correspondientes de acuerdo al retraso o falta y descuentos establecidos.

Una vez teniendo un registro actualizado y automatizado de datos del personal el Sistema podrá tener un control de descuentos por empleado.

- **Emitir Reportes.**

El Sistema brindara la posibilidad de generar reportes del personal de control de asistencia.

Con la finalidad de poder tener un conocimiento general de los distintos datos que maneja la gerencia referente al personal

- **Registro de la Asistencia del Personal.**

El Sistema brindara la posibilidad de registrar la asistencia del personal para un mejor control por parte de la gerencia.

II.1.1.3 Justificación

Durante la elaboración del presente documento, se procurará dar conocer de cómo se realiza actualmente el proceso de tratamiento de toda la información que se produce a consecuencia del proceso de control de asistencia del personal.

II.1.1.3.1 Justificación Social

Uno de los fines de la “Universidad Autónoma Juan Misael Saracho”, es formar profesionales idóneos en todas las áreas del conocimiento científico, tecnológico y cultural, que responda a las necesidades del desarrollo nacional y regional.

II.1.1.3.2 Justificación Económica

Para el desarrollo del presente Trabajo, se cuenta con todos los medios y recursos necesarios y suficientes por lo que se puede asegurar que el mismo es económicamente factible.

II.1.1.3.3 Justificación Académica

Académicamente el presente trabajo tiene como objetivo la aplicación de todos los conocimientos adquiridos durante la formación universitaria.

II.1.1.4 Alcances y Limitaciones

El Alcance General de presente Proyecto es implementar Sistema Informático para la gestión de información en el proceso de Control de Asistencia del Personal, que proporcione al usuario una interfaz amigable para que este pueda tener información actualizada y acorde a sus necesidades y requerimientos.

Consideremos que es preciso que el Sistema Informático debe gestionar el movimiento de:

- Gestión de Usuarios.
- Gestión de Áreas y Puestos.
- Gestión de Empleados.
- Gestión de Permisos y Vacaciones.
- Gestión de Planillas de Sueldos y Aguinaldos.
- Gestión de Edición Manual de Asistencia.
- Gestión de Horarios.
- Gestión de Días Inhábiles.
- Reportes (de personal, de control de asistencia).
- Registro de Asistencia.

II.1.1.5 Suposiciones y Restricciones

Las suposiciones y restricciones respecto de este Sistema, que derivan directamente de las entrevistas con los responsables de la institución son:

II.1.1.5.1 Suposiciones:

- El sistema será seguro por que almacenara en su base de datos la información de todas las actividades del Sistema.

- Información que podrá ser reutilizada en reportes y procesos futuros para la toma de decisiones a nivel estratégico.
- Seguridad de transacciones e intercambio de información.
- El Sistema verificara el estado del personal.
- El Sistema permitirá imprimir planillas de sueldos y salarios del personal.

II.1.1.5.2 Restricciones:

- Restricción de acceso al sistema mediante el uso de claves.
- Restricción de información al personal no autorizado.
- Cada personal autorizado tendrá un único Usuario de el Sistema en línea.
- Las gestiones de generar planillas de sueldos y salarios del personal se realizaran en gerencia.
- Un usuario tendrá acceso al sistema solo si la gerencia lo permite.

II.1.1.6 Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objetos de modificaciones a lo largo del proceso de desarrollo, con lo cual, solo al termino del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas, etc.). Permite situar al Sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos para este modelo.

3) Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada uso de negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados en el caso de uso del negocio.

Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

4) Requerimientos Funcionales

En este apartado se describen los requerimientos funcionales del sistema, la funcionalidad o los servicios que se espera que éste provea.

5) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada de acuerdo al proyecto.

6) Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

7) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

8) Especificaciones de Casos de Uso

Para los casos de uso que requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación grafica mediante un Diagrama de Actividad.

9) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia un diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

10) Modelo de Datos

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es solo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

11) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizaran como:

dibujos a mano en papel, dibujos con alguna herramienta grafica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto.

12) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen.

Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.

13) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados.

14) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Manual de Usuario y Manual de Instalación.

15) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación.

II.1.1.6.1 Organización del Proyecto

II.1.1.6.1.1 Participantes del Proyecto

De momento no se incluye el personal que designará “PROLAC” como Responsable del Proyecto, Comité de Control y Seguimiento, otros participantes que se estimen convenientes para proporcionar los requisitos y validar el sistema.

El resto del personal del proyecto, considerando las fases de Inicio, Elaboración y dos iteraciones de la fase de Construcción, estará formado por los siguientes puestos de trabajo y personal asociado:

Jefe de Proyecto: Universitario Leonardo Farfán Arroyo, alumno del último curso de la carrera de Ingeniería Informática de la Facultad de Ciencia y Tecnología de la UAJMS. Con una experiencia modesta en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y el proceso de desarrollo RUP.

Analista de Sistema: El perfil establecido pertenece al universitario Leonardo Farfán Arroyo con conocimientos de UML, con experiencia en sistemas afines a la línea del proyecto.

Programador: Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser los mas cercanos posibles al producto final. Este trabajo ha sido encomendado a Leonardo Farfán Arroyo.

Ingeniero de Software: El perfil establecido estará a cargo del universitario de Informática que participará realizando labores de gestión de requisitos, gestión de configuración, documentación y diseño de datos. Encargado de las pruebas funcionales del sistema, realizará la labor de Tester: Leonardo Farfán Arroyo.

II.1.1.6.1.2 Interfaces Externas

“PROLAC” definirá los participantes del proyecto que proporcionará los requisitos e información del sistema para el desarrollo del sistema, y entre ellos quienes serán los encargados de evaluar los artefactos de acuerdo a cada parte del sistema.

El equipo de desarrollo interactuará activamente con los participantes de “PROLAC” para especificación y validación del producto generado.

II.1.1.6.1.3 Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Los roles serán desempeñados con la responsabilidad y seriedad para llevar adelante el proyecto.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Planificación y Control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

Tabla N° 9: Roles y Responsabilidades

II.1.1.6.2 Gestión del Proceso

II.1.1.6.2.1 Estimación del Proyecto

II.1.1.6.2.1.1 Presupuesto / Justificación

Se desea estimar los costos que tendrá el Sistema Informático para el Control de Asistencia el Personal de PROLAC.

Utilizaremos el método **COCOMO** para estimación de costos y para esto tenemos los siguientes datos: el sistema estima producir 3,5 **MIFE** (miles de instrucciones fuentes) se pretende que el sistema tenga una complejidad media y para esto se contratara analistas y programadores que tengan conocimientos medios en este tipo de programación del sistema (cobrando alrededor de 400 US\$) pretende brindar una confiabilidad alta, utilizar recursos de base de datos muy altos y una complejidad normal.

Método Cocomo Básico

Desarrollo:

Datos: Tamaño=3.5 MIFE Tarifa Mensual=400 US\$ COSTO= EN*tarifa mensual

Para calcular el Esfuerzo Nominal vamos a utilizar la siguiente fórmula:

$$EN = 2.4 * (MIFE)^{1.05}$$

Reemplazando datos nos dará $EN = 2.4 * (3,5)^{1.05}$ entonces el esfuerzo nominal será:

$$EN = 9 \text{ meses de programador}$$

Para calcular el Tiempo de Desarrollo vamos a utilizar la siguiente fórmula:

$$TDES = 2.5 * (EN)^{0.38}$$

Reemplazando datos nos dará: $TDES = 2.5 * (9)^{0.38}$ entonces el tiempo de desarrollo será:

$$TDES = 6 \text{ meses para su desarrollo}$$

$$COSTO = 9 * 400 = US\$ \text{ Costo Estimado del Sistema}$$

Factor Multiplicador	Razonamiento	Valor
Confiabilidad	Nominal	1
Base de Datos	Nominal	1
Tiempo	Nominal	1
Memoria	Alto	1.06
Maquina Virtual	Nominal	1
Tiempo de Retorno	Bajo	0.87
Analistas	Alto	0.86
Programadores	Alto	0.86
Experiencia	Nominal	1
Practica	Muy Alto	0.82
Herramientas	Alto	0.91
Calendario	Alto	1.04
Complejidad	Muy Alto	1.30
Factor de Ajuste		0.7

Método Cocomo Intermedio

Desarrollo:

Con el fin de **ajustar las estimaciones** con respecto a desviaciones del valor inicial obtenido se utiliza el factor de ajuste el cual se calcula con los datos arriba descritos y el cual nos da: **FA=0.7**

Para calcular el esfuerzo nominal ajustado vamos a utilizar la siguiente la siguiente formula.

$$PM = 2.4 * (MIFE)^{1.05} * FA$$

Reemplazando datos nos dará $PM = 2.4 * (3,5)^{1.05} * 0.7$ entonces el esfuerzo nominal ajustado será:

$$PM = 6 \text{ meses de programador}$$

Para calcular el tiempo de desarrollo ajustado vamos a utilizar la siguiente fórmula:

$$TDES=2.5*(PM)^{0.38}$$

Reemplazando datos nos dará: $TDES=2.5*(6)^{0.38}$ entonces el tiempo de desarrollo ajustado será:

$$TDES=6 \text{ meses para su desarrollo}$$

$$COSTO=PM*costo \text{ mensual}=6*400=2400 \text{ US\$ costo estimado del proyecto.}$$

II.1.1.6.2.1.2 Presupuesto de Gastos

Para que el sistema logre los objetivos previstos anteriormente detallaremos el gasto que se realizará por cada mes del tiempo de desarrollo del sistema, como se realizó una estimación de costo del sistema y del tiempo que durará el sistema nos basaremos en esto para detallar el gasto del sistema por cada mes que dure el mismo. En el desarrollo de dicho sistema consideraremos los siguientes recursos básicos como ser Costo Analista, Costo Programador, Costo Hardware, Costo Software. A continuación describiremos cada uno de estos costos.

Costo analista.- Este costo se refiere al precio que pagaremos por el servicio del analista a lo largo del desarrollo del sistema.

Costo Programador.- Este costo se refiere al precio que pagaremos por el servicio del programador a lo largo del desarrollo del sistema.

Costo Hardware.- Este costo se refiere al precio que pagaremos por el alquiler del equipo como: computadoras, impresoras y otros, en el tiempo que dure el desarrollo del sistema.

Costo Software.- Este costo se refiere al precio que pagaremos por las diferentes herramientas de software que ocuparemos en el desarrollo del proyecto.

Meses	Costo Hardware	Costo Analista	Costo Programador	Costo Software	Costo Total
1	50	200	-	0	250
2	70	200	-	0	270
3	70	100	100	0	270
4	70	100	400	0	570
5	70	50	400	0	520
6	70	50	400	0	520
Total	400	700	1300	0	2400 US\$

II.1.1.6.2.2 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario de presentaciones del proyecto.

II.1.1.6.2.2.1 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la distribución de tiempos y el numero de iteraciones de cada fase (para las fases de Construcción y Transición es solo una aproximación muy preliminar).

Fase	Nro. Iteraciones	Duración	Fecha
Fase de Inicio	2	2 semana	1 Mayo – 15 Mayo
		2 semana	17 Mayo – 31 Mayo
Fase de Elaboración	4	2 semanas	1 Mayo – 30 Mayo
		3 semanas	1 Junio – 22 Junio

		3 semanas	22 Junio – 7 Julio
		3 semanas	8 Julio – 1 Agosto
Fase de Construcción	4	3 semanas	10 Agosto – 31 Agosto
		3 semanas	31 Agosto – 21 Septiembre
		4 semanas	21 Septiembre – 20 Octubre
		4 semanas	20 octubre – 20 Noviembre
Fase de Transición	-	Mesas	

Tabla N° 10: Plan de Fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase se desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera fase de Construcción

	deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado del producto hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Las cuatro iteraciones suman una duración de 9 semanas.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 4 iteraciones, cada una produciendo una versión a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión 3.0, con la capacidad operacional parcial del producto que se haya considerado como critica, lista para ser entregada a los usuarios.
Fase de Transición	El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto. Con la defensa respectiva del proyecto realizado.

Tabla N° 11: Descripción de Fases

II.1.1.6.2.2.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo solo las fases de Inicio. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase de iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

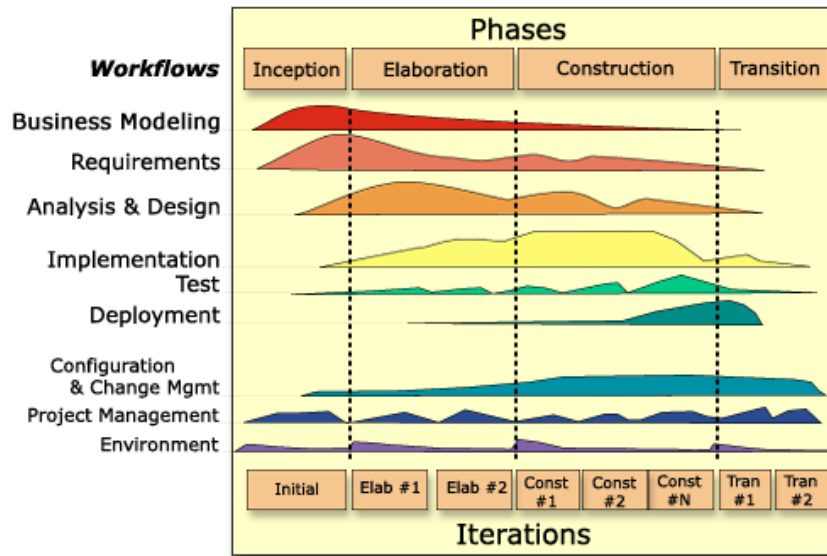


Figura N° 4: Metodología RUP

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuando el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 2 1/05 – 15/05	
Requisitos		
Glosario	Semana 2 1/05 – 22/05	
Visión	Semana 2 23/05 – 15/06	
Modelo de Casos de Uso	23/05 – 17/06	
Especificación de casos de Uso	18/06 – 29/06	
Especificaciones Adicionales	-----	
Análisis / Diseño		
Modelo de Análisis / Diseño	01/06 – 23/07	
Modelo de Datos	-----	
Implementación		
Prototipos de Interfaces de Usuario	-----	
Modelo de Implementación	-----	

Pruebas		
Casos de Pruebas Funcionales	-----	
Despliegue		
Modelo de Despliegue		Siguiente fase
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del Proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las iteraciones	-----	
Ambiente	Durante todo el proyecto	

Tabla N° 12: Disciplinas y Artefactos en la fase de Inicio

II.1.1.6.3 Referencias

- Documentación de Rational Unified Process, Manuales de ayuda, tutoriales, etc.
- Diseño de Aplicaciones de escritorio usando UML, por Miguel Reynolds.
- UML y Patrones Craig Larman.
- Visual Modeling with Rational Rose and UML, Terry Quatrani. – Addison – Wesley.

II.1.1.7 Requerimientos Funcionales

II.1.1.7.1 Introducción

En este apartado se describen los requerimientos funcionales del sistema, la funcionalidad o los servicios que se espera que éste provea.

II.1.1.7.2 Descripción Textual de los Requerimientos Funcionales

ID del Requerimiento:	REQ-A01
Nombre del Requerimiento:	Permitir al usuario administrar las áreas de trabajo.
Características:	El sistema permitirá al administrador crear, modificar y eliminar áreas.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A02
Nombre del Requerimiento:	Permitir al usuario administrar los días inhábiles de trabajo.
Características:	El sistema permitirá al administrador crear, modificar y eliminar días inhábiles.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A03
Nombre del Requerimiento:	Permitir al usuario administrar todos los datos del empleado.
Características:	El sistema permitirá al administrador crear, modificar y eliminar datos de los empleados.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A04
Nombre del Requerimiento:	Permitir al usuario administrar los horarios de trabajo.
Características:	El sistema permitirá al administrador modificar los horarios de trabajo.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A05
Nombre del Requerimiento:	Permitir al usuario administrar los movimientos de registro de los empleados.
Características:	El sistema permitirá al administrador crear,

	modificar y eliminar los movimientos de los empleados.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A06
Nombre del Requerimiento:	Permitir al usuario administrar los permisos de los empleados.
Características:	El sistema permitirá al administrador crear, modificar y eliminar los permisos de los empleados.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A07
Nombre del Requerimiento:	Permitir al usuario administrar las planillas de sueldos y salarios.
Características:	El sistema permitirá al administrador crear, modificar y eliminar planillas de sueldos y salarios.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A08
Nombre del Requerimiento:	Permitir al usuario administrar los puestos de trabajo.
Características:	El sistema permitirá al administrador crear, modificar y eliminar los puestos de trabajo.
Atributo: Prioridad • Normal	

ID del Requerimiento:	REQ-A09
Nombre del Requerimiento:	Permitir al usuario administrar los usuarios del sistema.
Características:	El sistema permitirá al administrador crear, modificar y eliminar los usuarios de acceso al sistema.
Atributo: Prioridad • Normal	

II.1.1.8 Glosario

II.1.1.8.1 Introducción

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada de acuerdo al proyecto.

II.1.1.8.2 Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo de un sistema para PROLAC. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores que se da en el proyecto.

II.1.1.8.3 Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos. De tal modo que la terminología empleada en PROLAC, el sistema y sus diferentes áreas, se refleja con claridad en este documento.

II.1.1.8.4 Referencias

El presente glosario hace referencia a los siguientes documentos:

- Documento Plan de Desarrollo Software del Proyecto.
- Documento Visión del Proyecto.
- Documentos de Especificación de Casos de Uso del Proyecto.

II.1.1.8.5 Organización del glosario

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional de la lengua español.

II.1.1.8.6 Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo del sistema.

Anticipo.- Se refiere a los anticipos que se puede proporcionar a los empleados.

Áreas.- Es la sección donde trabaja o desempeña su labor el personal.

Cargar detalle de Planilla de Sueldos.- Se carga un detalle de una planilla cancelada al personal.

Cargar detalle de Planilla de Aguinaldos.- Se carga un detalle de una planilla de Aguinaldo cancelada al personal.

Contrato.- Se refiere a los contratos realizados con los empleados.

Crear Reportes de Asistencia.- Es la creación de reportes de acuerdo a la selección del usuario.

Día Inhábil.- Se refiere a los días que el sistemas no controlara la asistencia.

Edición Manual.- Se refiere a listar todos los eventos que el personal a registrado.

Eliminar Anticipo.- Se refiere a dar de baja a un anticipo para que ya no pueda ser usada.

Eliminar Área.- Se refiere a dar de baja a un área para que ya no pueda ser usada.

Eliminar Contrato.- Se refiere a dar de baja a un contrato para que ya no pueda ser usado.

Eliminar Datos Día Inhábil.- Se refiere a dar de baja a un día inhábil para que ya no pueda ser usado.

Eliminar Empleado.- Se refiere a dar de baja a un empleado para que ya no pueda ser usado.

Eliminar Evento Edición Manual.- Se refiere a dar de baja a una edición manual para que ya no pueda ser usada.

Eliminar Horario.- Se refiere a dar de baja a un horario para que ya no pueda ser usado.

Eliminar Parámetro.- Se refiere a dar de baja a un parámetro para que ya no pueda ser usado.

Eliminar Permiso y Vacación.- Se refiere a dar de baja a un permiso o vacación para que ya no pueda ser usado.

Eliminar Puesto.- Se refiere a dar de baja a un puesto para que ya no pueda ser usado.

Eliminar Usuario.- Se refiere a dar de baja a un usuario para que ya no pueda ser usado.

Empleados.- Miembros de la empresa para quienes esta diseñado el control de asistencia.

Filtrar Área.- El sistema tendrá la capacidad de buscar y listar los datos de las áreas.

Filtrar Puesto.- El sistema tendrá la capacidad de buscar y listar los datos de los puestos.

Filtrar Usuario.- El sistema tendrá la capacidad de buscar y listar los datos de los usuarios.

Guardar Cambios Usuarios Módulos.- El sistema guarda los cambios efectuados por el usuario.

Horarios.- Se refiere a listar todos los días hábiles de trabajo con sus horarios de entrada y salida.

Imprimir Anticipo.- El sistema tendrá la capacidad de imprimir los reportes de las anticipos existentes en la base de datos.

Imprimir Área.- El sistema tendrá la capacidad de imprimir los reportes de las áreas existentes en la base de datos.

Imprimir Contrato.- El sistema tendrá la capacidad de imprimir los contratos de los empleados existentes en la base de datos.

Imprimir Credencial Empleado.- El sistema tendrá la capacidad de imprimir la credencial del empleado existente en la base de datos.

Imprimir Datos Día Inhábil.- El sistema tendrá la capacidad de imprimir los reportes de los días inhábiles existentes en la base de datos.

Imprimir Empleado.- El sistema tendrá la capacidad de imprimir los reportes de los empleados existentes en la base de datos.

Imprimir Eventos Edición Manual.- El sistema tendrá la capacidad de imprimir los reportes de los eventos de edición manual existentes en la base de datos.

Imprimir Permiso y Vacación.- El sistema tendrá la capacidad de imprimir los reportes de permisos y vacaciones existentes en la base de datos.

Imprimir Planilla de Sueldos.- El sistema tendrá la capacidad de imprimir las planillas de sueldos de los empleados.

Imprimir Planilla de Aguinaldos.- El sistema tendrá la capacidad de imprimir las planillas de aguinaldos de los empleados.

Imprimir Puesto.- El sistema tendrá la capacidad de imprimir los datos de los puestos existentes en la base de datos.

Imprimir Usuario.- El sistema tendrá la capacidad de imprimir los datos de los usuarios existentes en la base de datos.

Modificar Anticipo.- Se refiere a la modificación de los datos de los anticipos.

Modificar Área.- Se refiere a la modificación de los datos de las áreas.

Modificar Contrato.- Se refiere a la modificación de los datos de los contratos.

Modificar Datos Día Inhábil.- Se refiere a la modificación de los datos de los días inhábiles.

Modificar Empleado.- Se refiere a la modificación de los datos de los empleados.

Modificar Evento Edición Manual.- Se refiere a la modificación de los datos de los eventos de edición manual.

Modificar Parámetro.- Se refiere a la modificación de los datos de los parámetros.

Modificar Permiso y Vacación.- Se refiere a la modificación de los datos de los permisos y vacaciones.

Modificar Puesto.- Se refiere a la modificación de los datos de los puestos.

Modificar Usuario.- Se refiere a la modificación de los datos de los usuarios.

Nueva Anticipo.- Es la creación de un nuevo anticipo en el sistema informático para su uso posterior.

Nueva Área.- Es la creación de una nueva área en el sistema informático para su uso posterior.

Nuevo Contrato.- Es la creación de un nuevo contrato en el sistema informático para su uso posterior.

Nuevo Día Inhábil.- Es la creación de un nuevo día inhábil en el sistema informático para su uso posterior.

Nuevo Empleado.- Es la creación de un nuevo empleado en el sistema informático para su uso posterior.

Nuevo Evento Edición Manual.- Es la creación de un nuevo evento de edición manual en el sistema informático para su uso posterior.

Nueva Parámetro.- Es la creación de un nuevo parámetro en el sistema informático para su uso posterior.

Nuevo Permiso y Vacación.- Es la creación de un nuevo permiso o vacación en el sistema informático para su uso posterior.

Nueva Planilla de Sueldos.- Es la creación de una nueva planilla de sueldos de los empleados en el sistema informático para su uso posterior.

Nueva Planilla de Aguinaldos.- Es la creación de una nueva planilla de aguinaldo de los empleados en el sistema informático para su uso posterior.

Nuevo Puesto.- Es la creación de un nuevo puesto en el sistema informático para su uso posterior.

Nuevo Usuario.- Es la creación de un nuevo usuario en el sistema informático para su uso posterior.

Permisos y Vacaciones.- Se refiere a los permisos y vacaciones que están registrados en el sistema, para que no realice el control de asistencia de los empleados.

Planilla de Sueldos.- Se refiere a las planillas de sueldos y salarios de los empleados que el sistema emite.

Planilla de Aguinaldos.- Se refiere a las planillas de aguinaldos de los empleados que el sistema emite.

Puestos.- Es el puesto donde trabaja o desempeña su labor el personal.

Reporte de Asistencia.- El sistema tendrá la capacidad de generar reportes de asistencia de los empleados para su uso posterior.

Usuarios.- Son las personas que interactúan con el sistema.

Usuarios Módulos.- Se refiere a los módulos que el usuario puede tener acceso en el sistema.

II.1.1.9 Visión

II.1.1.9.1 Introducción

Esta funcionalidad se basa principalmente en el apoyo al área Administrativa de PROLAC, en la automatización y gestión en el registro de datos y utilización de estos en las actividades rutinarias que esta empresa realiza, además de brindar una mas rápida y confiable comunicación de PROLAC al control de asistencia del personal.

Apoyando también a para que sean capaces de atender los distintos requerimientos que realizan a diario.

II.1.1.9.1.1 Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel, llevando a un desempeño optimo de las características del Sistema automatizado de una empresa privada. El documento se centra en la funcionalidad requerida por los participantes en el proyecto y los usuarios finales de la empresa.

Los detalles de cómo el Sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.

II.1.1.9.1.2 Alcance

El documento Visión se ocupa, como ya se ha apuntado, a la automatización de los diferentes registros de control de asistencia del personal. Dicho Sistema será desarrollado por el grupo de desarrollo de software.

El sistema permitirá al Gerente y Secretarias de PROLAC controlar todo lo relativo a la gestión de información de control de asistencia del personal.

II.1.1.9.2 Definiciones, Acrónimos y Abreviaciones

RUP: Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.

II.1.1.9.3 Referencias

- Glosario.
- Plan de desarrollo de software.
- RUP (Rational Unified Process).
- Diagrama de Casos de Uso.

II.1.1.9.4 Posicionamiento

II.1.1.9.4.1 Oportunidad del Negocio

Este Sistema permitirá realizar gestión de control de la asistencia del personal, registrando toda esta información en una base de datos que permitirá la reutilización de estos registros en las diferentes actividades que se realizan en PROLAC.

II.1.1.9.4.2 Sentencia que define el problema

El problema de	Controlar coordinar la gestión de control de asistencia del personal, generar planillas de sueldos y salarios.
Afecta a	Gerente General e PROLAC Secretarias de PROLAC Personal de PROLAC
El impacto asociado es	Almacenar toda la información referente a controlar la asistencia del personal, facilitar el acceso a datos e una manera mas rápida y confiable, que esta información sea accesible y actualizada, para así tener mayor rapidez y mejor control de las actividades que estas realizan a diario, es un proceso prácticamente moroso y molesto de realizar en el caso de que no este automatizado.
Una solución	Automatizar los procesos, usando un sistema con una base de datos

adecuada seria	accesible y generar interfaces a dicha base de datos

Tabla N° 13: Sentencia que define el problema

II.1.1.9.4.3 Sentencia que define la posición del Producto

Para	1.- Gerente General de PROLAC 2.- Secretarias encargadas de generar reportes y planillas.
Quienes	1.- Tiene que administrar y llevar un control adecuado y confiable de las actividades comunes en la empresa. 2.- Controlan la asistencia del personal.
El nombre del producto	Es una herramienta software.
Que	Almacena la información necesaria para optimizar las actividades e la empresa.
No como	El Sistema actual que manual.
Nuestro producto	Permite optimizar las distintas actividades e la empresa mediante una interfaz grafica sencilla y amigable. Además proporciona un acceso rápido y actualizado y confiable a la información desde cualquier punto que tenga acceso autorizado al Sistema de Información.

Tabla N° 14: Sentencia que define la posición del Producto

II.1.1.9.5 Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades e los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los Usuarios del Sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que estos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que estos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por que estos requisitos son necesarios.

II.1.1.9.5.1 Resumen de Stakeholders

Nombre	Descripción	Responsabilidades
Sr. Víctor Cardozo	Encargado de PROLAC	El Stakeholder realiza: Representa a todos los usuarios posibles de la empresa. Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades-

Tabla N° 15: Resumen de Stakeholders

II.1.1.9.5.2 Resumen de Usuarios

Nombre	Descripción	Stakeholder
Encargado de	Supervisor del buen	Encargado de

Administración	funcionamiento del Sistema, gestión de todas las actividades y encargado de generar planillas de pago.	Administración de PROLAC
Secretaria	Responsable de generar los reportes en general	Secretaria

Tabla N° 16: Resumen de Usuarios

II.1.1.9.5.3 Entorno de Usuario

Los usuarios entraran al sistema identificándose sobre un ordenador, tras este paso entraran a la parte de aplicación diseñada para cada uno según su papel en la empresa.

II.1.1.9.5.4 Perfil de los Stakeholders

Representante del área técnica y sistemas de información

Representante	Sr. Víctor Cardozo
Descripción	Representante de PROLAC
Tipo	Gerente General
Responsabilidades	Encargado de mostrar las necesidades de cada usuario del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema.
Criterio de Éxito	A definir por el cliente.
Grado de participación	Revisión de requerimientos, estructura del sistema, brindar información.

Comentarios	Ninguno.

Tabla N° 17: Representante del Área Técnica y Sistemas de Información

II.1.1.9.5.5 Perfiles de Usuario

Secretarias

Representante	Secretaria encargada.
Descripción	Encargada de emitir reportes.
Tipo	Usuario casual del sistema.
Responsabilidades	Encargada de generar o emitir reportes de la asistencia del personal.
Criterio de Éxito	A definir por el cliente.
Grado de participación	A definir por el cliente.
Comentarios	Ninguno.

Tabla N° 18: Perfiles de usuario Secretarias

Personal

Representante	Personal.
Descripción	Personal.
Tipo	Personal de PROLAC.
Grado de participación	A definir por el cliente

Comentarios	Ninguno.

Tabla N° 19: Perfiles de usuario Personal

II.1.1.9.6 Descripción Global del Producto

II.1.1.9.6.1 Perspectiva del producto

El producto a desarrollar es un Sistema para PROLAC, con la intención de automatizar sus actividades más comunes. Las áreas a tratar por el Sistema son: La Administración de PROLAC y el Control de Asistencia del personal.

II.1.1.9.6.2 Resumen de Características

A continuación se mostrara un listado con los beneficios que obtendrá el cliente a partir del producto final:

Beneficio del cliente	Características que lo apoyan
Sistema automatizado para el Control de Asistencia del personal.	El sistema contara con un módulo para la administración de PROLAC.
Mayor facilidad para la gestión de los recursos humanos.	Base de Datos centralizada con la información de todo el personal.

Tabla N° 20: Resumen de Características

Gestión de Administración

Almacenar toda la información referente al personal, programación de descuentos, facilitar el acceso a datos de una manera más rápida y confiable, que esta información esté al instante accesible y actualizado en las oficinas de PROLAC.

Gestión de Secretarías

Tendrá el objetivo de facilitar datos actuales a las secretarías de PROLAC, pudiendo también estos emitir reportes de la asistencia del personal hacia la gerencia de PROLAC.

II.1.1.9.7 Seguimiento y Control del Proyecto

II.1.1.9.7.1 Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito.

II.1.1.9.7.2 Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por los integrantes del grupo.

II.1.1.9.7.3 Control de Calidad

Los defectos detectados en las revisiones tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión.

II.1.1.9.7.4 Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

II.1.1.9.7.5 Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de

Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto.

II.1.1.9.8 Referencias

- http://www.elguille.info/colabora/puntoNET/canchala_UML.htm
- <http://glosario.panamacom.com/>
- <http://www.ingenierosoftware.com/análisisydiseño/patrones-diseño.php>
- <http://www.itba.edu.ar/capis/webcapis/RGMITBA/articulosrgm/R-ITBA-22-estimacion.pdf>
- Libro UML gota a gota

II.1.2 MARCO TEORICO



II.1.2.1 Definiciones

A continuación se definen algunos términos que se manejarán a lo largo de todo el proyecto de desarrollo.

II.1.2.2 Sistema de Información Computarizado

Un Sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio, para proporcionar información básica para la conducción de la empresa. Esta información hace posible que las empresas lleven a cabo sus tareas con mayor calidad y facilidad.

Un Sistema de información realiza cuatro actividades básicas:

- **Entrada de Información:** Proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, diskettes, cintas magnéticas, código de barras, etc.
- **Almacenamiento de Información:** Es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión anterior.
- **Procesamiento de la Información:** Esta característica de los sistemas permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general en un año base.
- **Salida de Información:** Es la capacidad de un SI para sacar información procesada o bien datos de datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas y otros.

II.1.2.3 Sistemas de Gestión de Base de Datos

Los sistemas de gestión de base de datos (SGBD); son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Proporciona servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en base de datos de forma segura y eficiente. Un SGBD actúa como un intermediario entre las aplicaciones y los datos.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

Ventajas:

- Proveen facilidades para la manipulación de grandes volúmenes de datos.
Entre estas:
 - Simplifican la programación de chequeos de consistencia.
 - Manejando las políticas de respaldo adecuadas garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores en el disco, o hay muchos usuarios accediendo simultáneamente a los mismos datos, o se ejecutaron programas que no terminaron su trabajo correctamente, etc.
 - Permiten realizar modificaciones en la organización de los datos con un impacto mínimo en el código de los programas.
 - Permiten implementar un manejo centralizado de la seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- Las facilidades anteriores bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Inconvenientes:

1. Típicamente, es necesario disponer de una o mas personas que administren de la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o mas personas que administren de los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.
2. Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una planilla de cálculo.
3. Complejidad: Los SGBD son software muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
4. Tamaño: La complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder correr.
5. Coste del hardware adicional: Los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero

II.1.2.3.1 Funciones de los Sistemas de Gestión de Base de Datos

La función que desempeñan los SGBD, y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

II.1.2.3.2 Arquitectura de los Sistemas de Gestión de Base de Datos

Las bases de datos respetan la arquitectura de tres niveles definida, para cualquier tipo de base de datos, por el grupo ANSI/SPARC. En esta arquitectura la base de datos se divide en los niveles externo, conceptual e interno.

1. Nivel Interno: Es el nivel mas bajo de abstracción, y define como se almacenan los datos en el soporte físico, así como los métodos de acceso.
2. Nivel Conceptual: Es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.
3. Nivel Externo: Es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso.

II.1.2.4 Herramientas Tecnológicas de Desarrollo

II.1.2.4.1 JAVA

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma.

Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores del software.

II.1.2.4.2 iReport

La herramienta iReport es un constructor y diseñador de informes visual, poderoso, intuitivo y fácil de usar por JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc.

Características de iReport

- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los soporta internacionalización nativamente.
- Browser de la estructura del documento.
- Recopilador y exportador integrados.
- Soporta JavaBeans como orígenes de datos.
- Tiene asistentes como orígenes de datos.
- Tiene asistentes para las planillas.
- Facilidad de instalación.

II.1.2.4.2 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

Características

Algunas de sus principales características son:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversion) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por la tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee natividad soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foraneas (foreign keys).
- Disparadores (triggers): Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:
 - El nombre del disparador o trigger.
 - El momento en que el disparador debe arrancar.
 - El evento del disparador deberá activarse sobre.
 - La tabla donde el disparador se activara.
 - La frecuencia de la ejecución.
 - La función que podría ser llamada.

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

II.1.2.5 Especificación de Requerimientos

II.1.2.5.1 Metodologías de Desarrollo

Son métodos que indican como hacer mas eficiente el desarrollo de sistemas de información. Para ello suelen estructurar en fases la vida de dichos sistemas con el fin de facilitar su planificación, desarrollo y mantenimiento.

Los principales objetivos de una metodología de desarrollo son:

- ✦ Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en sí.
- ✦ Satisfacer las necesidades de los usuarios del sistema.
- ✦ Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- ✦ Ajustarse a los plazos y costes previstos en la planificación.
- ✦ Generar de forma adecuada la documentación asociada a los sistemas.
- ✦ Facilitar el mantenimiento posterior de los sistemas.

II.1.2.5.2 Proceso de Ingeniería de Software

Ingeniería de software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software.

- ✦ Mejorar la calidad de los productos de software.
- ✦ Aumentar la productividad y trabajo de los ingenieros del software.
- ✦ Facilitar el control del proceso de desarrollo de software.
- ✦ Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- ✦ Definir una disciplina que garantice la producción y mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

II.1.2.5.3 Lenguaje Unificado de Modelado UML

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

II.1.2.5.3.1 ¿Que es UML?

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica como crear y leer los modelos, pero no dice como crearlos. Esto último es el objetivo de las metodologías de desarrollo. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✦ **Visualizar:** UML permite expresar de una forma grafica un sistema de forma que otro lo puede entender.
- ✦ **Especificar:** UML permite especificar cuales son las características de un sistema antes de su construcción.
- ✦ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✦ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

II.1.2.5.3.2 ¿Que forma un modelo de UML?

Un modelo UML esta compuesto por tres clases de bloques de construcción:

- ✦ **Elementos:** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- ✦ **Relaciones:** Relacionan los elementos entre si.
- ✦ **Diagramas:** Son colecciones de elementos con sus relaciones. Veamos con mayor detalle los diagramas de UML.

Elementos

- ⤴ Pueden ser estructurales, de comportamiento de agrupación o de anotación.
- ⤴ Elementos estructurales: Actores, casos de uso, clases, objetos.
- ⤴ Elementos de comportamiento: Mensajes.
- ⤴ Elementos de agrupación: Paquetes.

Relaciones

Dependencia

- Es una relación semántica entre dos elementos, en la cual un cambio en un elemento puede afectar a la semántica de otro elemento.
- Existen varios tipos de dependencia predefinidas que se indican mediante estereotipos <<extend>> o <<include>> para casos de uso.

Asociación

- Es una relación estructural entre dos elementos, que describen las conexiones entre ellos (suele ser bidireccional).
- Puede presentarse como agregación o composición.

Generalización

- Es una relación taxonómica entre un elemento mas general (el padre) y un elemento mas específico (el hijo).
- Se utiliza tanto en diagramas de clases como en diagramas de casos de uso.

II.1.2.5.3.3 Diagramas de UML

Los diagramas proporcionan múltiples perspectivas del sistema bajo análisis. El modelo subyacente integra estas perspectivas de forma que se puede construir un sistema auto consistente. Estos diagramas, junto con la documentación de soporte, es lo primero que se ve el diseñador.

Por otro lado podemos ver el modelo de una forma estática o de una forma dinámica. Estas perspectivas nos dan la siguiente clasificación:

- ▲ Modelo Estático (Estructural):
 - Diagrama de Despliegue.
 - Diagrama de Componentes.
 - Diagrama de Clases.
 - Diagrama de Objetos.

- ▲ Modelo Dinámico (Componentes)
 - Diagrama de Estados.
 - Diagrama de Actividades.
 - Diagrama de Secuencia.
 - Diagrama de Colaboración.
 - Diagrama de Casos de Uso.

II.1.2.5.3.3.1 Diagramas de Clases

El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Este es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas “conocen” a que otras clases o que clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

II.1.2.5.3.3.2 Diagrama de Componentes

Un 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, librerías 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 sistemas.

Debido a que estos son más parecidos a los diagramas de casos de uso estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la

organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

II.1.2.5.3.3.3 Diagramas de Despliegue

El diagrama de despliegue es un tipo de diagrama que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Además muestra un conjunto de nodos y sus relaciones. Los diagramas de despliegue se utilizan para describir la vista de despliegue estática de un sistema. Los diagramas de despliegue se relacionan con los diagramas de componentes, ya que un nodo normalmente incluye uno o más componentes.

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

II.1.2.5.3.3.4 Diagramas de Casos de Uso

El diagrama de casos de uso representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo.

II.1.2.5.3.3.5 Diagramas de Interacción (Secuencia, Colaboración)

Los diagramas de interacción se dan este nombre colectivo a los diagramas de secuencia y los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se puede convertir de uno a otro sin pérdida de información.

- ⤴ Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Un diagrama de secuencia presenta un conjunto de objetos y los mensajes enviados y recibidos por ellos. Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden

representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.

- ✦ Un diagrama de colaboración es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos.

II.1.2.5.3.3.6 Diagramas de Actividades

Un diagrama de actividades es un tipo especial de diagrama de estados que muestra el flujo secuencial o ramificado de actividades en un sistema. Conjunto de actividades son especialmente importantes para modelar la función del sistema, así como para resaltar el flujo de control entre objetos.

II.1.2.6 Metodología de Desarrollo

II.1.2.6.1 Metodología de Desarrollo RUP (Rational Unified Process)

El Proceso Unificado Racional (Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Por lo que (RUP) describe como aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados “mejores prácticas” pues son utilizados en la industria por organizaciones exitosas.

La metodología RUP, Rational Unified Process, divide en 4 fases el desarrollo del software:

- ✧ **Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.
- ✧ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- ✧ **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ✧ **Transmisión:** El objetivo es llegar a obtener el reléase del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- ✧ Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- ✧ Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- ✧ Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- ✧ implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ✧ Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

Disciplina de Soporte

- ✧ Configuración y Administración del cambio: Guardando todas las versiones del proyecto.
- ✧ Administrando el proyecto: Administrando horarios y recursos.
- ✧ Ambiente: Administrando el ambiente de desarrollo.
- ✧ Distribución: Hacer todo lo necesario para la salida del proyecto.

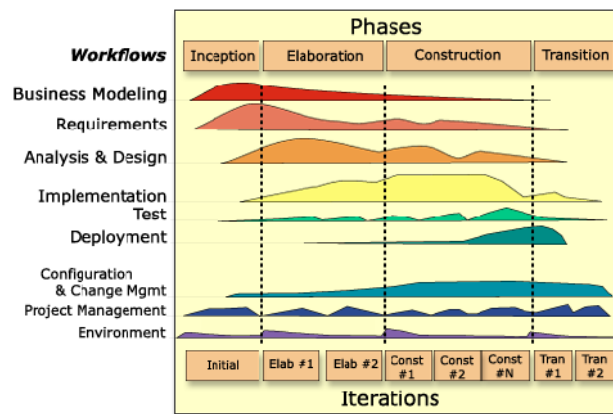


Figura N° 5 Fases de RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

II.1.3 INGENIERIA DE REQUERIMIENTOS



II.1.3.1 Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas, etc.). Permite situar al Sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos para este modelo.

II.1.3.1.1 Caso de Uso del Negocio General

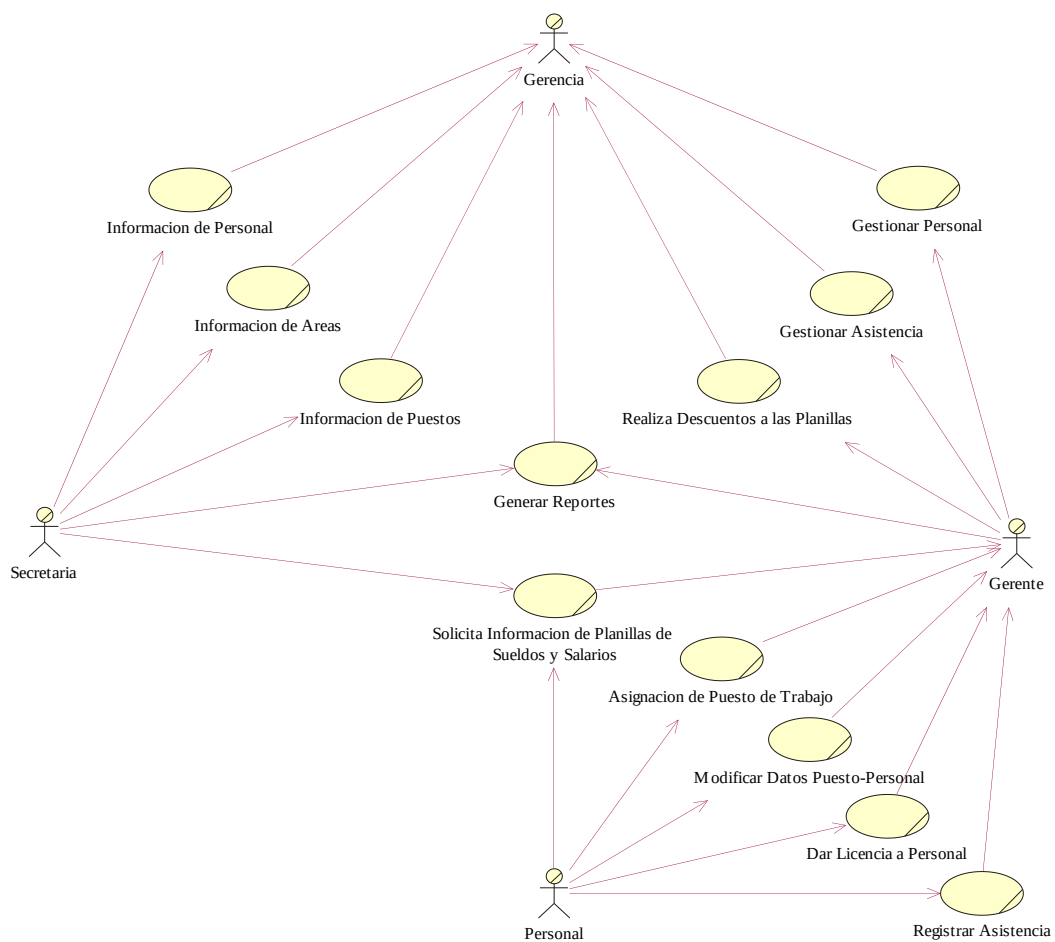


Figura N° 6: Caso de Uso del Negocio General

II.1.3.1.2 Descripción de Casos de Uso del Negocio

NOMBRE DEL CASO DE USO DEL NEGOCIO: Gestionar Personal

ACTORES: Gerente

DEFINICION: Este caso de uso engloba todo lo referente al registro de empleados de la empresa.

NOTAS:

- El gerente tiene la función de llevar un registro y reportes de datos de los empleados existentes.
- El gerente podrá añadir, modificar y eliminar datos de los empleados.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Gestionar Asistencia

ACTORES: Gerente

DEFINICION: Este caso de uso engloba todo lo referente al control de la asistencia del personal de la empresa.

NOTAS:

- El gerente tiene la función de llevar un control y generar reportes de la asistencia del personal existente.
- El gerente podrá añadir, modificar y eliminar los datos de la asistencia del personal.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Realiza Descuento a las Planillas

ACTORES: Gerente

DEFINICION: Este caso de uso engloba todo lo referente a generar los descuentos a las planillas de salario de los empleados.

NOTAS:

- El gerente tiene la función de realizar los descuentos a las planillas de sueldos y salarios.
- El gerente podrá añadir, modificar y eliminar los descuentos a las planillas de salarios.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Generar Reportes**ACTORES: Gerente y Secretaria**

DEFINICION: Este caso de uso engloba todo lo referente a los reportes que el Gerente o Secretaria deben generar.

NOTAS:

- El Gerente o Secretaria debe generar reportes de los diferentes procesos y actividades que se llevan a cabo para llevar un control de dichos procesos.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Información de Áreas**ACTORES: Gerente y Secretaria**

DEFINICION: Este caso de uso engloba todo lo referente a ver la información de las áreas de trabajo de la empresa.

NOTAS:

- El Gerente o Secretaria tiene entre sus registros información referente a las áreas que corresponden a la empresa para llevar un mejor control.
- Además de que ambos tienen entre sus funciones la de proporcionar información de las vacancias en los diferentes áreas.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Información de Puestos**ACTORES: Gerente y Secretaria**

DEFINICION: Este caso de uso engloba todo lo referente a ver la información de los puestos de trabajo de la empresa.

NOTAS:

- El Gerente o Secretaria tiene entre sus registros información referente a los puestos que corresponden a la empresa para llevar un mejor control.
- Además de que ambos tienen entre sus funciones la de proporcionar información de las vacancias en los diferentes puestos.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Información del personal**ACTORES: Gerente y Secretaria****DEFINICION:** Este caso de uso engloba todo lo referente a ver la información del personal de la empresa.**NOTAS:**

- El Gerente o Secretaria tiene entre sus registros información referente a los empleados para llevar un mejor control de asistencia.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Asignación de Licencias y vacaciones**ACTORES: Gerente****DEFINICION:** Este caso de uso engloba todo lo referente a la asignación de licencias y permisos al personal.**NOTAS:**

- El Gerente tiene la función de dar licencia o vacación al personal que lo solicite.
- Luego de una asignación de licencia o vacación el gerente registre el informe de la solicitud.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Asignación de Puestos de Trabajo**ACTORES: Gerente****DEFINICION:** Este caso de uso engloba todo lo referente a la asignación de puestos de trabajo.**NOTAS:**

- El Gerente tiene la función de asignar un puesto de trabajo en la empresa.
- El gerente lleva un registro de datos del personal vigente.

NOMBRE DEL CASO DE USO DEL NEGOCIO: Registrar Asistencia**ACTORES: Personal**

DEFINICION: Este caso de uso engloba todo lo referente al registro de asistencia del personal.

NOTAS:

- El personal tiene la función de registrar el ingreso y salida para el control de la asistencia.

II.1.3.2 Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados en el caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

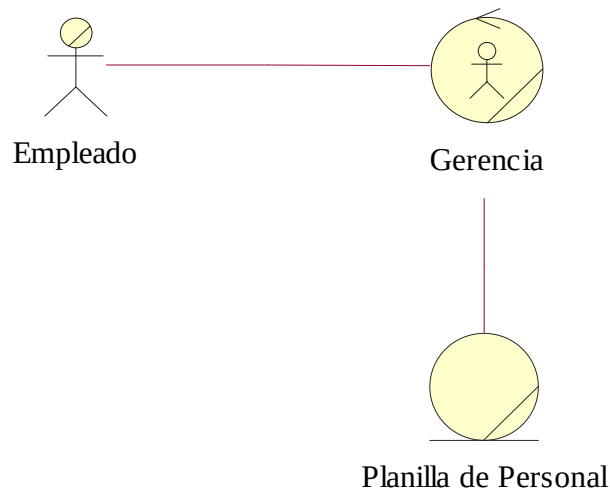
Caso de Uso: Registrar Personal

Figura N° 7: Modelo del Negocio del Caso de Uso Registro de Personal

Caso de Uso: Asignar Personal a Puesto

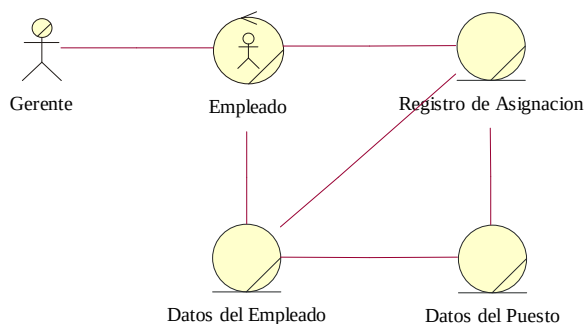


Figura N° 8: Modelo del Negocio del Caso de Uso Asignar Personal a Puesto

Caso de Uso: Ver Disponibilidad de Puesto

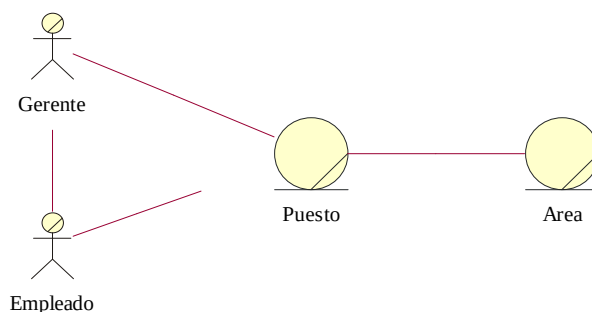


Figura N° 9: Modelo del Negocio del Caso de Uso Ver Disponibilidad de Puesto

Caso de Uso: Solicitar Puesto

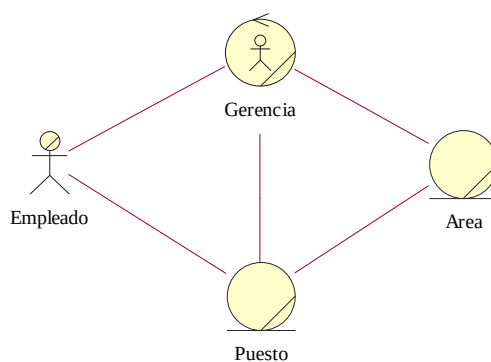


Figura N° 10: Modelo del Negocio del Caso de Uso Solicitar Puesto

II.1.3.3 Diagrama de Modelo de Casos de Uso

II.1.3.3.1 Introducción

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

II.1.3.3.2 Propósito

- ✦ Comprender la estructura y la dinámica del sistema deseado para la organización.
- ✦ Identificar posibles mejoras.

II.1.3.3.3 Alcance

- ✦ Describe los procesos de sistema y los usuarios.
- ✦ Identificar y definir los procesos del sistema según los objetivos de la organización.
- ✦ Definir un caso de uso para cada proceso del sistema (el diagrama de casos de uso puede mostrar el contexto y los límites de la organización).

II.1.3.3.4 Casos de Uso Configurar Conexión



Figura N° 11: Caso de Uso Configurar Conexión

II.1.3.3.5 Casos de Uso Configurar Conexión

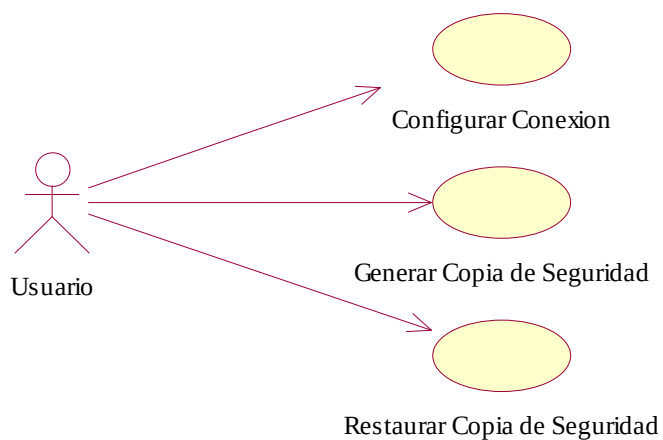


Figura N° 12: Caso de Uso Configurar Conexión

II.1.3.3.6 Ingresar al Sistema



Figura N° 13: Caso de Uso Ingresar al Sistema

II.1.3.3.7 MODULO I: SISTEMA

II.1.3.3.7.1 Casos de Uso General del Sistema

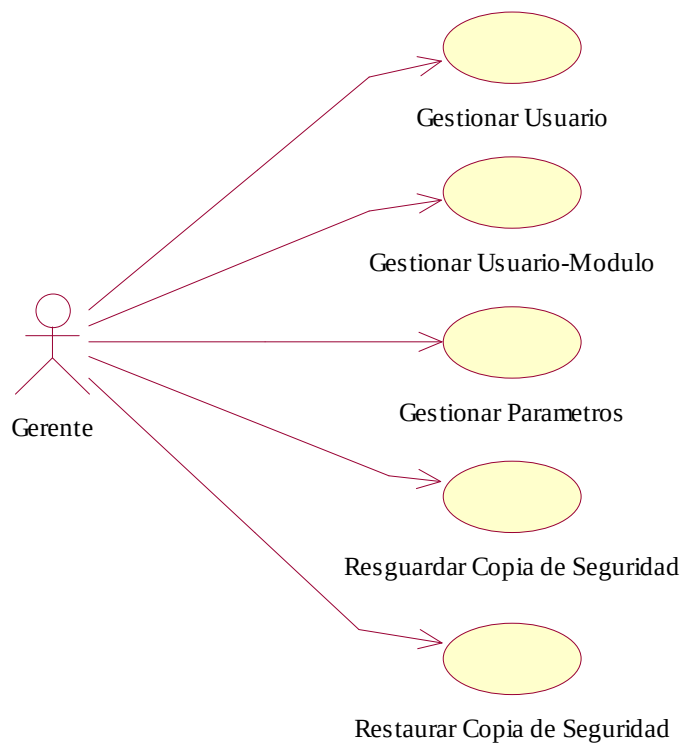


Figura N° 14: Caso de Uso General del Sistema

II.1.3.3.7.2 Gestionar Usuarios

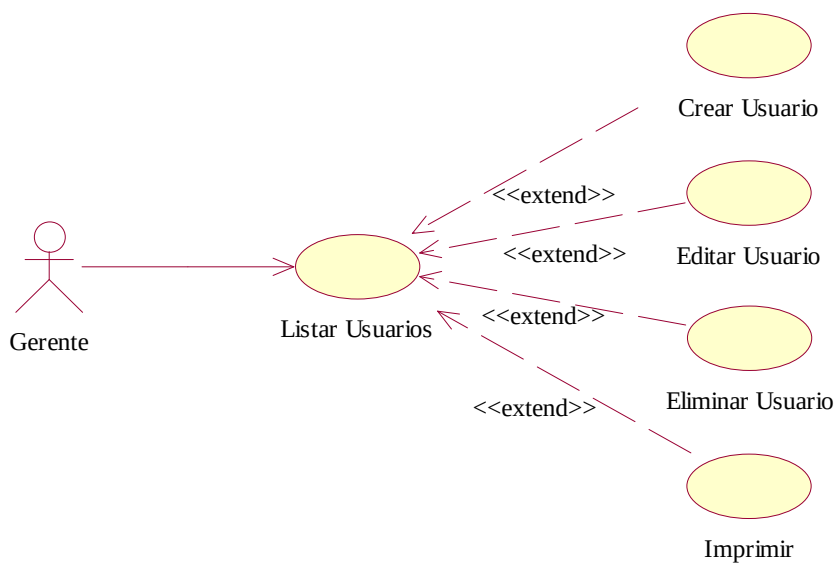


Figura N° 15: Caso de Uso Gestionar Usuarios

II.1.3.3.7.3 Gestionar Usuarios Módulos



Figura N° 16: Caso de Uso Gestionar Usuarios Módulos

II.1.3.3.7.4 Gestionar Parámetros

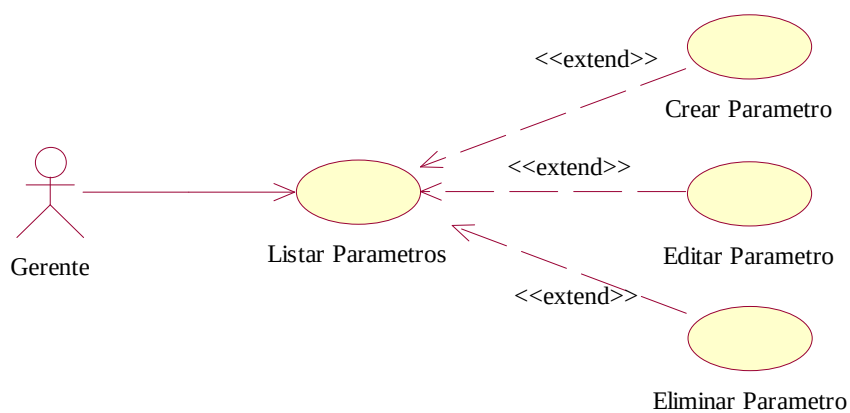


Figura N° 17: Caso de Uso Gestionar Parámetros

II.1.3.3.8 MODULO II: PERSONAL

II.1.3.3.8.1 Casos de Uso General del Personal

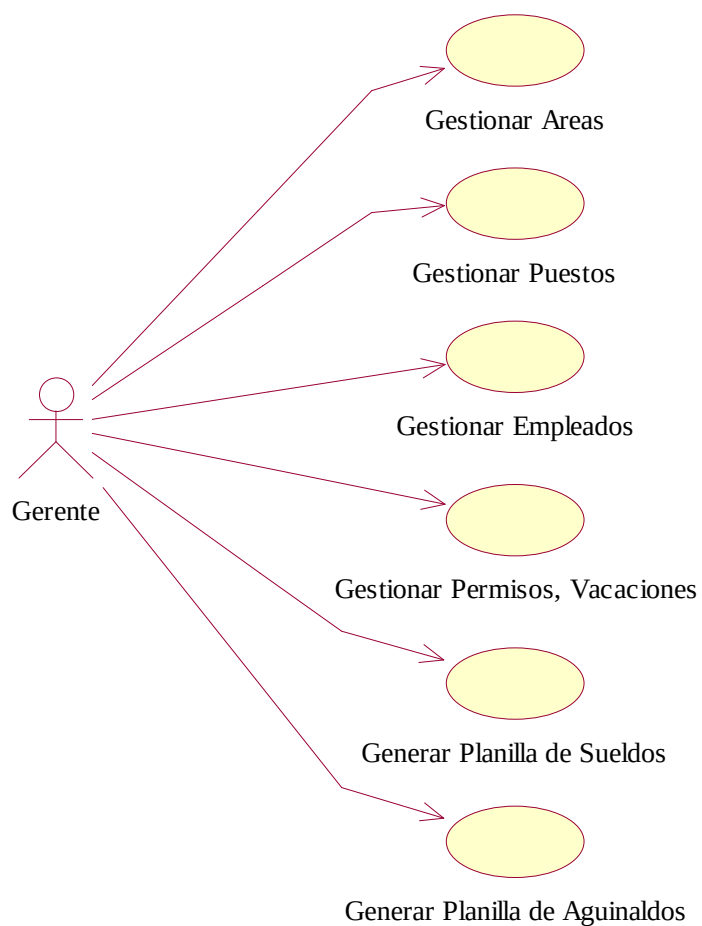


Figura N° 18: Caso de Uso General del Personal

II.1.3.3.8.2 Gestionar Empleados

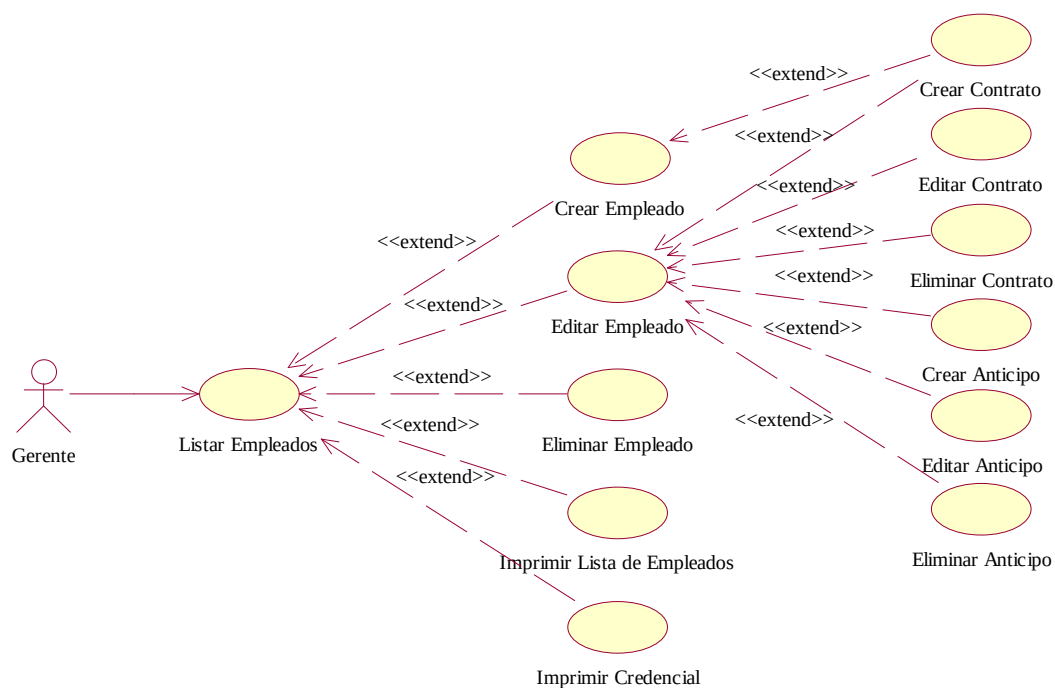


Figura N° 19: Caso de Uso Gestionar Empleados

II.1.3.3.8.3 Gestionar Áreas

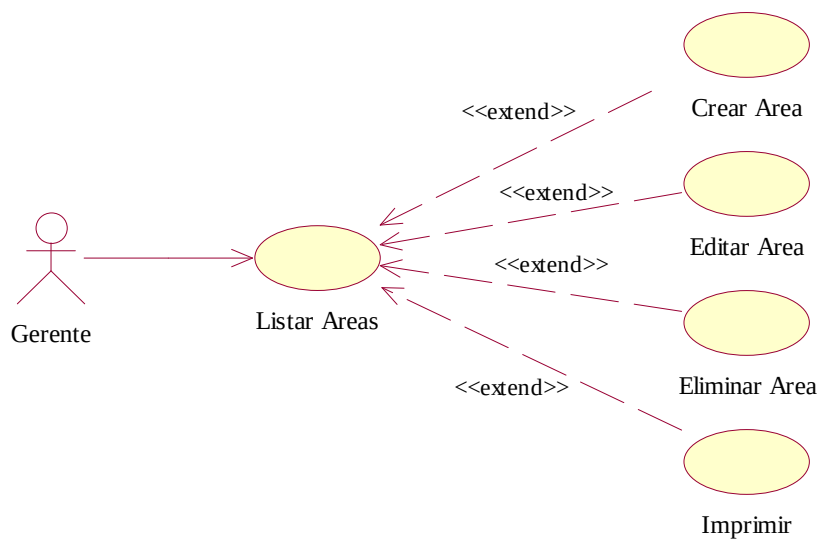


Figura N° 20: Caso de Uso Gestionar Áreas

II.1.3.3.8.4 Gestionar Puestos

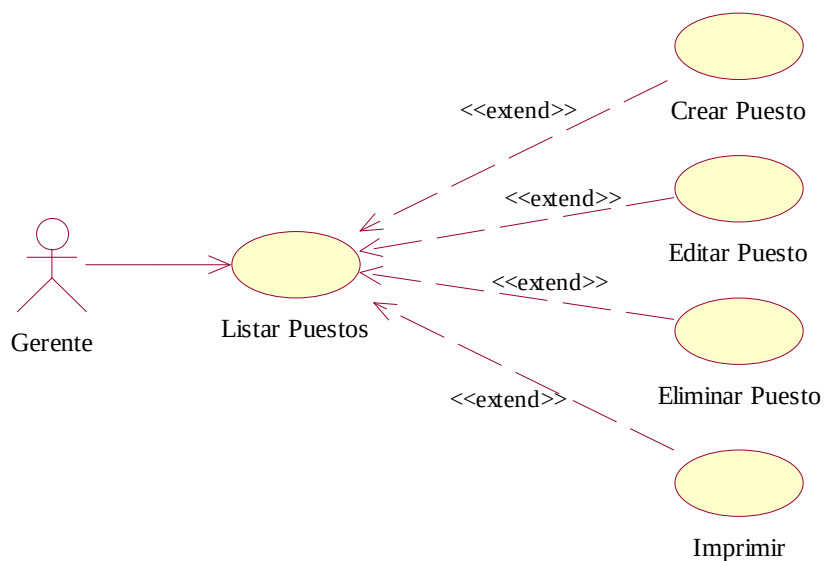


Figura N° 21: Caso de Uso Gestionar Puestos

II.1.3.3.8.5 Gestionar Permisos-Vacaciones

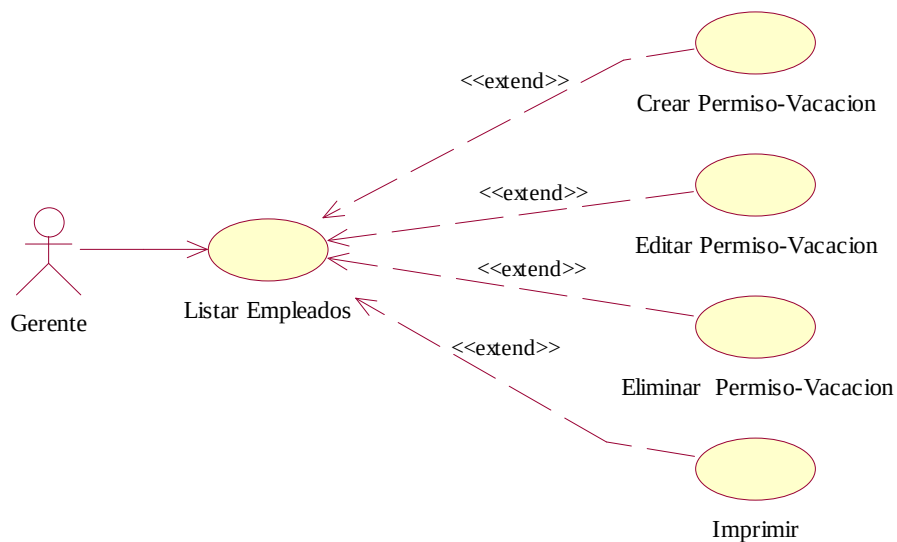


Figura N° 22: Caso de Uso Gestionar Permisos, Vacaciones

II.1.3.3.8.6 Generar Planillas de Sueldos

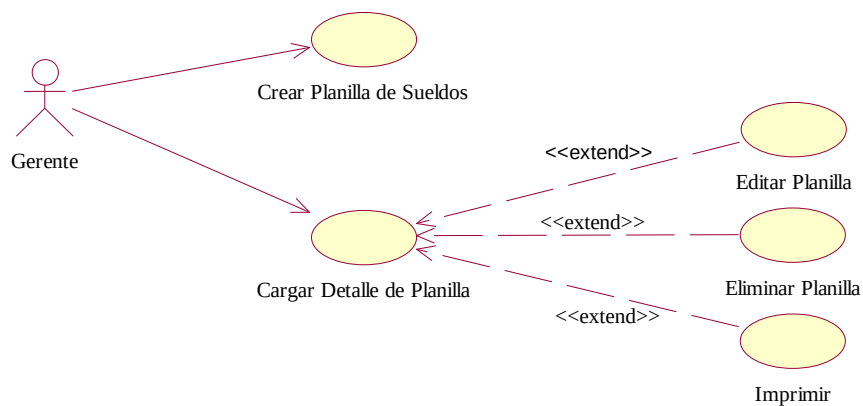


Figura N° 23: Caso de Uso Generar Planillas de Sueldos

II.1.3.3.8.7 Generar Planillas de Aguinaldos

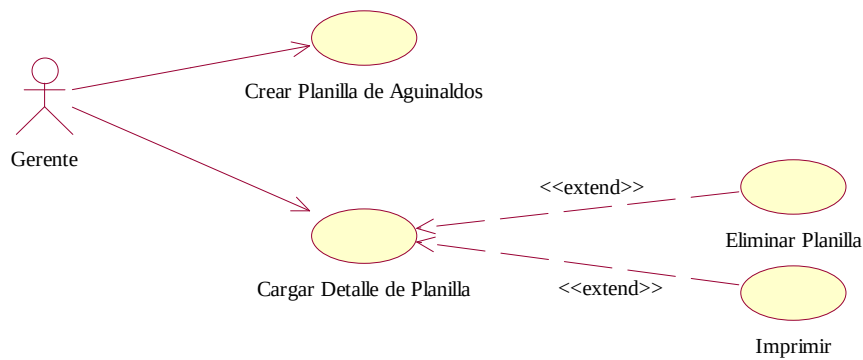


Figura N° 24: Caso de Uso Generar Planillas de Aguinaldos

II.1.3.3.9 MODULO III: CONTROL ASISTENCIA

II.1.3.3.9.1 Casos de Uso General de Control de Asistencia

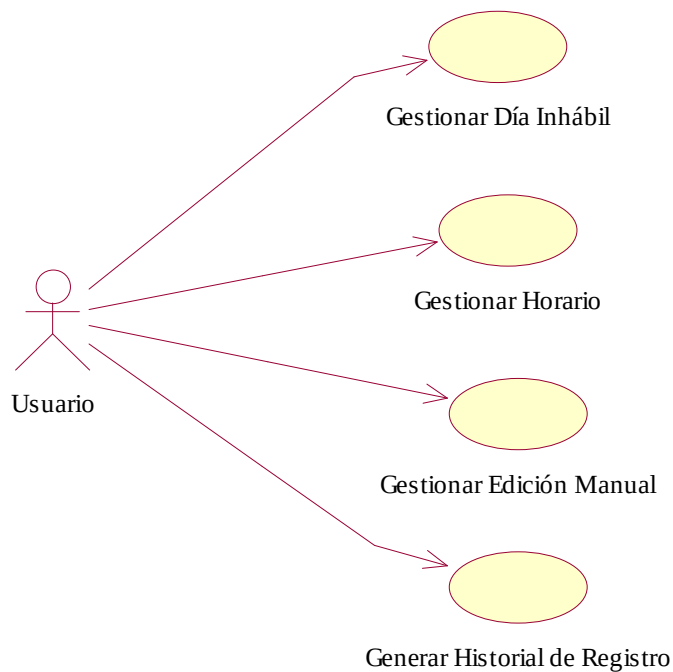


Figura N° 25: Caso de Uso Generales de Control de Asistencia

II.1.3.3.9.2 Gestionar Día Inhábil

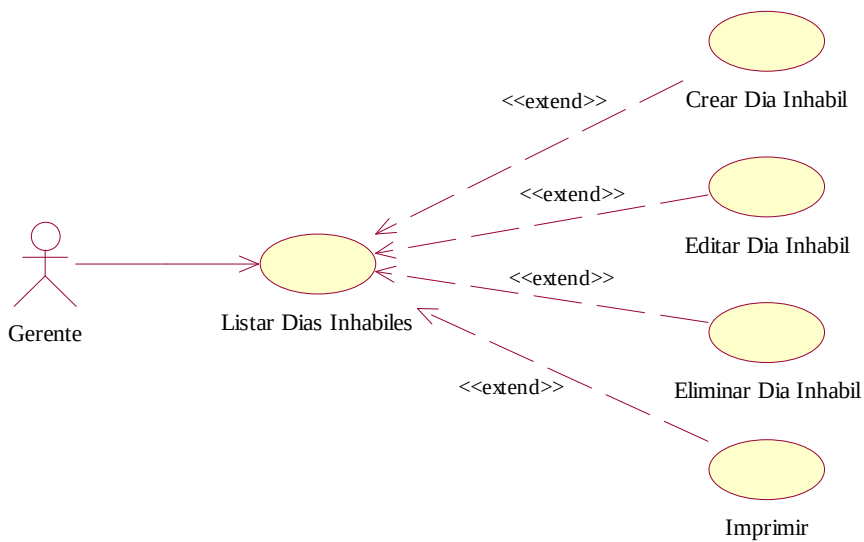


Figura N° 26: Caso de Uso Gestionar Día Inhábil

II.1.3.3.9.3 Gestionar Horario

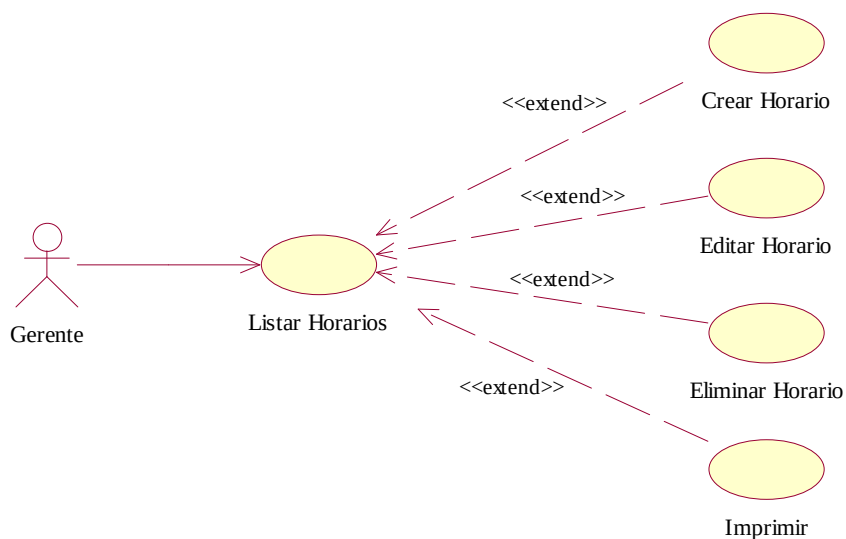


Figura N° 27: Caso de Uso Gestionar Horario

II.1.3.3.9.4 Gestionar Edición Manual

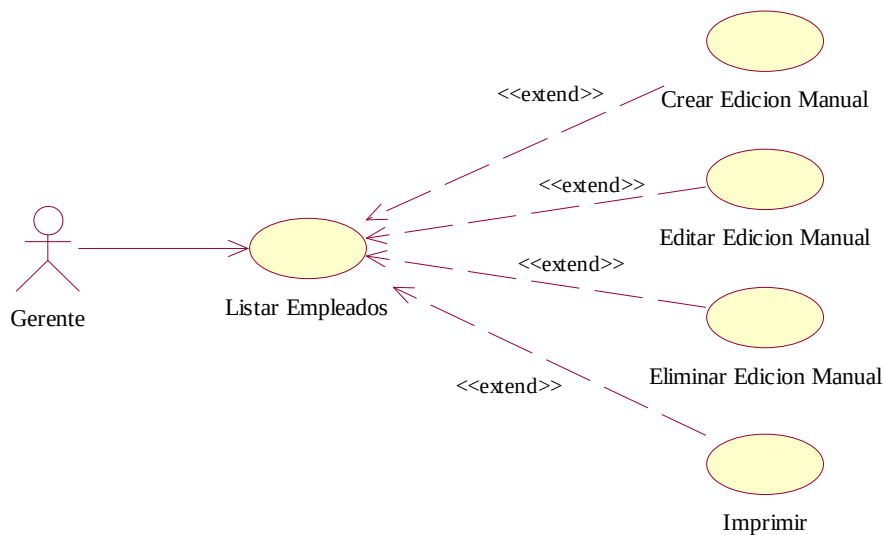


Figura N° 28: Caso de Uso Gestionar Edición Manual

II.1.3.3.9.5 Generar Historial de Registro

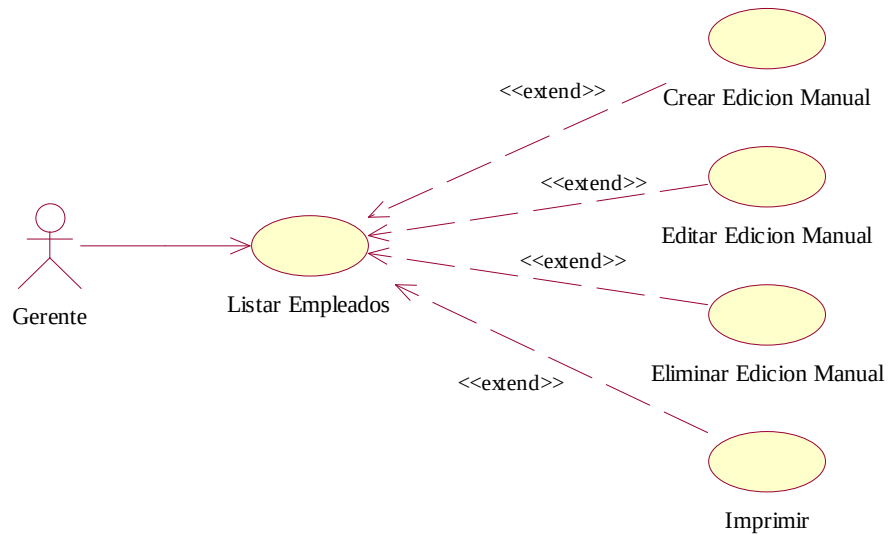


Figura N° 29: Caso de Uso Generar Historial de Registro

II.1.3.3.10 MODULO IV: REPORTE

II.1.3.3.10.1 Generar Reportes



Figura N° 30: Caso de Uso Generar Reportes

II.1.3.4 Especificaciones de Casos de Uso

Para los casos de uso que requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

CASO DE USO:	Configurar Conexión
ACTORES:	Usuario
DESCRIPCION:	Este caso de uso permite configurar la conexión en red.
FLUJO BASICO:	1.- El usuario introduce los datos. 2.- Si el usuario selecciona aceptar el sistema guarda la conexión.
PRECONDICIONES	Los datos introducidos deben ser validos. El sistema debe estar instalado en maquinas que estén conectados en red.
POSTCONDICIONES:	El sistema actualiza la base de datos.

Tabla N° 21: Caso de Uso Configurar Conexión

CASO DE USO:	Generar Copia de Seguridad
ACTORES:	Usuario
DESCRIPCION:	Este caso de uso permite generar una copia de seguridad de la base de datos.
FLUJO BASICO:	1.- El usuario ubica la dirección donde se resguardara la copia de seguridad de la base de datos.
PRECONDICIONES	-----
POSTCONDICIONES:	Los datos del Usuario quedan sesionados para su uso durante la iteración con el sistema.

Tabla N° 22: Caso de Uso Generar Copia de Seguridad

CASO DE USO:	Restaurar Copia de Seguridad
ACTORES:	Usuario
DESCRIPCION:	Este caso de uso permite restaurar una copia de seguridad de la base de datos.
FLUJO BASICO:	1.- El usuario ubica la dirección donde se encuentra la copia de seguridad de la base de datos.
PRECONDICIONES	Todas las aplicaciones del sistema deben estar cerradas.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 23: Caso de Uso Restaurar Copia de Seguridad

CASO DE USO:	Registrar Asistencia
ACTORES:	Usuario
DESCRIPCION:	Este caso de uso permite registrar nuevos eventos
FLUJO BASICO:	1.- Los usuarios introduce los datos del Evento 2.- Si el usuario selecciona aceptar el sistema agrega el evento.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 24: Caso de Uso Registrar Asistencia

CASO DE USO:	Ingresar al sistema
ACTORES:	Usuario
DESCRIPCION:	Este caso de uso permite el acceso al sistema.
FLUJO BASICO:	1.- El usuario ingresa su usuario y clave. 2.- El sistema verifica los datos del usuario. 3.- Si los datos son correctos el sistema despliega el menú principal del usuario, caso contrario lanza un mensaje de

	error.
PRECONDICIONES	-----
POSTCONDICIONES:	Los datos del Usuario quedan sesionados para su uso durante la iteración con el sistema.

Tabla N° 25: Caso de Uso Ingresar al sistema

II.1.3.4.1 ESPECIFICACION TEXTUAL DE CASO DE USO MODULO I

CASO DE USO:	Listar Usuarios
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de usuarios con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción usuario. 2.- El sistema lista los usuarios existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de usuarios debe quedar actualizada después de alguna operación.

Tabla N° 26: Caso de Uso Listar Usuarios

CASO DE USO:	Crear Usuario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos usuarios.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo usuario. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del usuario. 4.- Si el gerente selecciona aceptar el sistema agrega el usuario, caso contrario regresa a la pantalla anterior.

PRECONDICIONES	Los datos introducidos deben ser validos. Se debe asignar la clave obligatoriamente.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 27: Caso de Uso Crear Usuario

CASO DE USO:	Editar Usuario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los usuarios.
FLUJO BASICO:	1.- El gerente selecciona un usuario. 2.- El gerente selecciona la opción editar usuario. 3.- El sistema muestra un formulario con los datos actuales del usuario seleccionado. 4.- El gerente introduce nuevos datos del usuario. 5.- Si el gerente acepta la modificación el sistema modifica los datos del usuario, caso contrario regresa a la pantalla del listado de los usuarios.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 28: Caso de Uso Editar Usuario

CASO DE USO:	Eliminar Usuario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los usuarios.
FLUJO BASICO:	1.- El gerente selecciona un usuario. 2.- El gerente selecciona la opción eliminar usuario. 3.- Si el gerente acepta la eliminación el sistema elimina

	los datos del usuario, caso contrario regresa a la pantalla de listado de los usuarios.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del usuario.
POSTCONDICIONES:	Es sistema realizara un borrado lógico del usuario. El sistema actualiza la base de datos.

Tabla N° 29: Caso de Uso Eliminar Usuario

CASO DE USO:	Usuarios Módulos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite habilitar módulos a los usuarios.
FLUJO BASICO:	1.- El gerente debe seleccionar un usuario. 2.- El gerente selecciona los módulos que desea que el usuario tenga acceso. 3.- Si el gerente acepta habilitar los módulos el sistema guarda los datos, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 30: Caso de Uso Usuarios Módulos

CASO DE USO:	Listar Parámetros
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de parámetros (Descuento General, Bono Antigüedad y Vacación) con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción parámetro. 2.- El sistema lista los parámetros existentes.

	3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de parámetros debe quedar actualizada después de alguna operación.

Tabla N° 31: Caso de Uso Listar Parámetros

CASO DE USO:	Crear Parámetro
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos parámetros (Descuento General, Bono Antigüedad y Vacación).
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo parámetro. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del parámetro. 4.- Si el gerente selecciona aceptar el sistema agrega el parámetro, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 32: Caso de Uso Crear Parámetro

CASO DE USO:	Editar Parámetro
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los parámetros.
FLUJO BASICO:	1.- El gerente selecciona un parámetro. 2.- El gerente selecciona la opción editar parámetro. 3.- El sistema muestra un formulario con los datos

	actuales del parámetro seleccionado. 4.- El gerente introduce nuevos datos del parámetro. 5.- Si el gerente acepta la modificación el sistema modifica los datos del parámetro, caso contrario regresa a la pantalla del listado de los parámetros.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 33: Caso de Uso Editar Parámetro

CASO DE USO:	Eliminar Parámetro
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los parámetros.
FLUJO BASICO:	1.- El gerente selecciona un parámetro. 2.- El gerente selecciona la opción eliminar parámetro. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del parámetro, caso contrario regresa a la pantalla de listado de los parámetros.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del parámetro.
POSTCONDICIONES:	Es sistema realizara un borrado lógico del parámetro. El sistema actualiza la base de datos.

Tabla N° 34: Caso de Uso Eliminar Parámetro

II.1.3.4.2 ESPECIFICACION TEXTUAL DE CASO DE USO MODULO II

CASO DE USO:	Listar Empleados
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de empleados con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción personal. 2.- El sistema lista los empleados existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de empleados debe quedar actualizada después de alguna operación

Tabla N° 35: Caso de Uso Listar Empleados

CASO DE USO:	Crear Empleado
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos empleados.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo empleado. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del empleado. 4.- Si el gerente selecciona aceptar el sistema agrega el empleado, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 36: Caso de Uso Crear Empleado

CASO DE USO:	Editar Empleado
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los empleados.
FLUJO BASICO:	1.- El gerente selecciona un empleado. 2.- El gerente selecciona la opción editar empleado. 3.- El sistema muestra un formulario con los datos del empleado seleccionado. 4.- El gerente introduce nuevos datos del empleado. 5.- Si el gerente acepta la modificación el sistema modifica los datos del empleado, caso contrario regresa a la pantalla del listado de los empleados.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 37: Caso de Uso Editar Empleados

CASO DE USO:	Eliminar Empleado
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los empleados.
FLUJO BASICO:	1.- El gerente selecciona un empleado. 2.- El gerente selecciona la opción eliminar empleado. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del empleado, caso contrario regresa a la pantalla del listado de los empleados.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para la eliminación del empleado.
POSTCONDICIONES:	El sistema realizara un borrado lógico del empleado. El sistema actualiza la base de datos.

Tabla N° 38: Caso de Uso Eliminar Empleados

CASO DE USO:	Listar Contratos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de contratos con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la pestaña Contratos de la pantalla Nuevo o Modificar Empleado. 2.- El sistema lista los contratos existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema. Si el gerente está creando un nuevo empleado solo estará habilitada la opción nuevo contrato. Si el gerente está modificando los datos de un empleado estarán habilitadas las opciones: nuevo, editar y eliminar.
POSTCONDICIONES:	La lista de contratos debe quedar actualizada después de alguna operación

Tabla N° 39: Caso de Uso Listar Contratos

CASO DE USO:	Crear Contrato
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos contratos.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo contrato. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del contrato. 4.- Si el gerente selecciona aceptar el sistema agrega el contrato, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos. Para crear un nuevo contrato debe haber finalizado el

	anterior contrato.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 40: Caso de Uso Crear Contrato

CASO DE USO:	Editar Contrato
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los contratos.
FLUJO BASICO:	1.- El gerente selecciona un contrato. 2.- El gerente selecciona la opción editar contrato. 3.- El sistema muestra un formulario con los datos del contrato seleccionado. 4.- El gerente introduce nuevos datos del contrato. 5.- Si el gerente acepta la modificación el sistema modifica los datos del contrato, caso contrario regresa a la pantalla del listado de los contratos.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 41: Caso de Uso Editar Contratos

CASO DE USO:	Eliminar Contrato
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los contratos.
FLUJO BASICO:	1.- El gerente selecciona un contrato. 2.- El gerente selecciona la opción eliminar contrato. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del contrato, caso contrario regresa a la pantalla

	del listado de los contratos.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para la eliminación del contrato.
POSTCONDICIONES:	El sistema realizara un borrado lógico del contrato. El sistema actualiza la base de datos.

Tabla N° 42: Caso de Uso Eliminar Contratos

CASO DE USO:	Listar Anticipos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de anticipos con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la pestaña Anticipos de la pantalla Modificar Empleado. 2.- El sistema lista los anticipos existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema. Solo si el gerente está modificando los datos de un empleado estarán habilitadas las opciones: nuevo, editar y eliminar.
POSTCONDICIONES:	La lista de anticipos debe quedar actualizada después de alguna operación.

Tabla N° 43: Caso de Uso Listar Anticipos

CASO DE USO:	Crear Anticipo
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos anticipos.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo anticipo.

	2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del anticipo. 4.- Si el gerente selecciona aceptar el sistema agrega el anticipo, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos. El total del anticipo no debe superar el saldo de su sueldo mensual del empleado.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 44: Caso de Uso Crear Anticipo

CASO DE USO:	Editar Anticipo
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los anticipos.
FLUJO BASICO:	1.- El gerente selecciona un anticipo. 2.- El gerente selecciona la opción editar anticipo. 3.- El sistema muestra un formulario con los datos del anticipo seleccionado. 4.- El gerente introduce nuevos datos del anticipo. 5.- Si el gerente acepta la modificación el sistema modifica los datos del anticipo, caso contrario regresa a la pantalla del listado de los contratos.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 45: Caso de Uso Editar Anticipo

CASO DE USO:	Eliminar Anticipo
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los anticipos.
FLUJO BASICO:	1.- El gerente selecciona un anticipo. 2.- El gerente selecciona la opción eliminar anticipo. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del anticipo, caso contrario regresa a la pantalla del listado de los anticipos.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para la eliminación del anticipo.
POSTCONDICIONES:	El sistema realizara un borrado lógico del anticipo. El sistema actualiza la base de datos.

Tabla N° 46: Caso de Uso Eliminar Anticipo

CASO DE USO:	Listar Áreas
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de áreas con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción área. 2.- El sistema lista las áreas existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de áreas debe quedar actualizada después de alguna operación.

Tabla N° 47: Caso de Uso Listar Áreas

CASO DE USO:	Crear Área
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevas áreas.
FLUJO BASICO:	1.- El gerente selecciona la opción nueva área. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos de la zona. 4.- Si el gerente selecciona aceptar el sistema agrega el área, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 48: Caso de Uso Crear Área

CASO DE USO:	Editar Área
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de las áreas.
FLUJO BASICO:	1.- El gerente selecciona un área. 2.- El gerente selecciona la opción editar área. 3.- El sistema muestra un formulario con los datos del área seleccionada. 4.- El gerente introduce nuevos datos del área. 5.- Si el gerente acepta la modificación el sistema modifica los datos del área, caso contrario regresa a la pantalla del listado de las áreas.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 49: Caso de Uso Editar Área

CASO DE USO:	Eliminar Área
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de las áreas.
FLUJO BASICO:	1.- El gerente selecciona un área. 2.- El gerente selecciona la opción eliminar área. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del área, caso contrario regresa a la pantalla del listado de las áreas.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del área.
POSTCONDICIONES:	El sistema realizara un borrado lógico del área. El sistema actualiza la base de datos.

Tabla N° 50: Caso de Uso Eliminar Área

CASO DE USO:	Listar Puestos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de puestos con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción puestos. 2.- El sistema lista los puestos existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de puestos debe quedar actualizada después de alguna operación.

Tabla N° 51: Caso de Uso Listar Puestos

CASO DE USO:	Crear Puesto
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos puestos.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo puesto. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos del puesto. 4.- Si el gerente selecciona aceptar el sistema agrega el puesto, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 52: Caso de Uso Crear Puesto

CASO DE USO:	Editar Puesto
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de los puestos.
FLUJO BASICO:	1.- El gerente selecciona un puesto. 2.- El gerente selecciona la opción editar puesto. 3.- El sistema muestra un formulario con los datos del puesto seleccionada. 4.- El gerente introduce nuevos datos del puesto. 5.- Si el gerente acepta la modificación el sistema modifica los datos del puesto, caso contrario regresa a la pantalla del listado de los puestos.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 53: Caso de Uso Editar Puesto

CASO DE USO:	Eliminar puesto
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los puestos.
FLUJO BASICO:	1.- El gerente selecciona un puesto. 2.- El gerente selecciona la opción eliminar puesto. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del puesto, caso contrario regresa a la pantalla del listado de los puestos.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del puesto.
POSTCONDICIONES:	El sistema realizara un borrado lógico del puesto. El sistema actualiza la base de datos.

Tabla N° 54 Caso de Uso Eliminar Puesto

CASO DE USO:	Listar Empleados
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de permiso, vacación con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción Permisos-Vacaciones. 2.- El sistema lista los empleados existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de empleados debe quedar actualizada después de alguna operación.

Tabla N° 55: Caso de Uso Listar Empleados

CASO DE USO:	Crear Permiso-Vacación
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos Permiso-vacación.
FLUJO BASICO:	1.- El gerente selecciona un empleado del registro 2.- El gerente selecciona la opción nuevo permiso-vacación. 3.- El sistema muestra un formulario de entrada de datos. 4.- El gerente introduce los datos del permiso. 5.- Si el gerente seleccionar aceptar el sistema agrega el permiso-vacación caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 56: Caso de Uso Crear Permiso-Vacación

CASO DE USO:	Editar Permiso-Vacación
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar datos de los permisos-vacaciones.
FLUJO BASICO:	1.- El gerente selecciona un empleado. 2.- El gerente selecciona la opción editar permiso-vacación. 3.- El sistema muestra un formulario con los datos del permiso-vacación seleccionado. 4.- El gerente introduce nuevos datos del permiso-vacación. 5.- Si el gerente acepta la modificación el sistema

	modifica los datos del permiso-vacación, caso contrario regresa a la pantalla del listado de los empleados.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 57: Caso de Uso Editar Permiso-Vacación

CASO DE USO:	Eliminar Permiso-Vacación
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los permisos-vacaciones.
FLUJO BASICO:	1.- El gerente selecciona un permiso-vacación. 2.- El gerente selecciona la opción eliminar permiso-vacación. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del permiso-vacación, caso contrario regresa a la pantalla permiso-vacación.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del permiso.
POSTCONDICIONES:	El sistema realizara un borrado lógico del permiso-vacación. El sistema actualiza la base de datos.

Tabla N° 58 Caso de Uso Eliminar Permiso-Vacación

CASO DE USO:	Crear Planilla de Sueldos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite generar nuevas planillas de sueldos.
FLUJO BASICO:	1.- El gerente selecciona la opción nueva planilla de

	sueldos. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos de la planilla de sueldos. 4.- Si el gerente selecciona aceptar el sistema agrega la planilla de sueldos, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 59: Caso de Uso Crear Planilla de Sueldos

CASO DE USO:	Cargar Detalle de Planilla
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite cargar las planillas de Sueldos almacenadas en la base de datos, tiene las opciones de editar planilla y eliminar planilla.
FLUJO BASICO:	1.- El gerente selecciona el mes, año. 2.- El sistema lista la planilla de sueldos existente. 3.- El gerente selecciona la opción cargar detalle de planilla.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de planillas debe quedar actualizada después de alguna operación.

Tabla N° 60: Caso de Uso Cargar Detalle Planilla

CASO DE USO:	Eliminar Planilla de Sueldos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar las planillas de sueldos.

FLUJO BASICO:	1.- El gerente selecciona la planilla de sueldos. 2.- El gerente selecciona la opción eliminar planilla. 3.- Si el gerente acepta la eliminación el sistema elimina los datos de la planilla, caso contrario regresa a la pantalla planillas de sueldos.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado de la planilla.
POSTCONDICIONES:	El sistema realizara un borrado lógico de la planilla. El sistema actualiza la base de datos.

Tabla N° 61: Caso de Uso Eliminar Planilla de Sueldos

CASO DE USO:	Crear Planilla de Aguinaldos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite generar nueva planilla de aguinaldos.
FLUJO BASICO:	1.- El gerente selecciona la opción nueva planilla de aguinaldos. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos de la planilla de aguinaldos. 4.- Si el gerente selecciona aceptar el sistema agrega la planilla de aguinaldos, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 62: Caso de Uso Crear Planilla de Aguinaldos

CASO DE USO:	Cargar Detalle de Planilla
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite cargar las planillas de Aguinaldos almacenadas en la base de datos, tiene la opción de eliminar planilla.
FLUJO BASICO:	1.- El gerente selecciona el año. 2.- El sistema lista la planilla de aguinaldo existente. 3.- El gerente selecciona la opción cargar detalle de planilla.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de planillas debe quedar actualizada después de alguna operación.

Tabla N° 63: Caso de Uso Cargar Detalle Planilla

CASO DE USO:	Eliminar Planilla de Aguinaldos
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar las planillas de Aguinaldos.
FLUJO BASICO:	1.- El gerente selecciona la planilla de Aguinaldos. 2.- El gerente selecciona la opción eliminar planilla. 3.- Si el gerente acepta la eliminación el sistema elimina los datos de la planilla, caso contrario regresa a la pantalla planillas de Aguinaldos.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado de la planilla.
POSTCONDICIONES:	El sistema realizara un borrado lógico de la planilla. El sistema actualiza la base de datos.

Tabla N° 64: Caso de Uso Eliminar Planilla de Aguinaldos

II.1.3.4.3 ESPECIFICACION TEXTUAL DE CASO DE USO MODULO III

CASO DE USO:	Listar Día Inhábil
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de días inhábiles con la opción de nuevo, editar, eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción Día Inhábil. 2.- El sistema lista los Días Inhábiles existentes. 3.- El sistema lista las opciones de nuevo, editar, eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema.
POSTCONDICIONES:	La lista de Día Inhábil debe quedar actualizada después de alguna operación.

Tabla N° 65: Caso de Uso Listar Día Inhábil

CASO DE USO:	Crear Día Inhábil
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos Días Inhábiles
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo Día Inhábil 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos de Día Inhábil. 4.- Si el gerente seleccionar aceptar el sistema agrega el Día Inhábil, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 66: Caso de Uso Crear Día Inhábil

CASO DE USO:	Editar Día Inhábil
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de Día Inhábil.
FLUJO BASICO:	1.- El gerente selecciona la opción editar Día Inhábil. 2.- El sistema muestra un formulario con los datos del Día Inhábil seleccionado. 3.- El gerente introduce nuevos datos del Día Inhábil. 4.- Si el gerente acepta la modificación el sistema modifica los datos del Día Inhábil, caso contrario regresa a la pantalla del listado de los días inhábiles.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 67: Caso de Uso Editar Día Inhábil

CASO DE USO:	Eliminar Día Inhábil
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos de los Día Inhábil.
FLUJO BASICO:	1.- El gerente selecciona un Día Inhábil. 2.- El gerente selecciona la opción eliminar Día Inhábil. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del Día Inhábil, caso contrario regresa a la pantalla Día Inhábil.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del Día Inhábil.
POSTCONDICIONES:	El sistema realizara un borrado lógico del Día Inhábil. El sistema actualiza la base de datos.

Tabla N° 68: Caso de Uso Eliminar Día Inhábil

CASO DE USO:	Listar Horarios
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de horarios con la opción de nuevo, editar, eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción horarios. 2.- El sistema lista los Horarios existentes. 3.- El sistema lista las opciones de nuevo, editar, eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema.
POSTCONDICIONES:	La lista de Horario debe quedar actualizada después de alguna operación.

Tabla N° 69: Caso de Uso Listar Horarios

CASO DE USO:	Crear Horario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos Horarios.
FLUJO BASICO:	1.- El gerente selecciona la opción nuevo Horario. 2.- El sistema muestra un formulario de entrada de datos. 3.- El gerente introduce los datos de Horario. 4.- Si el gerente seleccionar aceptar el sistema agrega el Horario, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 70: Caso de Uso Crear Horario

CASO DE USO:	Editar Horario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos de Horario.

FLUJO BASICO:	1.- El gerente selecciona la opción editar Horario. 2.- El sistema muestra un formulario con los datos del Horario seleccionado. 3.- El gerente introduce nuevos datos del Horario. 4.- Si el gerente acepta la modificación el sistema modifica los datos del Horario, caso contrario regresa a la pantalla del listado de los Horarios.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros

Tabla N° 71: Caso de Uso Editar Horario

CASO DE USO:	Eliminar Horario
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos del Horario.
FLUJO BASICO:	1.- El gerente selecciona un Horario. 2.- El gerente selecciona la opción eliminar Horario. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del Horario, caso contrario regresa a la pantalla Horario.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del Horario.
POSTCONDICIONES:	El sistema realizara un borrado lógico del Horario. El sistema actualiza la base de datos.

Tabla N° 72: Caso de Uso Eliminar Horario

CASO DE USO:	Listar Empleados
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite la gestión de Edición Manual

	con las opciones de nuevo, editar y eliminar.
FLUJO BASICO:	1.- El gerente selecciona la opción Edición Manual 2.- El sistema lista los empleados existentes. 3.- El sistema lista las opciones de nuevo, editar y eliminar.
PRECONDICIONES	El gerente tiene que haber pasado el control de acceso al sistema
POSTCONDICIONES:	La lista de eventos debe quedar actualizada después de alguna operación.

Tabla N° 73: Caso de Uso Listar Empleados

CASO DE USO:	Crear Evento
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite agregar nuevos Eventos
FLUJO BASICO:	1.- El gerente selecciona un empleado del registro. 2.- El gerente selecciona la opción nuevo Evento. 3.- El sistema muestra un formulario de entrada de datos. 4.- El gerente introduce los datos del Evento. 5.- Si el gerente selecciona aceptar el sistema agrega el evento, caso contrario regresa a la pantalla anterior.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 74: Caso de Uso Crear Evento

CASO DE USO:	Editar Evento
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite editar los datos del Evento.
FLUJO BASICO:	1.- El gerente selecciona un empleado.

	2.- El gerente selecciona la opción editar Evento. 3.- El sistema muestra un formulario con los datos del evento seleccionado. 4.- El gerente introduce nuevos datos del evento. 5.- Si el gerente acepta la modificación el sistema modifica los datos del evento, caso contrario regresa a la pantalla de Edición Manual.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 75: Caso de Uso Editar Evento

CASO DE USO:	Eliminar Evento
ACTORES:	Gerente
DESCRIPCION:	Este caso de uso permite eliminar los datos del Evento.
FLUJO BASICO:	1.- El gerente selecciona un Evento. 2.- El gerente selecciona la opción eliminar evento. 3.- Si el gerente acepta la eliminación el sistema elimina los datos del evento, caso contrario regresa a la pantalla Edición Manual.
PRECONDICIONES	Antes de la eliminación el gerente debe aceptar el mensaje de confirmación para el borrado del evento.
POSTCONDICIONES:	El sistema realizara un borrado lógico del evento. El sistema actualiza la base de datos.

Tabla N° 76: Caso de Uso Eliminar Evento

II.1.3.4.3 ESPECIFICACION TEXTUAL DE CASO DE USO MODULO IV

CASO DE USO:	Generar Reportes
ACTORES:	Usuario (Gerente, Secretaria)
DESCRIPCION:	Este caso de uso permite generar diferentes tipos de reportes.
FLUJO BASICO:	1.- El usuario selecciona un tipo de reporte, fecha, área, puesto y empleado(s). 2.- El sistema muestra un formulario de entrada de datos. 3.- Si el usuario seleccionar aceptar el sistema agrega el despliegue del reporte.
PRECONDICIONES	Los datos introducidos deben ser validos.
POSTCONDICIONES:	El sistema actualiza la base de datos con los nuevos registros.

Tabla N° 77: Caso de Uso Generar Reportes

II.1.4 ANALISIS Y DISEÑO



II.1.4.1 Modelado de Diagrama de Actividades

II.1.4.1.1 Introducción

Una vista de actividades está compuesta por los diagramas de actividades que reflejan las actividades de computación implicadas en la ejecución del sistema.

Los diagramas de actividades se utilizan para modelar los aspectos dinámicos de un sistema, esto implica modelar los pasos secuenciales de un proceso.

II.1.4.1.2 Propósito

- ✧ Comprender la estructura y la dinámica del sistema deseado para la organización.
- ✧ Identificar posibles mejoras.

II.1.4.1.3 Alcance

- ✧ Describe los procesos de sistema.
- ✧ Identificar y definir los procesos de los casos de uso según los objetivos de la organización.
- ✧ Definir un diagrama de actividad para cada caso de uso del sistema.

II.1.4.1.4 Diagramas de Actividades

II.1.4.1.4.1 Registrar Asistencia

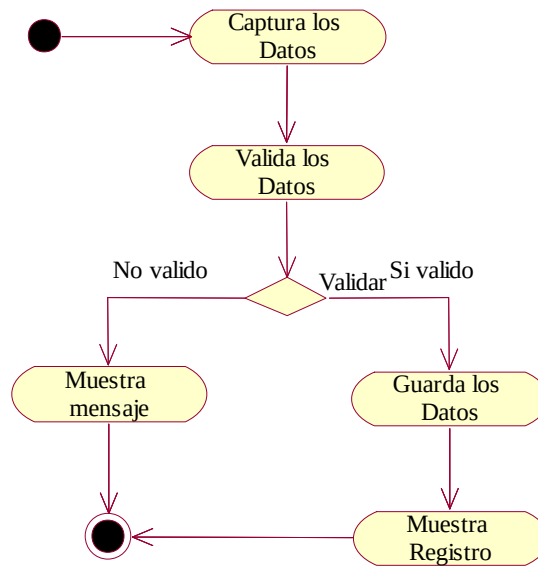


Figura N° 31: Diag. de Actividades Registrar Asistencia

II.1.4.1.4.2 Ingresar Sistema

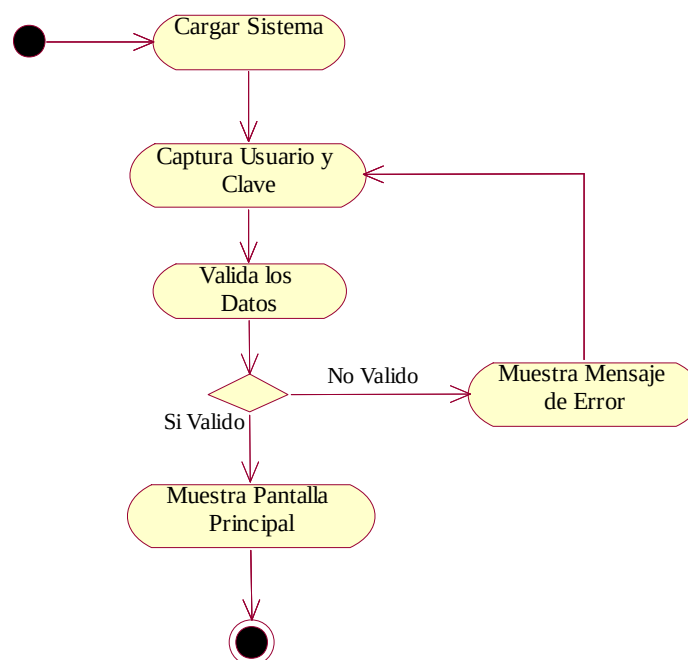


Figura N° 32: Diag. de Act. Ingresar Sistema

II.1.4.1.4.3 MODULO I: SISTEMA

II.1.4.1.4.3.1 GESTIONAR USUARIOS

Listar Usuarios

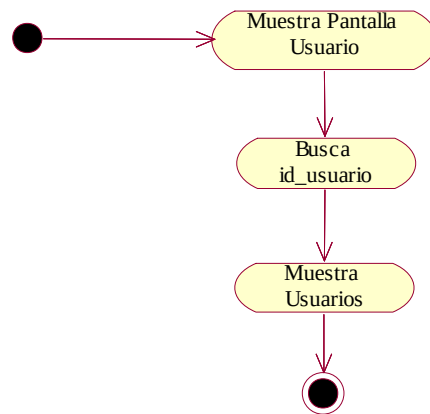


Figura N° 33: Diag. de Act. Listar Usuarios

Crear Usuario

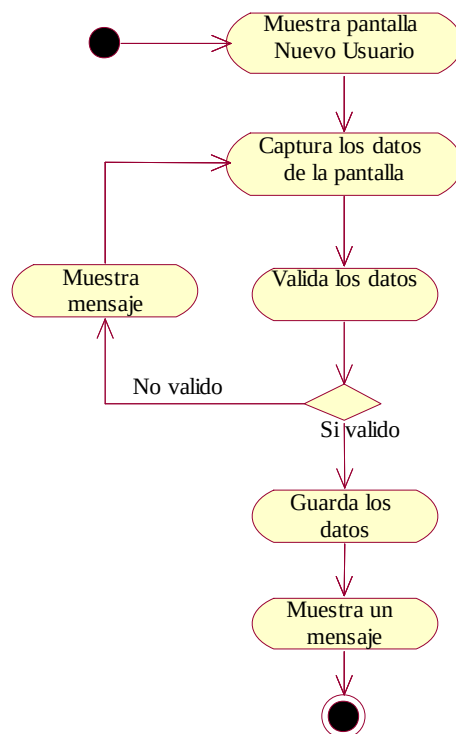


Figura N° 34: Diag. de Act. Crear Usuario

Editar Usuario

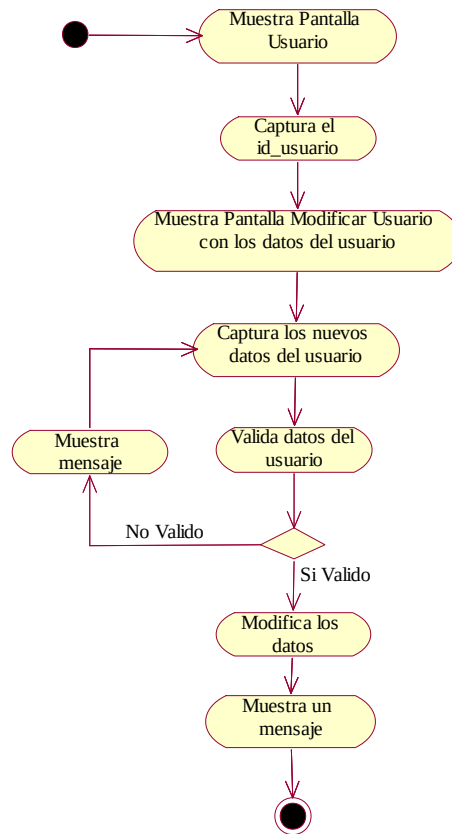


Figura N° 35: Diag. de Act. Editar Usuario

Eliminar Usuario

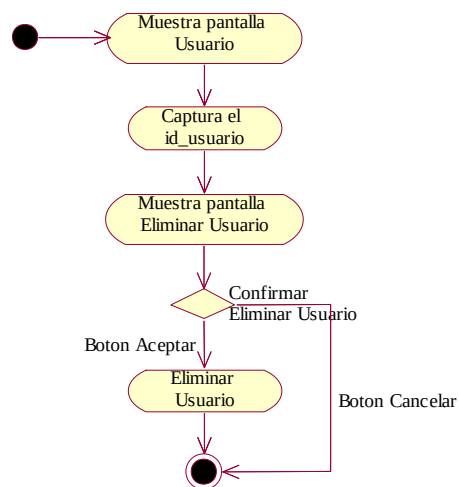


Figura N° 36: Diag. de Act. Eliminar Usuario

II.1.4.1.4.3.2 GESTIONAR USUARIOS MÓDULOS

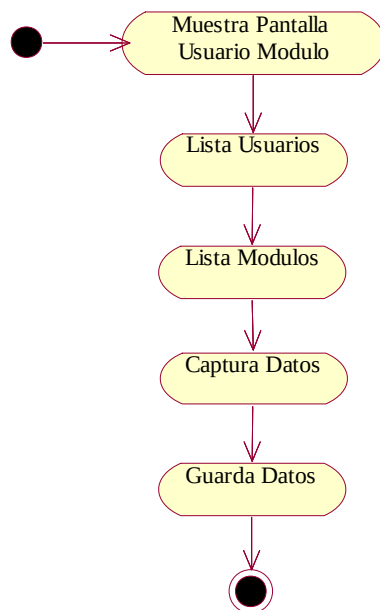


Figura N° 37: Diag. de Act. Gestionar Usuario Modulo

II.1.4.1.4.3.3 GESTIONAR PARAMETROS

Listar Parámetros

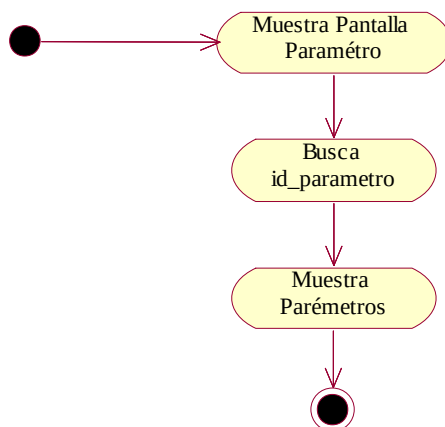


Figura N° 38: Diag. de Act. Listar Parámetros

Crear Parámetro

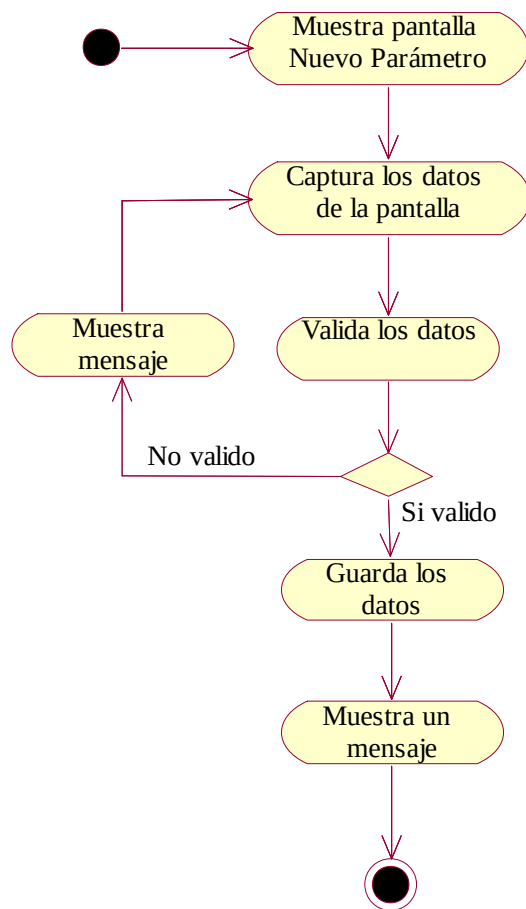


Figura N° 39: Diag. de Act. Crear Parámetro

Editar Parámetro

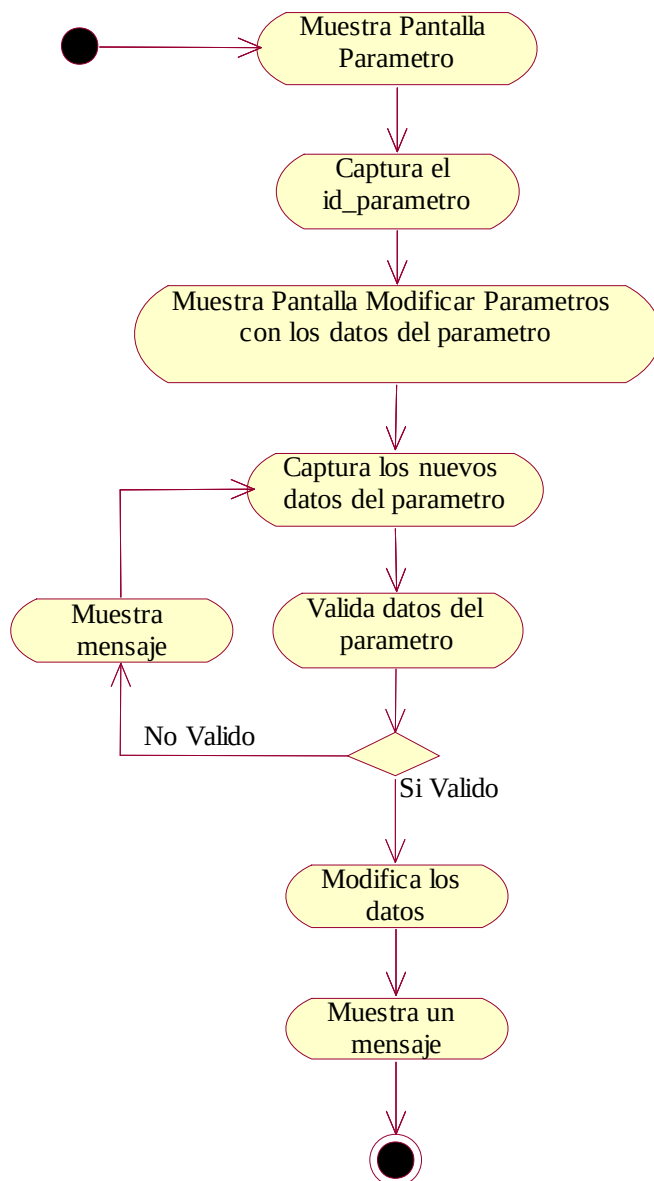


Figura N° 40: Diag. de Act. Editar Parámetro

Eliminar Parámetro

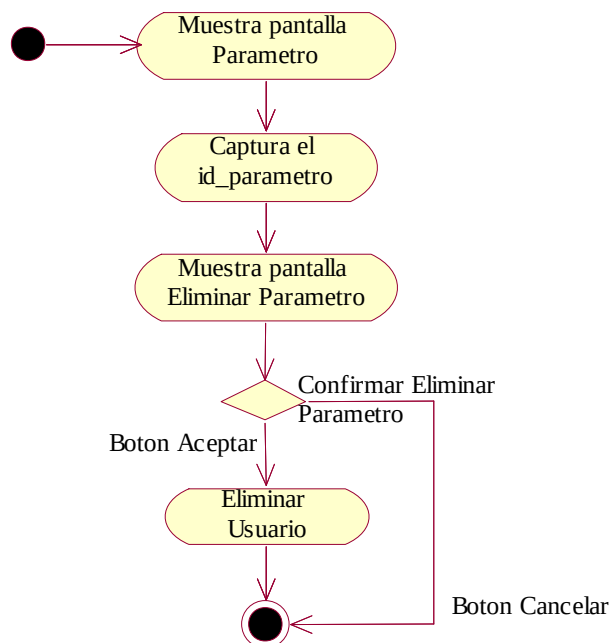


Figura N° 41: Diag. de Act. Eliminar Parámetro

II.1.4.1.4.4 MODULO II: PERSONAL

II.1.4.1.4.4.1 GESTIONAR EMPLEADOS

Listar Empleados

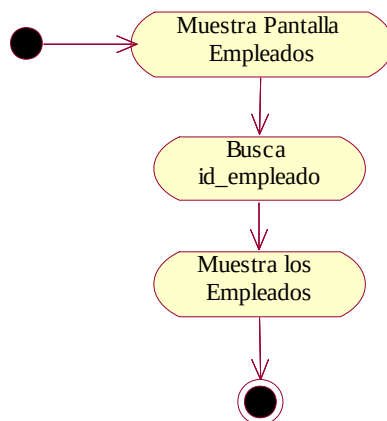


Figura N° 42: Diag. de Actividades Listar Empleado

Crear Empleado

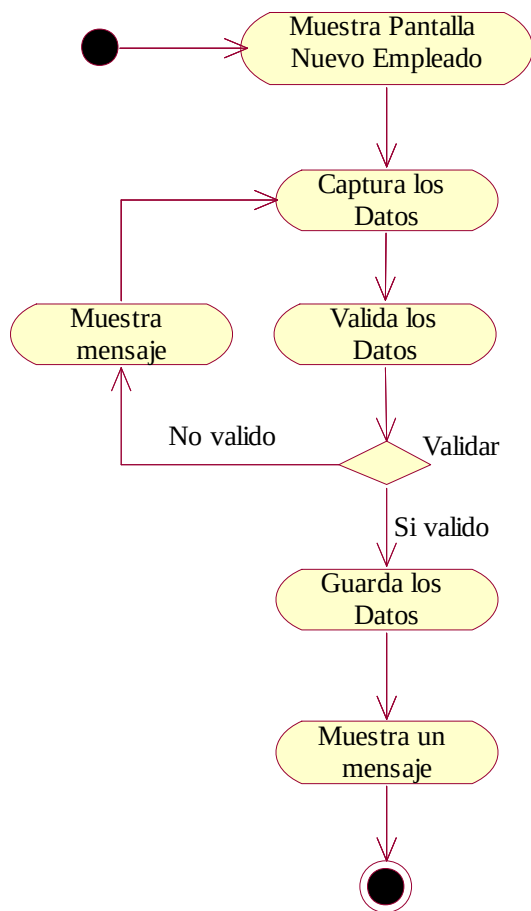


Figura N° 43: Diag. de Actividades Crear Empleado

Editar Empleado

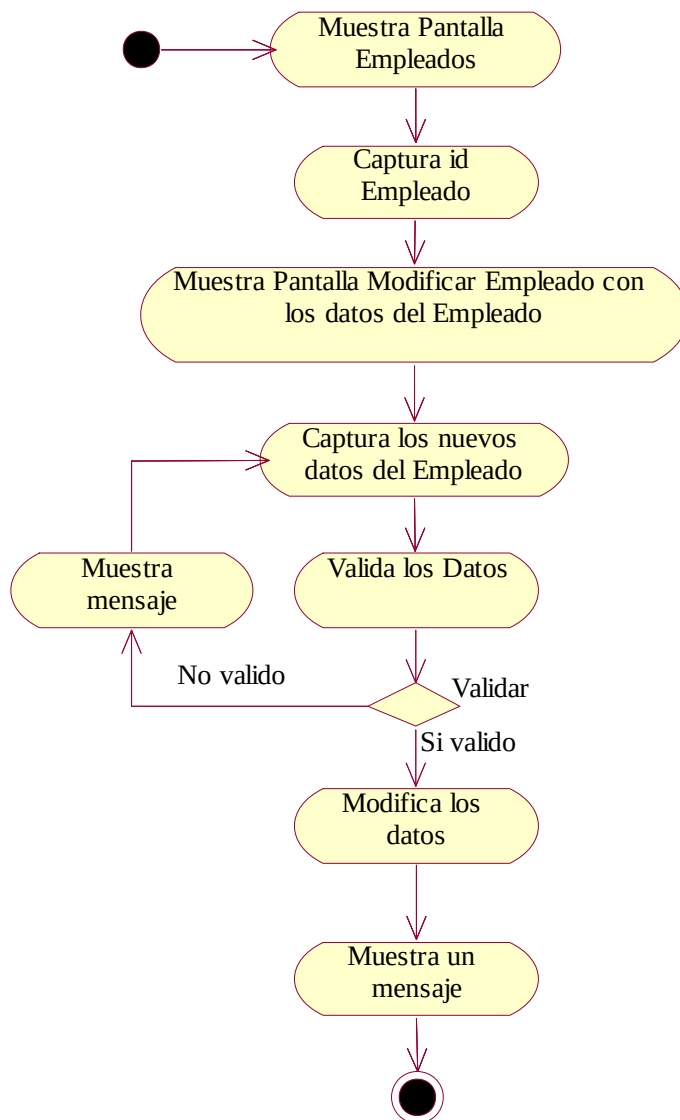


Figura N° 44: Diag. de Actividades Editar Empleado

Eliminar Empleado

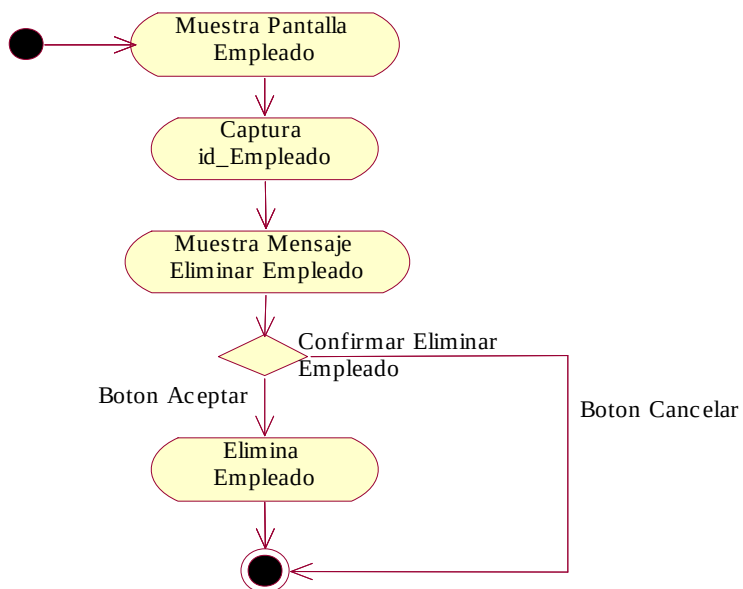


Figura N° 45: Diag. de Actividades Eliminar Empleado

Listar Contrato

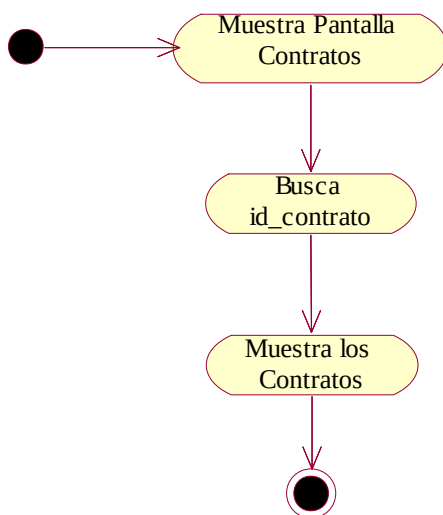


Figura N° 46: Diag. de Actividades Listar Contrato

Crear Contrato

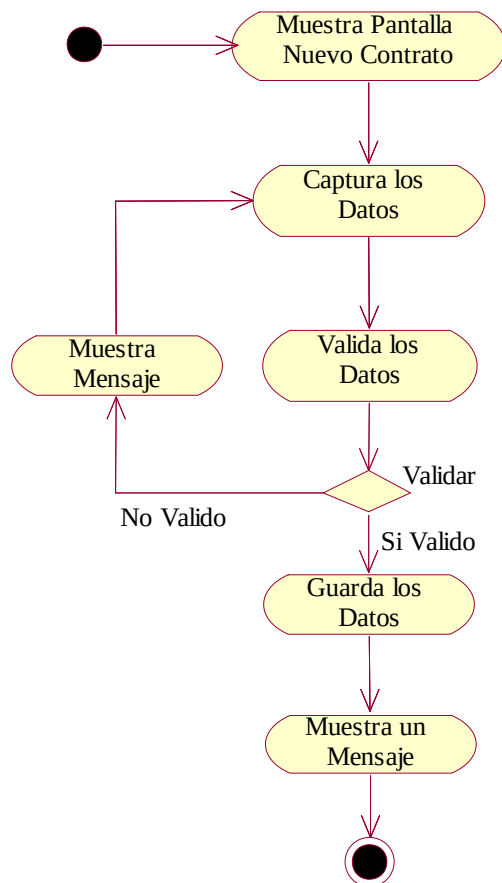


Figura N° 47: Diag. de Actividades Crear Contrato

Editar Contrato

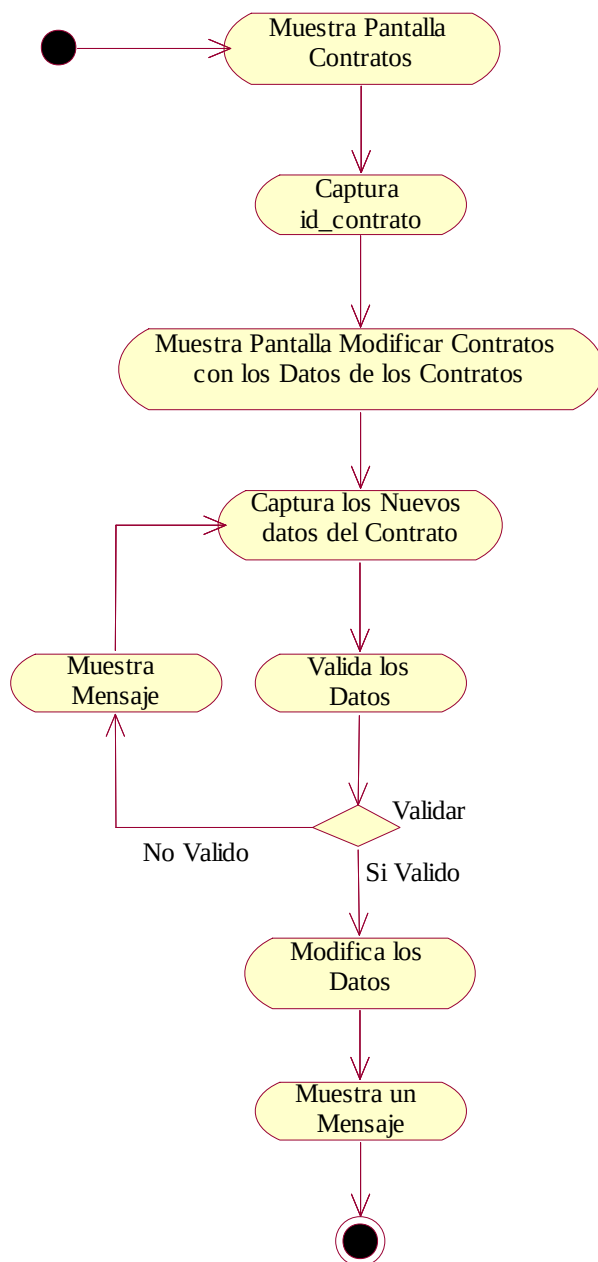


Figura N° 48: Diag. de Actividades Editar Contrato

Eliminar Contrato

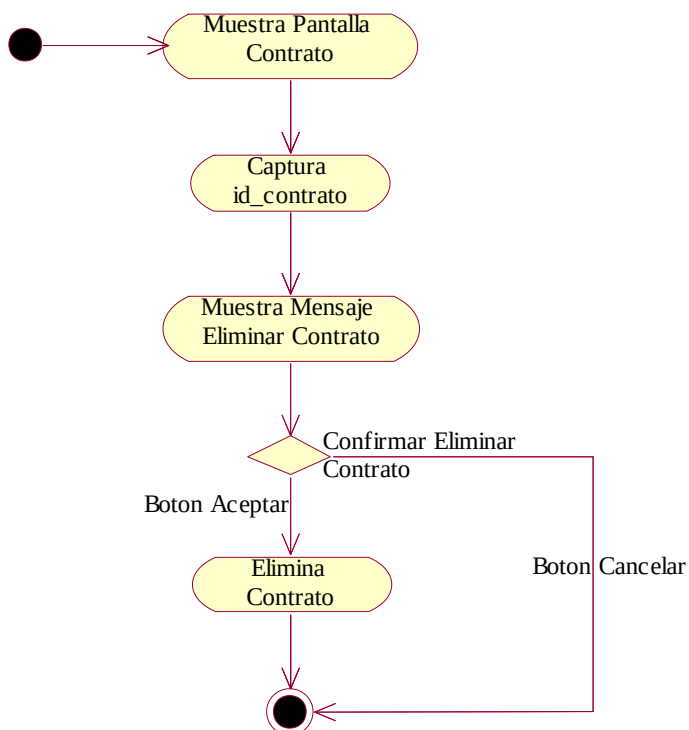


Figura N° 49: Diag. de Actividades Eliminar Contrato

Listar Anticipo

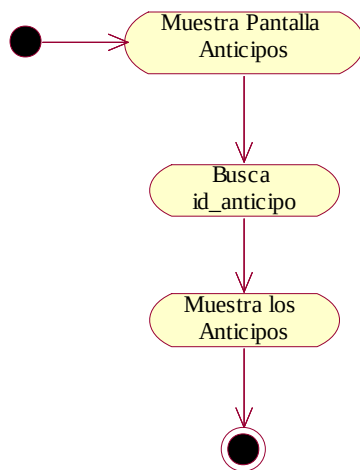


Figura N° 50: Diag. de Actividades Listar Anticipo

Crear Anticipo

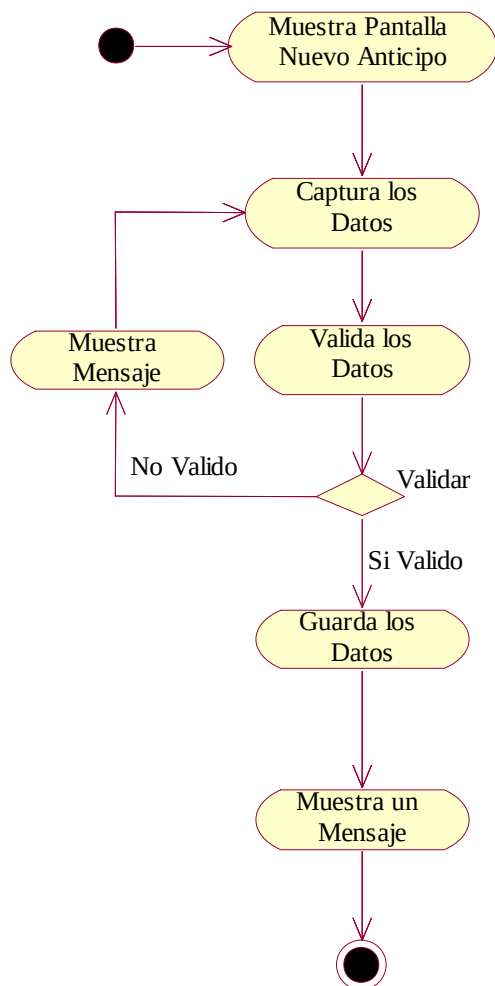


Figura N° 51: Diag. de Actividades Crear Anticipo

Editar Anticipo

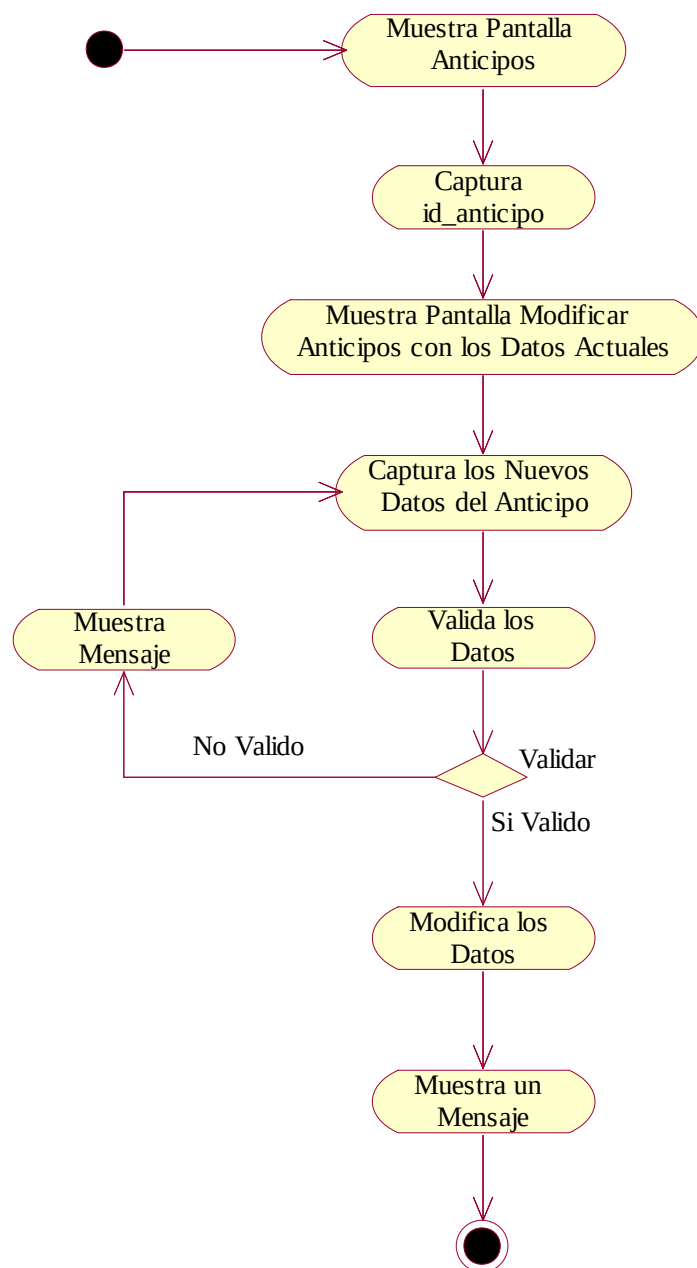


Figura N° 52: Diag. de Actividades Editar Anticipo

Eliminar Anticipo

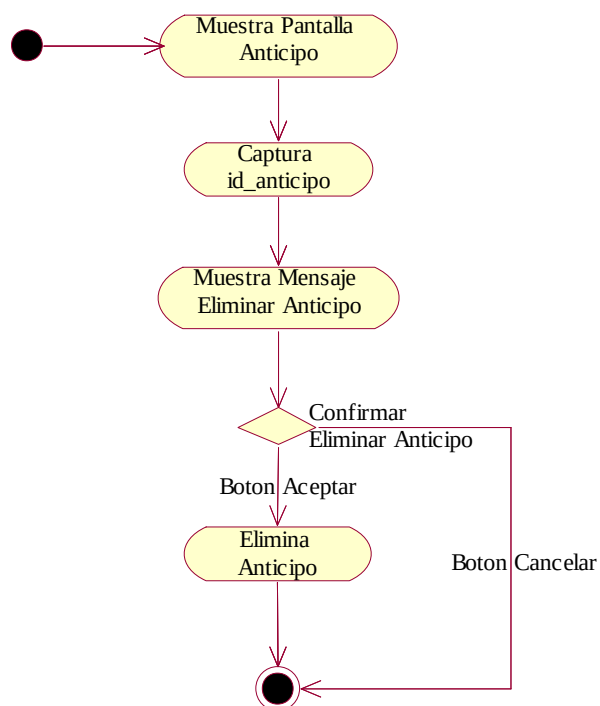


Figura N° 53: Diag. de Actividades Eliminar Anticipo

II.1.4.1.4.2 GESTIONAR AREAS

Listar Áreas

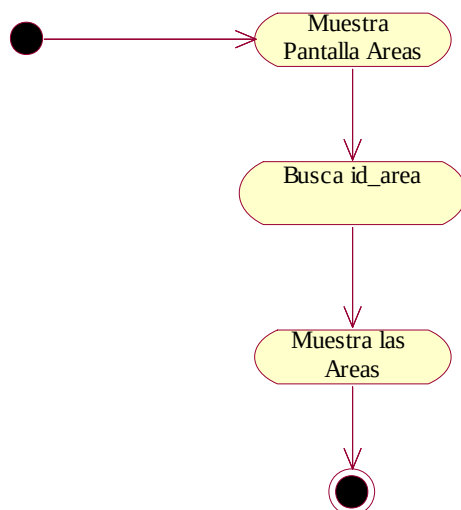


Figura N° 54 Diag. de Actividades Listar Áreas

Crear Área

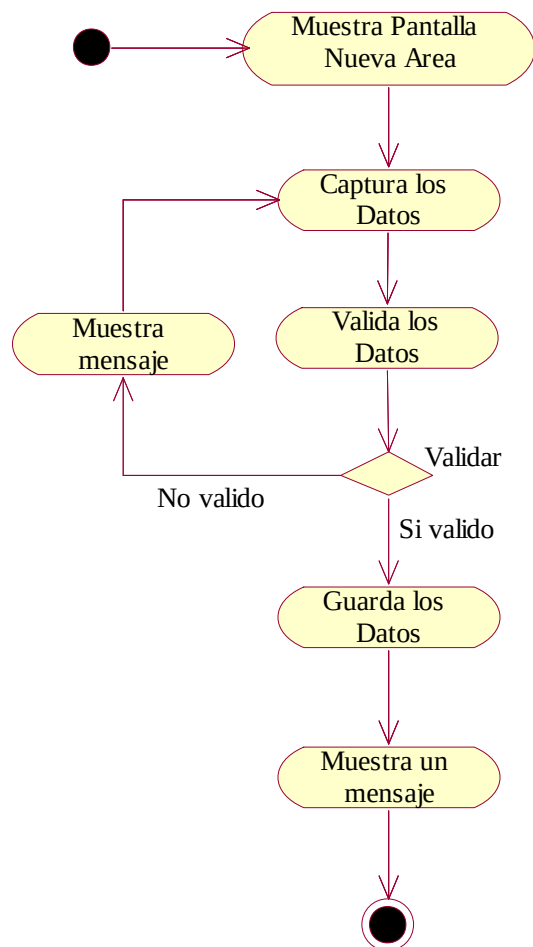


Figura N° 55: Diag. de Actividades Crear Área

Editar Área

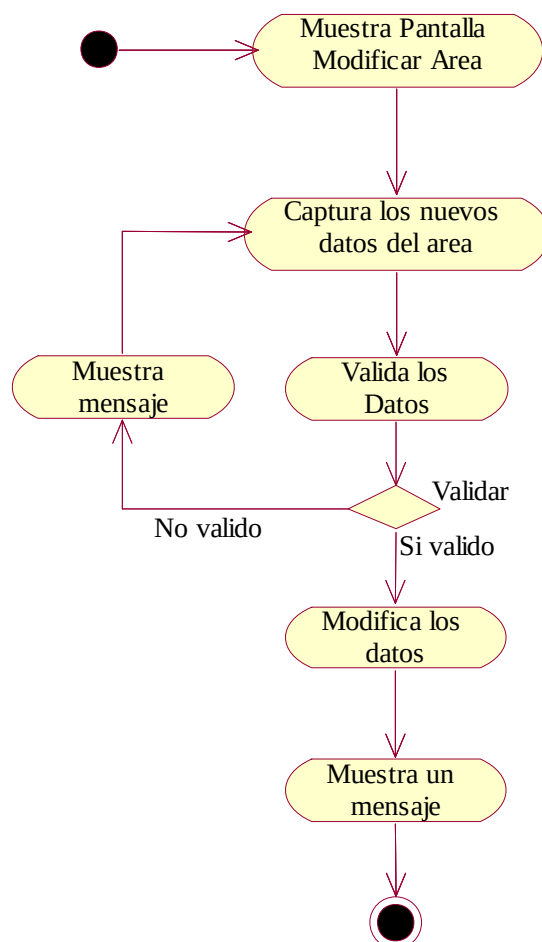


Figura N° 56 Diag. de Actividades Editar Área

Eliminar Área

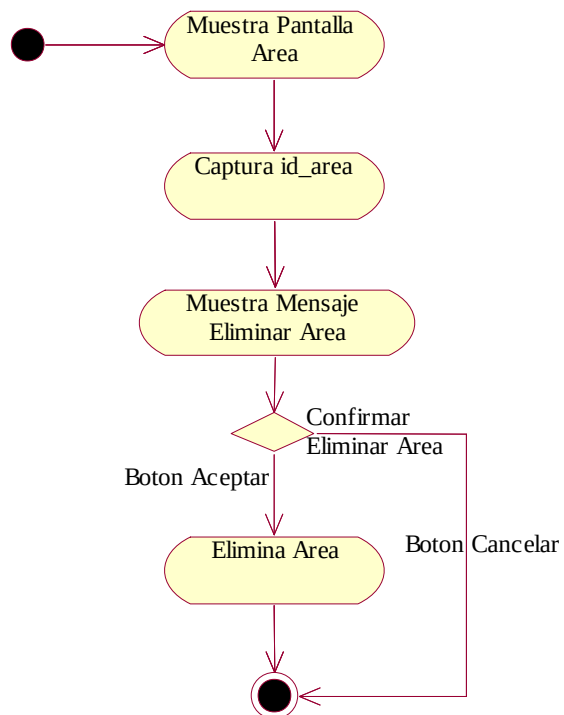


Figura N° 57: Diag. de Actividades Eliminar Área

II.1.4.1.4.3 GESTIONAR PUESTO

Listar Puestos

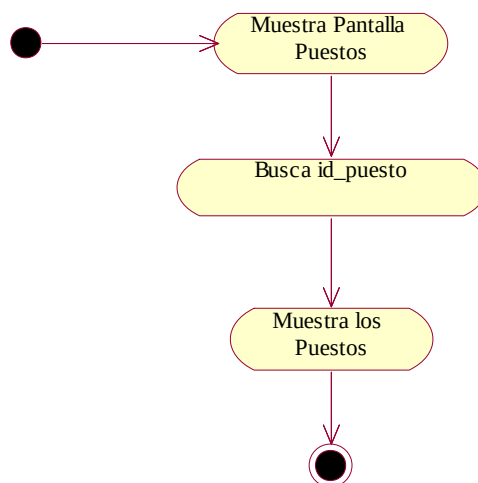


Figura N° 58: Diag. de Actividades Listar Puestos

Crear Puesto

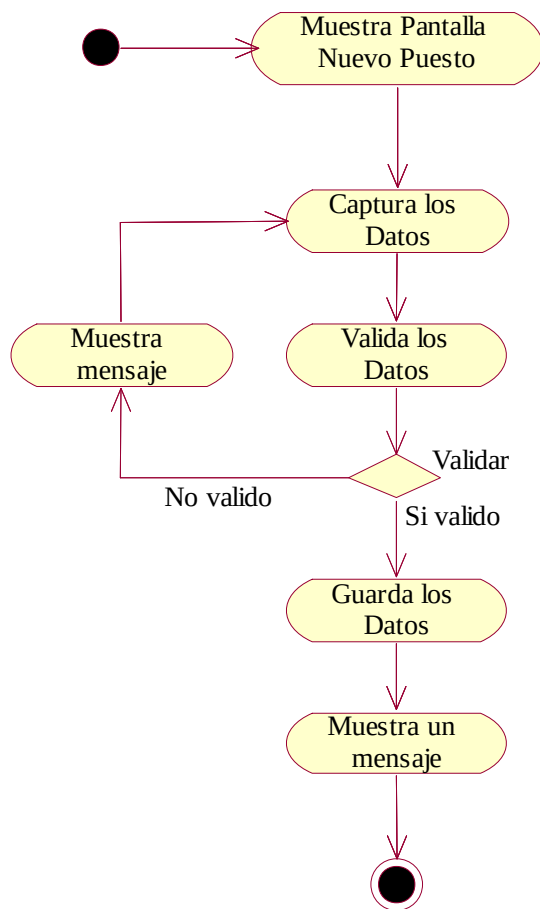


Figura N° 59: Diag. de Actividades Crear Puesto

Editar Puesto

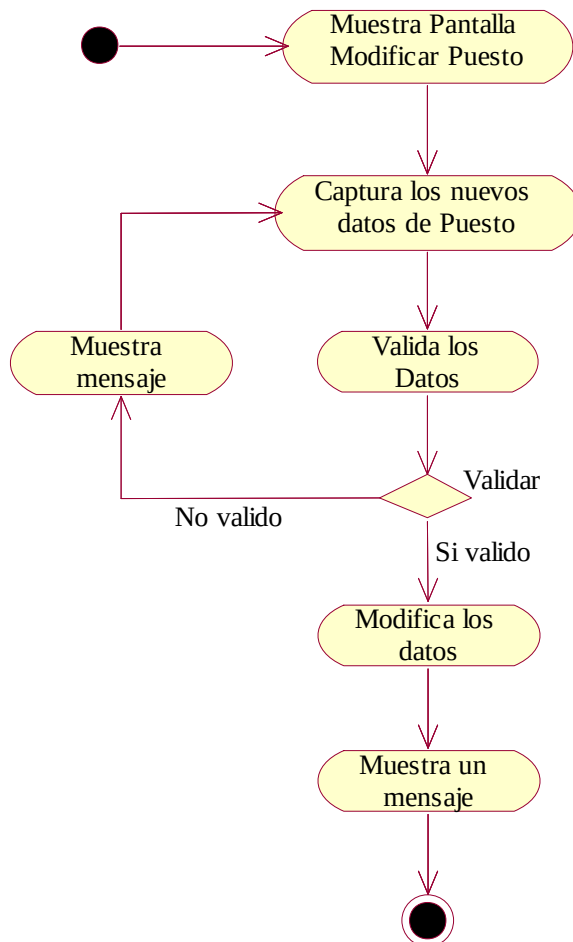


Figura N° 60: Diag. de Actividades Editar Puesto

Eliminar Puesto

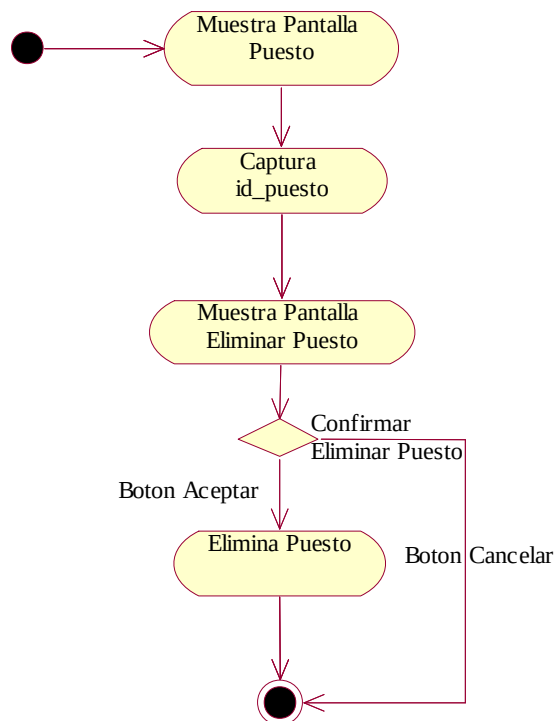


Figura N°61: Diag. de Actividades Eliminar Puesto

II.1.4.1.4.4 GESTIONAR PERMISOS-VACACIONES

Listar Empleados

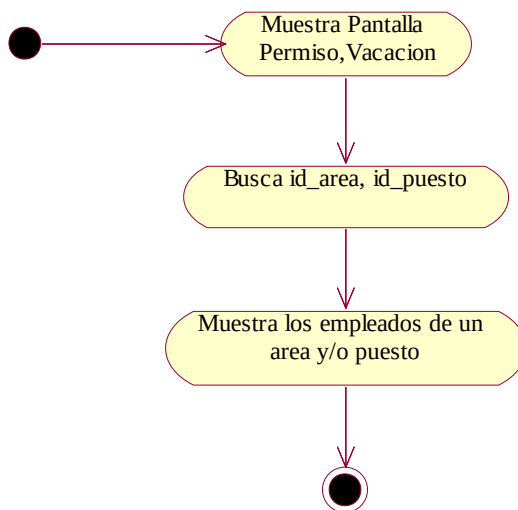


Figura N° 62: Diag. de Actividades Listar Empleados

Crear Permiso-Vacación

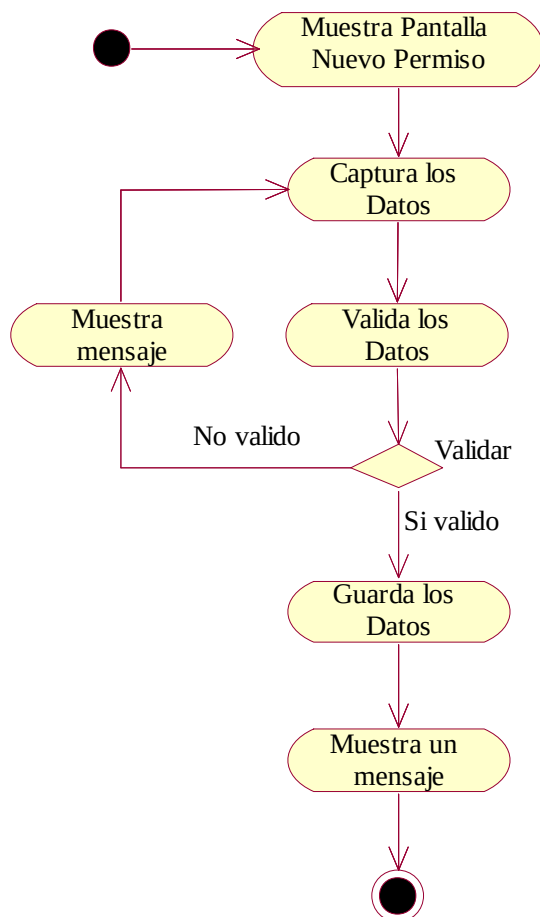


Figura N° 63: Diag. de Actividades Crear Permiso-Vacación

Editar Permiso-Vacación

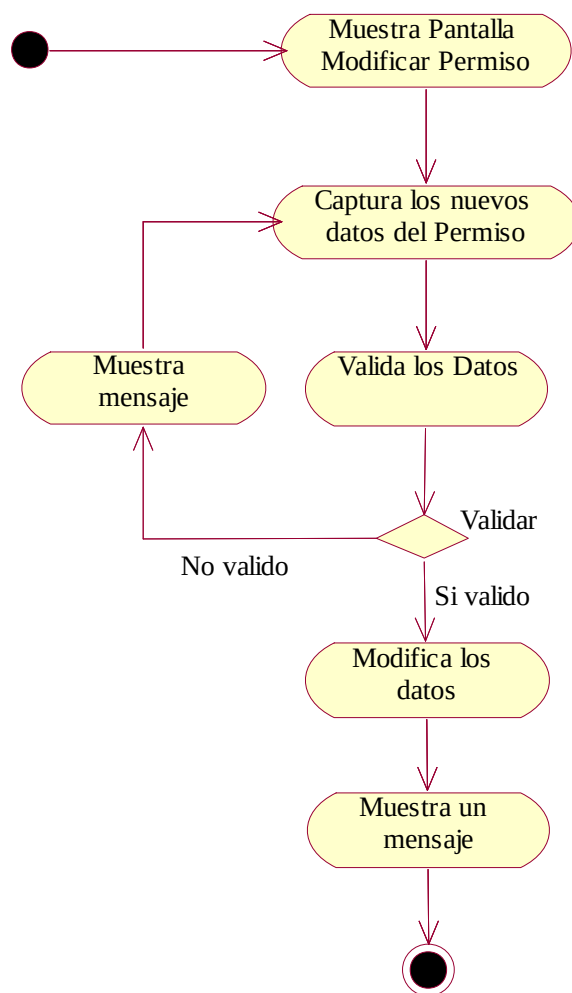


Figura N° 64: Diag. de Actividades Editar Permiso-Vacación

Eliminar Permiso-Vacación

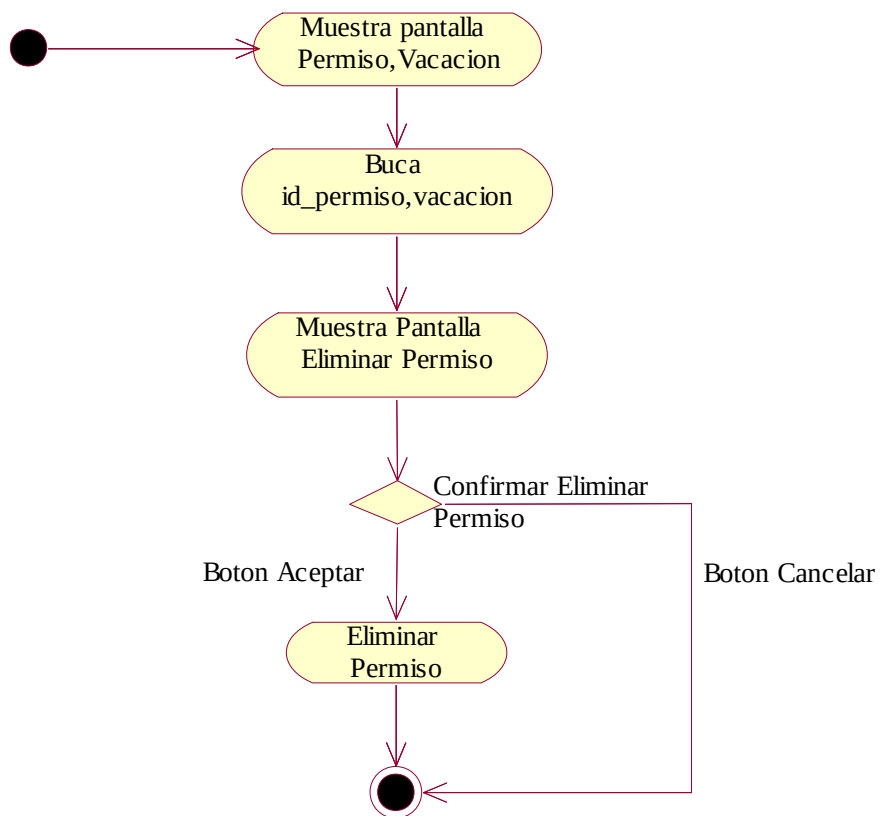


Figura N° 65: Diag. de Actividades Eliminar Permiso-Vacación

II.1.4.1.4.4.5 GESTIONAR PLANILLAS DE SUELDOS

Crear Planilla de Sueldo

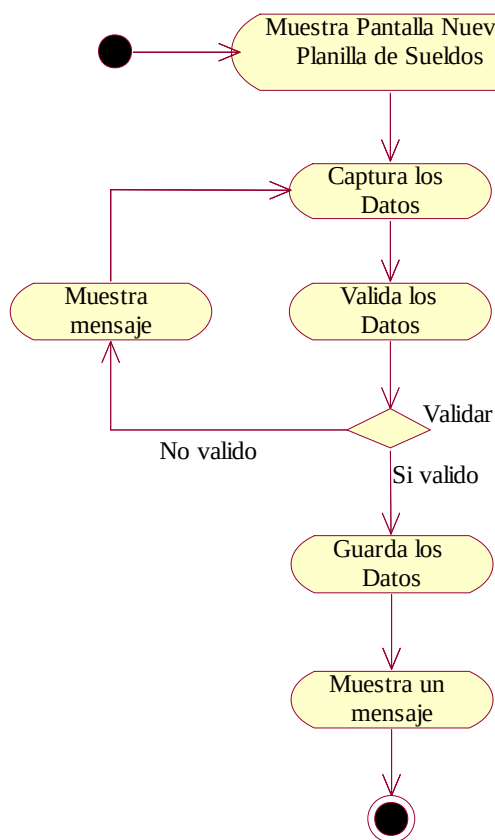


Figura N° 66: Diag. de Actividades Crear Planilla de Sueldo

Cargar Detalle de Planilla

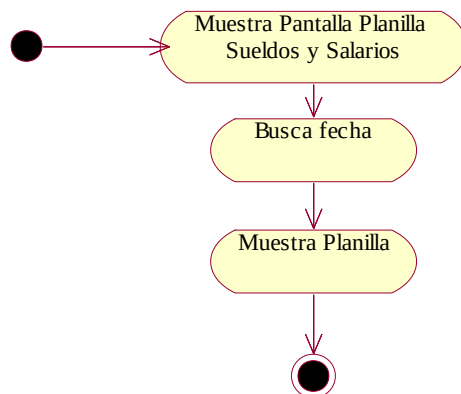


Figura N° 67: Diag. de Actividades Cargar Detalle de Planilla

Editar Planilla de Sueldos

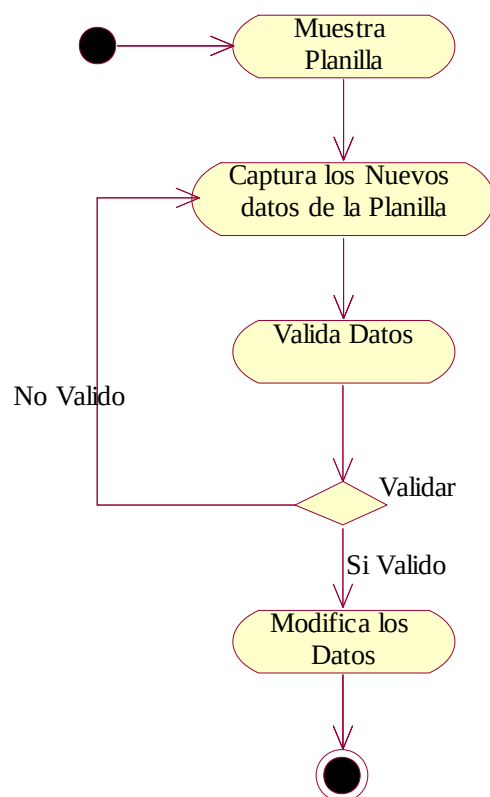


Figura N° 68: Diag. de Actividades Editar Planilla de Sueldos

Eliminar Planilla de Sueldos

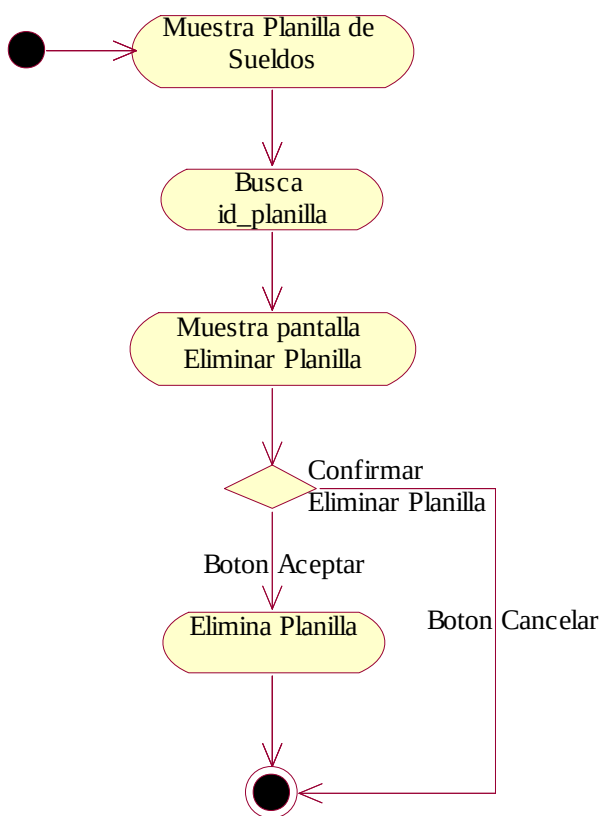


Figura N° 69: Diag. de Actividades Eliminar Planilla de Sueldos

II.1.4.1.4.4.6 GESTIONAR PLANILLAS DE AGUINALDOS

Crear Planilla de Aguinaldo

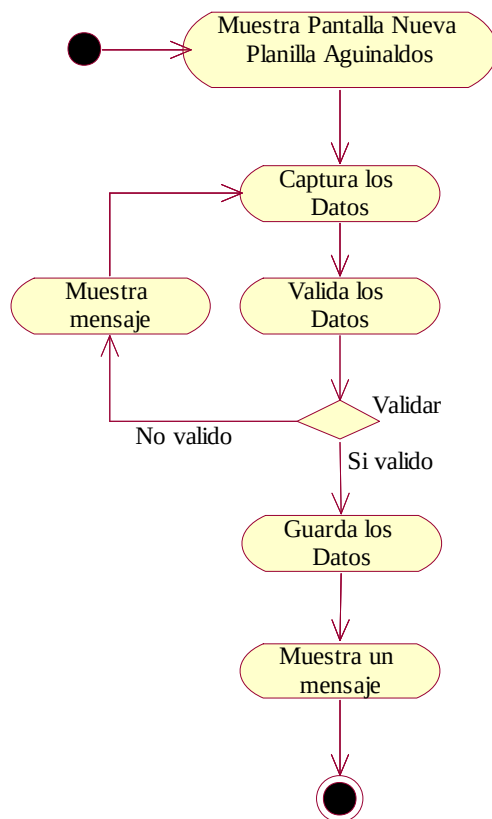


Figura N° 70: Diag. de Actividades Crear Planilla de Aguinaldo

Cargar Detalle de Planilla

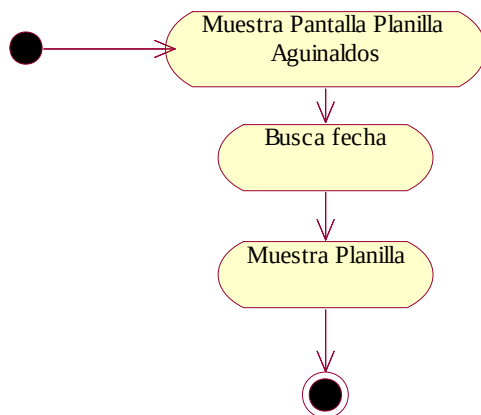


Figura N° 71: Diag. de Actividades Cargar Detalle de Planilla

Eliminar Planilla de Aguinaldos

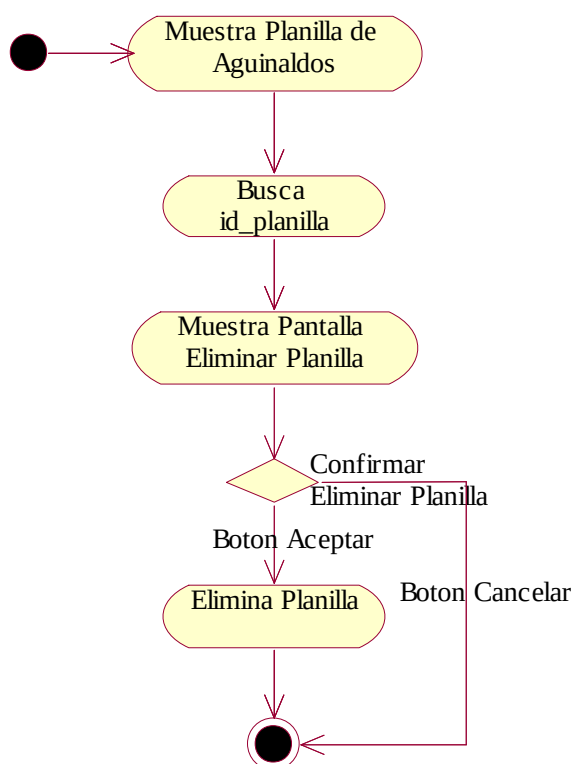


Figura N° 72: Diag. de Actividades Eliminar Planilla de Aguinaldos

II.1.4.1.4.5 MODULO III: CONTROL ASISTENCIA

II.1.4.1.4.5.1 GESTIONAR DIA INHABIL

Listar Día Inhábil

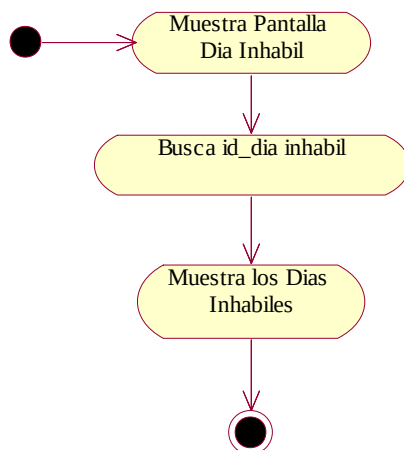


Figura N° 73: Diag. de Actividades Listar Día Inhábil

Crear Día Inhábil

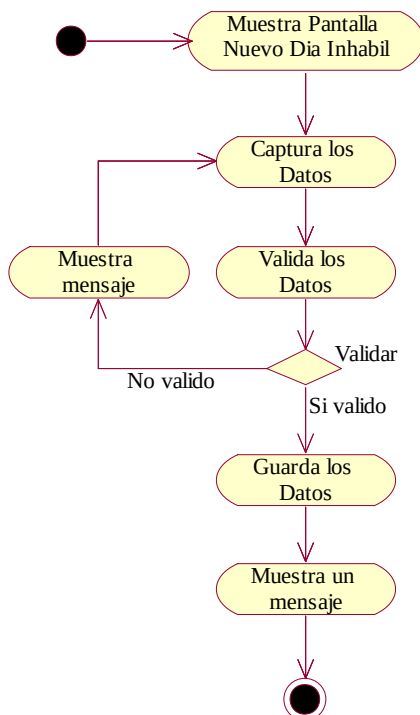


Figura N° 74: Diag. de Actividades Crear Día Inhábil

Editar Día Inhábil

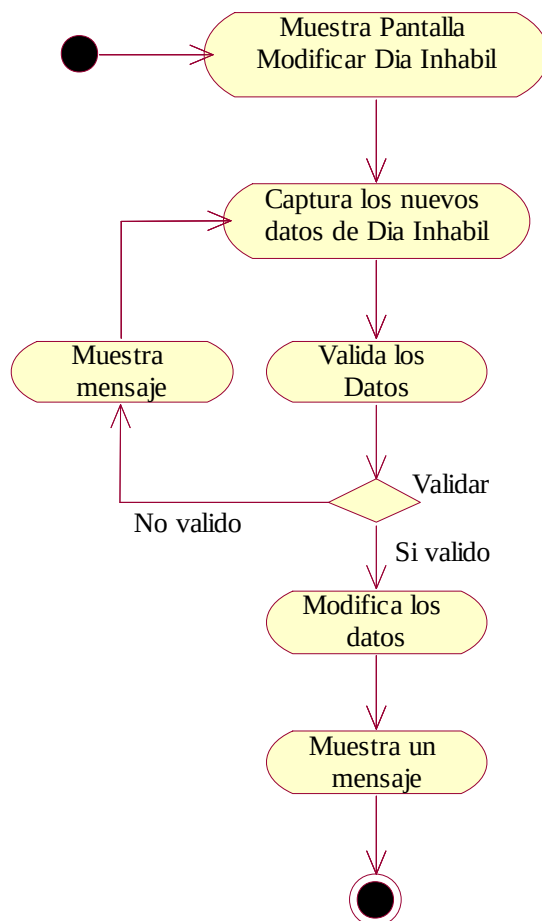


Figura N° 75: Diag. de Actividades Editar Día Inhábil

Eliminar Día Inhábil

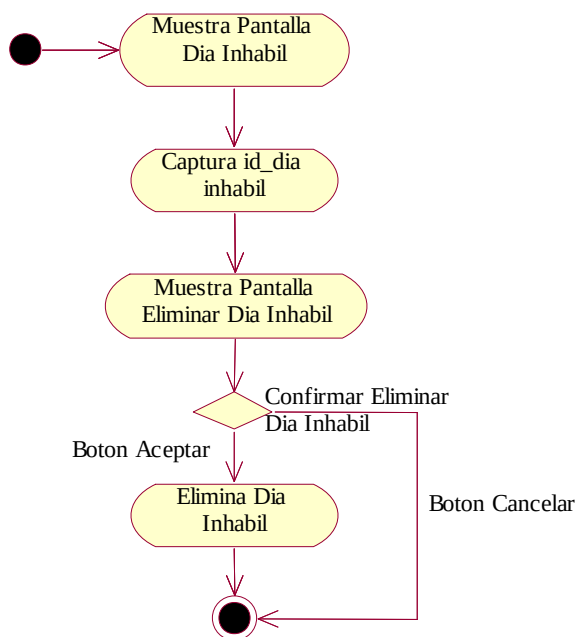


Figura N° 76: Diag. de Actividades Eliminar Día Inhábil

II.1.4.1.4.5.2 GESTIONAR HORARIO

Listar Horarios

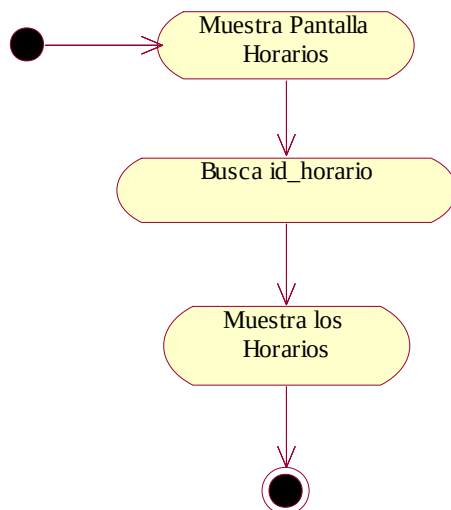


Figura N° 77: Diag. de Actividades Listar Horarios

Crear Horario

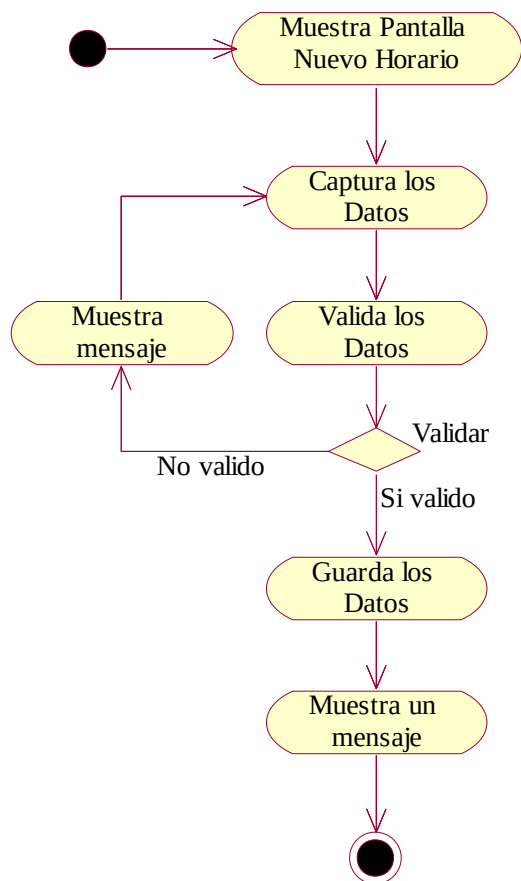


Figura N° 78: Diag. de Actividades Crear Horario

Editar Horario

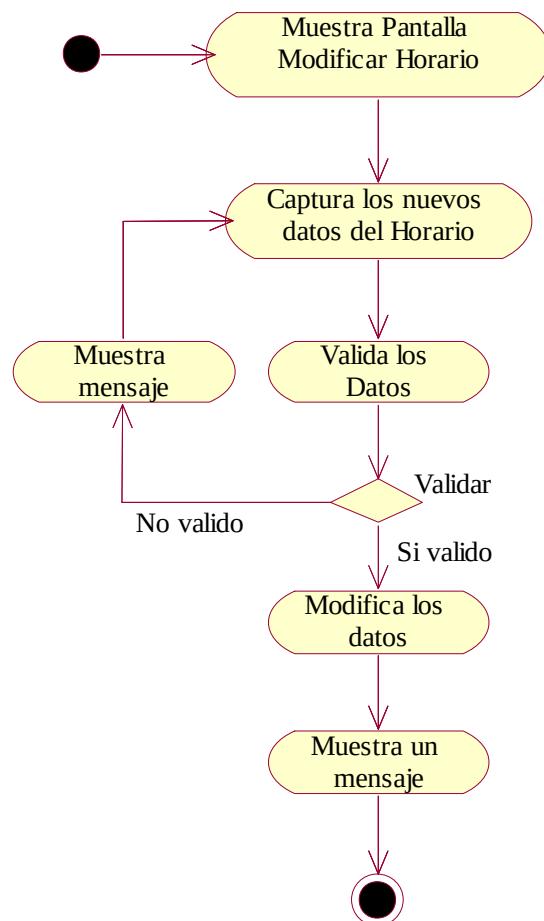


Figura N° 79: Diag. de Actividades Editar Horario

Eliminar Horario

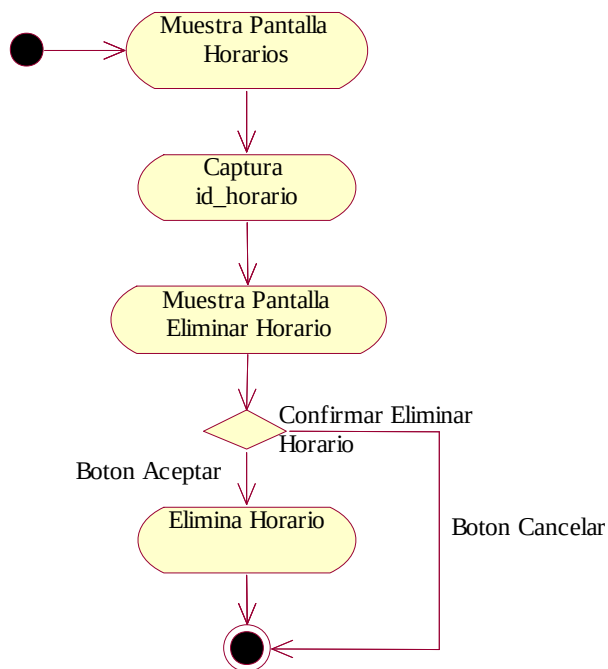


Figura N° 80: Diag. de Actividades Eliminar Horario

II.1.4.1.4.5.3 GESTIONAR EDICION MANUAL

Listar Empleados

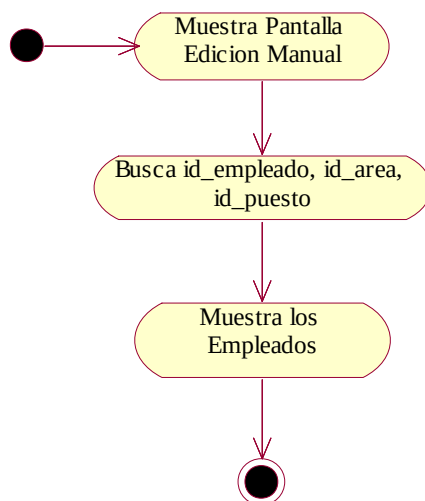


Figura N° 81: Diag. de Actividades Listar Empleados

Crear Evento

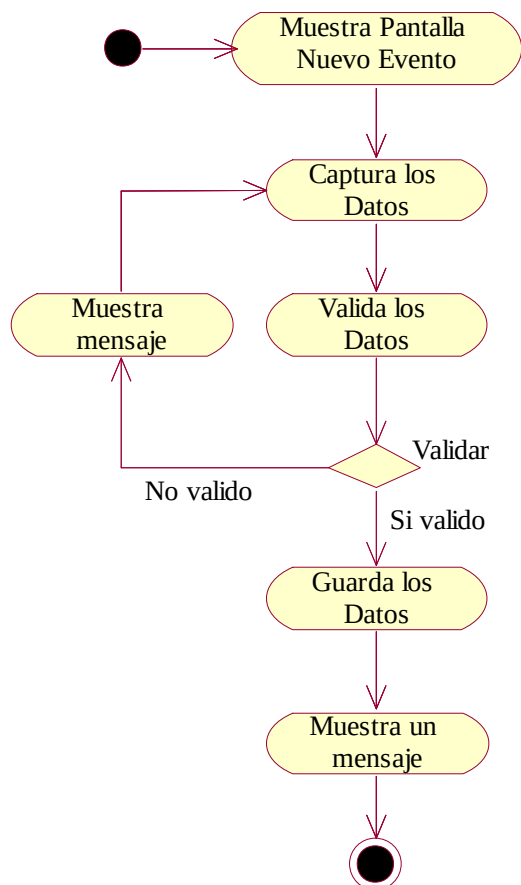


Figura N° 82: Diag. de Actividades Crear Evento

Editar Evento

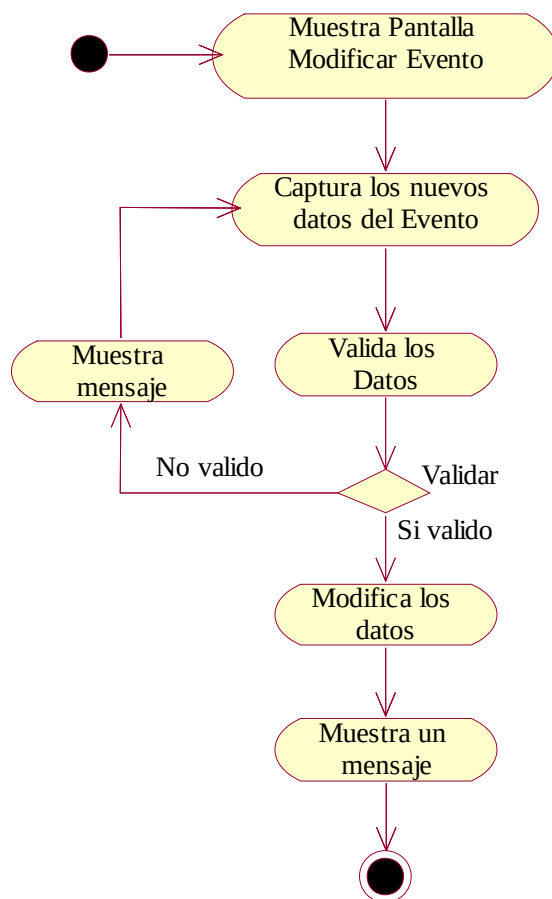


Figura N° 83: Diag. de Actividades Editar Evento

Eliminar Evento

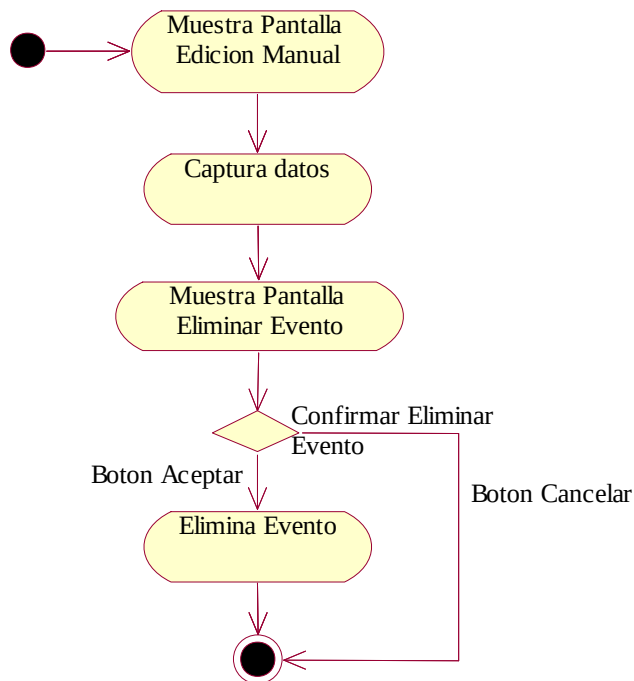


Figura N° 84: Diag. de Actividades Eliminar Evento

II.1.4.1.4.6 MODULO IV: REPORTES

II.1.4.1.4.6.1 Generar Reportes

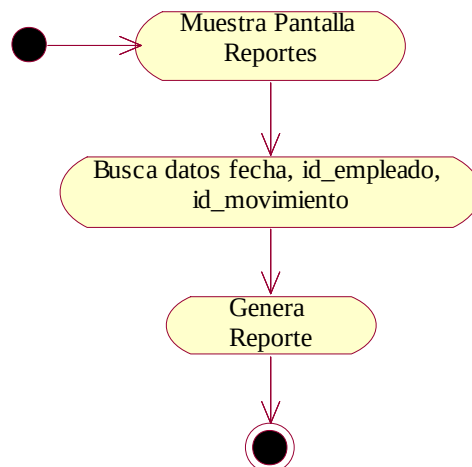


Figura N° 85: Diag. de Actividades Generar Reportes

II.1.4.2 Modelado de Diagrama de Interacción

II.1.4.2.1 Introducción

Los diagramas de interacción consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

II.1.4.2.2 Propósito

- ✧ Comprender la dinámica del sistema deseado para la organización.
- ✧ Identificar clases de análisis y diseño.

II.1.4.2.3 Alcance

- ✧ Describir la dinámica de sistema en el tiempo de vida de las clases u objetos.
- ✧ Definir un diagrama de secuencia para cada caso de uso del sistema.
- ✧ Definir un diagrama de colaboración para cada caso de uso del sistema.

II.1.4.2.4 Diagramas de Secuencia

II.1.4.2.4.1 Registrar Asistencia

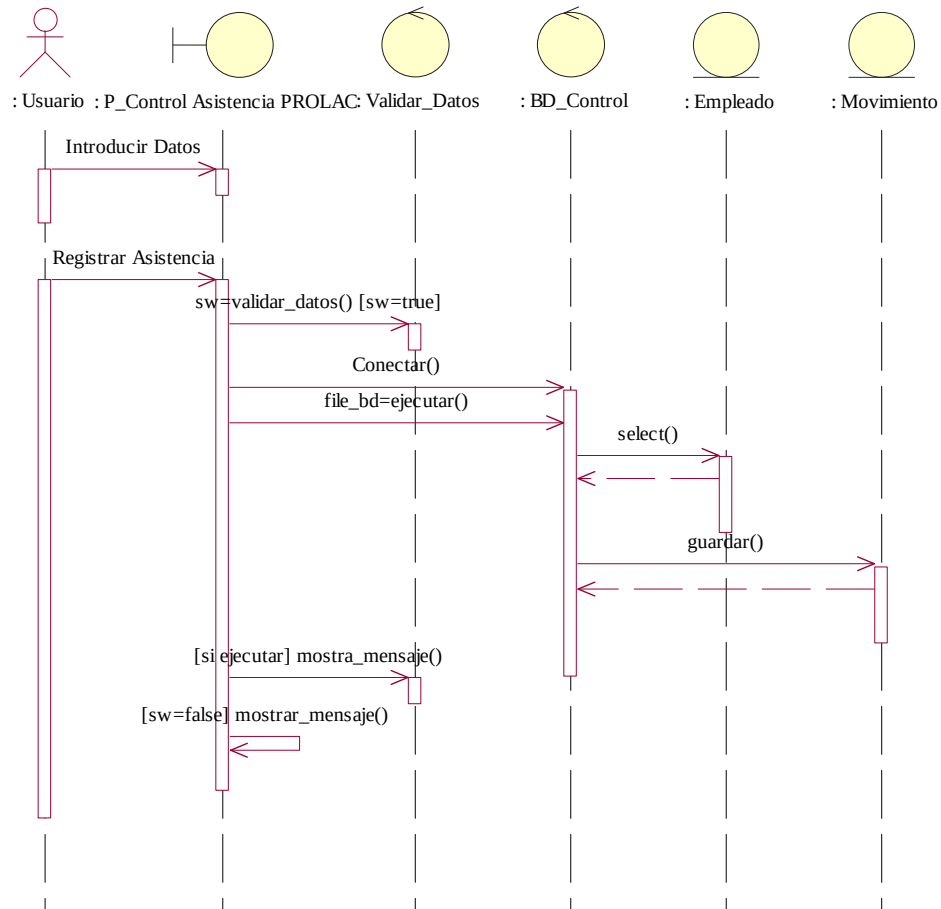


Figura N° 86: Diag. de Secuencia Registrar Asistencia

II.1.4.2.4.2 Ingresar Sistema

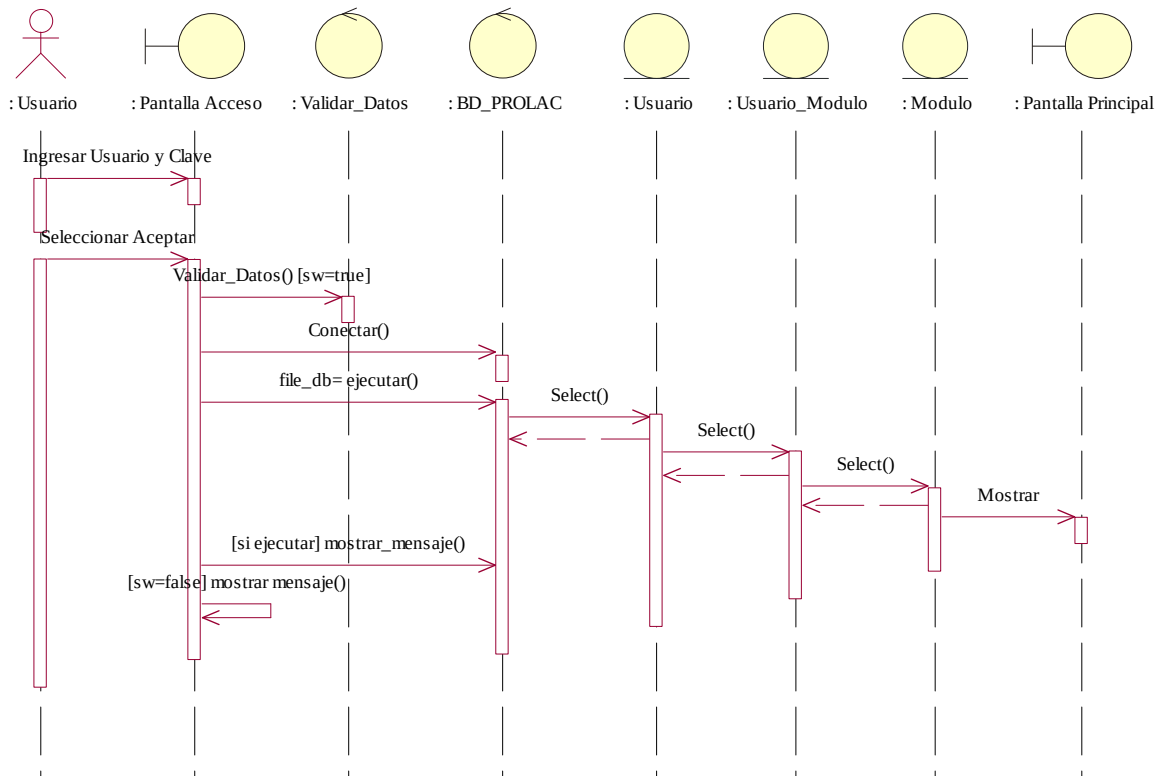


Figura N° 87: Diag. de Secuencia Ingresar Sistema

II.1.4.2.4.3 MODULO I: SISTEMA

II.1.4.2.4.3.1 GESTIONAR USUARIOS

Listar Usuarios

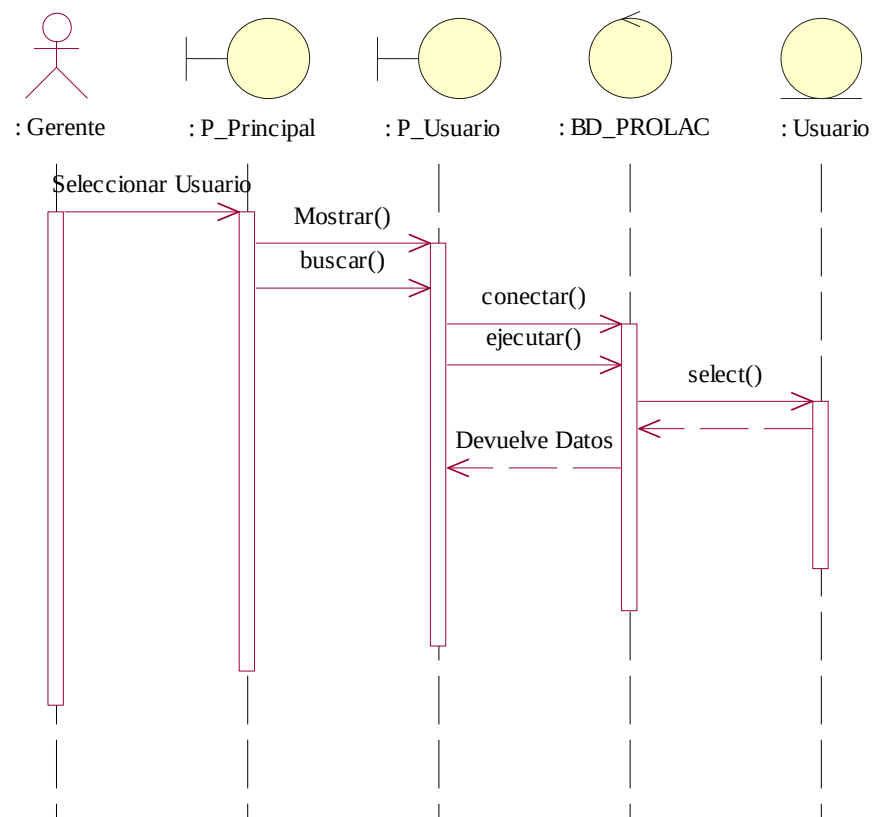


Figura N° 88: Diag. de Secuencia Listar Usuarios

Crear Usuario

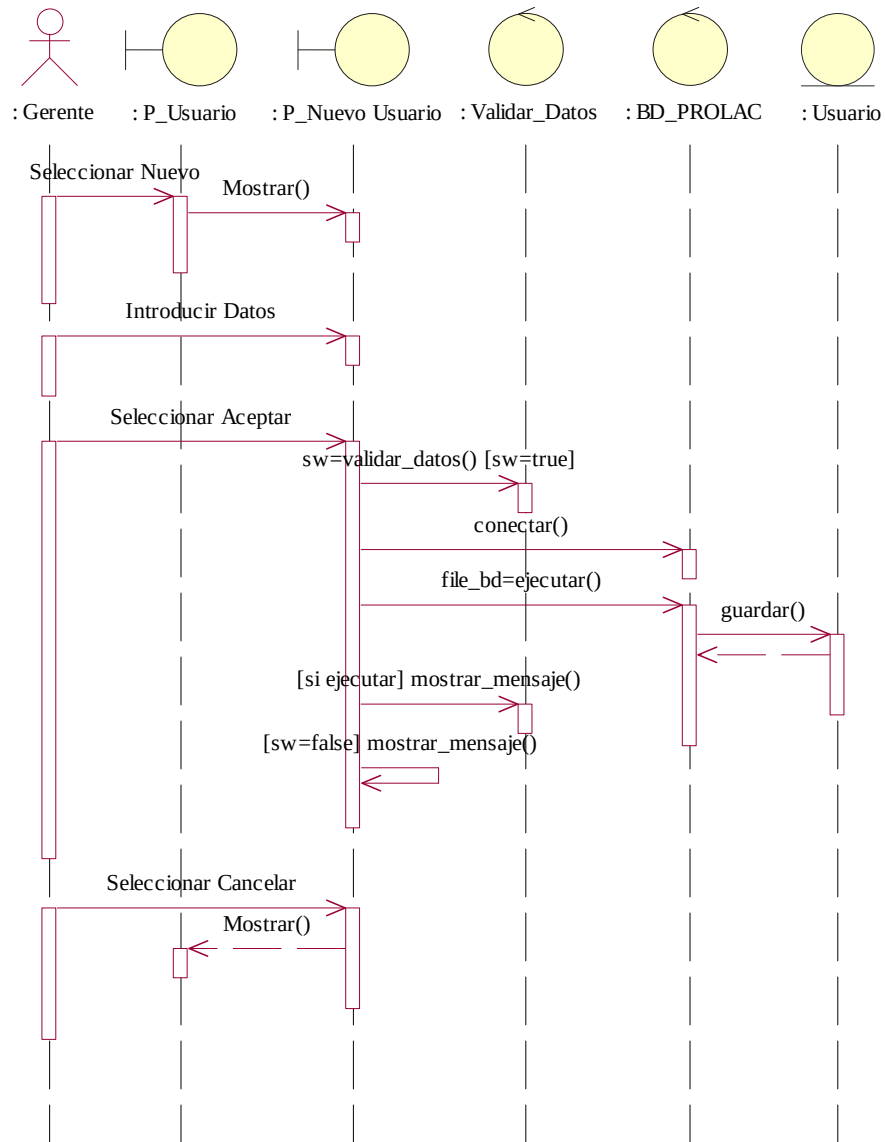


Figura N° 97: Diag. de Secuencia Crear Usuario

Editar Usuario

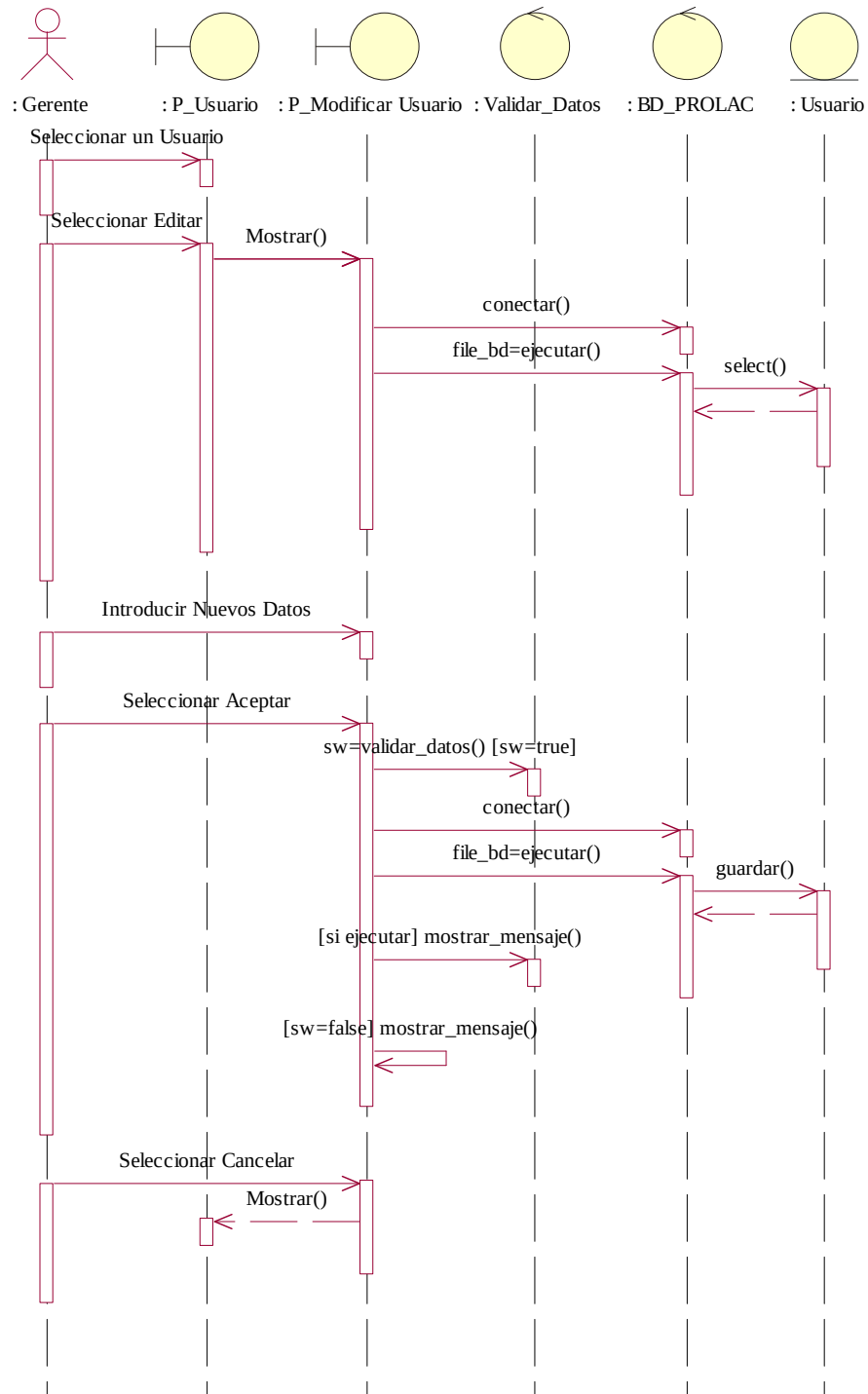


Figura N° 90: Diag. de Secuencia Editar Usuario

Eliminar Usuario

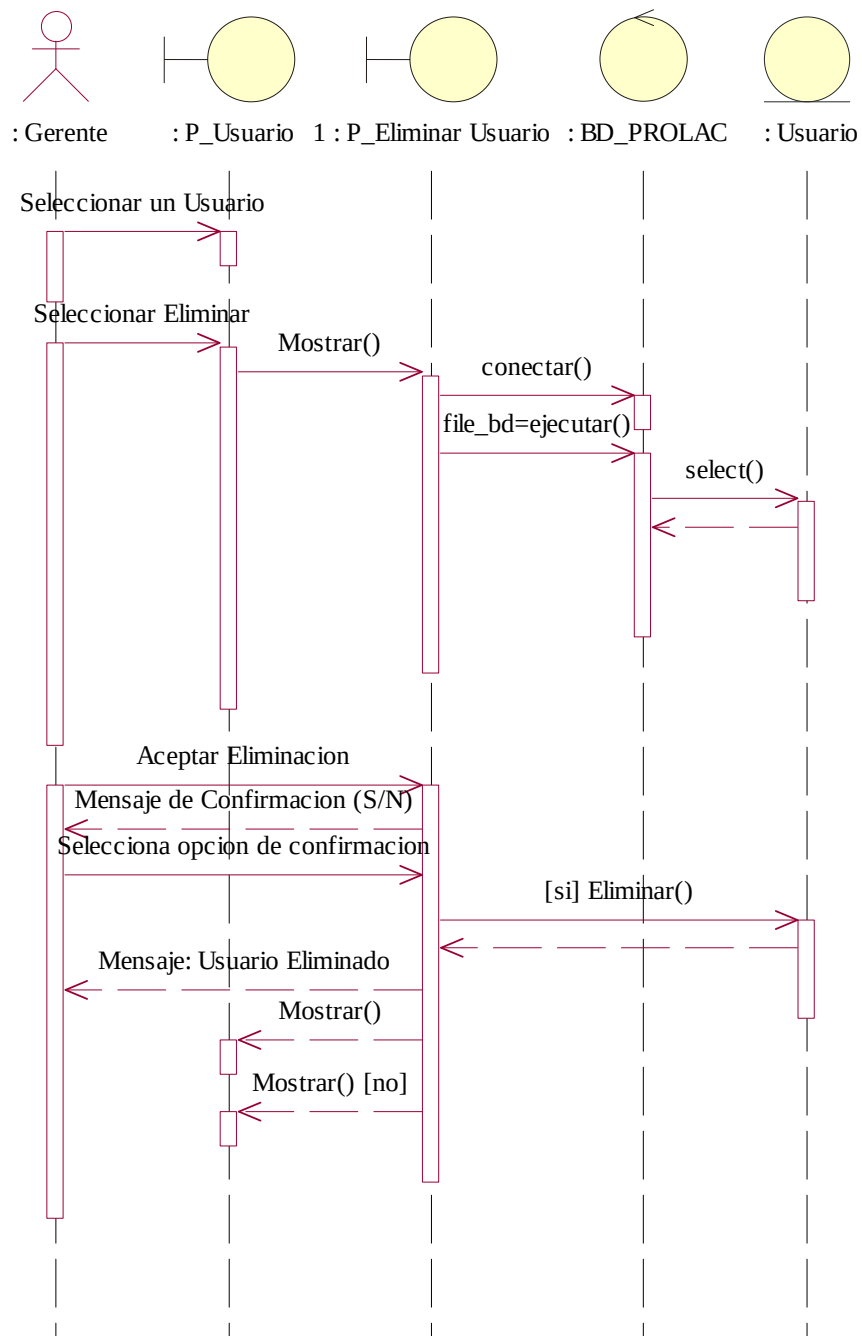


Figura N° 91: Diag. de Secuencia Eliminar Usuario

II.1.4.2.4.3.2 GESTIONAR USUARIOS MODULOS

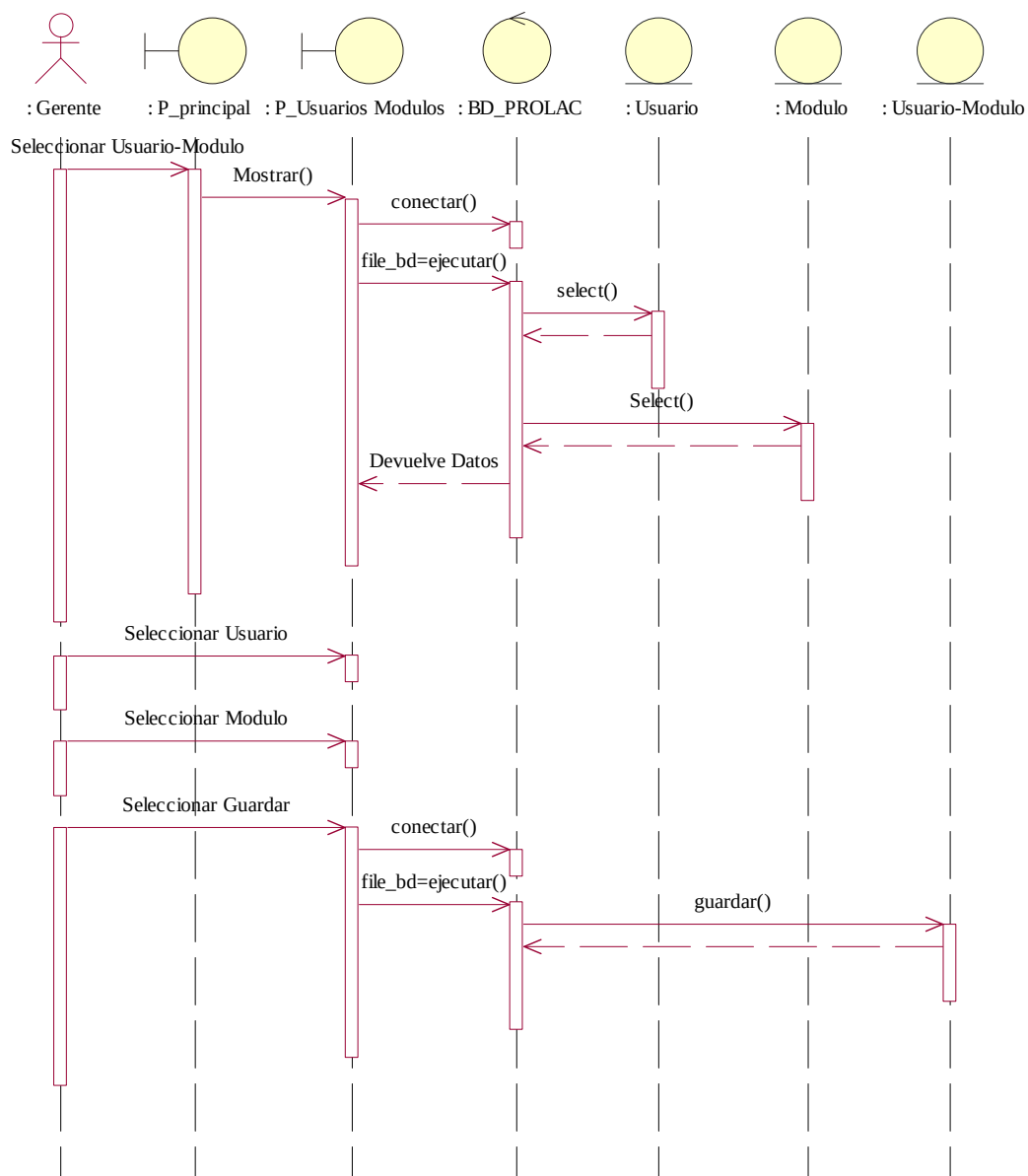


Figura N° 92: Diag. de Secuencia Gestionar Usuarios-Módulos

II.1.4.2.4.3.3 GESTIONAR PARAMETROS

Listar Parámetros

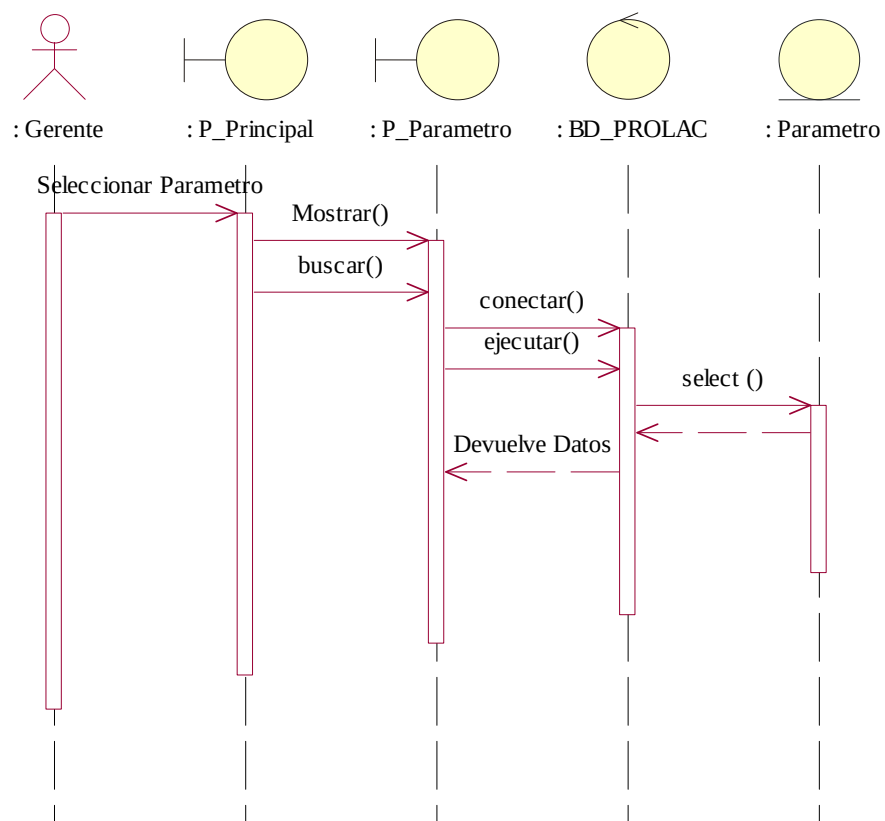


Figura N° 93: Diag. de Secuencia Listar Parámetros

Crear Parámetro

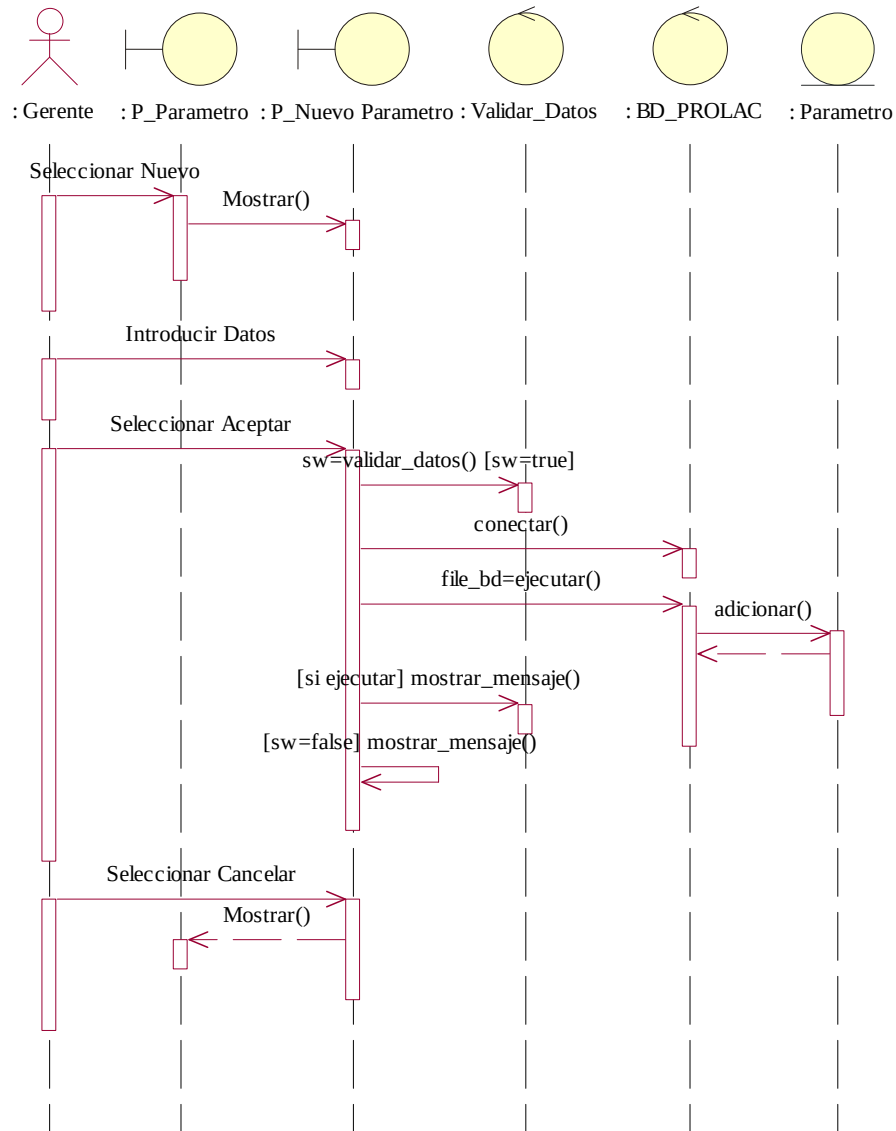


Figura N° 94: Diag. de Secuencia Crear Parámetro

Editar Parámetro

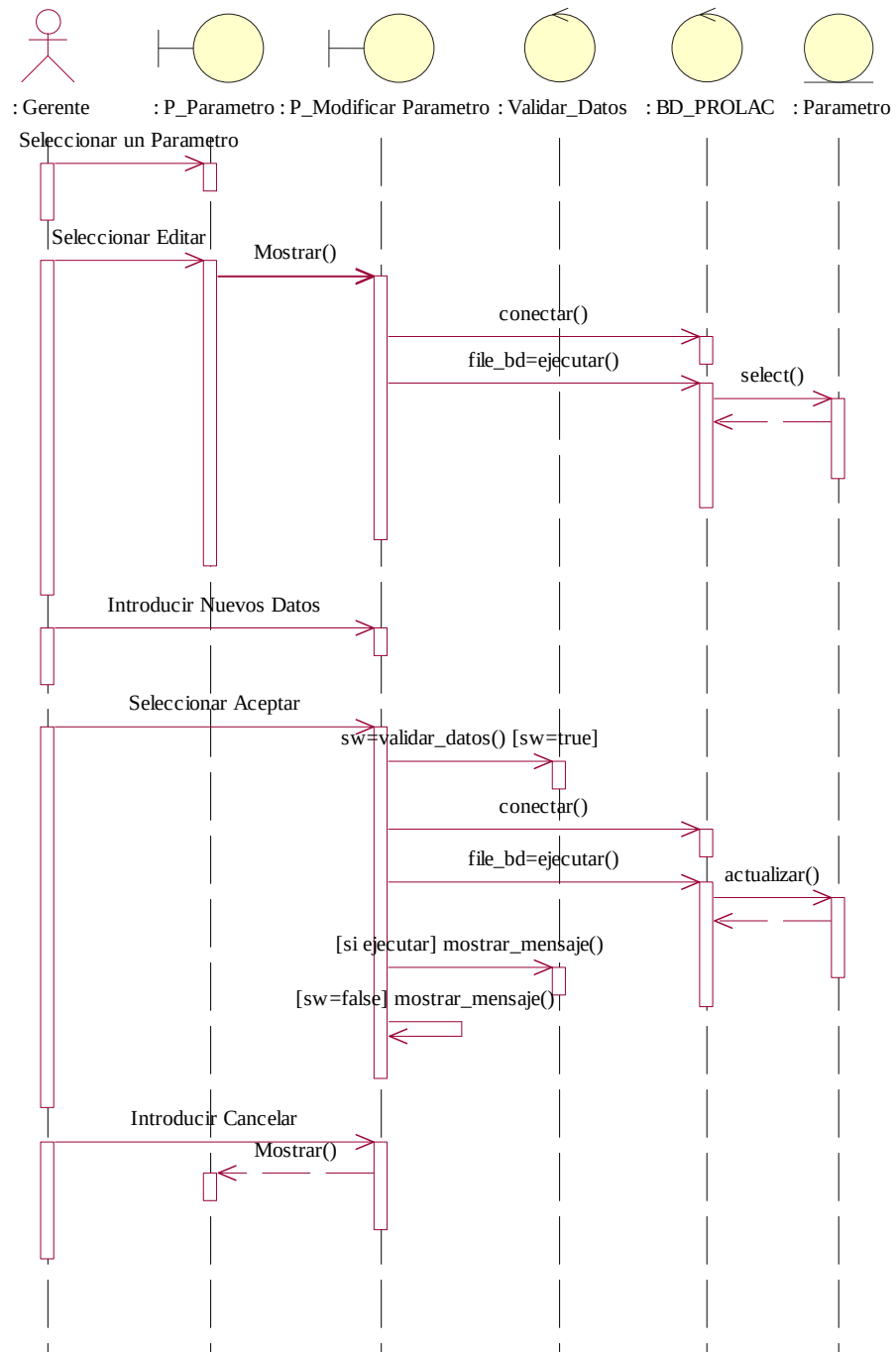


Figura N° 95: Diag. de Secuencia Editar Parámetro

Eliminar Parámetro

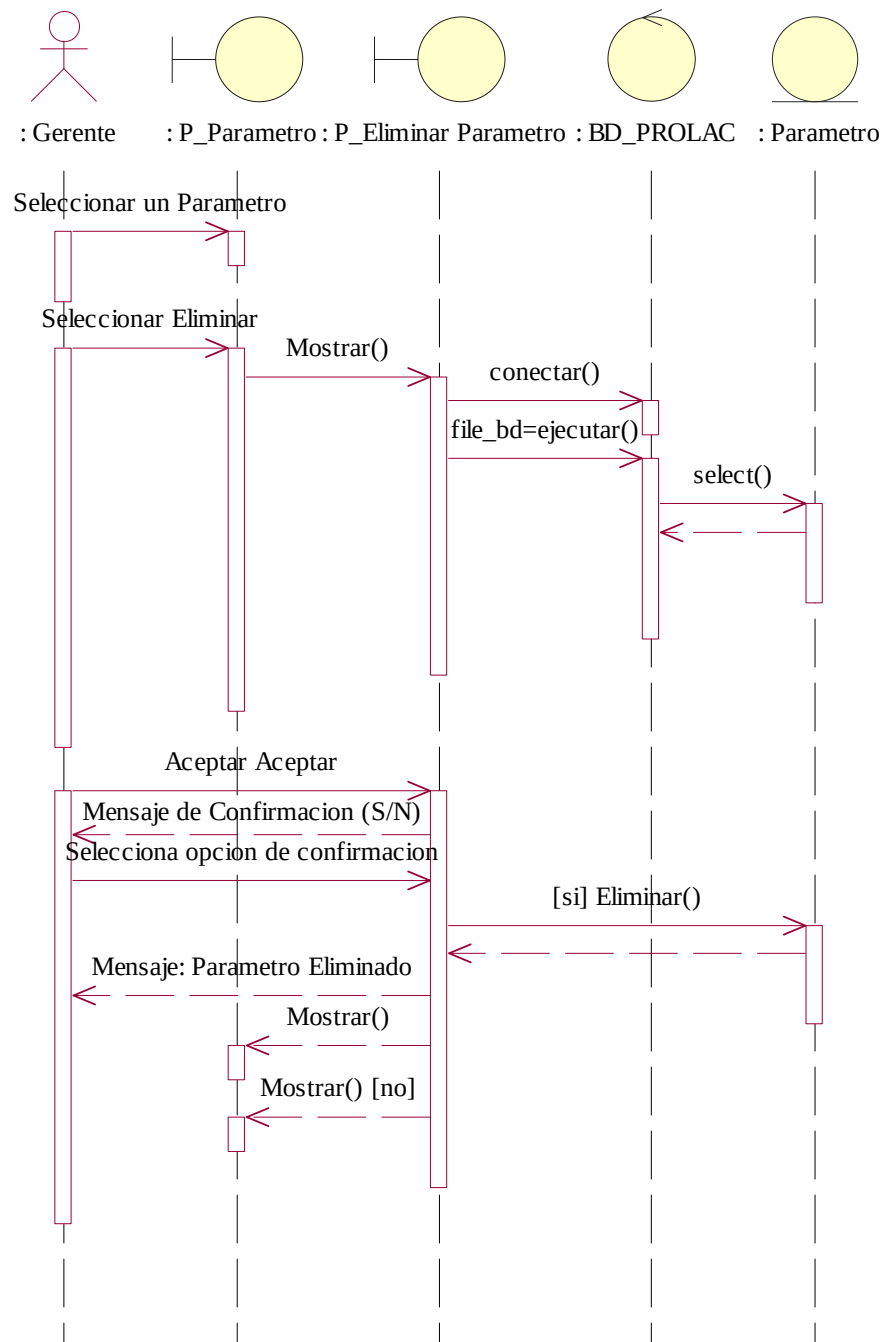


Figura N° 96: Diag. de Secuencia Eliminar Parámetro

II.1.4.2.4.4 MODULO II: PERSONAL

II.1.4.2.4.4.1 GESTIONAR EMPLEADOS

Listar Empleados

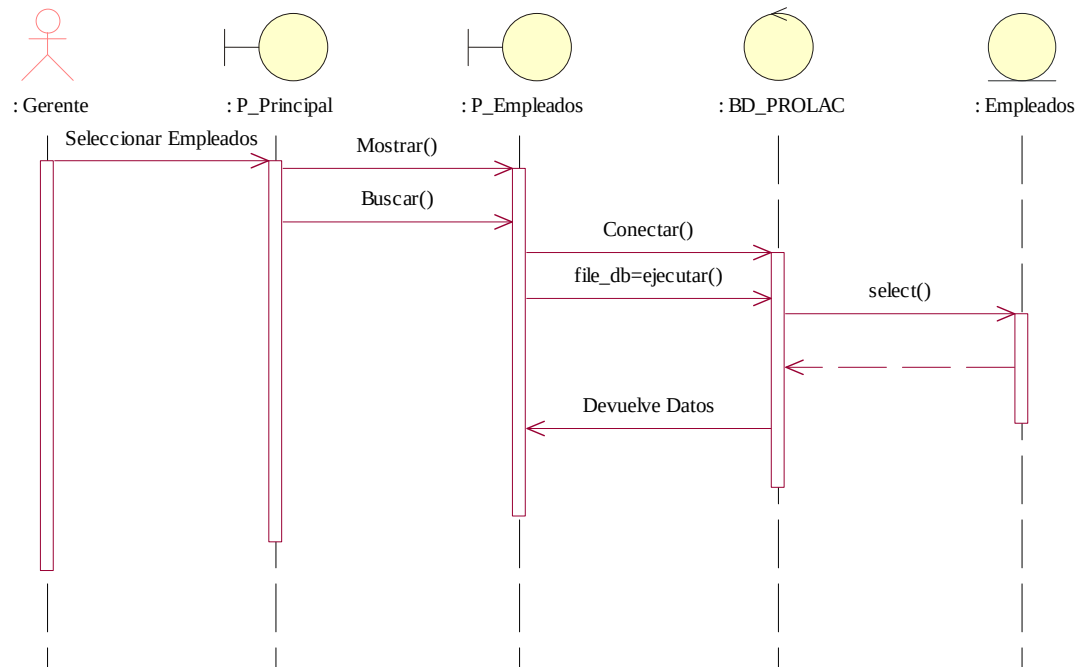


Figura N° 97: Diag. de Secuencia Listar Empleados

Crear Empleado

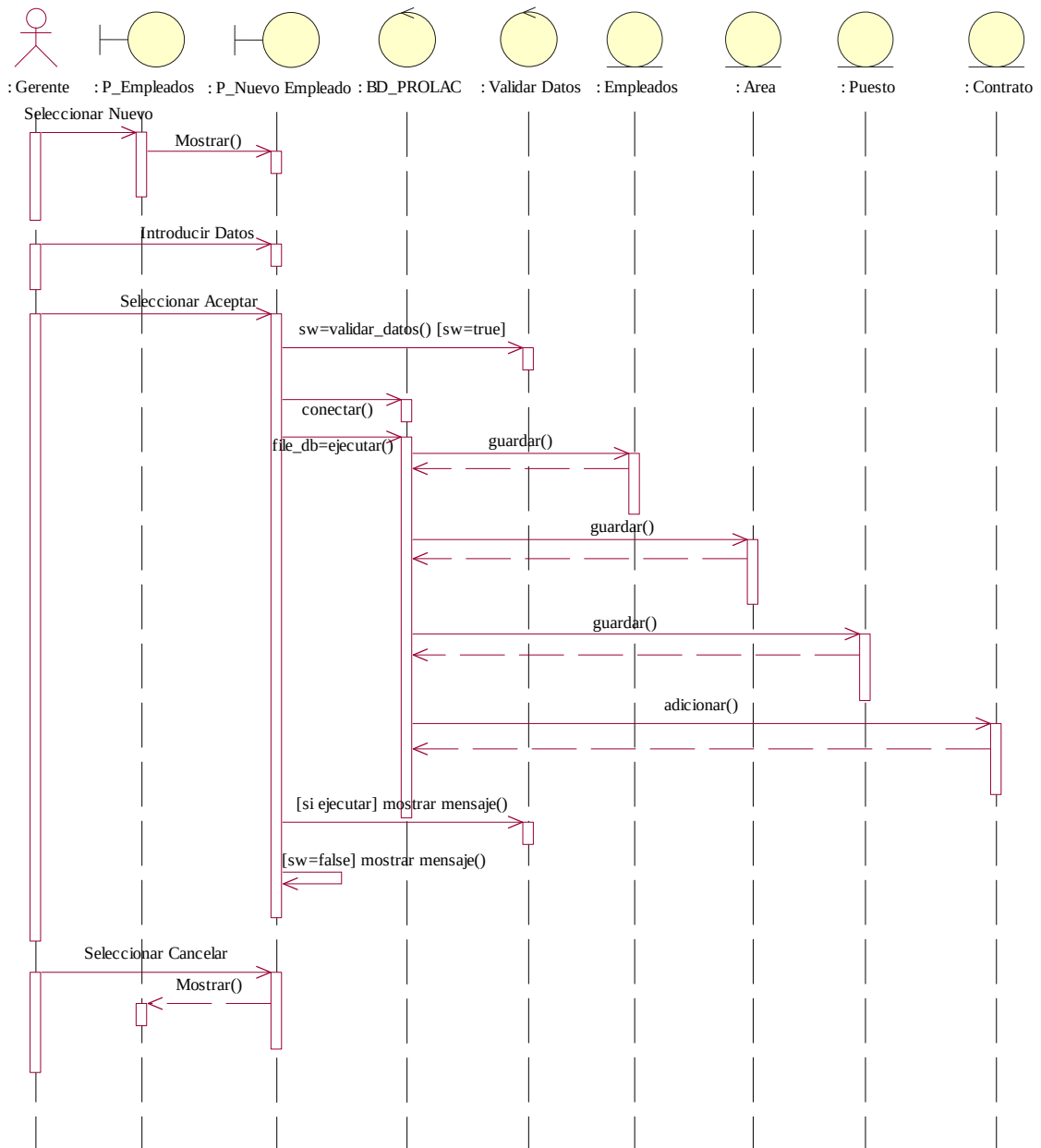


Figura N° 98: Diag. de Secuencia Crear Empleado

Editar Empleado

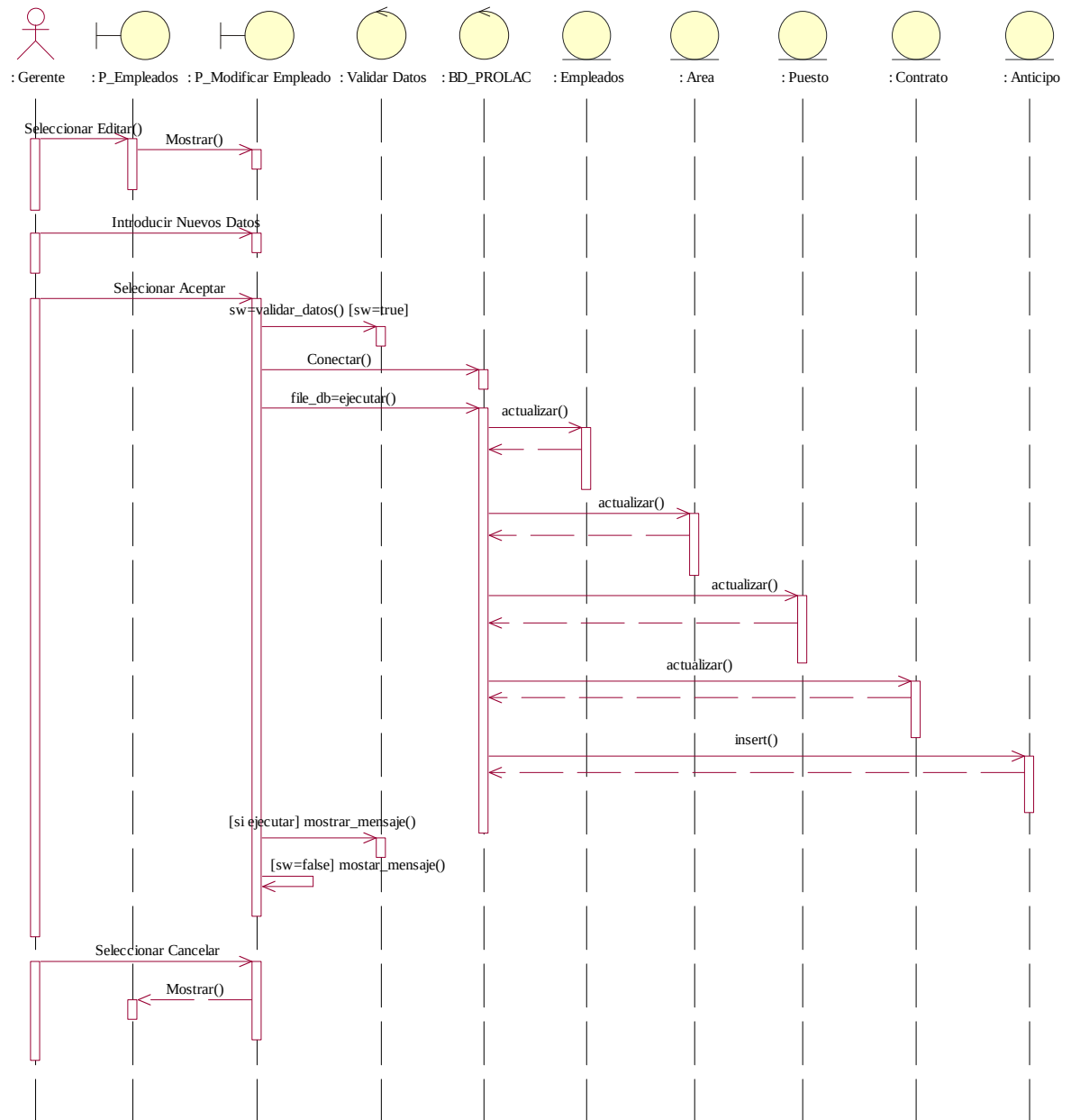


Figura N° 99: Diag. de Secuencia Editar Empleado

Eliminar Empleado

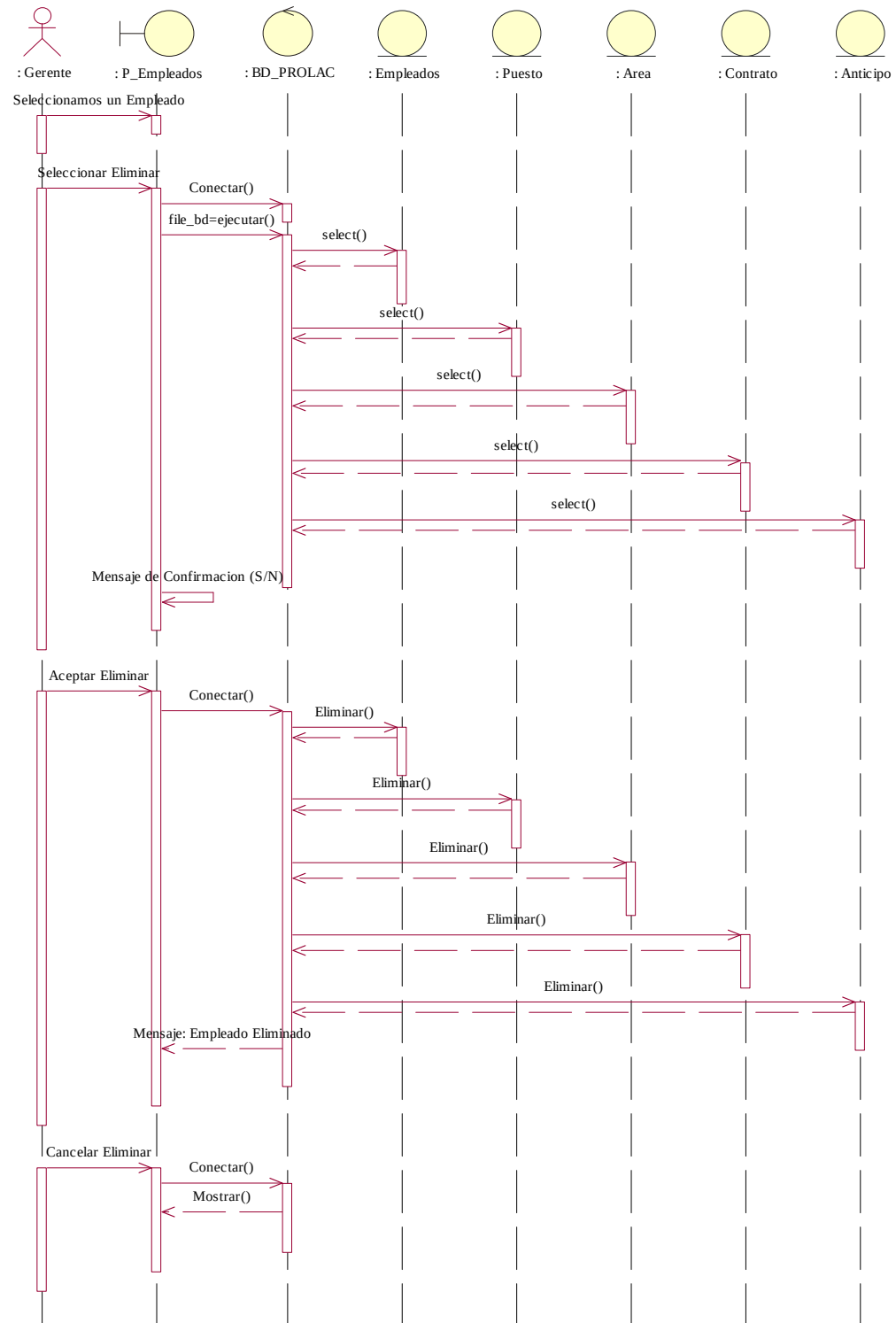


Figura N° 100: Diag. de Secuencia Eliminar Empleado

II.1.4.2.4.4.2 GESTIONAR AREAS

Listar Áreas

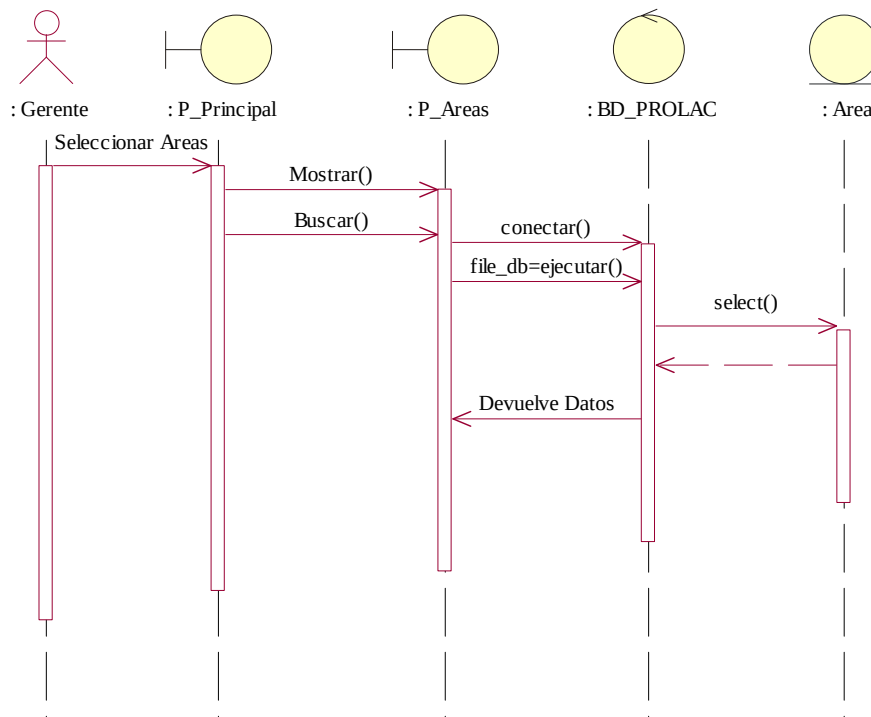


Figura N° 101: Diag. de Secuencia Listar Áreas

Crear Área

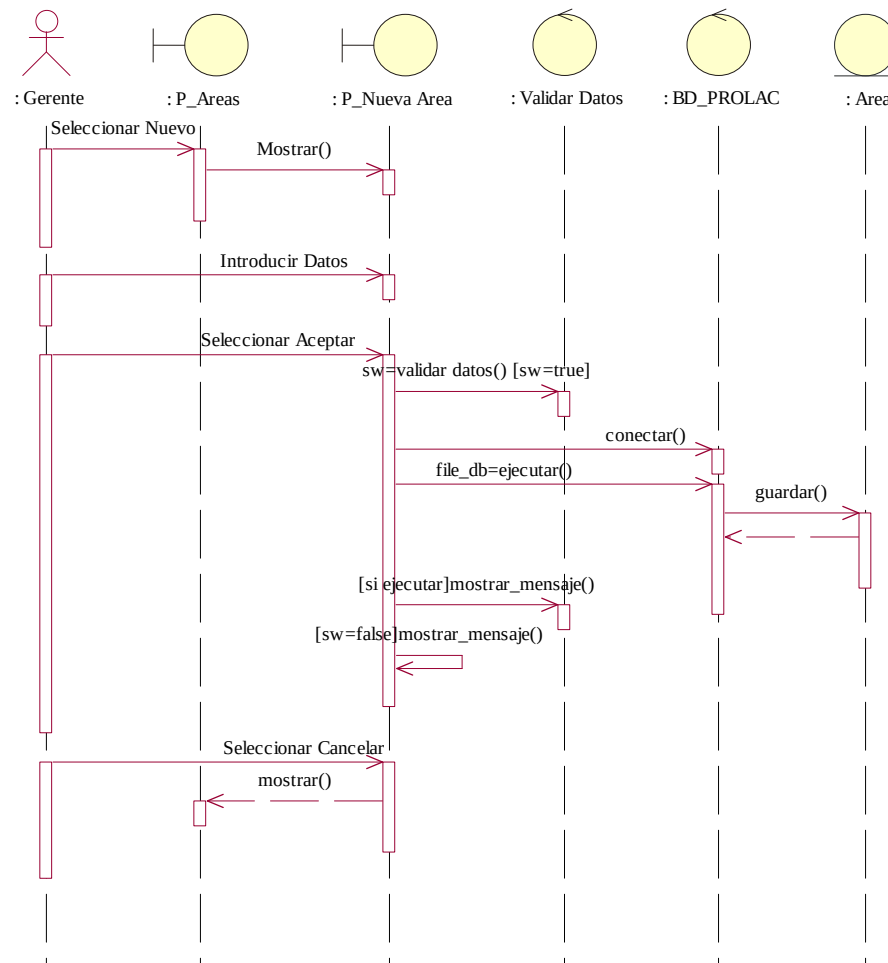


Figura N° 102: Diag. de Secuencia Crear Área

Editar Área

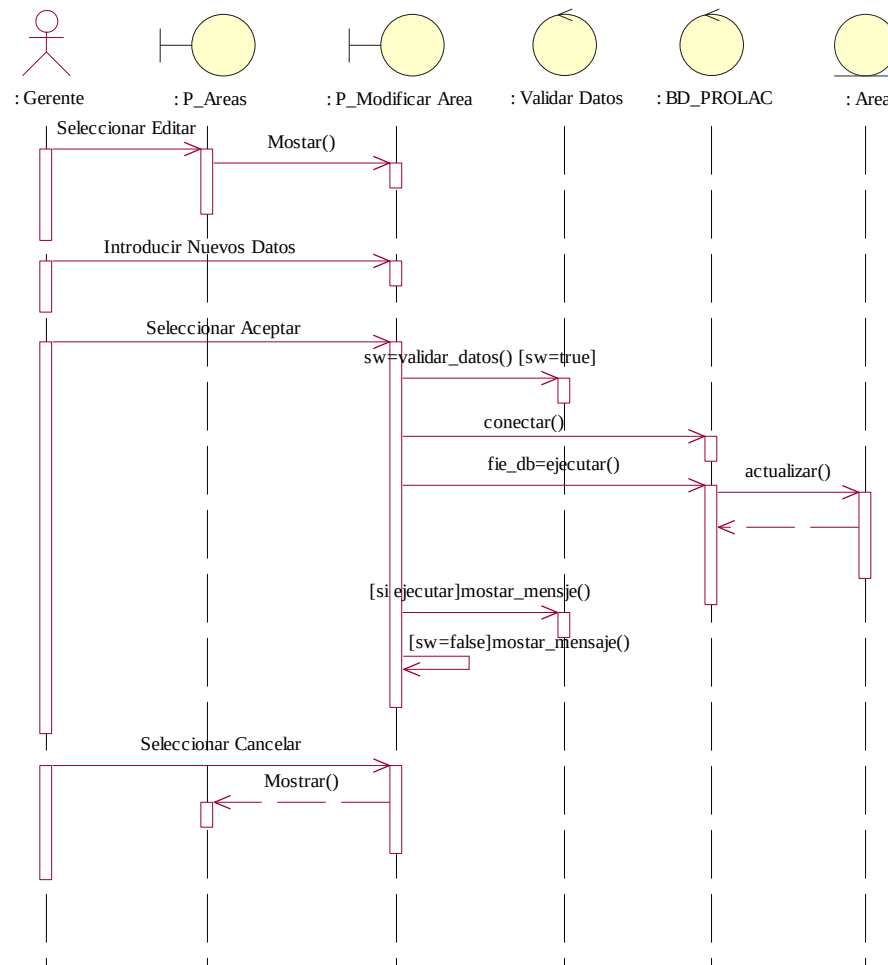


Figura N° 103: Diag. de Secuencia Editar Área

Eliminar Área

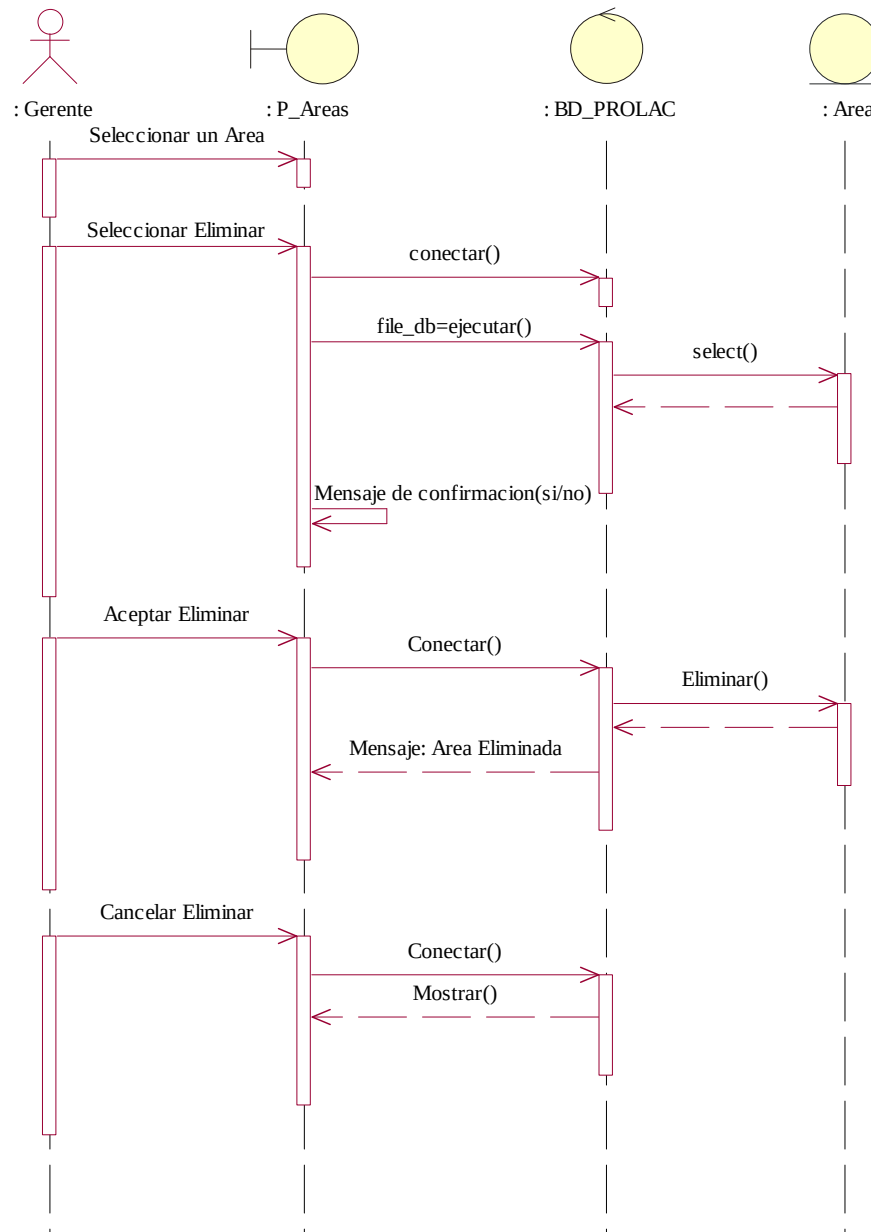


Figura N° 104: Diag. de Secuencia Eliminar Área

II.1.4.2.4.4.3 GESTIONAR PUESTOS

Listar Puestos

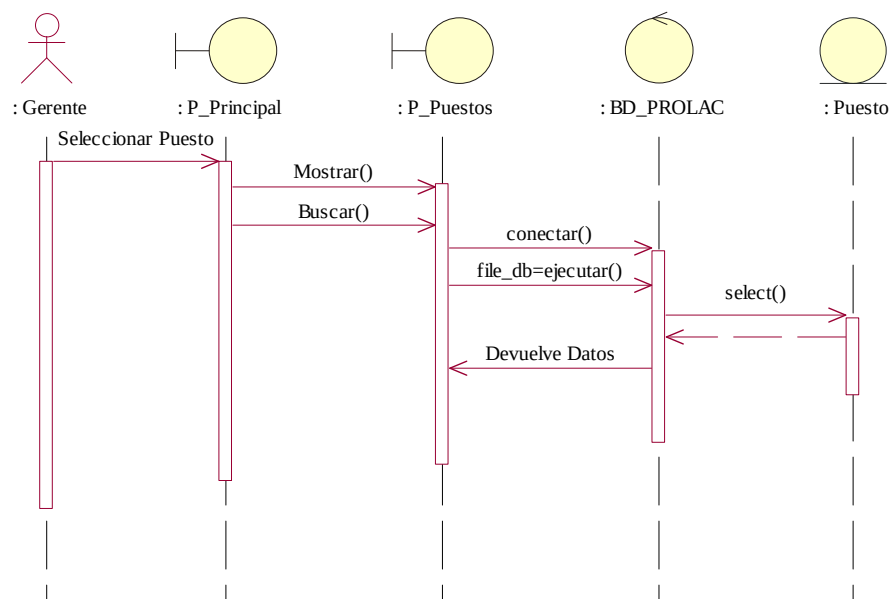


Figura N° 105: Diag. de Secuencia Listar Puestos

Crear Puesto

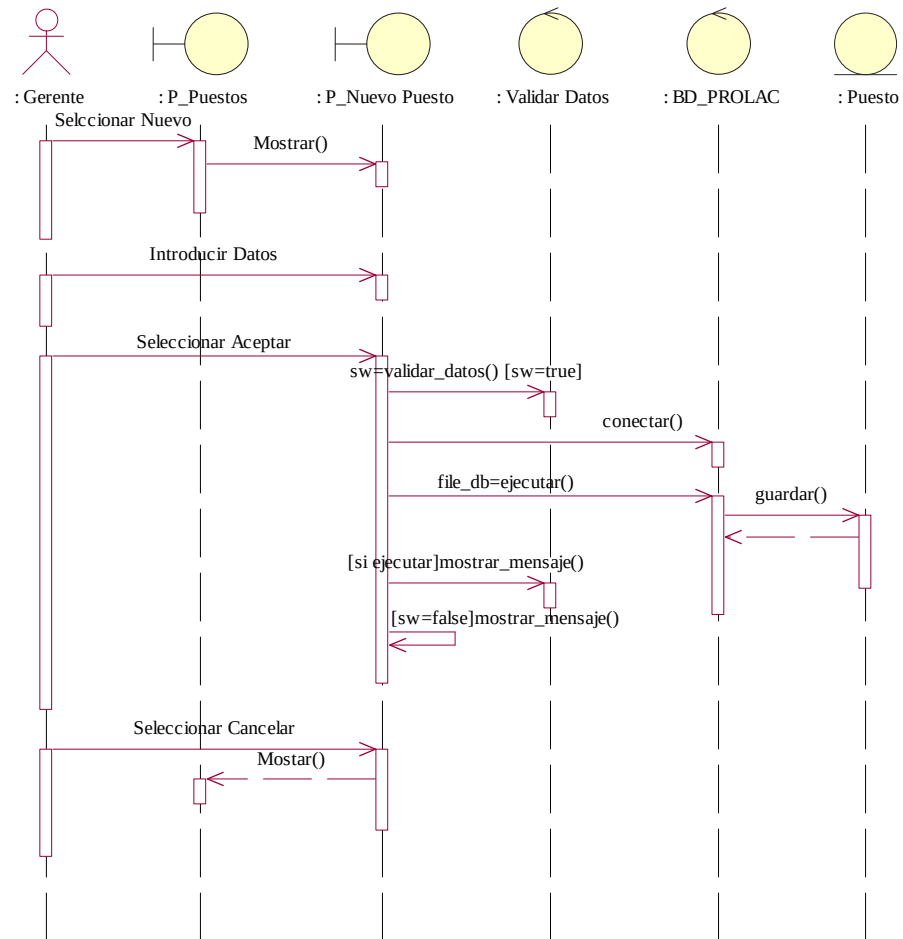


Figura N° 106: Diag. de Secuencia Crear Puesto

Editar Puesto

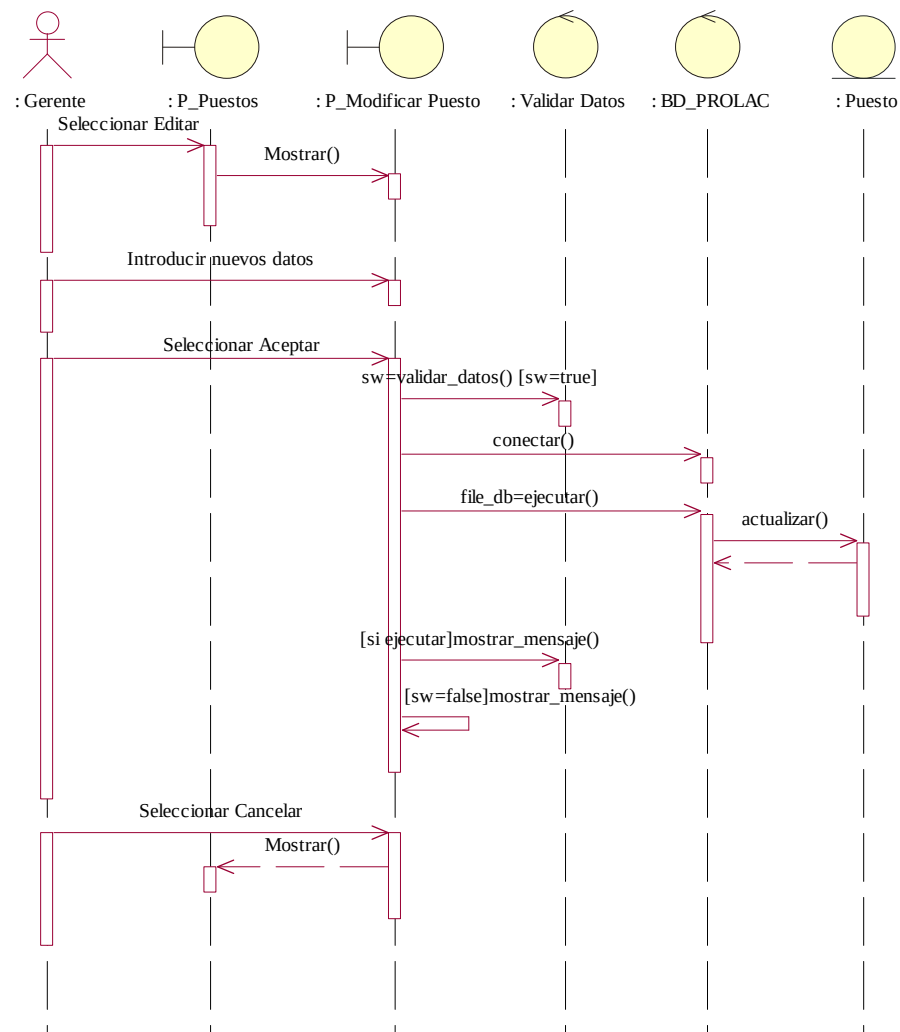


Figura N° 107: Diag. de Secuencia Editar Puesto

Eliminar Puesto

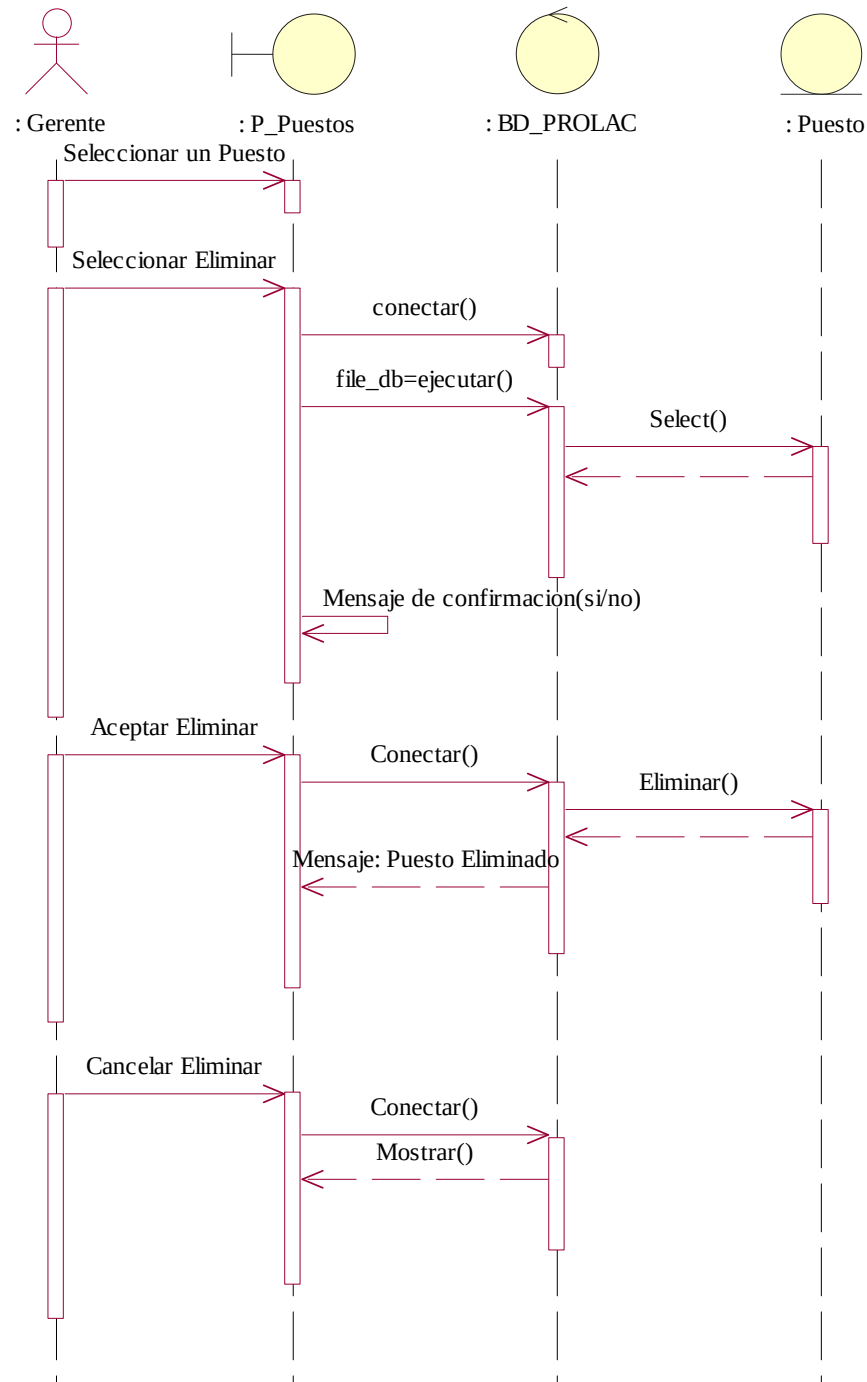


Figura N° 108: Diag. de Secuencia Eliminar Puesto

II.1.4.2.4.4.4 GESTIONAR PERMISOS-VACACIONES

Listar Empleados

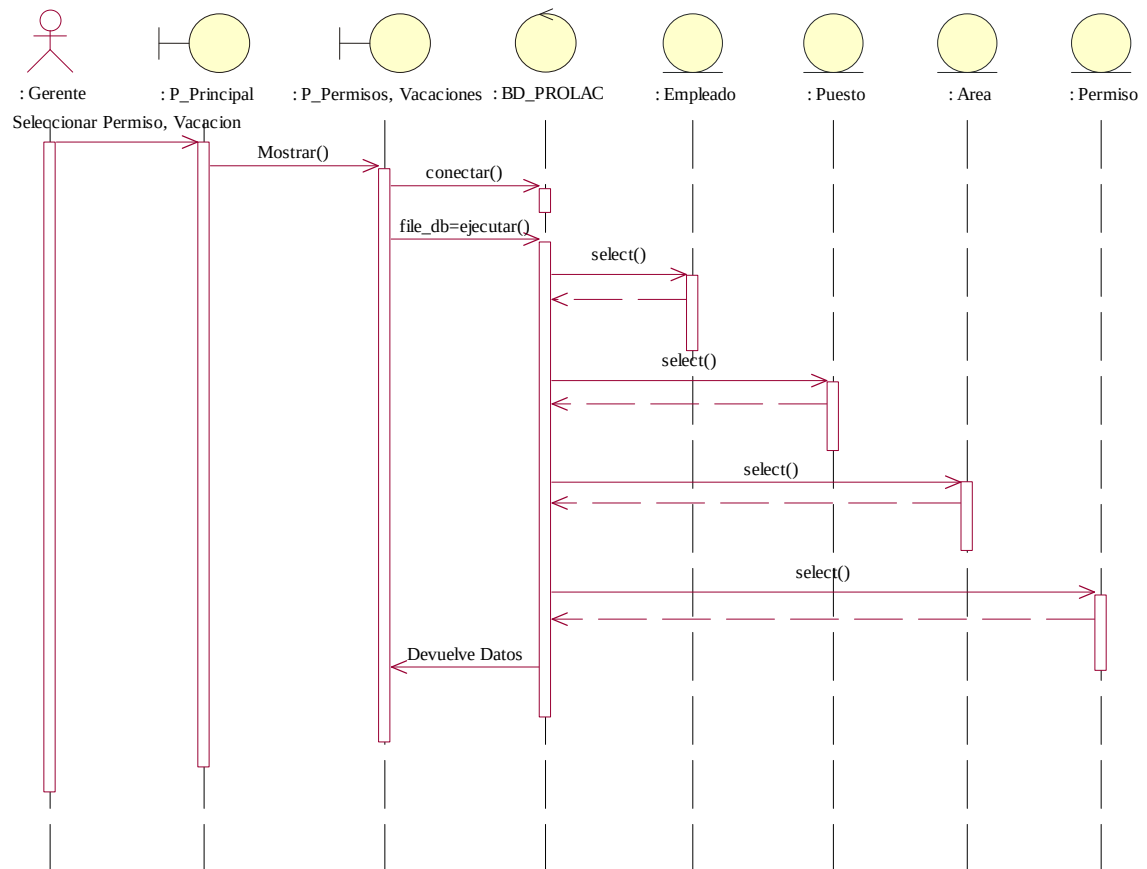


Figura N° 109 Diag. de Secuencia Listar Empleados

Crear Permiso-Vacación

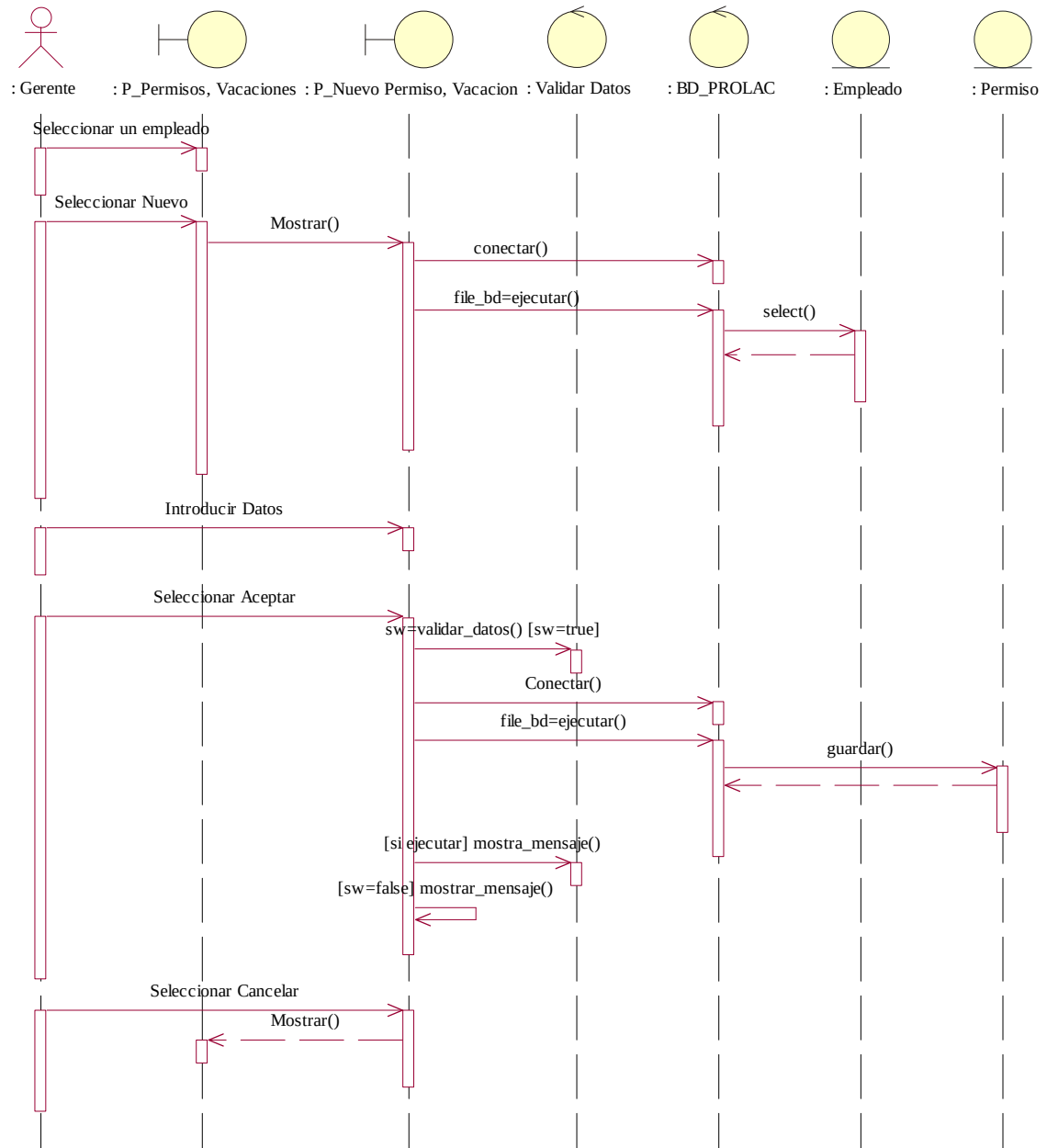


Figura N° 110: Diag. de Secuencia Crear Permiso-Vacación

Editar Permiso-Vacación

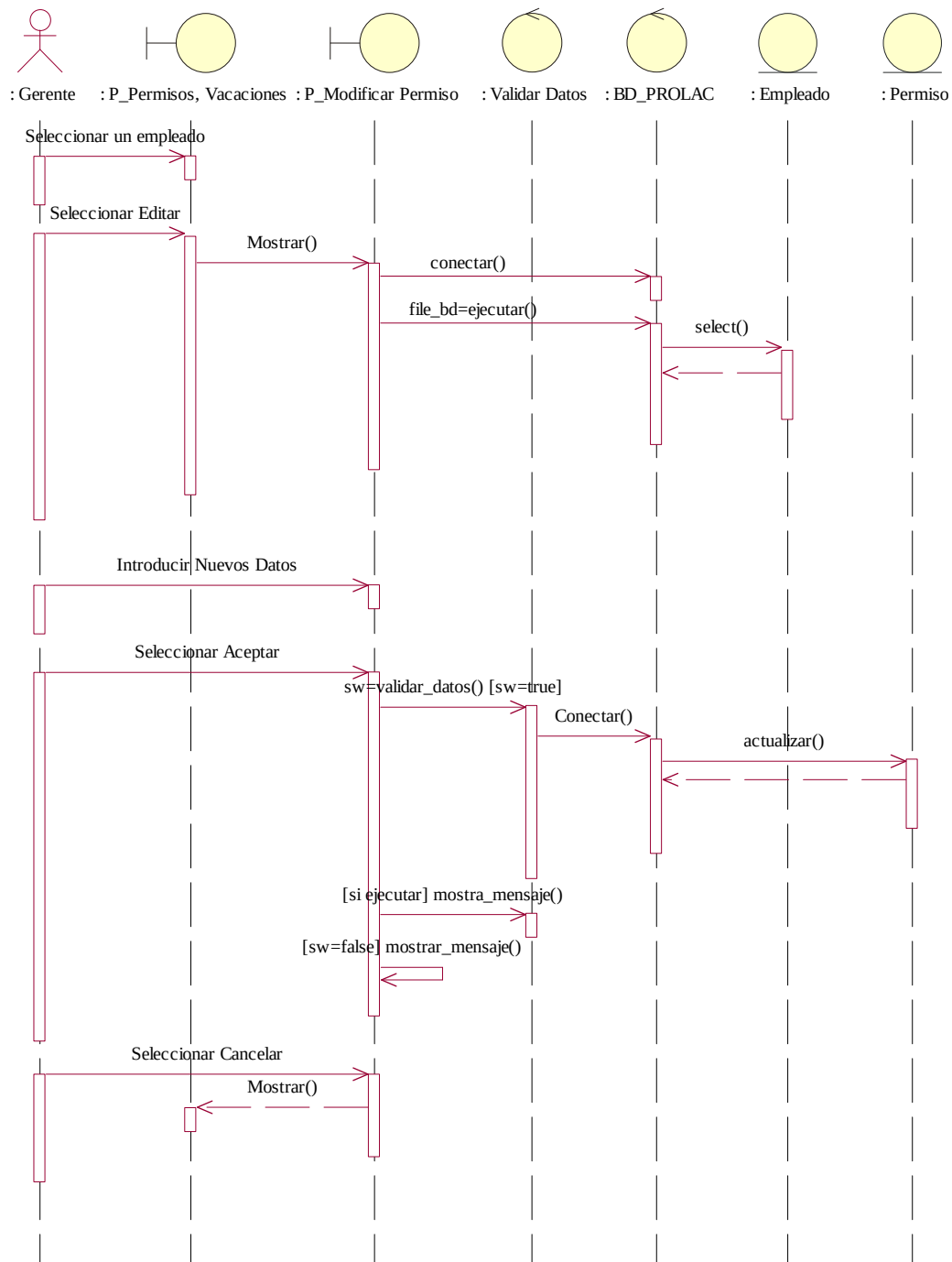


Figura N° 111: Diag. de Secuencia Editar Permiso-Vacación

Eliminar Permiso-Vacación

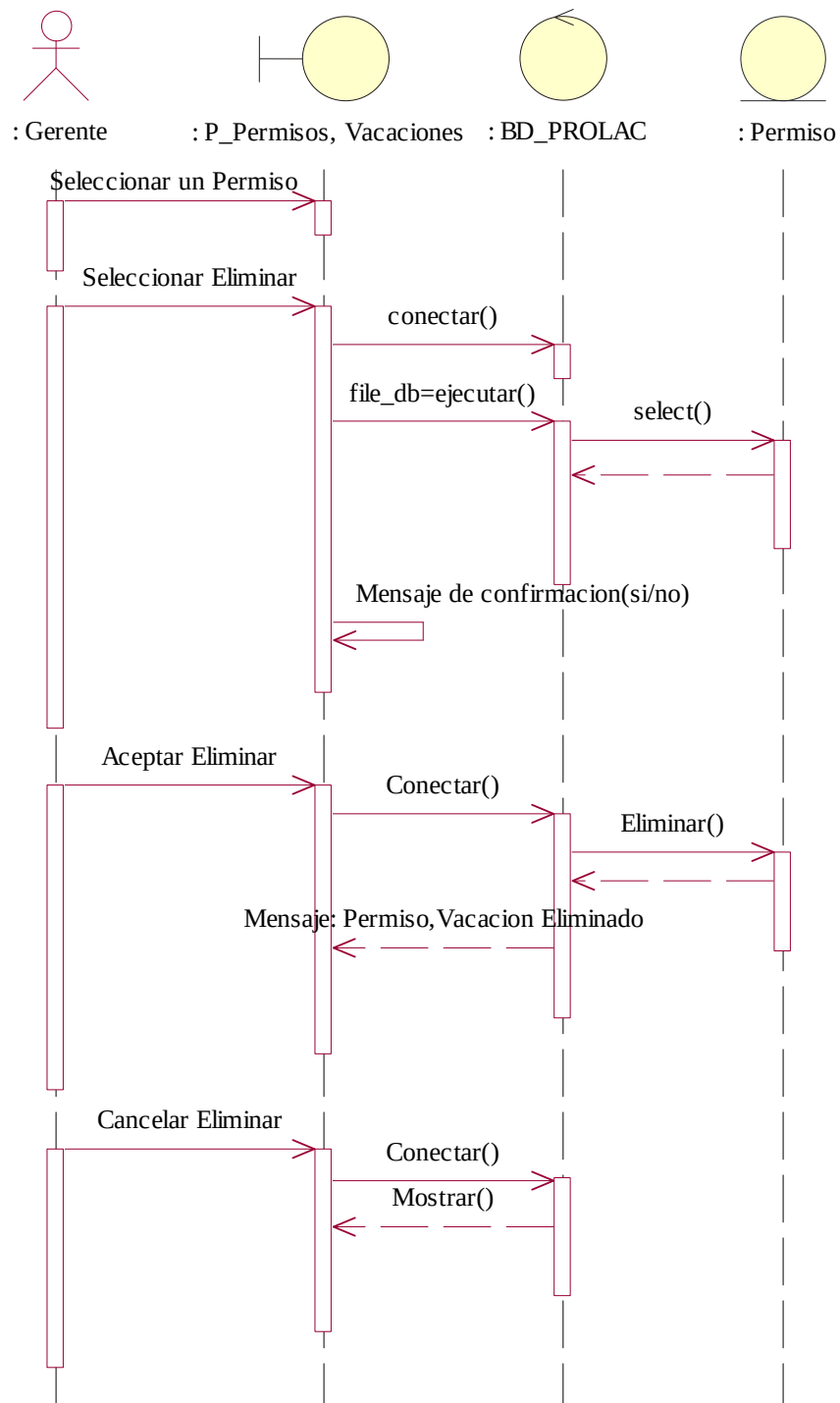


Figura N° 112: Diag. de Secuencia Eliminar Permiso-Vacación

II.1.4.2.4.4.5 GENERAR PLANILLAS DE SUELDOS

Crear Planilla de Sueldo

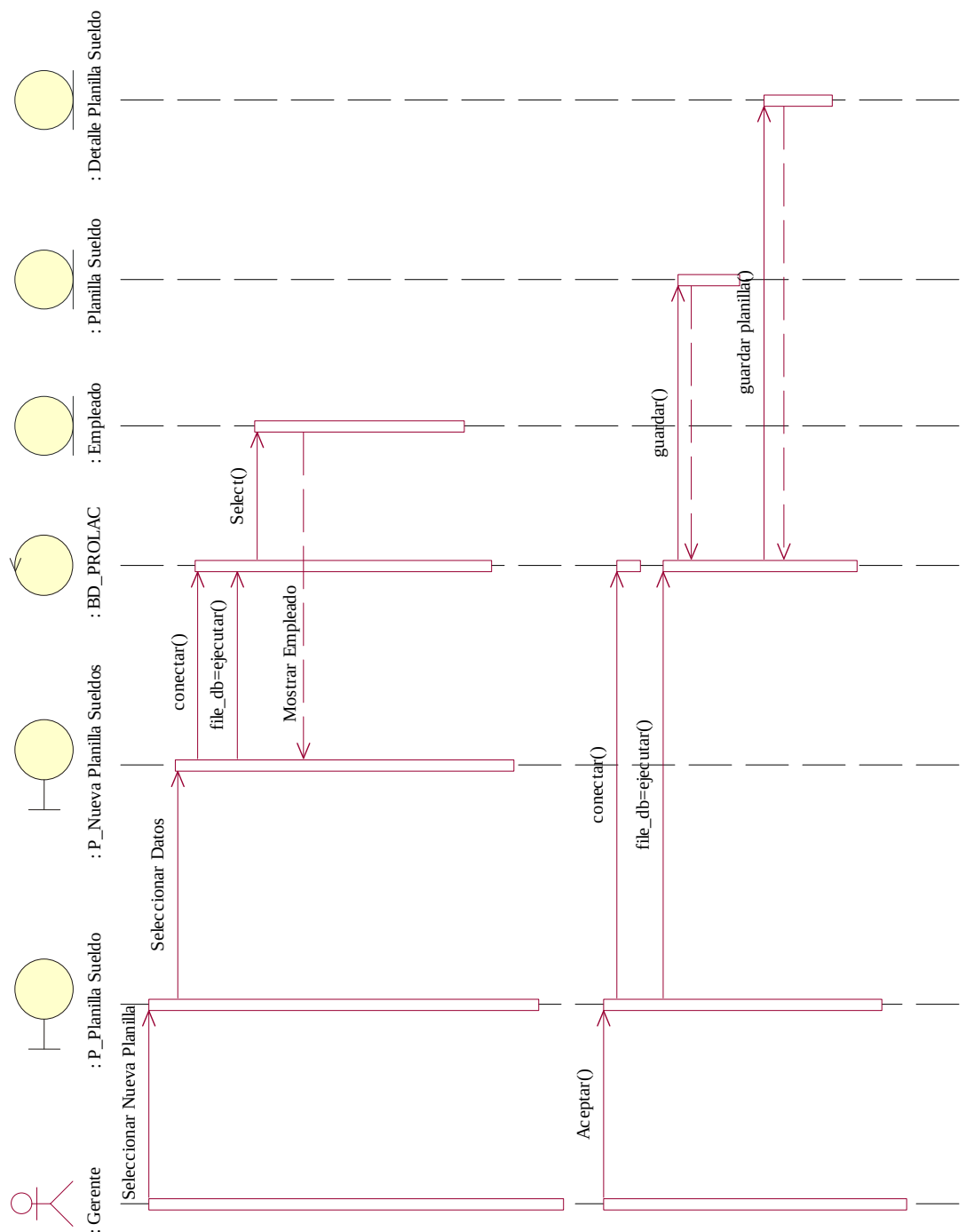


Figura N° 113: Diag. de Secuencia Crear Planilla de Sueldo

Editar Planilla de Sueldo

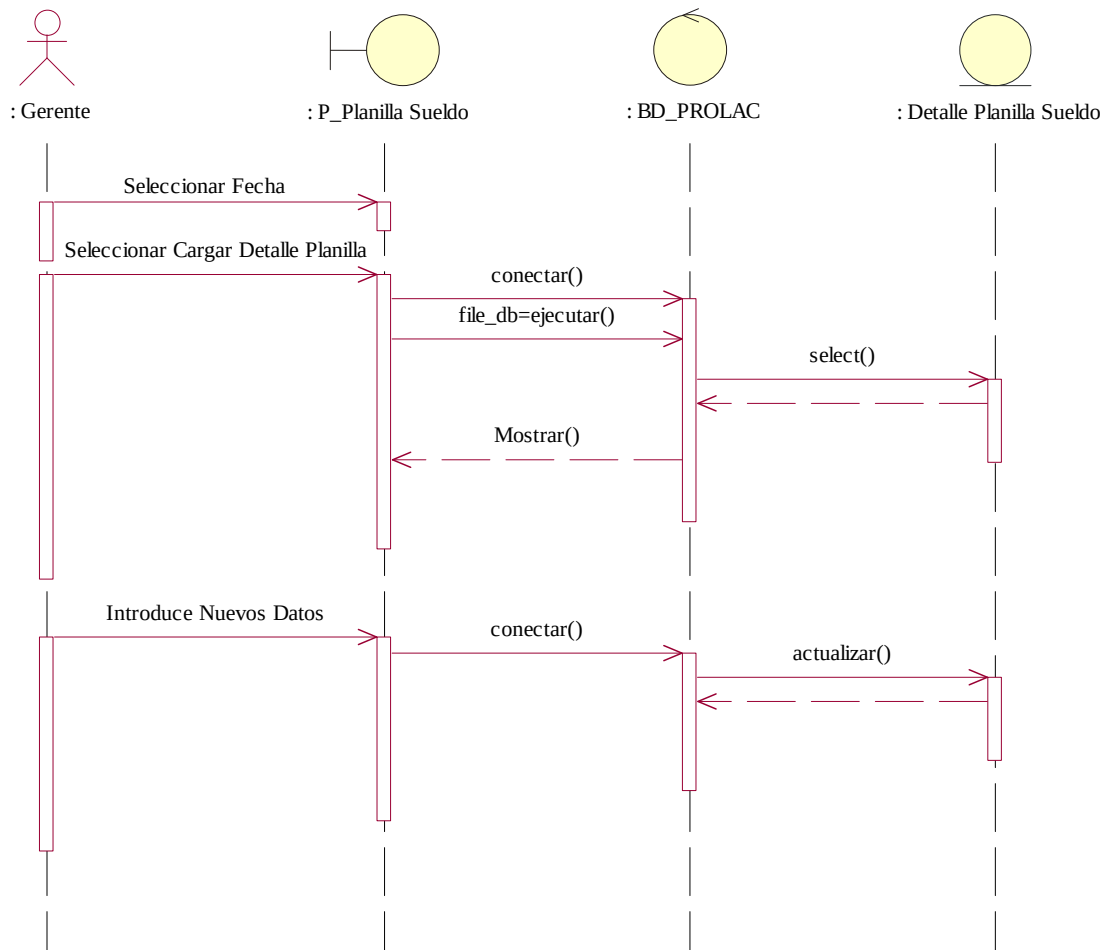


Figura N° 114: Diag. de Secuencia Editar Planilla de Sueldo

Cargar Detalle Planilla de Sueldo

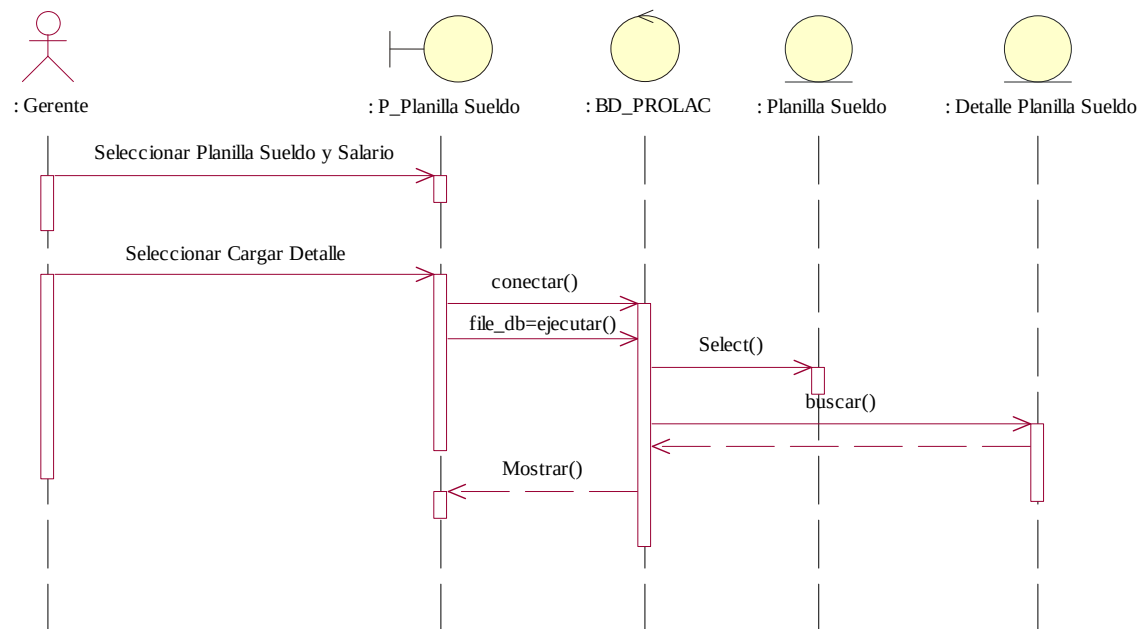


Figura N° 115: Diag. de Secuencia Cargar Detalle Planilla de Sueldo

II.1.4.2.4.4.6 GENERAR PLANILLA DE AGUINALDOS

Crear Planilla de Aguinaldo

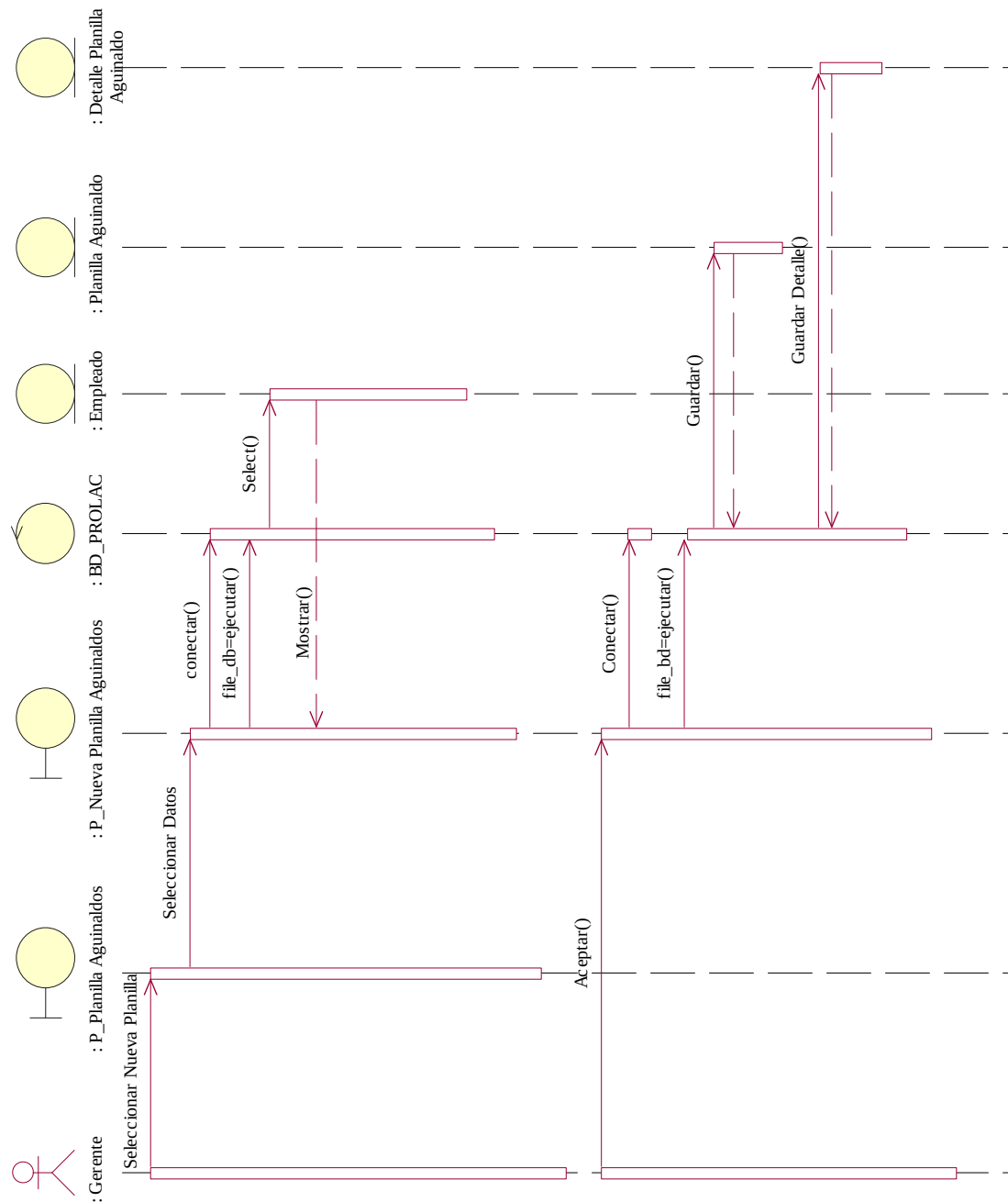


Figura N° 116: Diag. de Secuencia Crear Planilla de Aguinaldo

Cargar Detalle Planilla de Aguinaldo

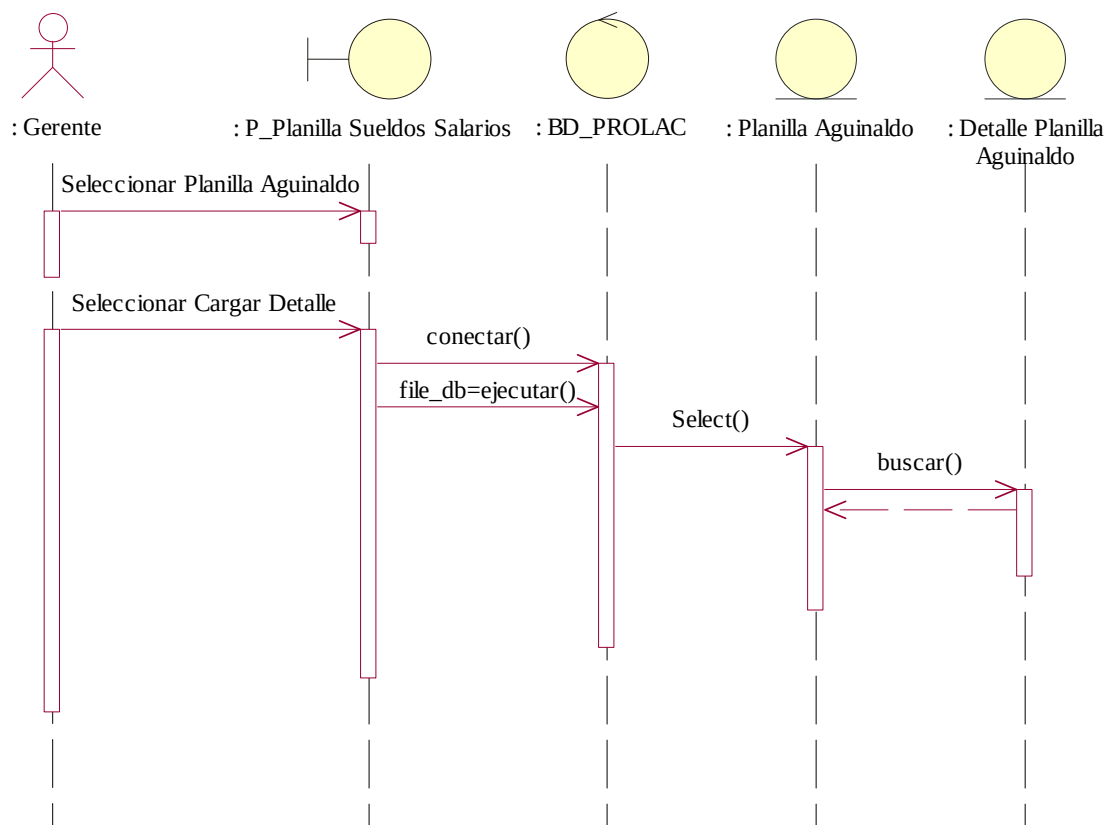


Figura N° 117: Diag. de Secuencia Cargar Detalle Planilla de Aguinaldo

II.1.4.2.4.5 MODULO III: CONTROL PERSONAL

II.1.4.2.4.5.1 GESTIONAR DIA INHABIL

Listar Día Inhábil

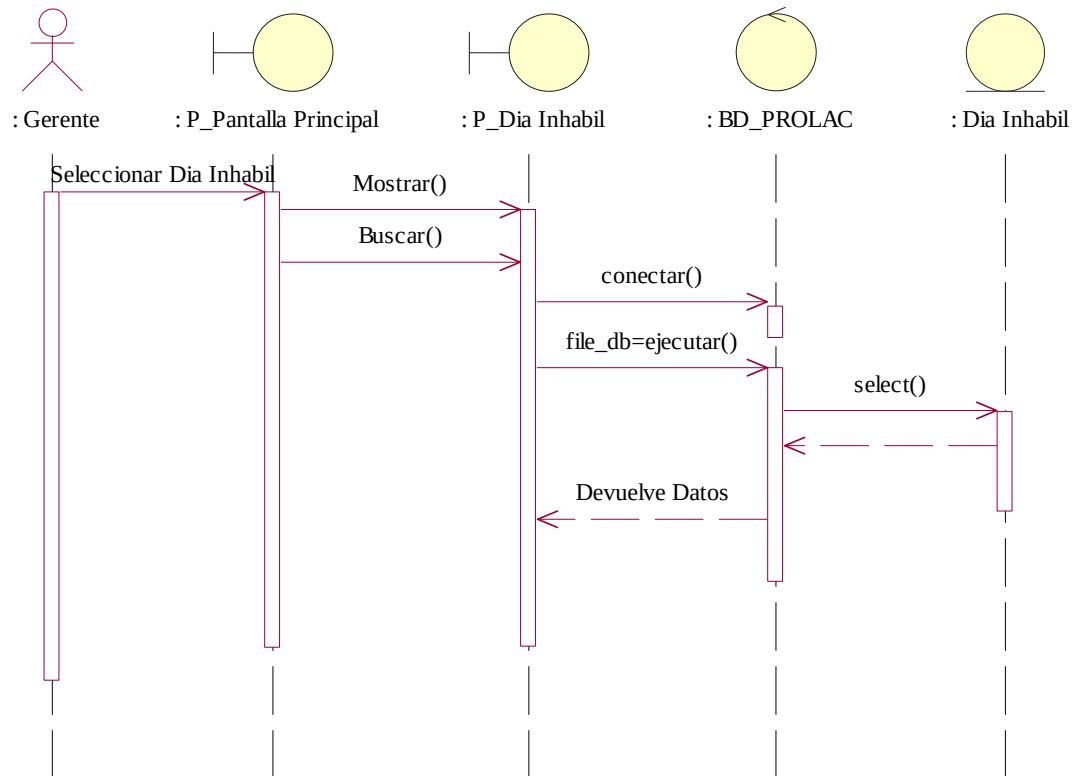


Figura N° 118: Diag. de Secuencia Listar Día Inhábil

Crear Día Inhábil

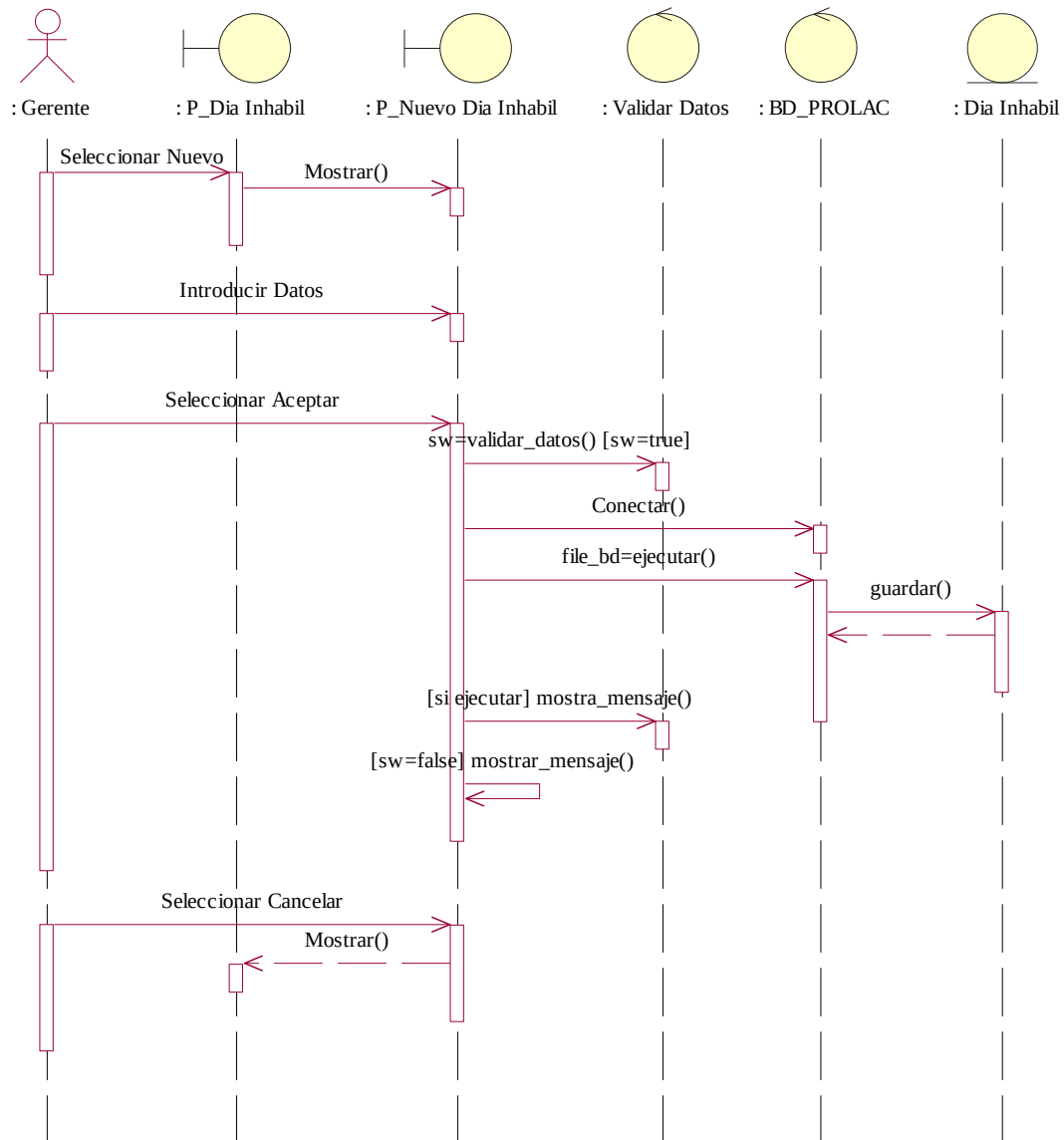


Figura N° 119: Diag. de Secuencia Crear Día Inhábil

Editar Día Inhábil

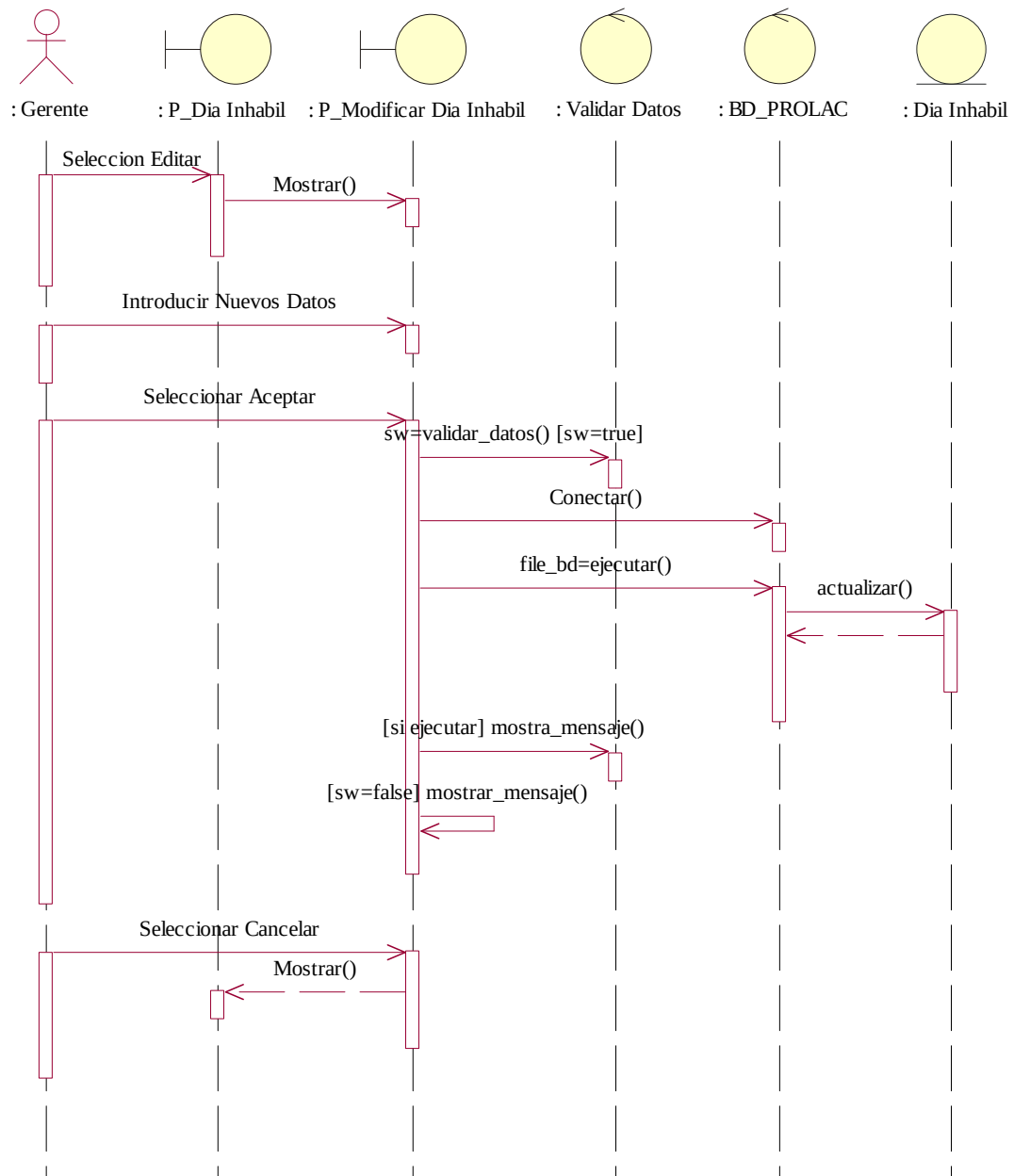


Figura N° 120: Diag. de Secuencia Editar Día Inhábil

Eliminar Día Inhábil

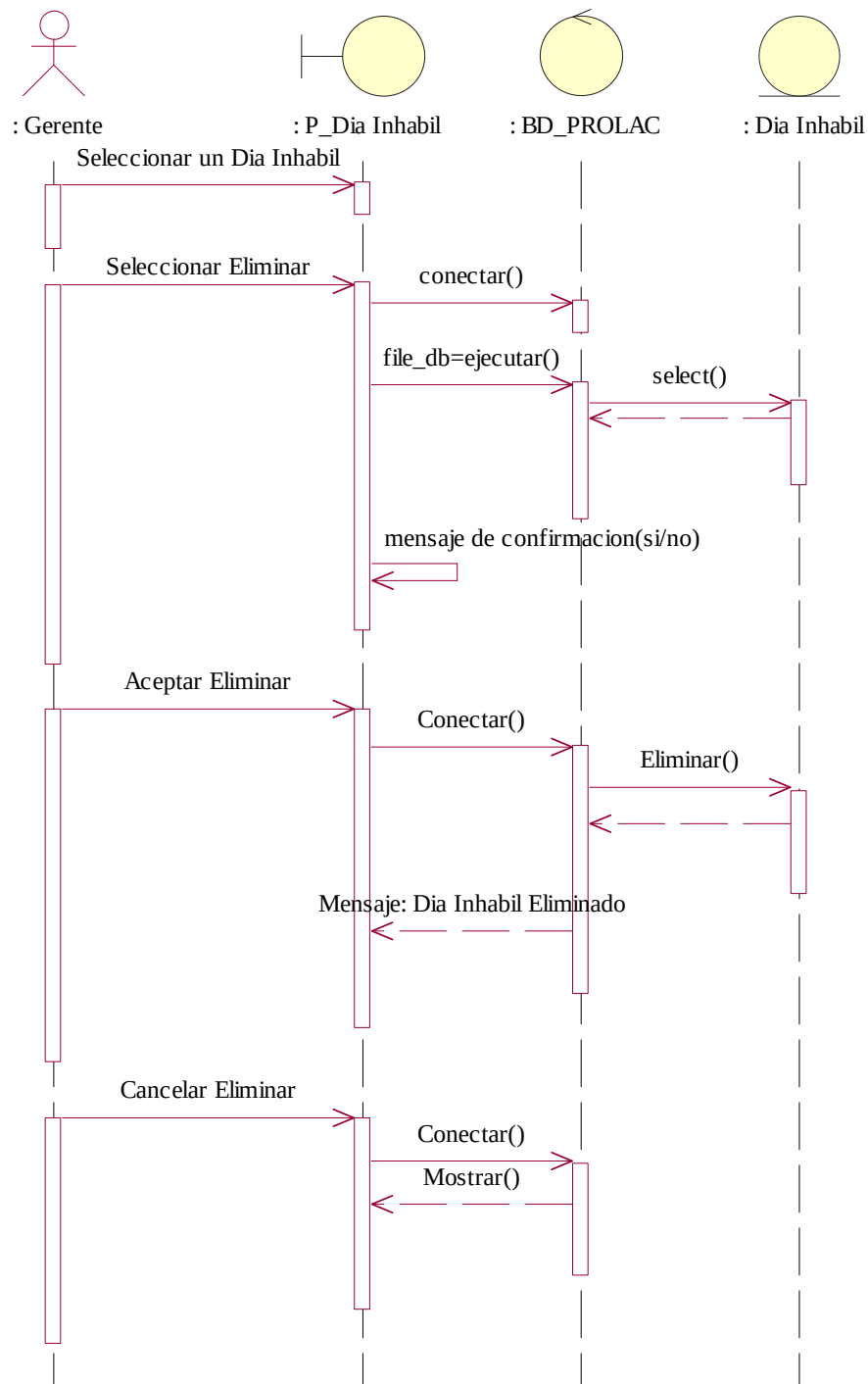


Figura N° 121: Diag. de Secuencia Eliminar Día Inhábil

II.1.4.2.4.5.2 GESTIONAR HORARIO

Listar Horario

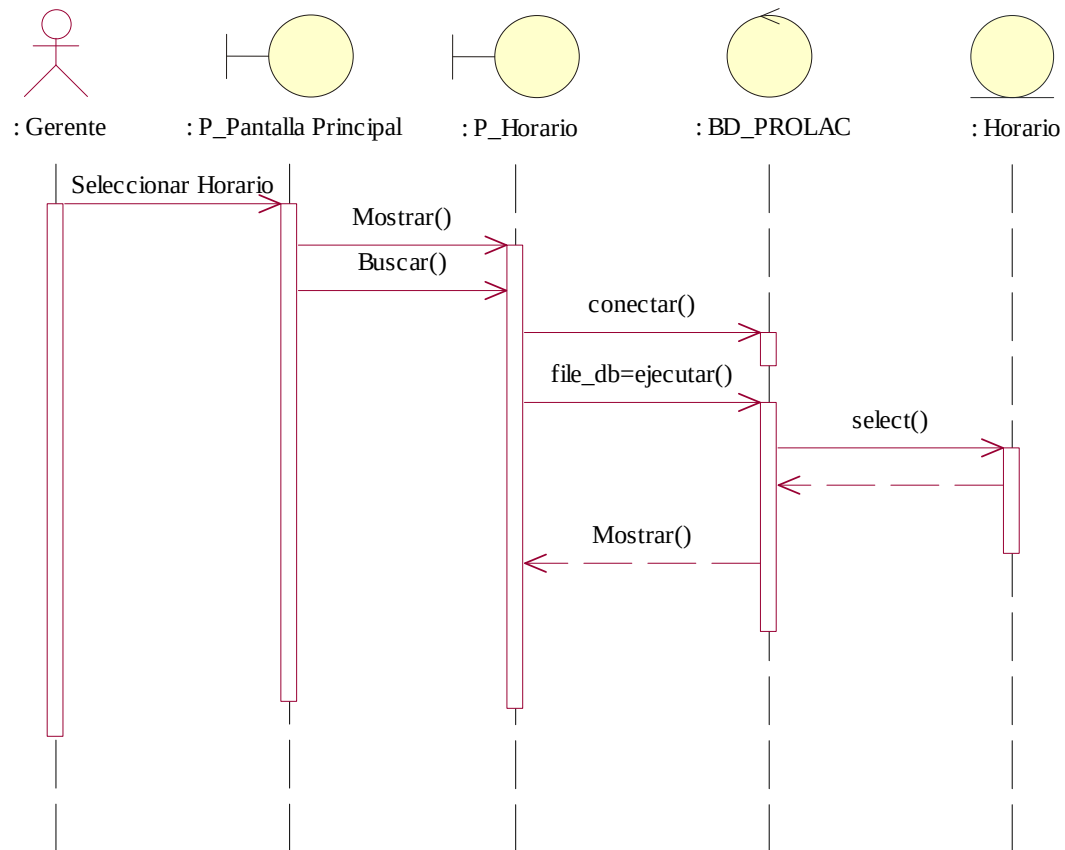


Figura N° 122: Diag. de Secuencia Listar Horario

Crear Horario

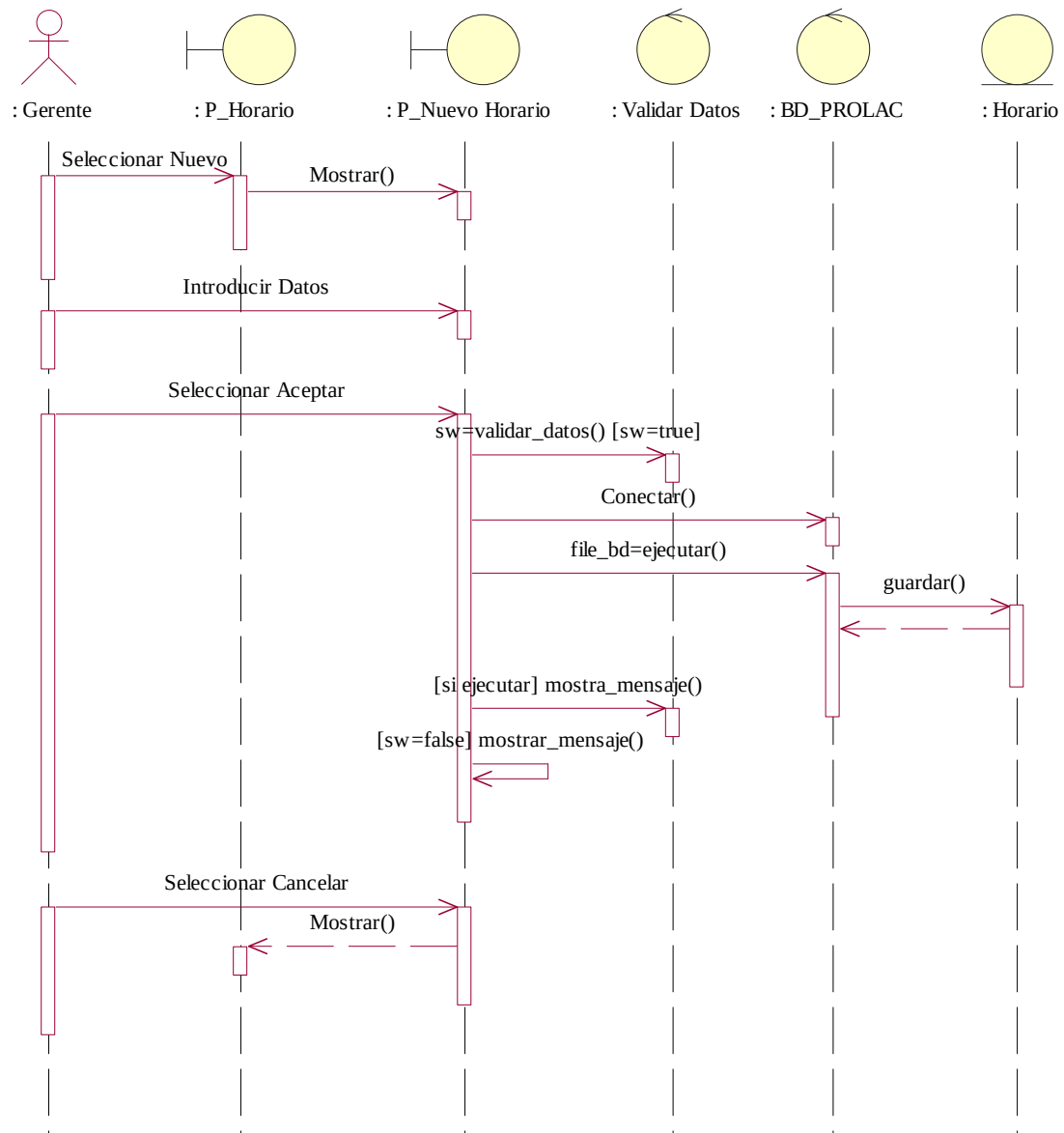


Figura N° 123: Diag. de Secuencia Crear Horario

Editar Horario

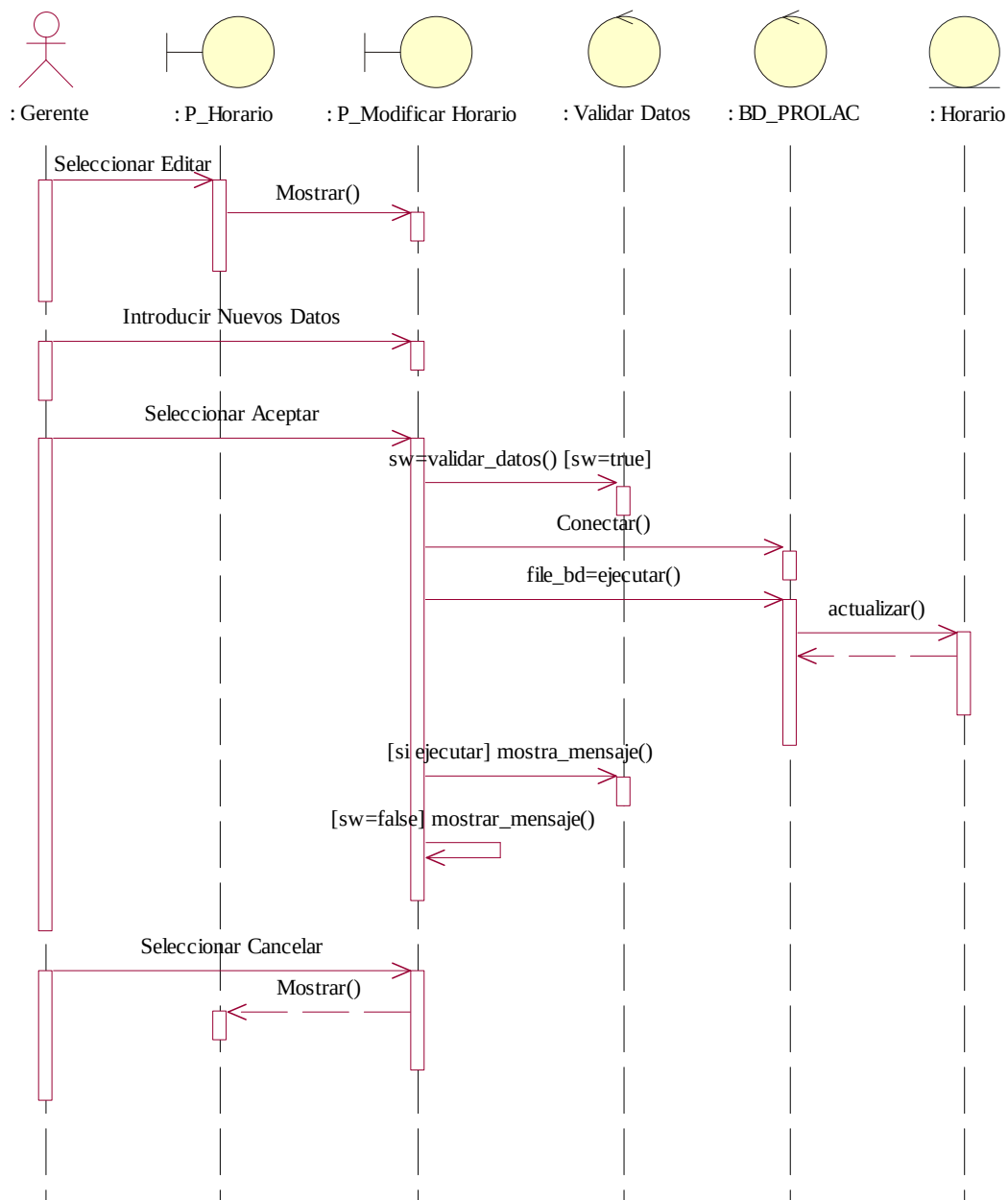


Figura N° 124: Diag. de Secuencia Editar Horario

Eliminar Horario

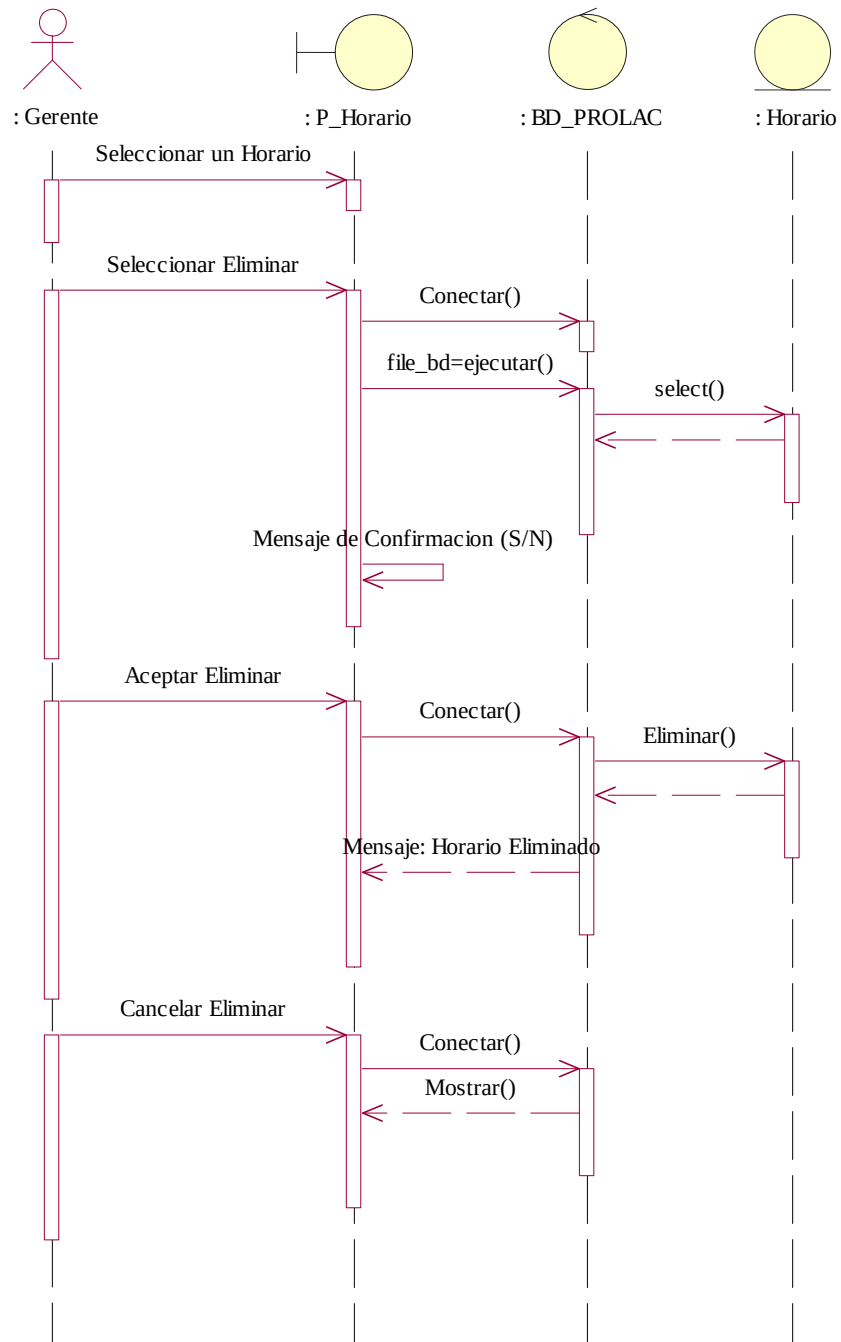


Figura N° 125: Diag. de Secuencia Eliminar Horario

II.1.4.2.4.5.3 GESTIONAR EDICION MANUAL

Listar Empleados

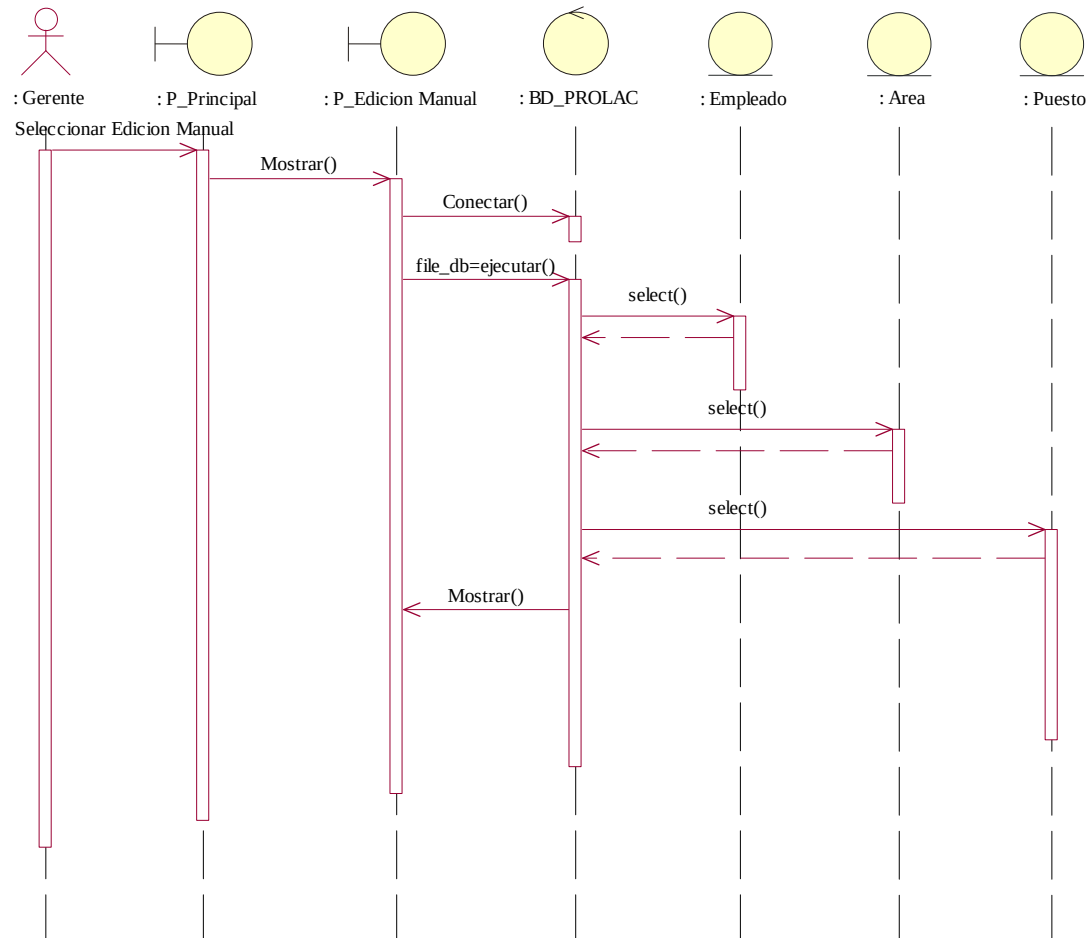


Figura N° 126: Diag. de Secuencia Listar Empleados

Crear Evento

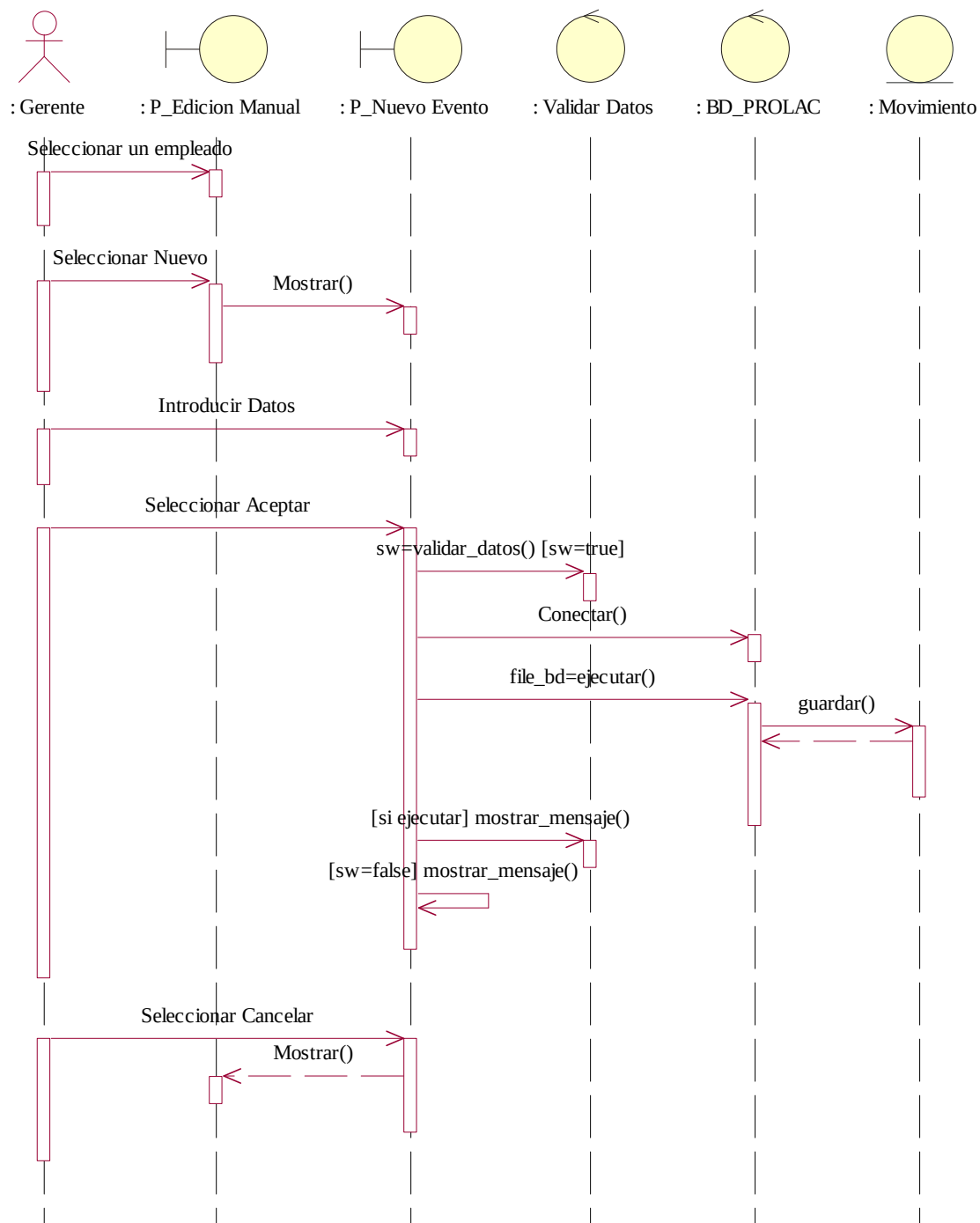


Figura N° 127: Diag. de Secuencia Crear Evento

Editar Evento

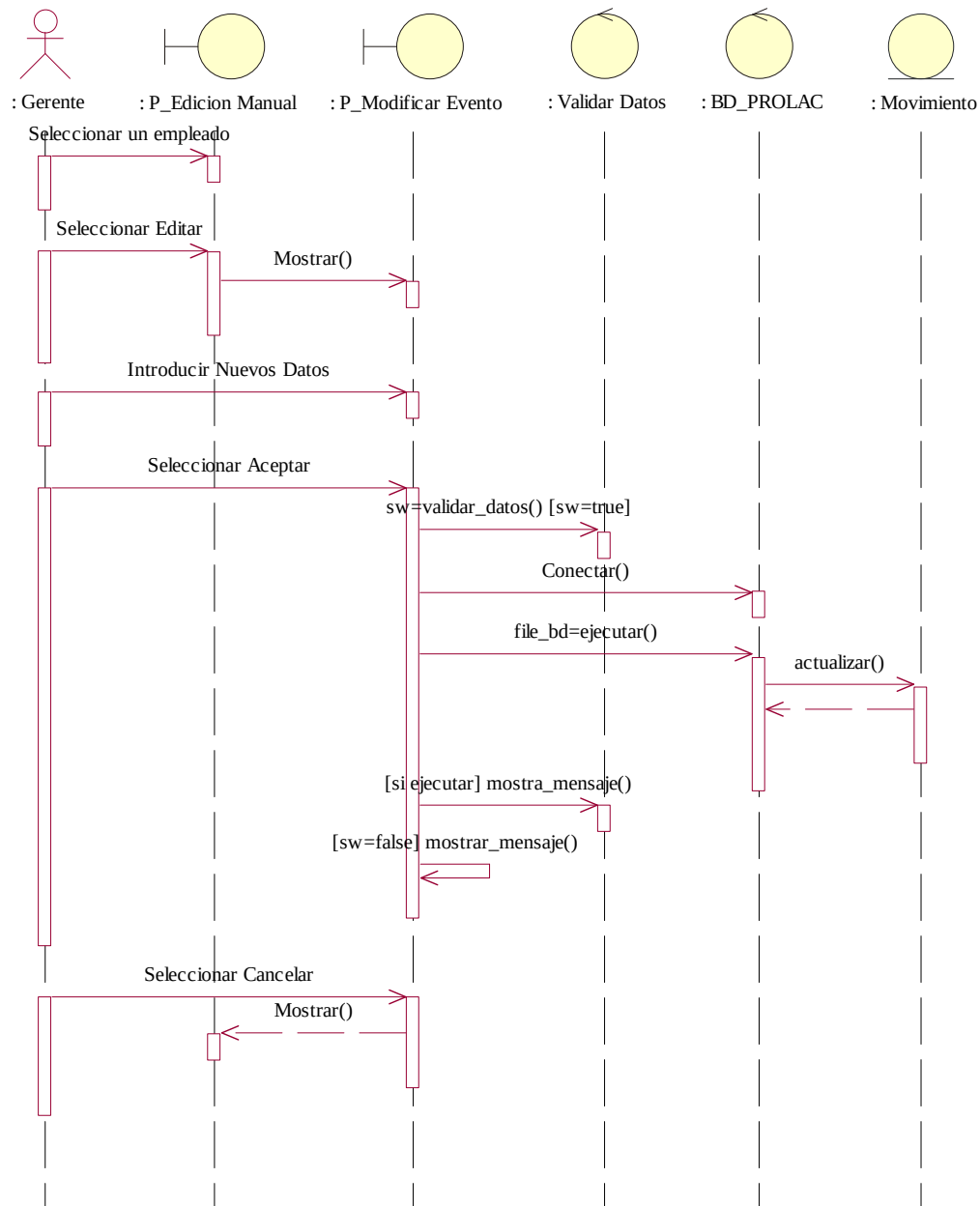


Figura N° 128: Diag. de Secuencia Editar Evento

Eliminar Evento

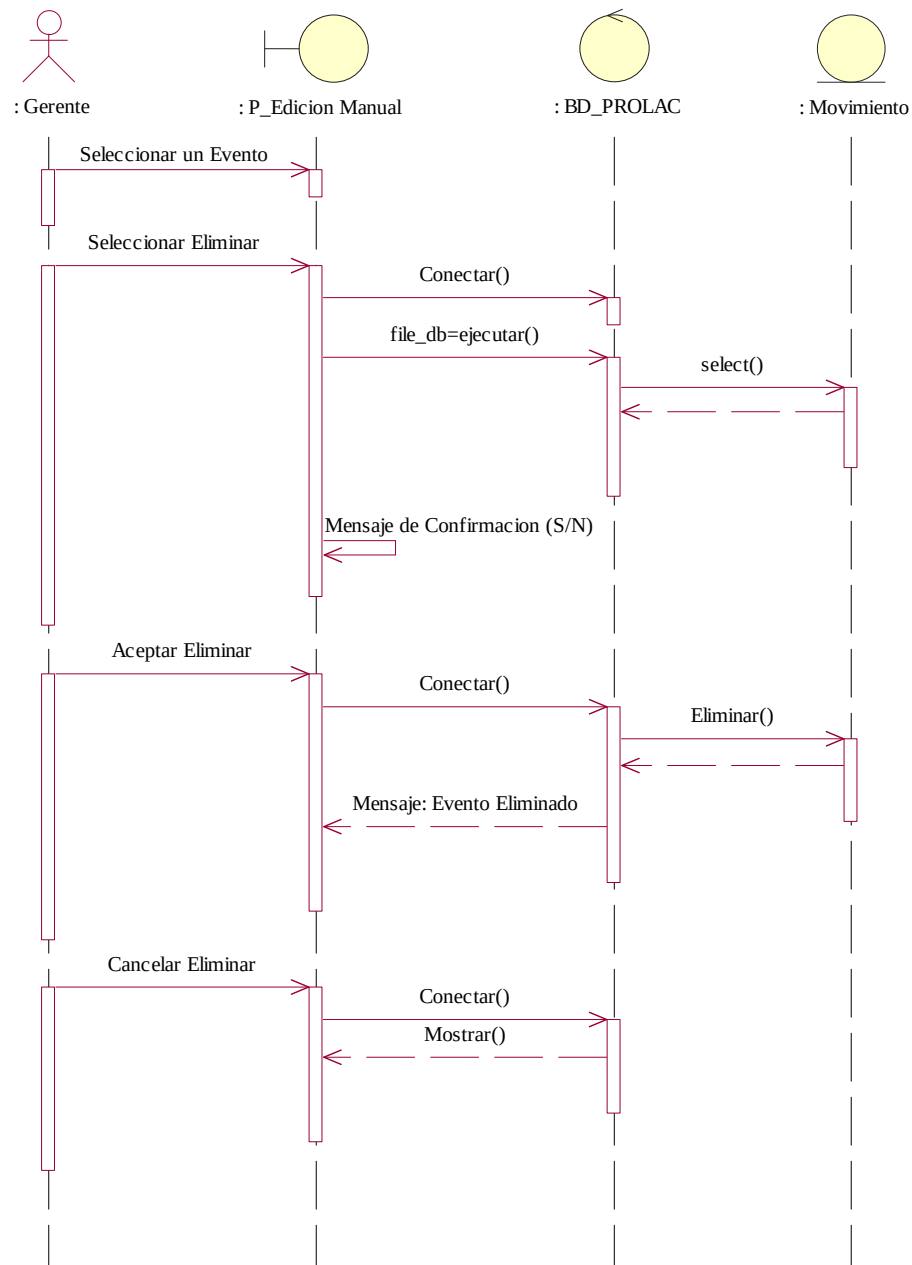


Figura N° 129: Diag. de Secuencia Eliminar Evento

II.1.4.2.4.6 MODULO IV: REPORTES

II.1.4.2.4.6.1 GENERAR REPORTES

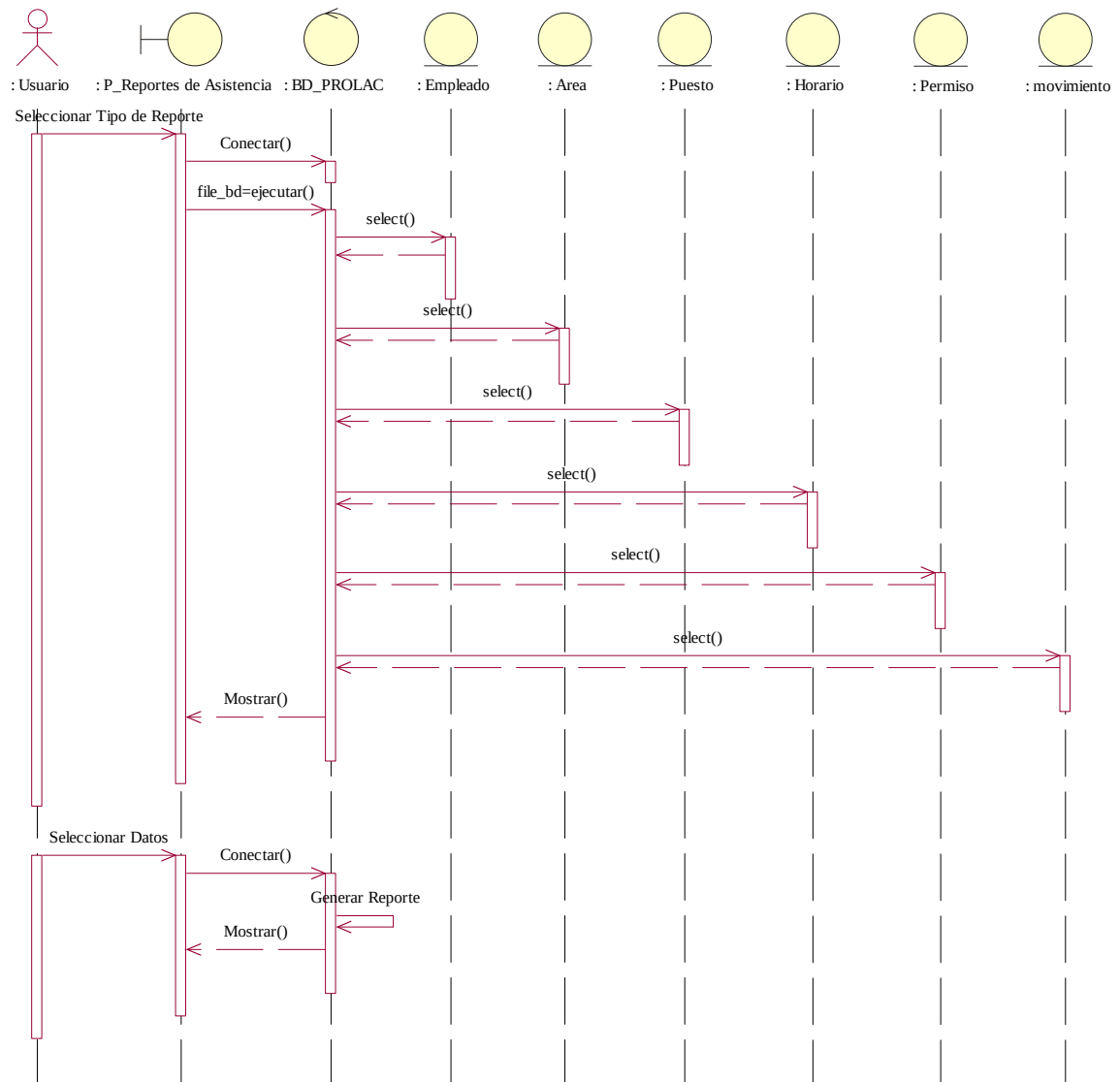


Figura N° 130: Diag. de Secuencia Generar Reportes

II.1.4.2.5 Diagramas de Colaboración

II.1.4.2.5.1 Registrar Asistencia

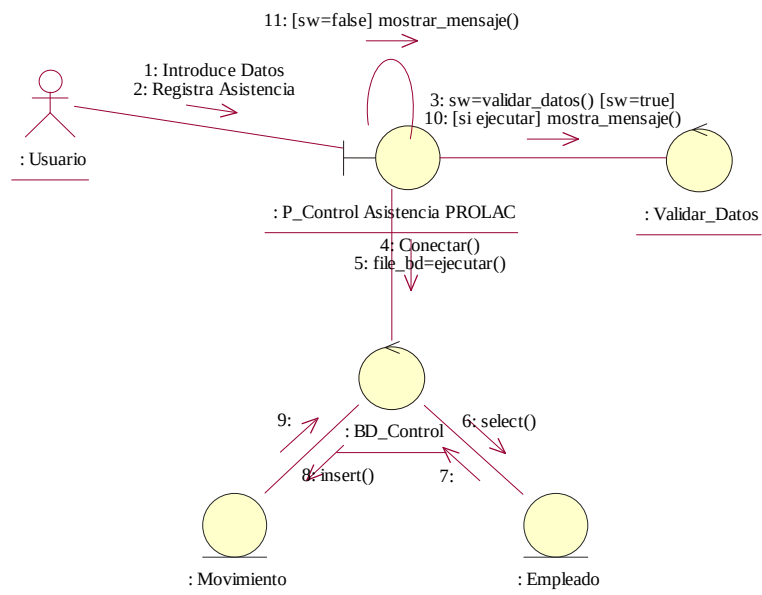


Figura N° 131: Diag. de Colaboración Registrar Asistencia

II.1.4.2.5.2 Ingresar Sistema

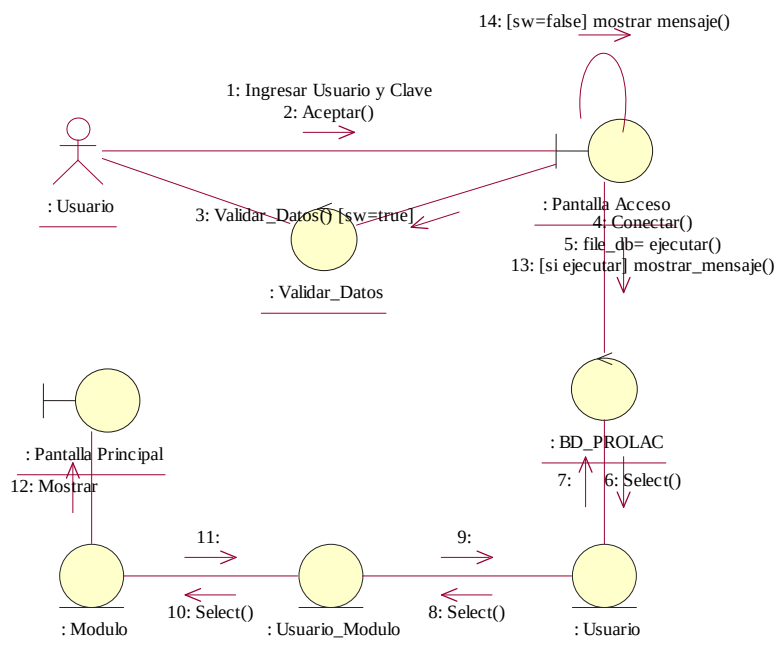


Figura N° 132: Diag. de Colaboración Ingresar al Sistema

II.1.4.2.5.3 MODULO I: SISTEMA

II.1.4.2.5.3.1 GESTIONAR USUARIOS

Listar Usuarios

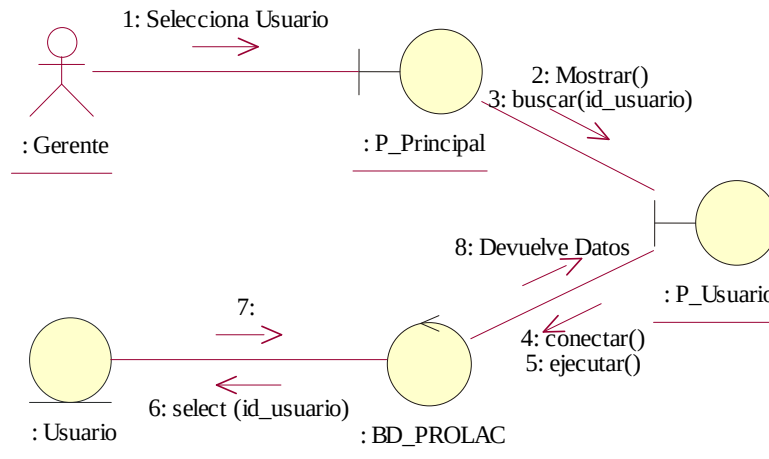


Figura N° 133: Diag. de Colaboración Listar Usuarios

Crear Usuario

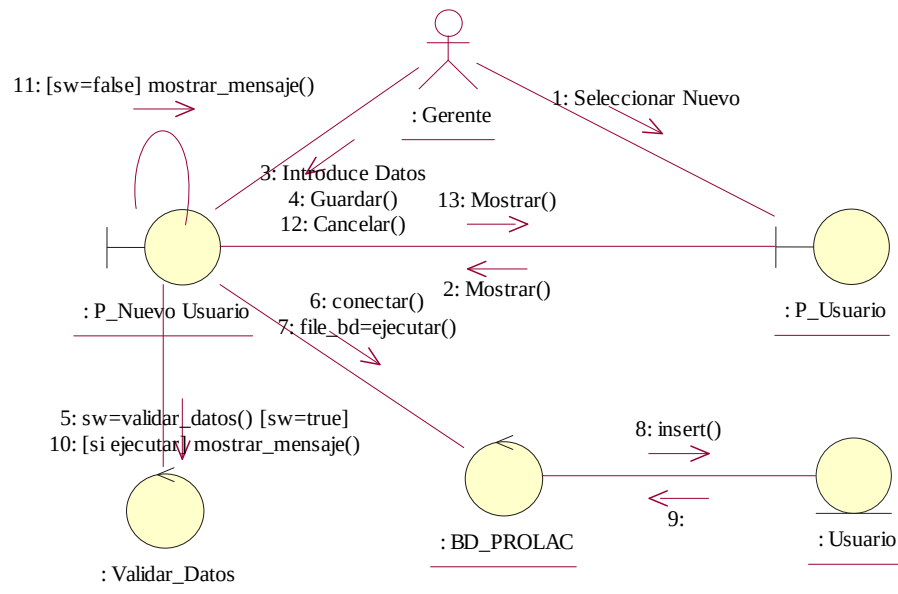


Figura N° 134: Diag. de Colaboración Crear Usuario

Editar Usuario

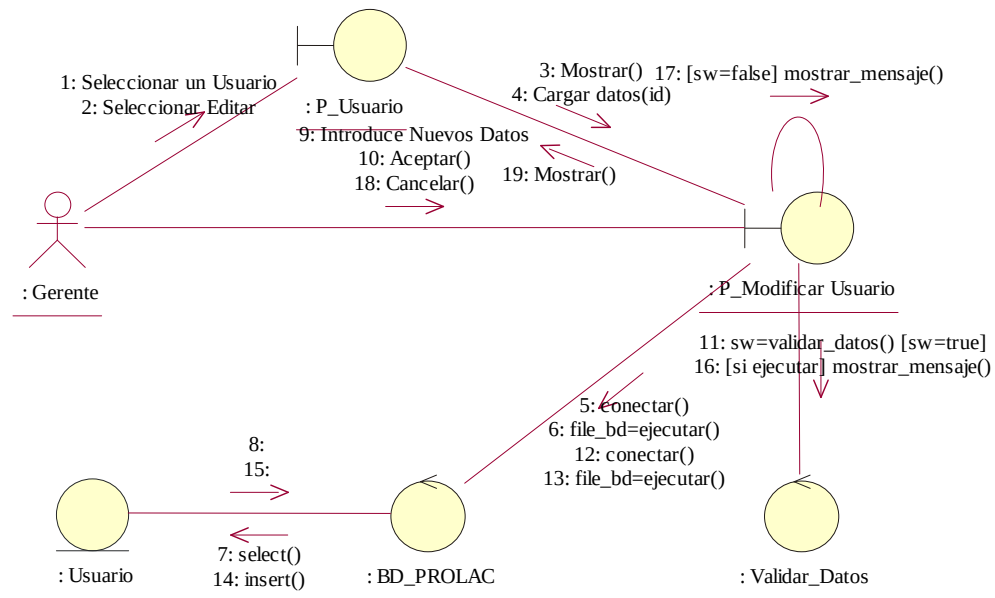


Figura N° 135: Diag. de Colaboración Editar Usuario

Eliminar Usuario

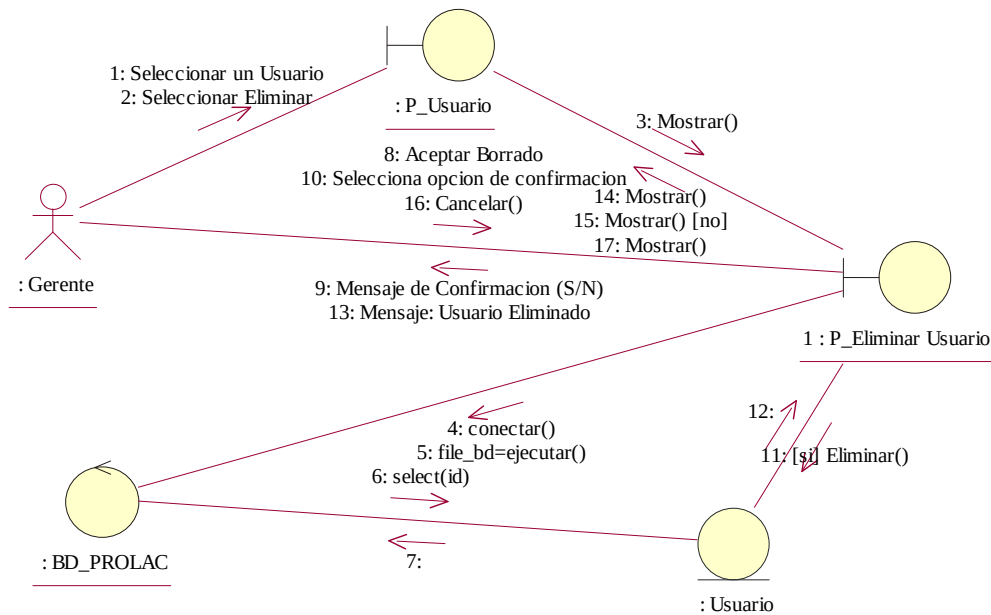


Figura N° 136: Diag. de Colaboración Eliminar Usuario

II.1.4.2.5.3.2 GESTIONAR USUARIOS MODULOS

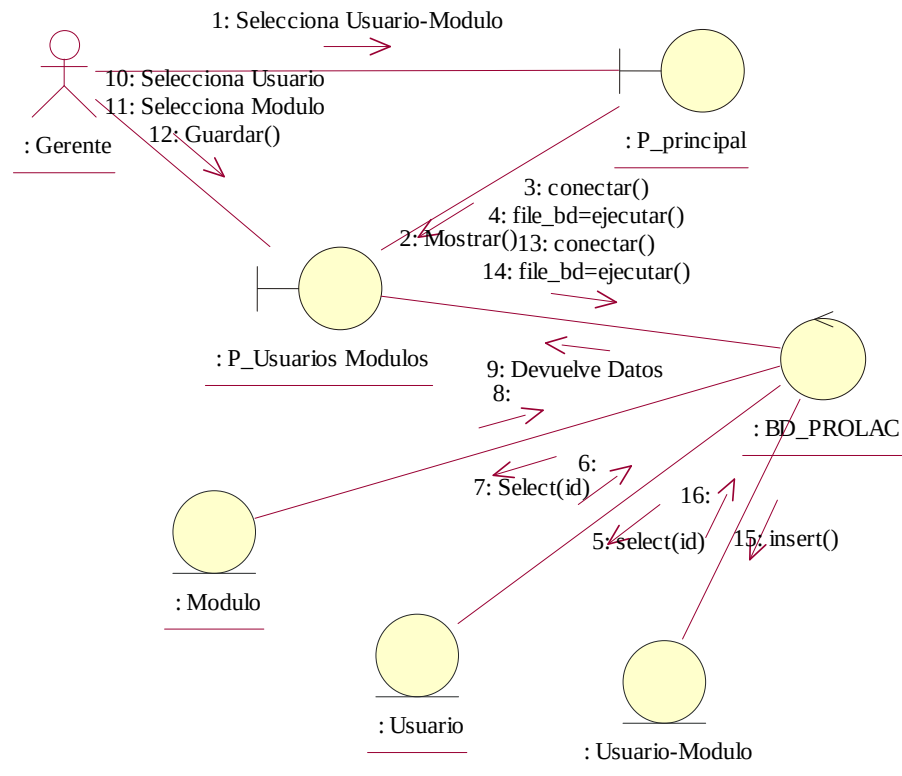


Figura N° 137: Diag. de Colaboración Gestionar Usuarios-Módulos

II.1.4.2.5.3.3 GESTIONAR PARAMETROS

Listar Parámetros

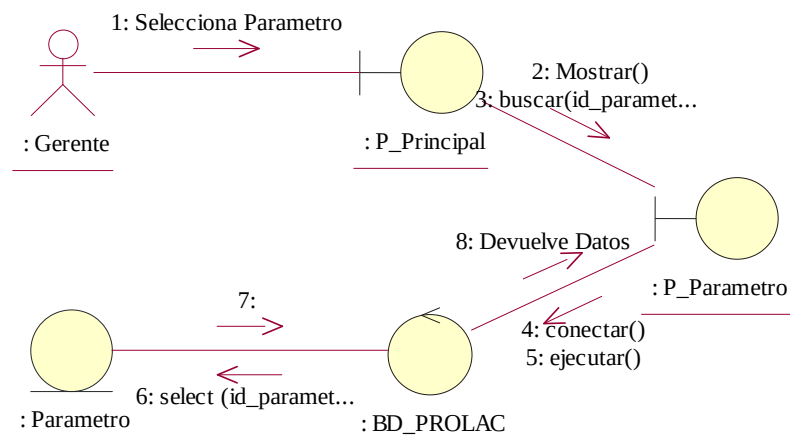


Figura N° 138: Diag. de Colaboración Listar Parámetros

Crear Parámetro

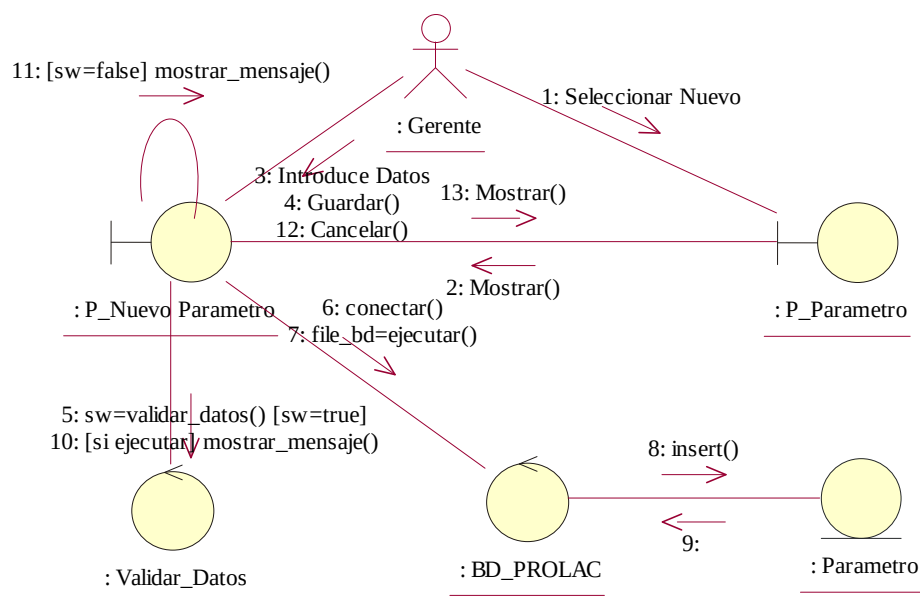


Figura N° 139: Diag. de Colaboración Crear Parámetro

Editar Parámetro

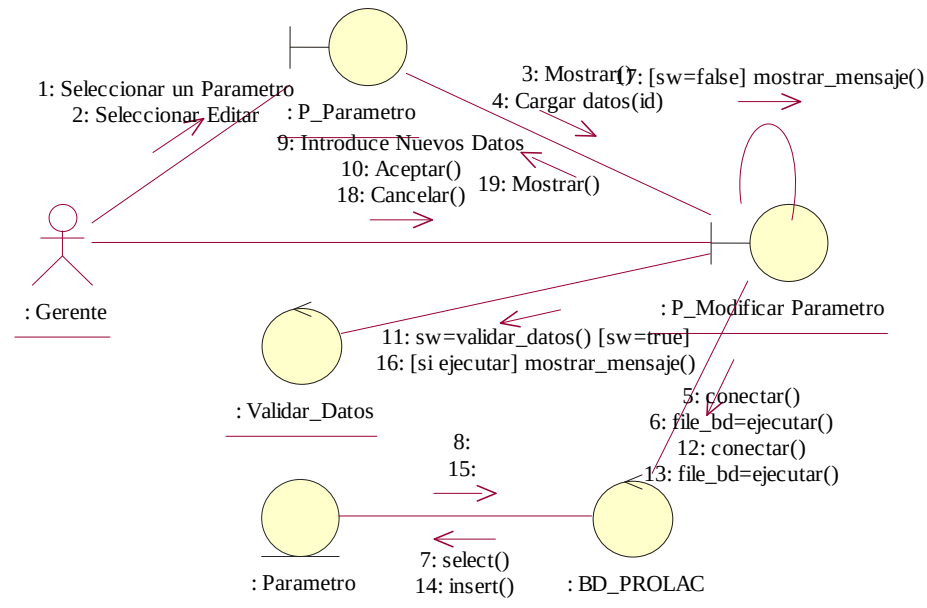


Figura N° 140: Diag. de Colaboración Editar Parámetro

Eliminar Parámetro

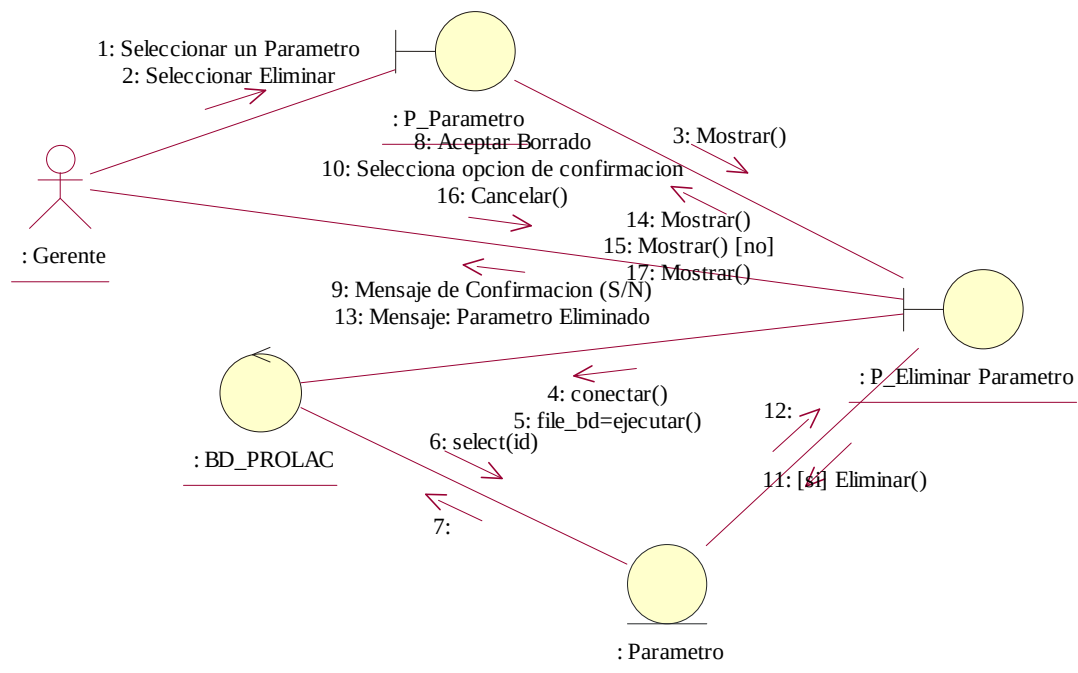


Figura N° 141: Diag. de Colaboración Eliminar Parámetro

II.1.4.2.5.4 MODULO II: PERSONAL

II.1.4.2.5.4.1 GESTIONAR EMPLEADOS

Listar Empleados

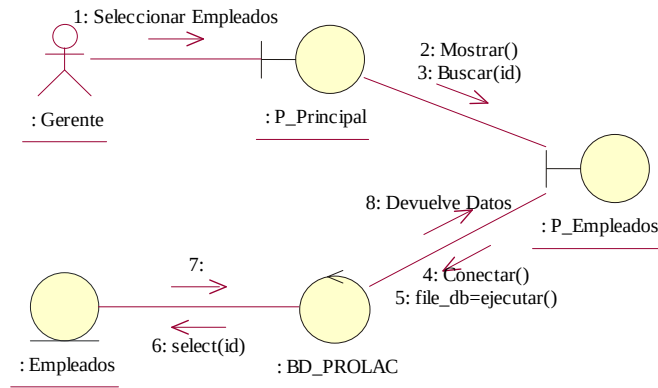


Figura N° 142: Diag. de Colaboración Listar Empleados

Crear Empleado

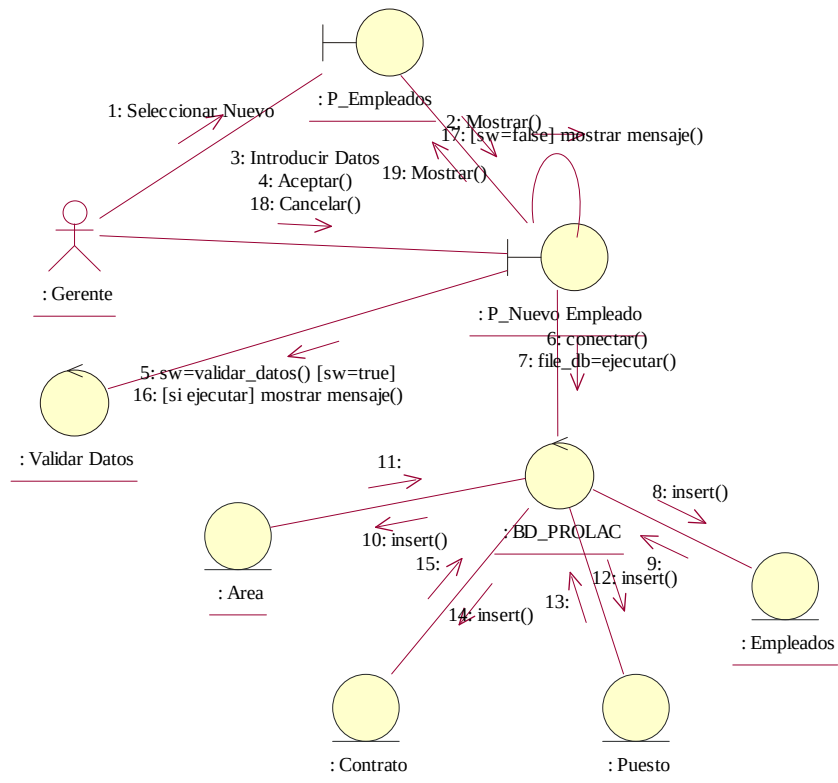


Figura N° 143: Diag. de Colaboración Crear Empleado

Editar Empleado

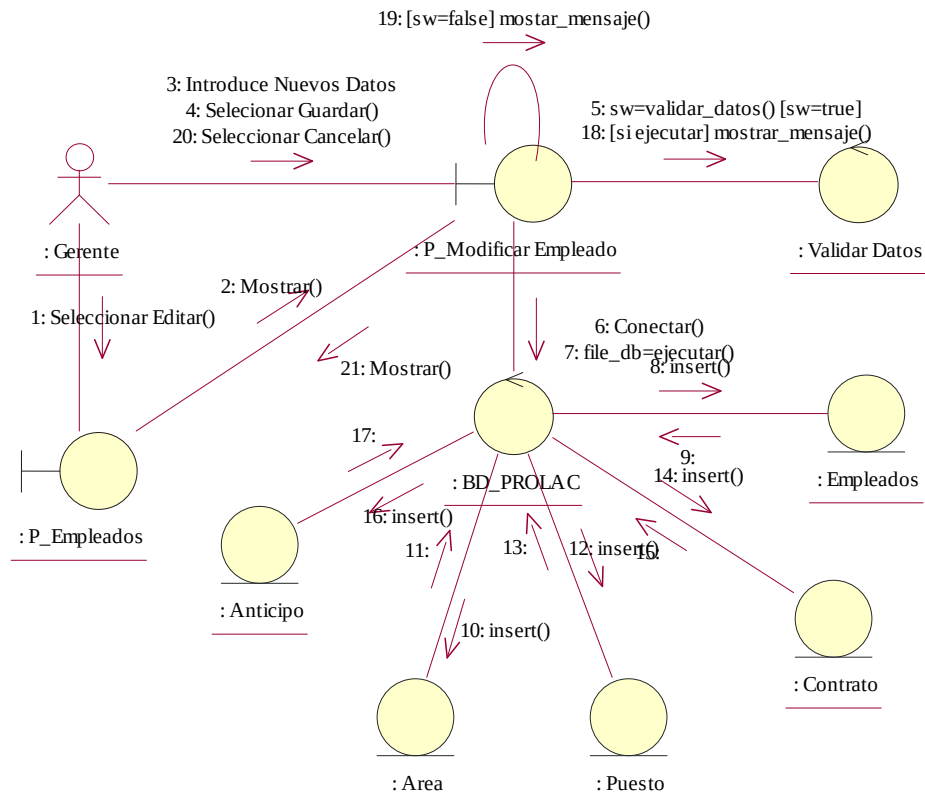


Figura N° 144: Diag. de Colaboración Editar Empleado

Eliminar Empleado

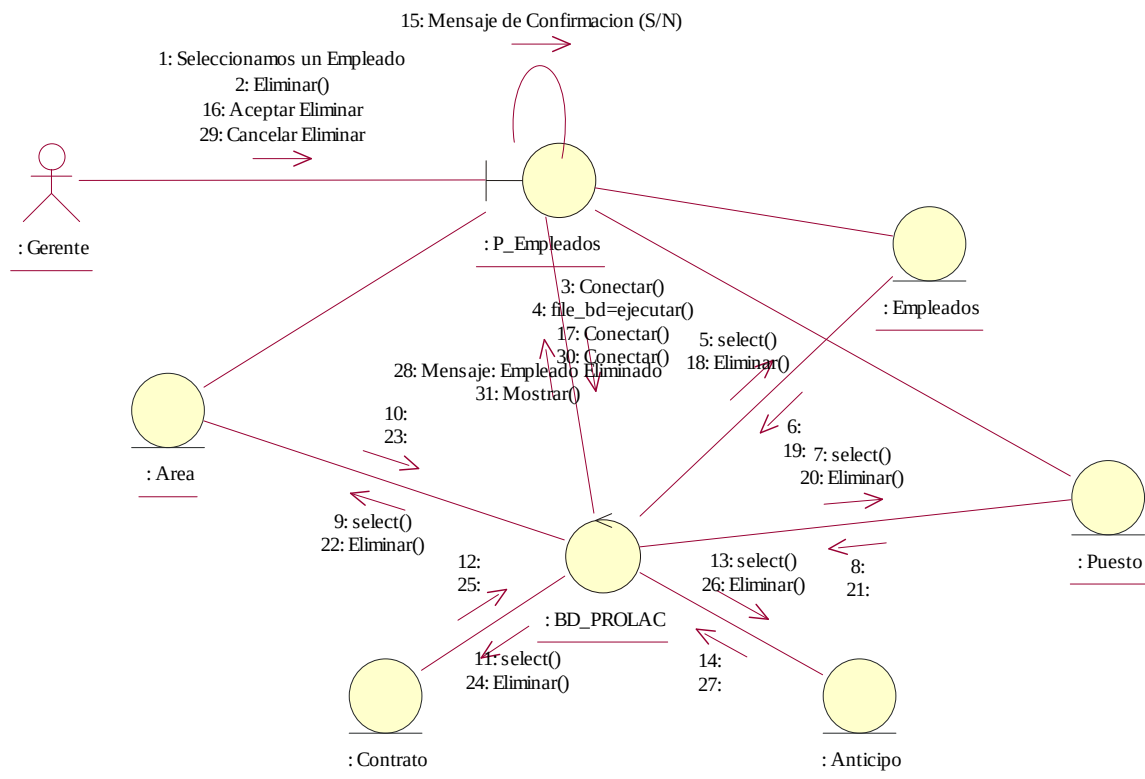


Figura N° 145: Diag. de Colaboración Eliminar Empleado

II.1.4.2.5.4.2 GESTIONAR AREAS

Listar Áreas

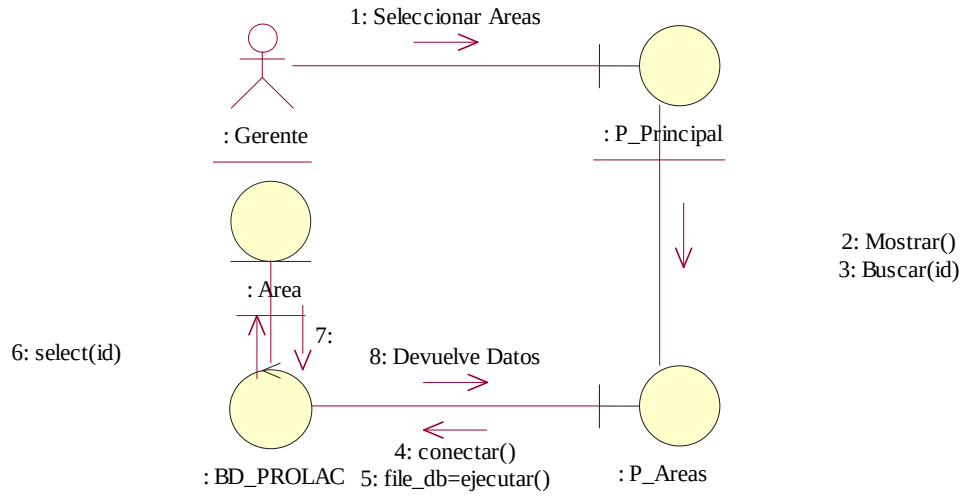


Figura N° 146: Diag. de Colaboración Listar Áreas

Crear Área

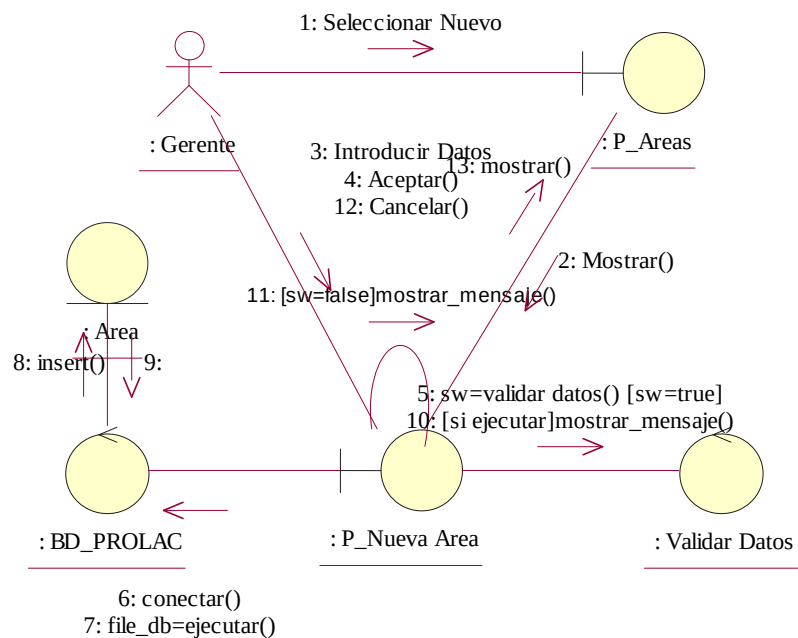


Figura N° 147: Diag. de Colaboración Crear Área

Editar Área

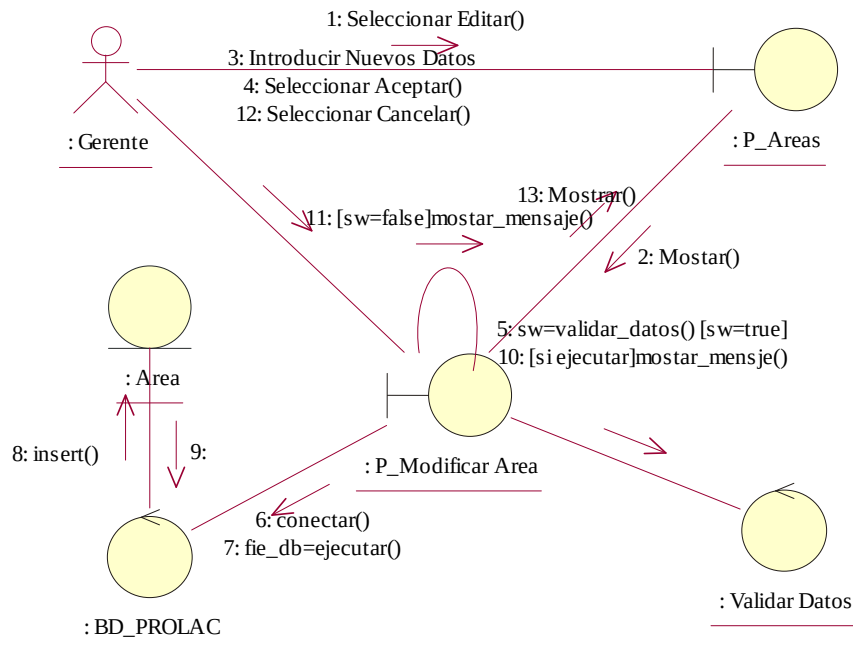


Figura N° 148: Diag. de Colaboración Editar Área

Eliminar Área

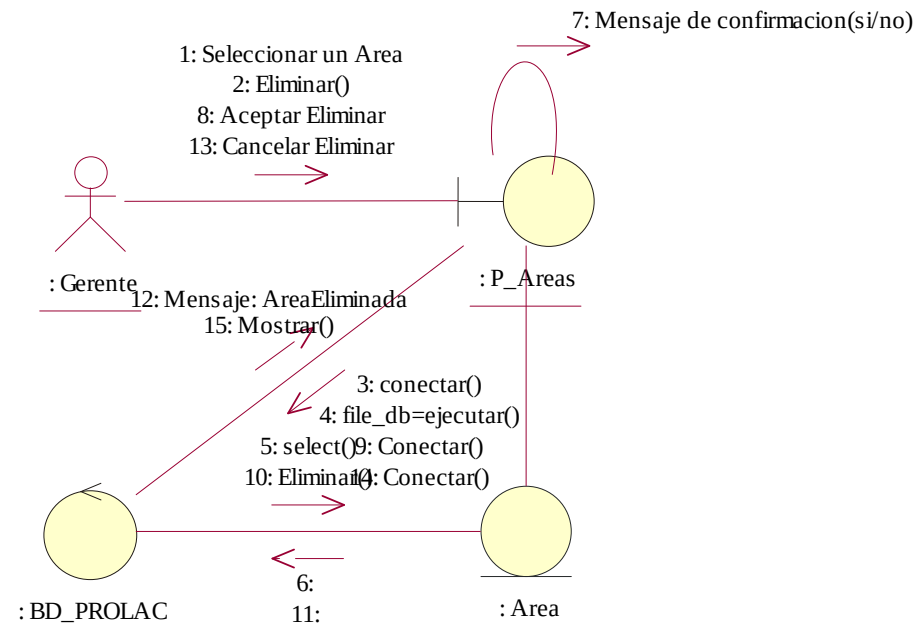


Figura N° 149: Diag. de Colaboración Eliminar Área

II.1.4.2.5.4.3 GESTIONAR PUESTOS

Listar Puestos

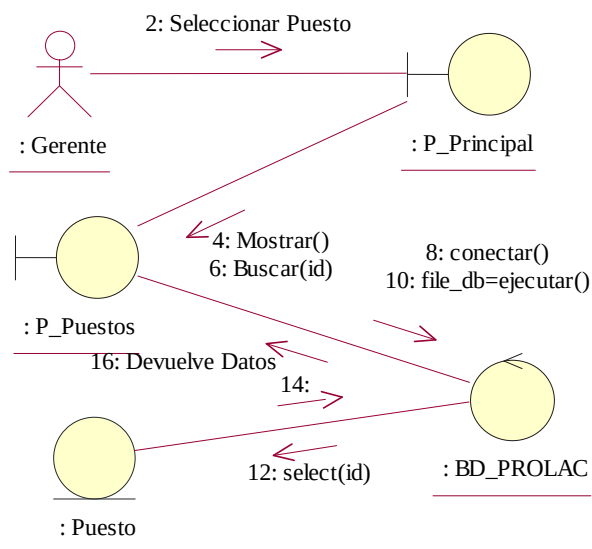


Figura N° 150: Diag. de Colaboración Listar Puestos

Crear Puesto

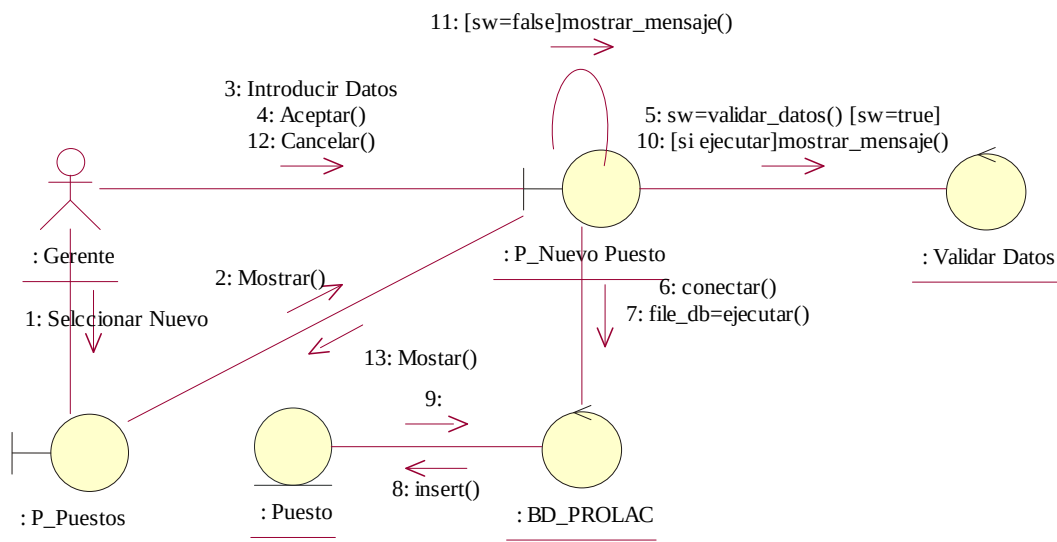


Figura N° 151: Diag. de Colaboración Crear Puesto

Editar Puesto

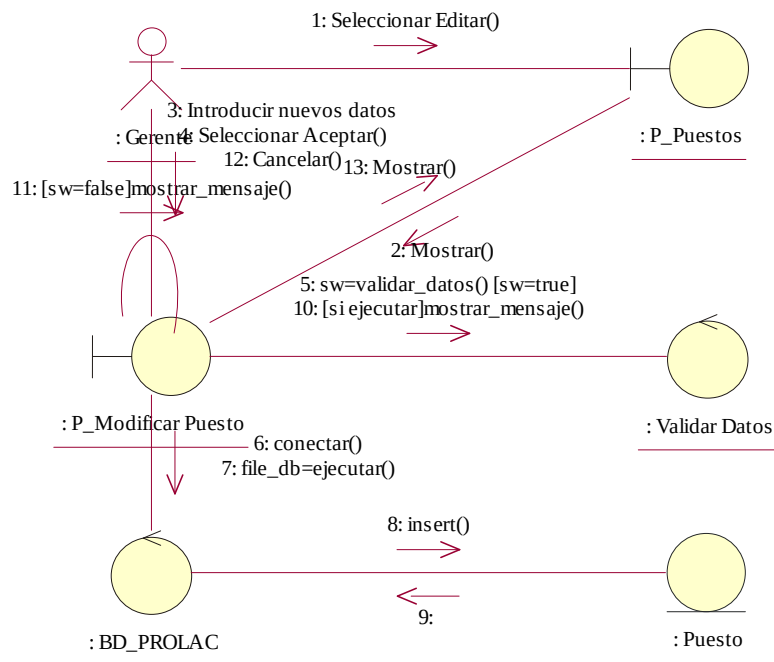


Figura N° 152: Diag. de Colaboración Editar Puesto

Eliminar Puesto

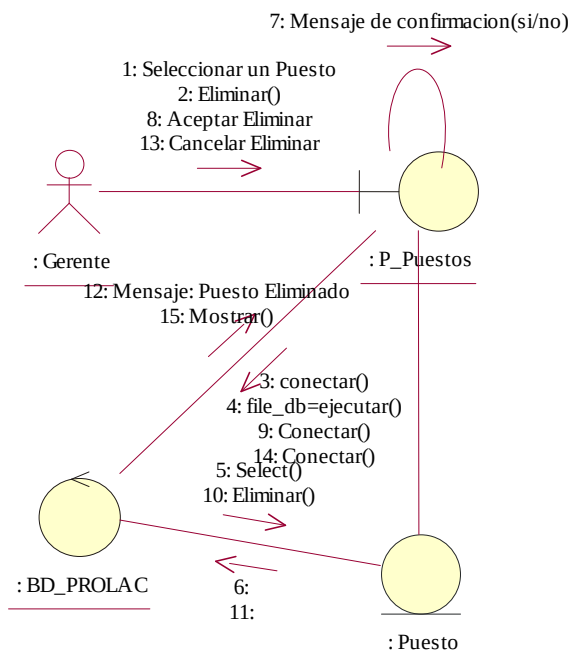


Figura N° 153: Diag. de Colaboración Eliminar Puesto

II.1.4.2.5.4.4 GESTIONAR PERMISOS-VACACIONES

Listar Empleados

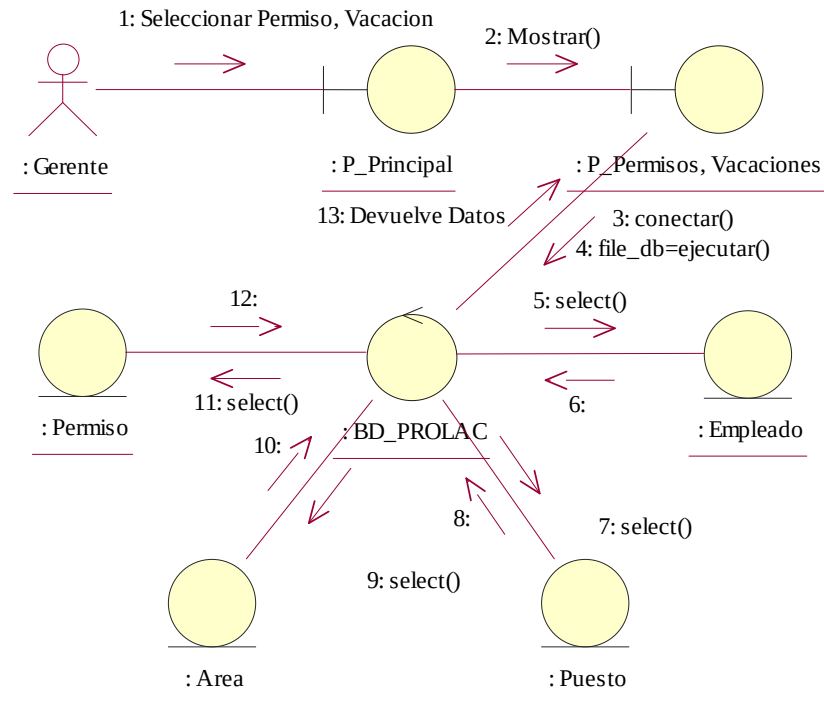


Figura N° 154: Diag. de Colaboración Listar Empleado

Crear Permiso-Vacación

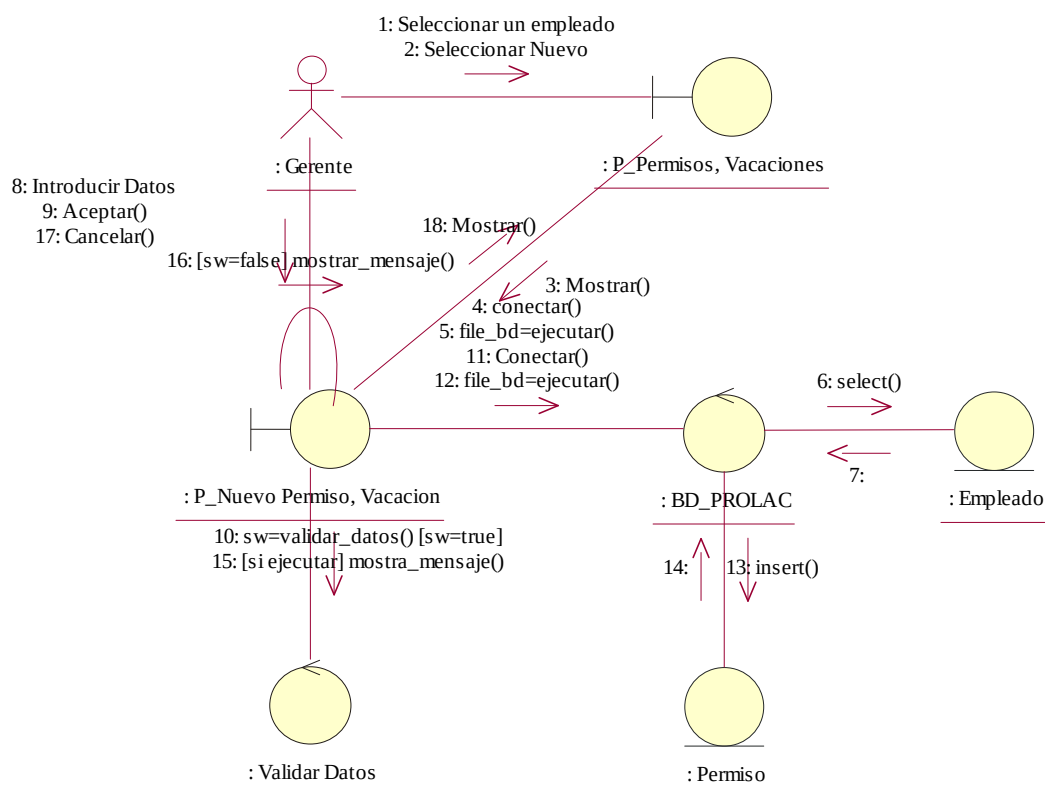


Figura N° 155: Diag. de Colaboración Crear Permiso-Vacación

Editar Permiso-Vacación

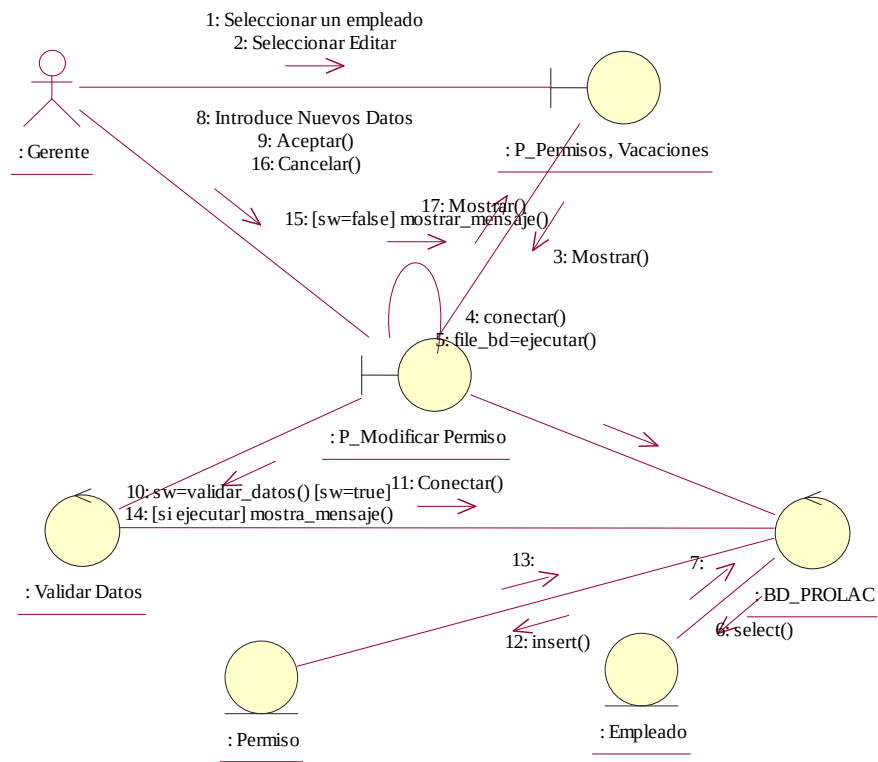


Figura N° 156: Diag. de Colaboración Editar Permiso-Vacación

Eliminar Permiso-Vacación

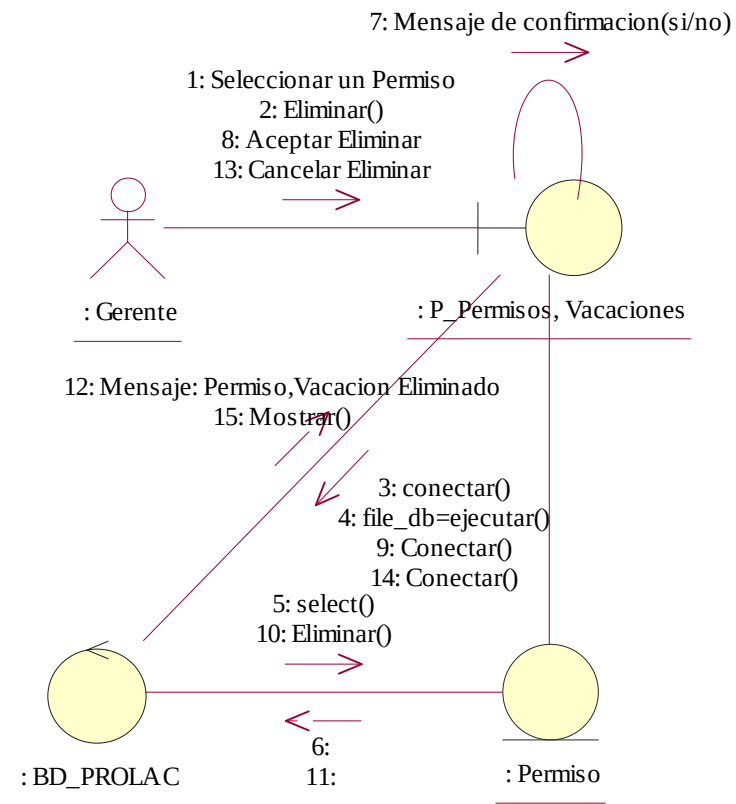


Figura N° 157: Diag. de Colaboración Eliminar Permiso-Vacación

II.1.4.2.5.4.5 GENERAR PLANILLAS DE SUELDOS

Crear Planilla de Sueldo

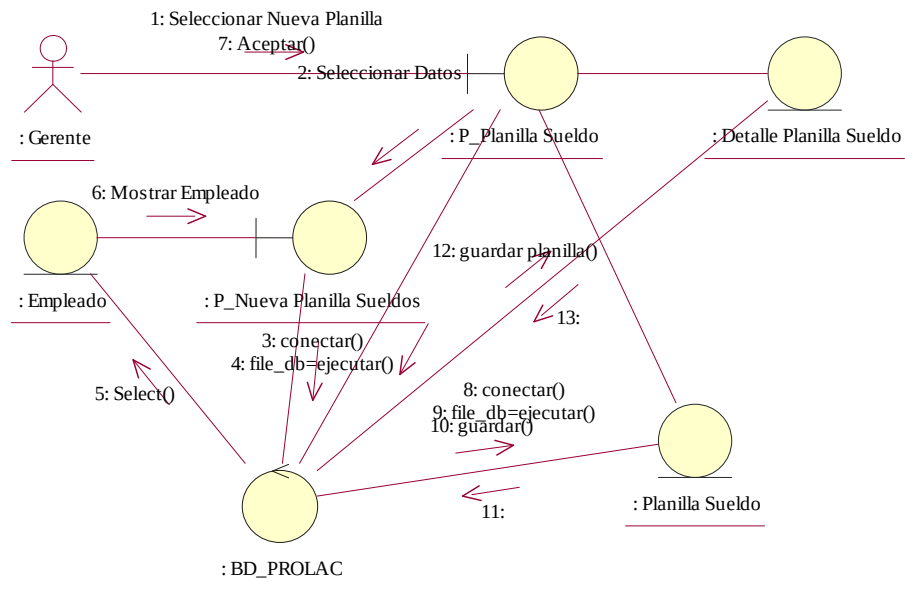


Figura N° 158: Diag. de Colaboración Crear Planilla de Sueldos

Editar Planilla de Sueldos

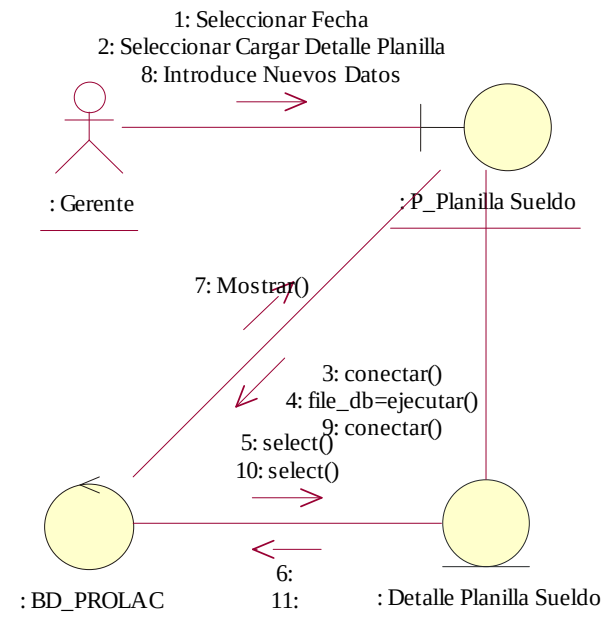


Figura N° 159: Diag. de Colaboración Editar Planilla de Sueldos

Cargar Detalle Planilla de Sueldos

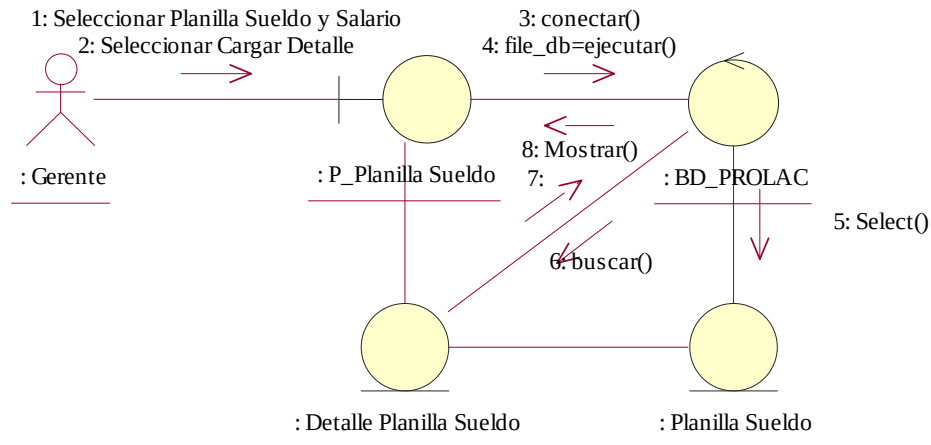


Figura N° 160: Diag. de Colaboración Cargar Detalle Planilla de Sueldos

II.1.4.2.5.4.6 GENERAR PLANILLAS DE AGUINALDOS

Crear Planilla de Aguinaldo

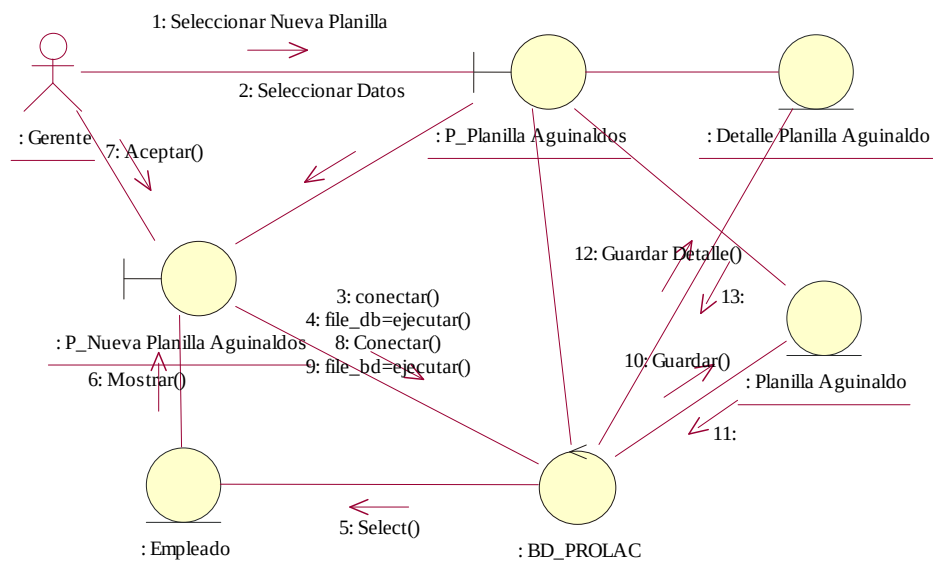


Figura N° 161: Diag. de Colaboración Crear Planilla de Aguinaldo

Cargar Detalle Planilla de Aguinaldos

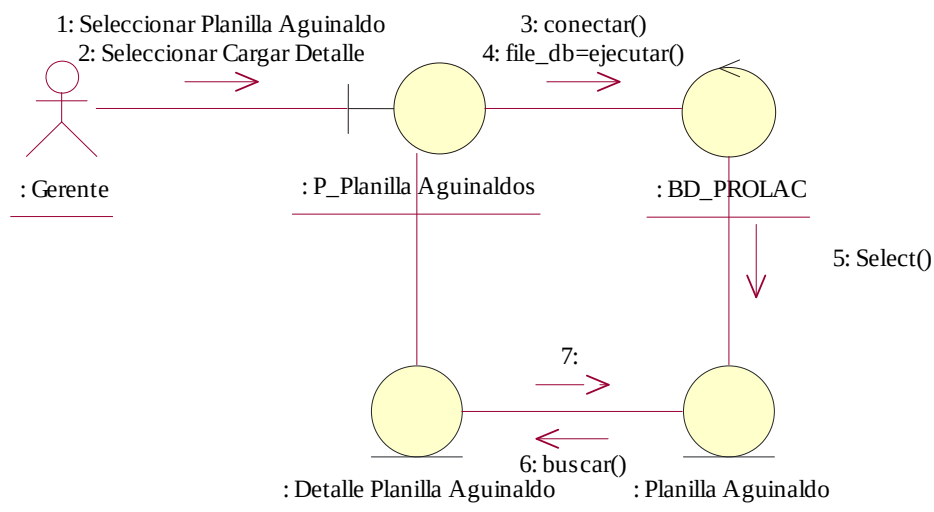


Figura N° 162: Diag. de Colaboración Cargar Detalle Planilla de Aguinaldos

II.1.4.2.5.5 MODULO III: CONTROL PERSONAL

II.1.4.2.5.5.1 GESTIONAR DIA INHABIL

Listar Día Inhábil

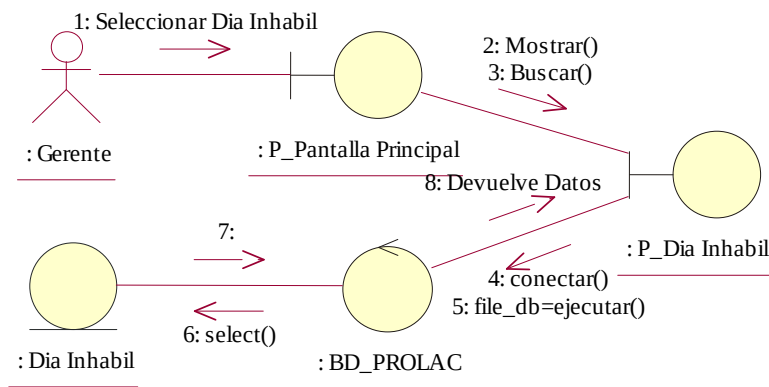


Figura N° 163: Diag. de Colaboración Listar Día Inhábil

Crear Día Inhábil

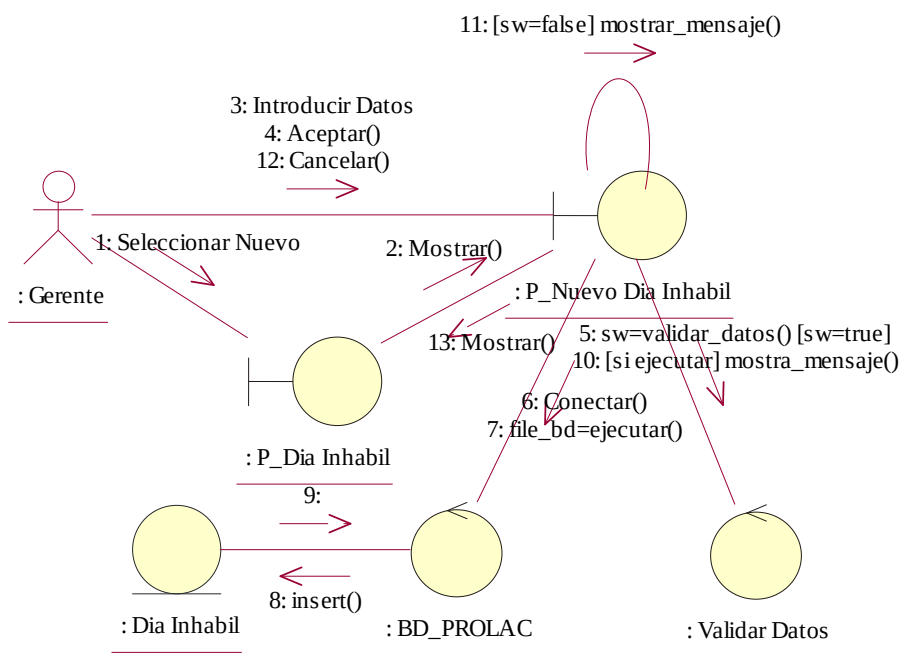


Figura N° 164: Diag. de Colaboración Crear Día Inhábil

Editar Día Inhábil

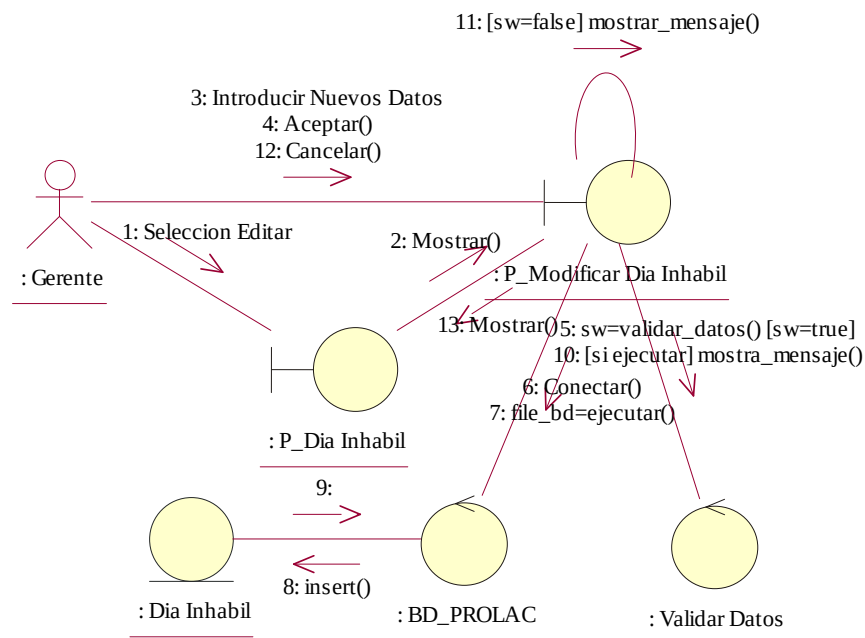


Figura N° 165: Diag. de Colaboración Editar Día Inhábil

Eliminar Día Inhábil

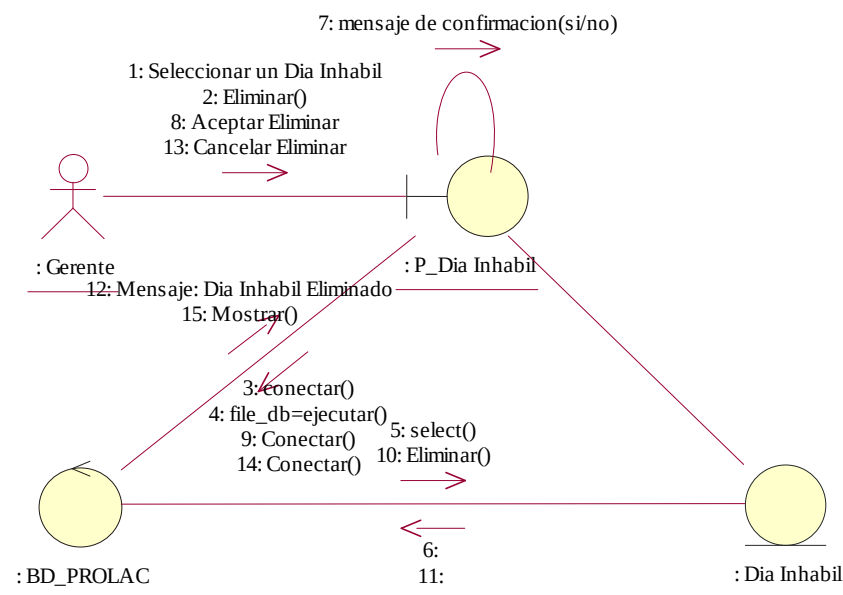


Figura N° 166: Diag. de Colaboración Eliminar Día Inhábil

II.1.4.2.5.2 GESTIONAR HORARIO

Listar Horarios

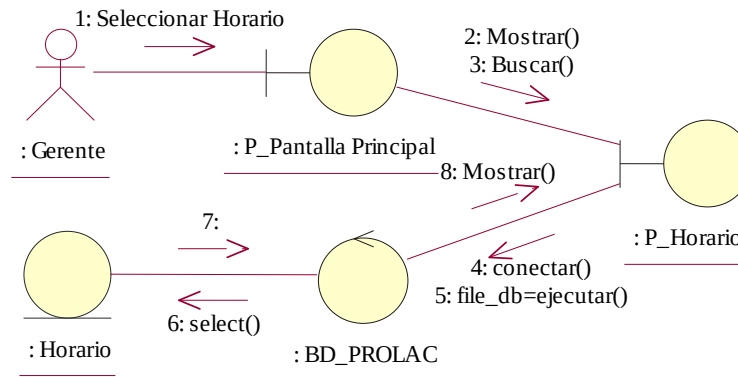


Figura N° 167: Diag. de Colaboración Listar Horarios

Crear Horario

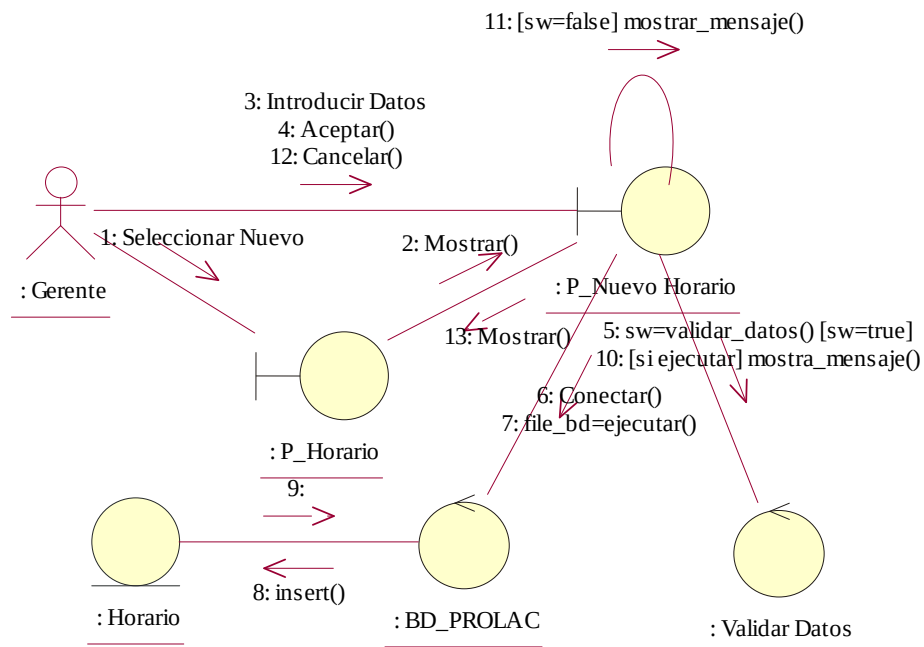


Figura N° 168: Diag. de Colaboración Crear Horario

Editar Horario

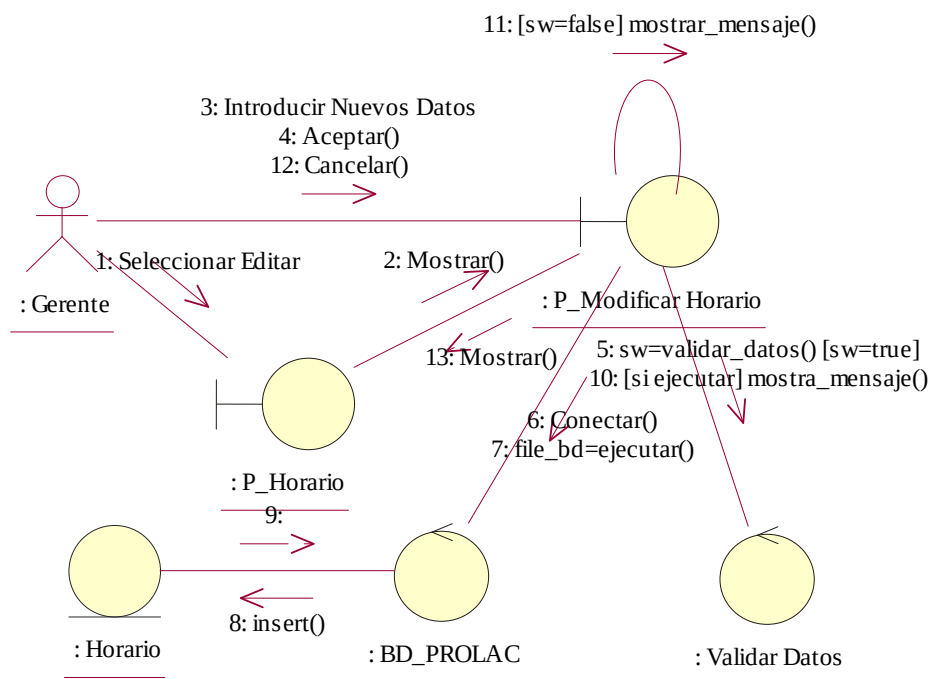


Figura N° 169: Diag. de Colaboración Editar Horario

Eliminar Horario

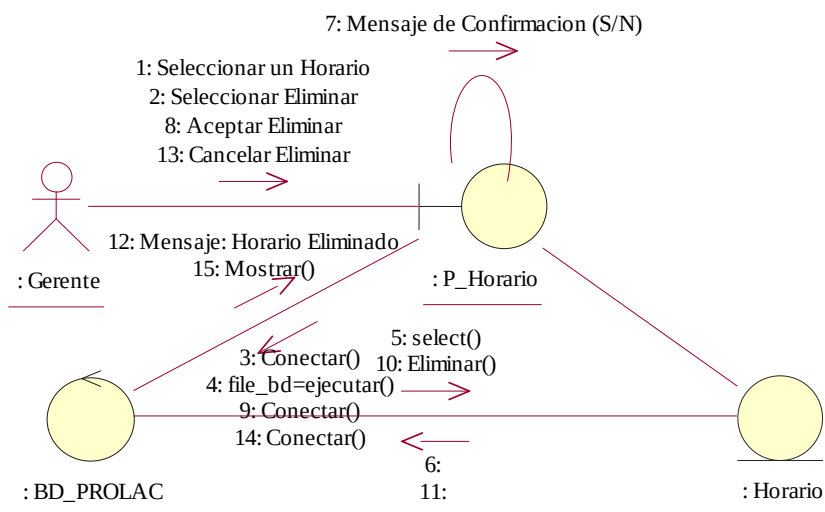


Figura N° 170: Diag. de Colaboración Eliminar Horario

II.1.4.2.5.3 GESTIONAR EDICION MANUAL

Listar Empleados

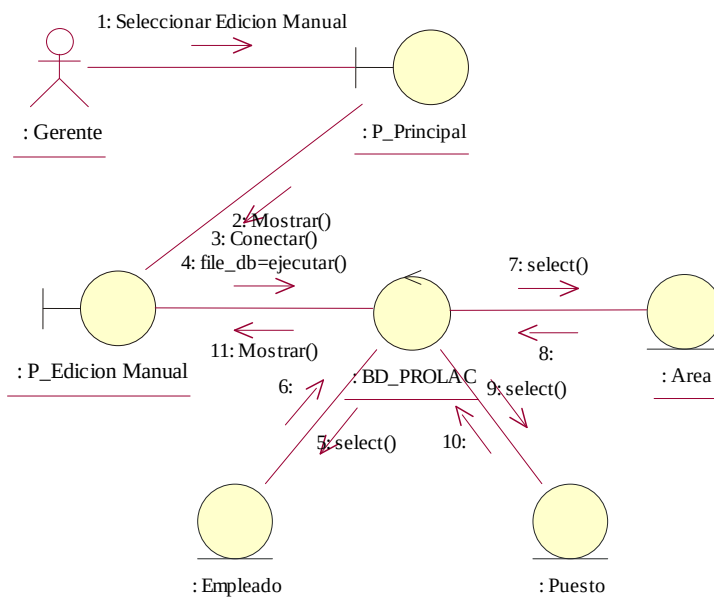


Figura N° 171: Diag. de Colaboración Listar Empleados

Crear Evento

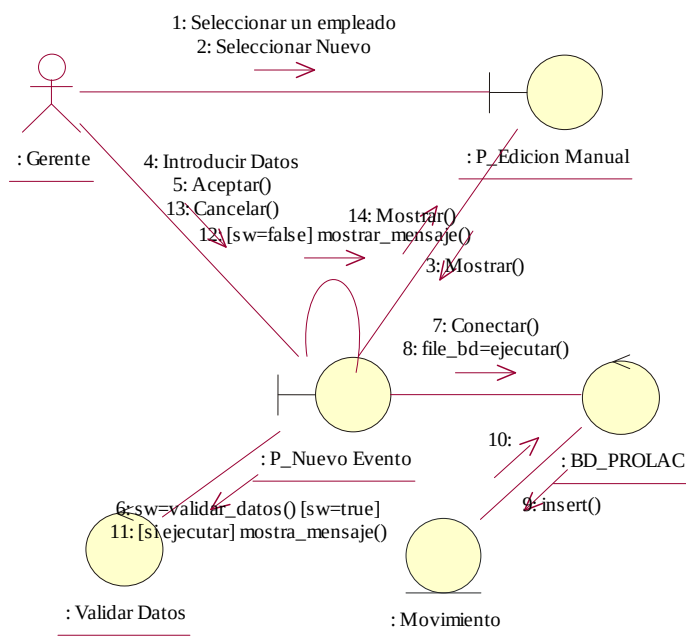


Figura N° 172: Diag. de Colaboración Crear Evento

Editar Evento

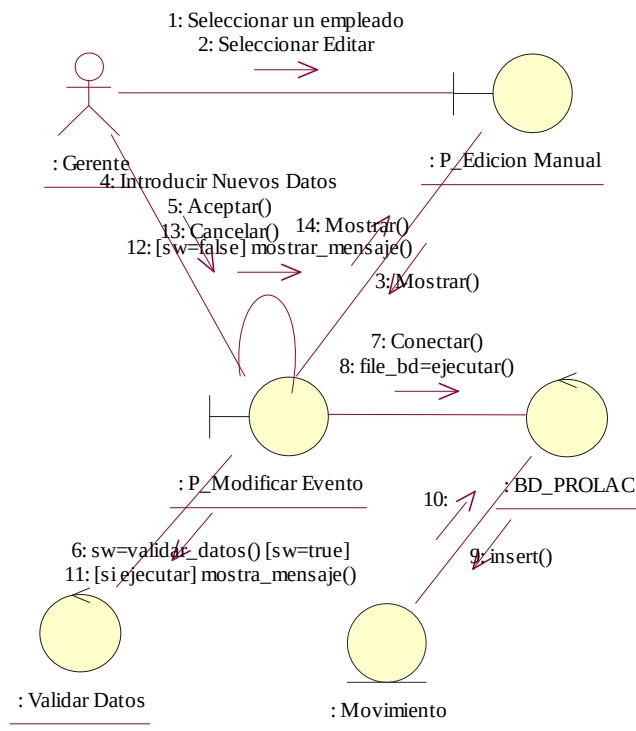


Figura N° 173: Diag. de Colaboración Editar Evento

Eliminar Evento

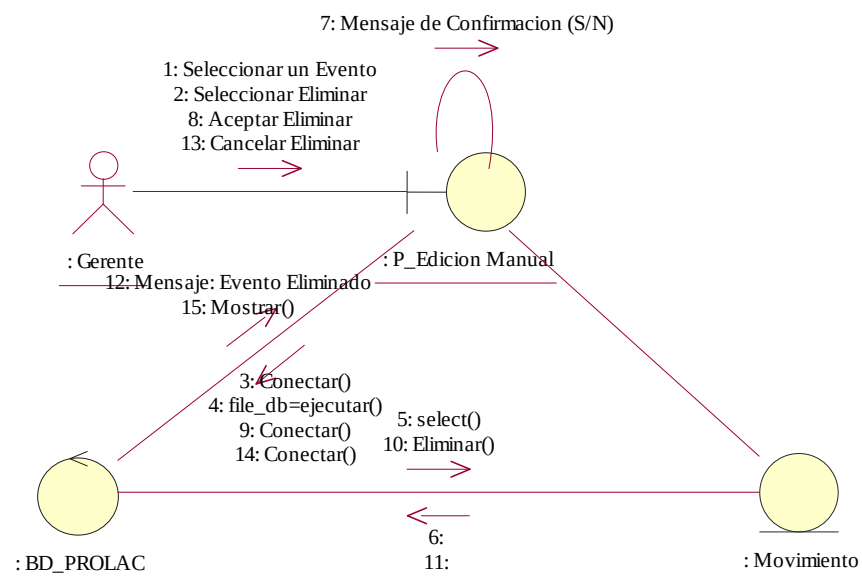


Figura N° 174: Diag. de Colaboración Eliminar Evento

II.1.4.2.5.6 MODULO IV: REPORTES

II.1.4.2.5.6.1 GENERAR REPORTE

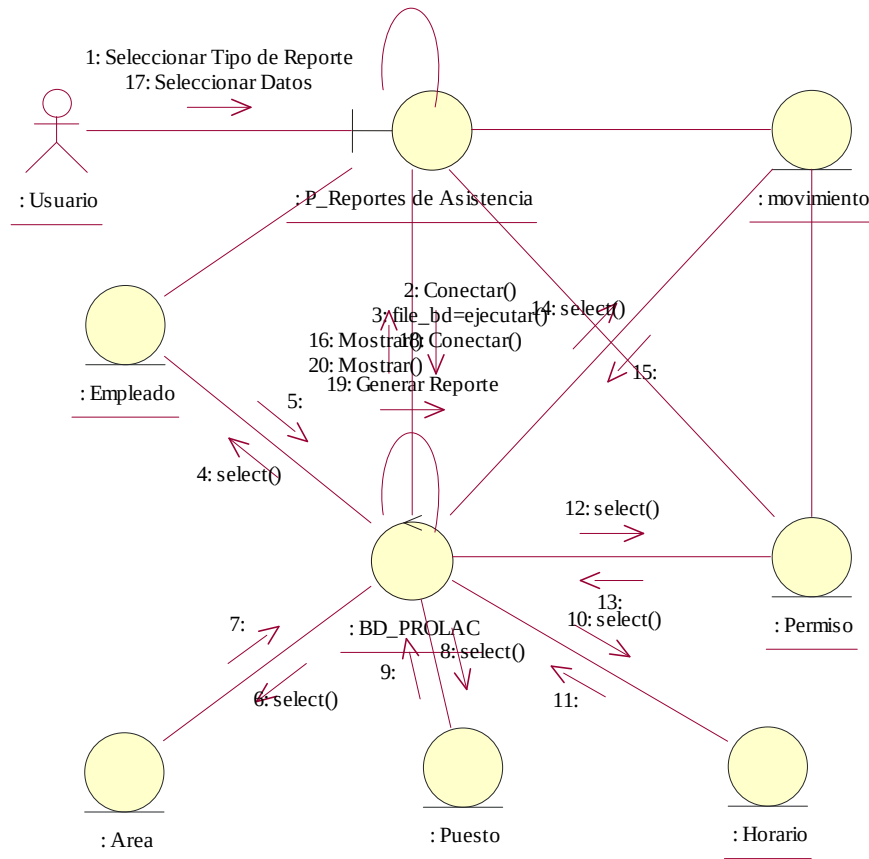


Figura N° 175: Diag. de Colaboración Generar Reporte

II.1.4.3 Prototipos de Interfaces de Usuarios

II.1.4.3.1 Introducción

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema.

II.1.4.3.2 Propósito

- ✦ Presentar los prototipos de pantallas para que el usuario tenga una idea de la interfaz que presentara el sistema.

II.1.4.3.3 Alcance

- ✦ Mostrar los prototipos de pantallas, solamente el diseño que adoptaran todas.

II.1.4.3.4 Prototipos de Pantallas

II.1.4.3.4.1 PANTALLA CONFIGURAR CONEXION

En esta pantalla se visualiza un formulario donde se ingresa datos, para configurar la conexión en red, cuenta con campos como: Host, Puerto, Base de Datos, Usuario y Clave.

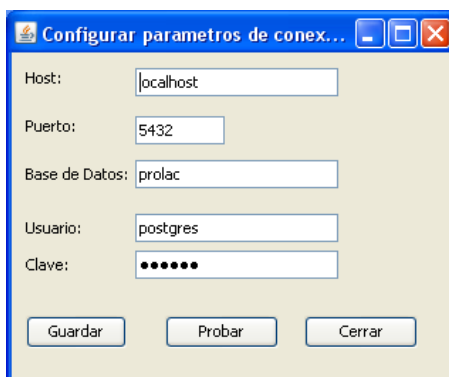


Figura N° 176: Pantalla Configurar Conexión

II.1.4.3.4.2 PANTALLA RESPALDO DE BASE DE DATOS

En esta pantalla se visualiza un formulario donde se selecciona datos, para generar una copia de seguridad de la base de datos.

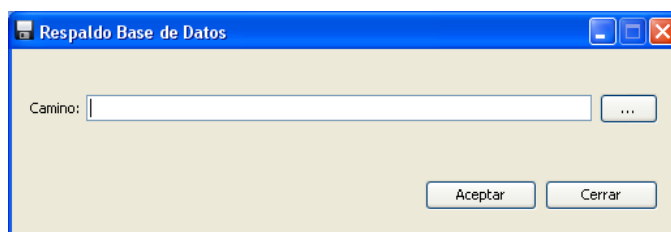


Figura N° 177: Pantalla Respaldo de Base de Datos

II.1.4.3.4.3 PANTALLA RESTAURACION DE BASE DE DATOS

En esta pantalla se visualiza un formulario donde se selecciona datos, para restaurar una copia de seguridad de la base de datos.

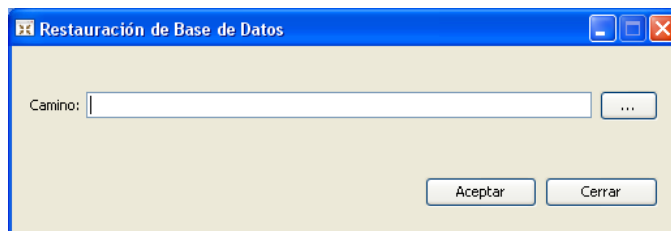


Figura N° 178: Pantalla Restauración de Base de Datos

II.1.4.3.4.4 PANTALLA REGISTRAR ASISTENCIA

En esta pantalla se visualiza un formulario para ingresar datos, para registrar la asistencia del personal, cuenta con campos como: Numero de Empleado y Clave.

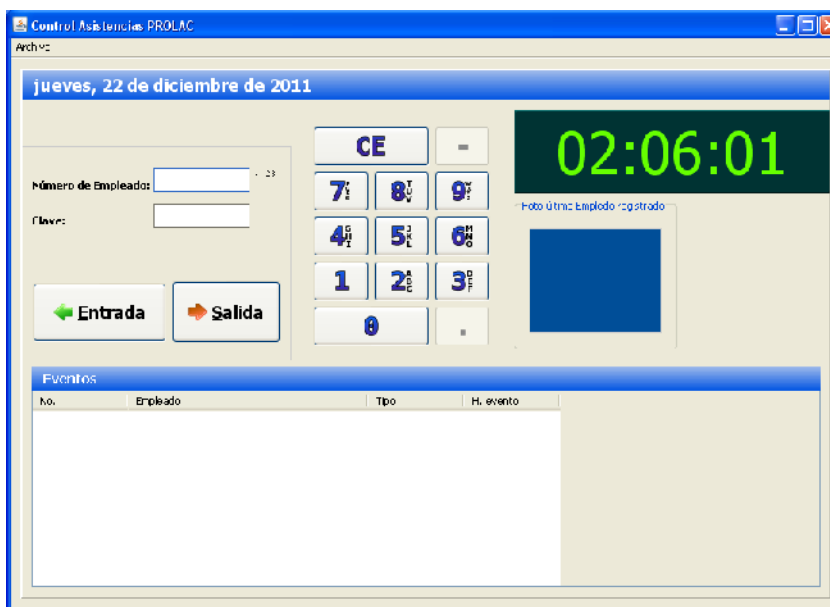


Figura N° 179: Pantalla Registrar Asistencia

PANTALLA ERROR DE REGISTRO DE ASISTENCIA

Esta pantalla contiene un mensaje que se le muestra al usuario cuando este introduce los campos en la pantalla registrar asistencia y estos datos no se encuentran en la base de datos del sistema, haciendo clic en la opción Aceptar visualiza la pantalla Registrar Asistencia.

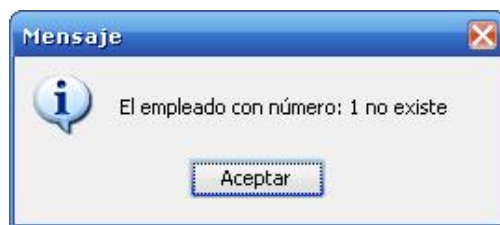


Figura N° 180: Pantalla Mensaje Empleado No Existe

II.1.4.3.4.5 PANTALLA INGRESAR SISTEMA

En esta pantalla es el inicio del sistema, muestra los campos de usuario y clave, introduciendo estos campos el sistema dará paso a la pantalla principal dependiendo el usuario.

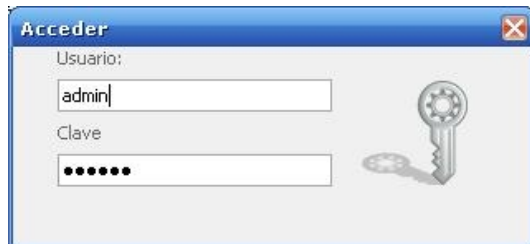


Figura N° 181: Pantalla Ingresar Sistema

PANTALLA MENSAJE ERROR DE ACCESO

Esta pantalla contiene un mensaje que se le muestra al usuario cuando este introduce los campos en la pantalla acceso y estos datos no se encuentran en la base de datos del sistema, haciendo clic en la opción Aceptar visualiza la pantalla Acceso.

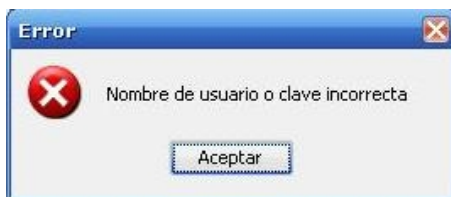


Figura N° 182: Pantalla Mensaje Error de Acceso

II.1.4.3.4.6 MODULO I: SISTEMA

En este modulo se encuentran todos los procesos que puede realizar un usuario que tenga acceso al modulo.

PANTALLA PRINCIPAL MODULO SISTEMA

Esta pantalla muestra una vez verificados los datos en la base de datos del usuario que quiera ingresar al sistema y que quiera acceder al modulo sistema, la pantalla visualiza el menú correspondiente al modulo.

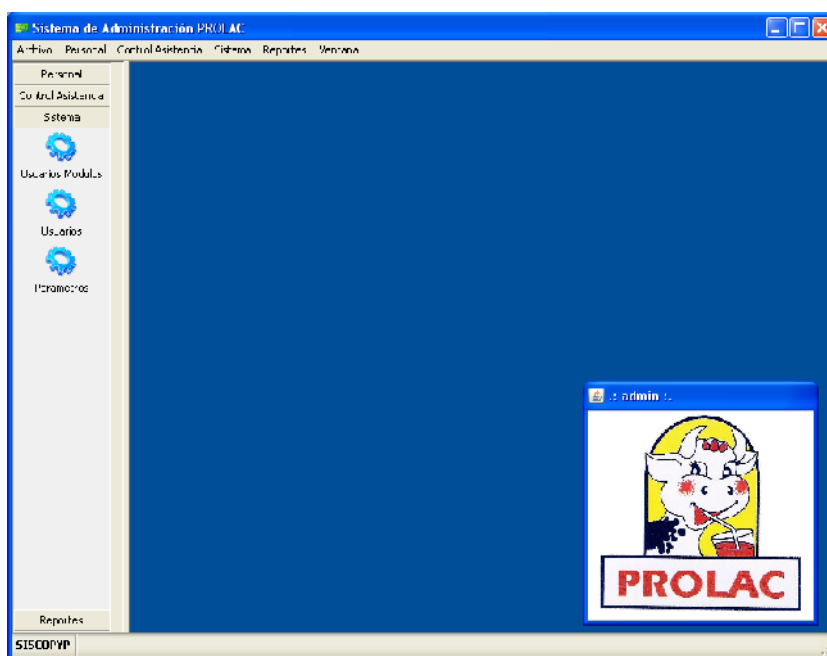


Figura N° 183: Pantalla Principal Modulo Sistema

II.1.4.3.4.6.1 PANTALLA USUARIOS

En esta pantalla al hacer click en el enlace Usuarios se visualiza una lista de los datos de los usuarios que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar, Filtrar e Imprimir, también muestra un paginador que va visualizando los datos.

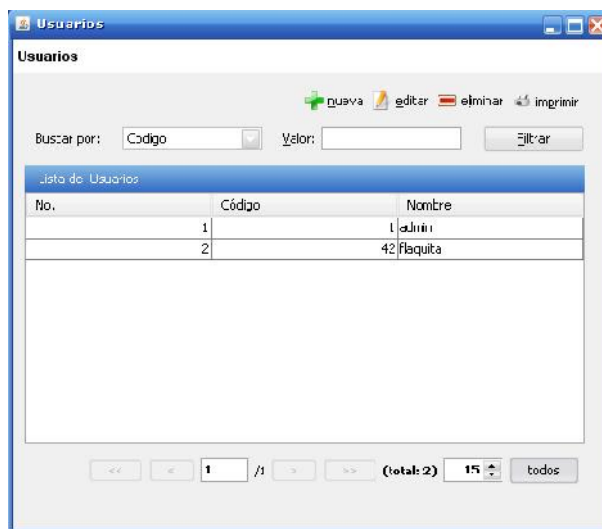


Figura N° 184: Pantalla Usuarios

PANTALLA NUEVO USUARIO

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Usuarios, en esta se visualiza un formulario para introducir datos del usuario que queremos agregar, cuenta con campos como: Nombre, Clave y Repetir Clave.

The 'Nuevo Usuario' form contains the following fields and buttons:

- Nombre:
- Clave:
- Repetir Clave:
- Buttons: Aceptar, Cancelar

Figura N° 185: Pantalla Nuevo Usuario

PANTALLA DE MENSAJE USUARIO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo usuario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla usuarios.

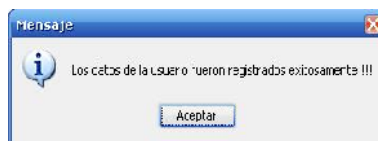


Figura N° 186: Pantalla de Mensaje Usuario Adicionado

PANTALLA MODIFICAR USUARIO

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Usuarios, en esta se visualiza un formulario con los datos actuales del usuario seleccionado, da la opción de editar los campos que quiera el usuario, cuenta con campos como: Nombre, Clave y Repetir Clave.



Figura N° 187: Pantalla Modificar Usuario

PANTALLA MENSAJE USUARIO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del usuario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Usuarios.

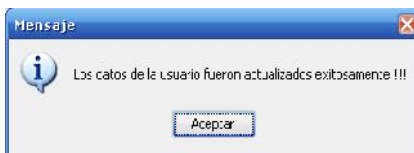


Figura N° 188: Pantalla Mensaje Usuario Modificado

PANTALLA MENSAJE DE CONFIRMACION ELIMINAR USUARIO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Usuarios, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el usuario seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al usuario de la base de datos, caso contrario visualiza la pantalla Usuarios.

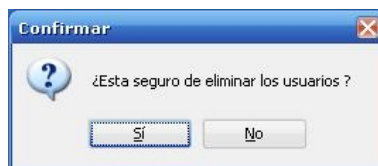


Figura N° 189: Pantalla Mensaje de Confirmación Eliminar Usuario

PANTALLA MENSAJE USUARIO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Usuario, haciendo click en la opción Aceptar visualiza la pantalla Usuarios.

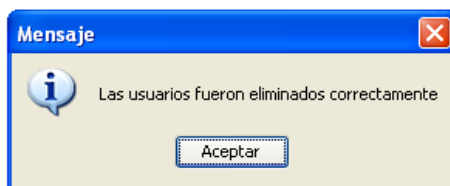


Figura N° 190: Pantalla Mensaje Usuario Eliminado

II.1.4.3.4.6.2 PANTALLA USUARIOS MODULOS

En esta pantalla al hacer click en el enlace Usuarios Módulos se visualiza una lista de los usuarios y de los datos de los módulos que se encuentran en la base de datos el cual el usuario puede seleccionar, en esta hay opción Guardar Cambios.

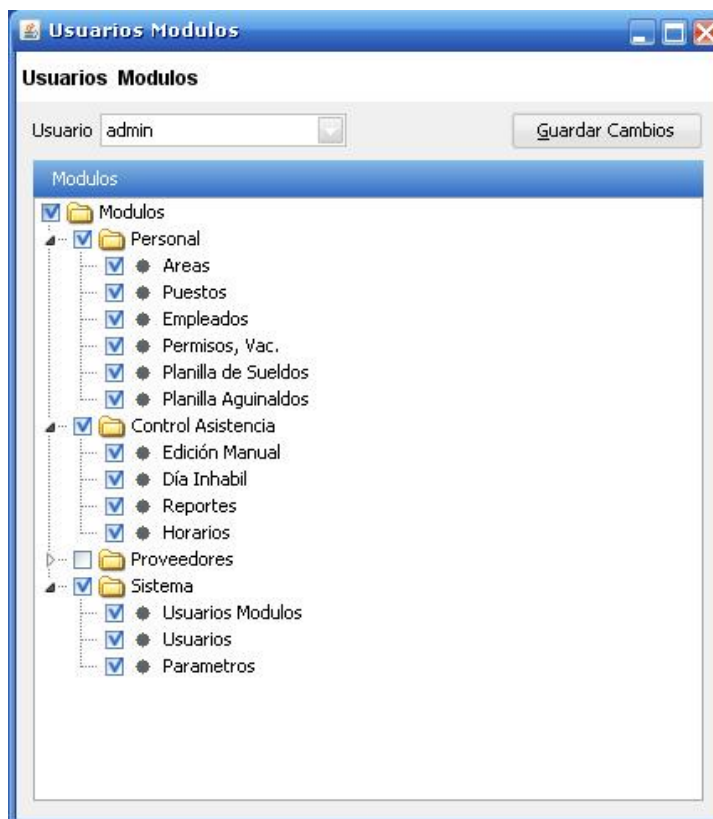


Figura N° 191: Pantalla Usuarios Módulos

PANTALLA MENSAJE DE CONFIRMACION GUARDAR CAMBIOS USUARIOS MODULOS

Esta pantalla se muestra después de hacer click en el enlace Guardar Cambios de la pantalla Usuarios Módulos, esta pantalla muestra un mensaje para que el usuario confirme si desea o no guardar los cambios del usuario seleccionado, al hacer click en la opción Si visualiza la pantalla Usuarios Módulos.

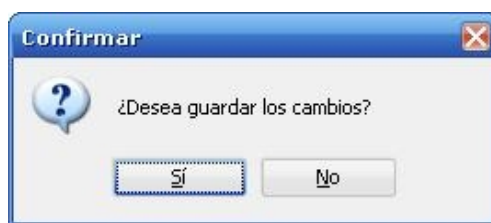


Figura 192: Pantalla Mensaje de Confirmación Guardar Cambios Usuarios Módulos

II.1.4.3.4.6.3 PANTALLA PARAMETROS (GENERAL)

En esta pantalla al hacer click en el enlace Parámetros se visualiza una lista de los datos de los descuentos generales que se realizan en la planilla de sueldos, como así también se establece el salario mínimo nacional, datos que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar.

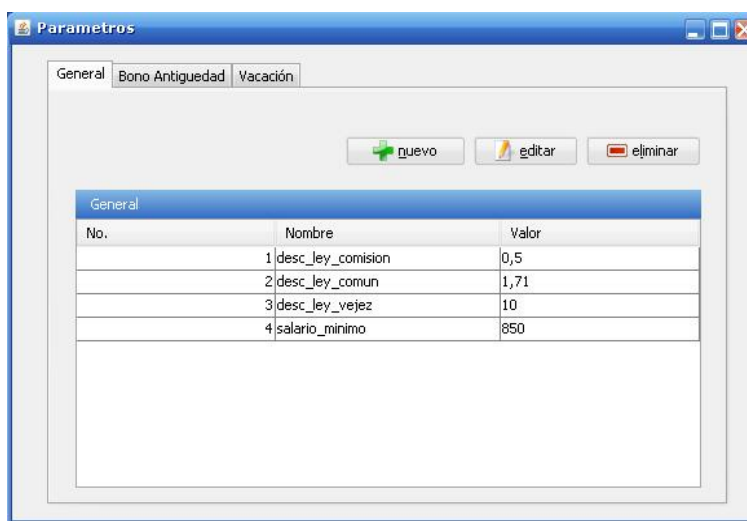


Figura N° 193: Pantalla Parámetros (General)

PANTALLA ADICIONAR PARAMETROS (GENERAL)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Parámetros (General), esta visualiza un formulario para introducir datos del parámetro general que se desea agregar, cuenta con campos como: Nombre y Valor.

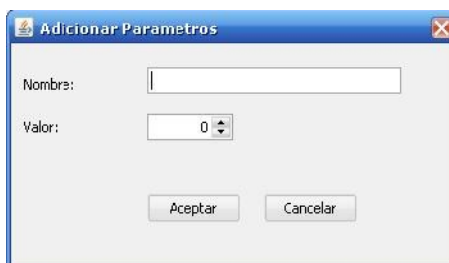


Figura N° 194: Pantalla Adicionar Parámetros (General)

PANTALLA DE MENSAJE PARAMETRO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo usuario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla usuarios.

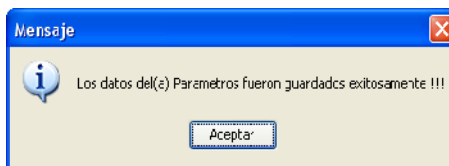


Figura N° 195: Pantalla de Mensaje Parámetro Adicionado

PANTALLA MODIFICAR PARAMETRO (GENERAL)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Parámetros (General), esta visualiza un formulario con los datos actuales del parámetro seleccionado, permite editar los campos que desea el usuario, cuenta con campos como: Nombre y Valor.

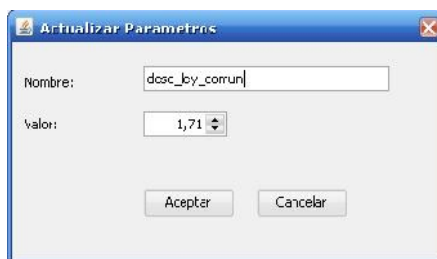


Figura N° 196: Pantalla Modificar Parámetro (General)

PANTALLA MENSAJE PARAMETRO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del parámetro en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Parametros (General).

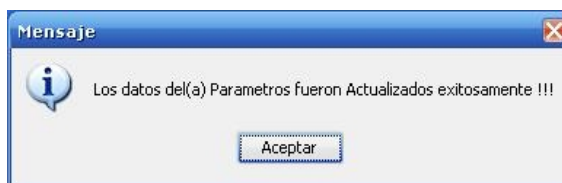


Figura N° 197: Pantalla Mensaje Parámetro Modificado

PANTALLA MENSAJE DE CONFIRMACION ELIMINAR PARAMETRO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Parámetro (General), esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el parámetro seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al parámetro de la base de datos, caso contrario visualiza la pantalla Parámetro (General).

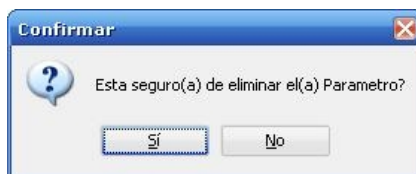


Figura N° 198: Pantalla Mensaje de Confirmación Eliminar Parámetro

PANTALLA MENSAJE PARAMETRO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Parámetro, haciendo click en la opción Aceptar visualiza la pantalla Parámetro (General).

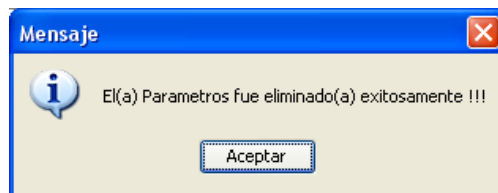


Figura N° 199: Pantalla Mensaje Parámetro Eliminado

PANTALLA PARAMETROS (BONO ANTIGÜEDAD)

En esta pantalla al hacer click en el enlace Parámetros se visualiza una lista de los datos de los bonos de antigüedad que se realizan en la planilla de sueldos, datos que se encuentran en la base de datos, en esta pantalla hay opciones como Nuevo, Editar, Eliminar.

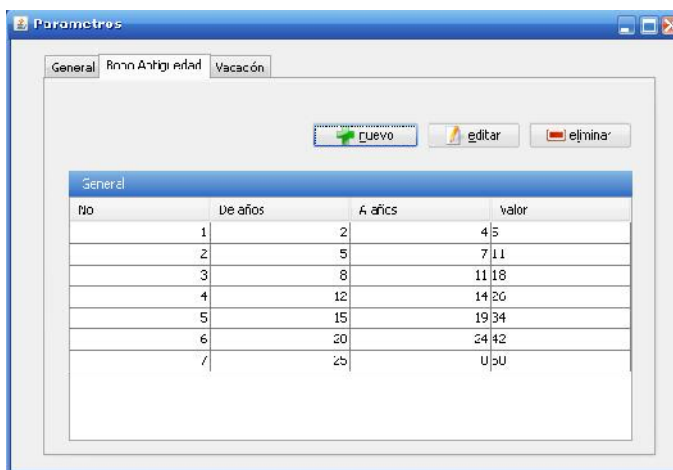


Figura N° 200: Pantalla Parámetros (Bono Antigüedad)

PANTALLA ADICIONAR PARAMETROS (BONO ANTIGÜEDAD)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Parámetros (Bono Antigüedad), esta visualiza un formulario para introducir datos del parámetro que se desea agregar, cuenta con campos como: De Años, A Años y Valor.

The screenshot shows a window titled 'Adicionar Bono'. It contains three input fields with spinners: 'De años' (set to 0), 'A años' (set to 0), and 'Valor' (set to 0). At the bottom are two buttons: 'Aceptar' and 'Cancelar'.

Figura N° 201: Pantalla Adicionar Parámetros (Bono Antigüedad)

PANTALLA DE MENSAJE PARAMETRO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo usuario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Parámetros (Bono Antigüedad).

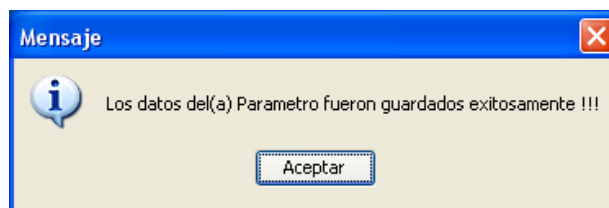


Figura N° 202: Pantalla de Mensaje Parámetro Adicionado

PANTALLA MODIFICAR PARAMETRO (BONO ANTIGÜEDAD)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Parámetros (Bono Antigüedad), esta visualiza un formulario con los datos actuales del parámetro seleccionado, permite editar los campos que desea el usuario, cuenta con campos como: De Años, A Años y Valor.

Figura N° 203: Pantalla Modificar Parámetro (Bono Antigüedad)

PANTALLA MENSAJE PARAMETRO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del parámetro en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Parametros (Bono Antigüedad).

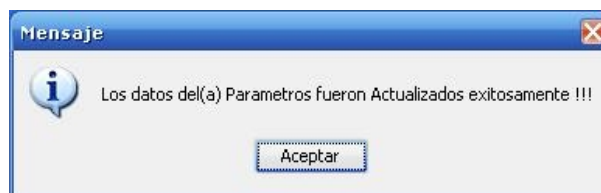


Figura N° 204: Pantalla Mensaje Parámetro Modificado

PANTALLA MENSAJE DE CONFIRMACION ELIMINAR PARAMETRO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Parámetro (Bono Antigüedad), esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el parámetro seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al parámetro de la base de datos, caso contrario visualiza la pantalla Parámetro (Bono Antigüedad).

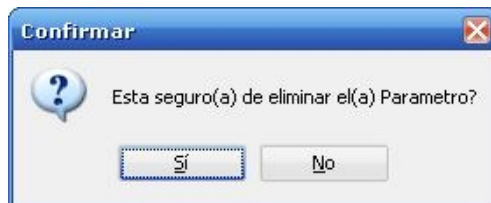


Figura N° 205: Pantalla Mensaje de Confirmación Eliminar Parámetro

PANTALLA MENSAJE PARAMETRO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Parámetro, haciendo click en la opción Aceptar visualiza la pantalla Parámetro (Bono Antigüedad).

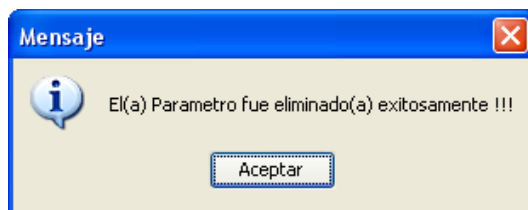


Figura N° 206: Pantalla Mensaje Parámetro Eliminado

PANTALLA PARAMETROS (VACACION)

En esta pantalla al hacer click en el enlace Parámetros se visualiza una lista de los datos que establece parámetros para la vacación de los empleados, datos que se encuentran en la base de datos, en esta pantalla hay opciones como Nuevo, Editar, Eliminar.

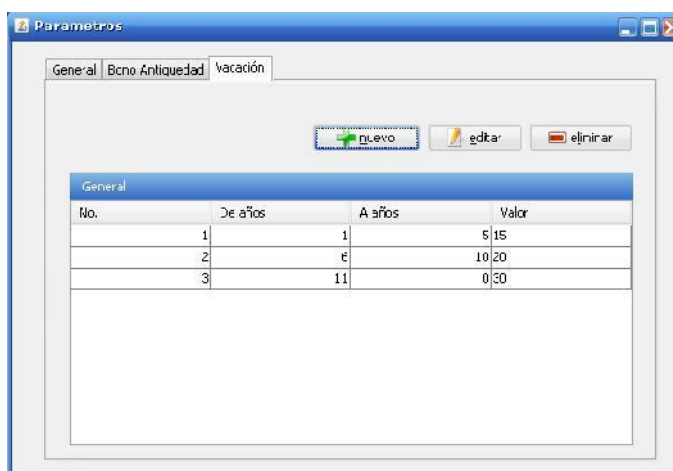


Figura N° 207: Pantalla Parámetros (Vacación)

PANTALLA ADICIONAR PARAMETROS (VACACION)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Parámetros (Vacación), esta visualiza un formulario para introducir datos del parámetro que se desea agregar, cuenta con campos como: De Años, A Años y Valor.

Figura N° 208: Pantalla Adicionar Parámetros (Vacación)

PANTALLA DE MENSAJE PARAMETRO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo usuario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Parámetros (Vacación).

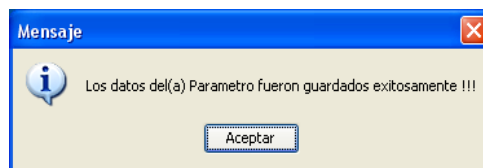


Figura N° 209: Pantalla de Mensaje Parámetro Adicionado

PANTALLA MODIFICAR PARAMETRO (VACACION)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Parámetros (Vacación), esta visualiza un formulario con los datos actuales del parámetro seleccionado, permite editar los campos que desea el usuario, cuenta con campos como: De Años, A Años y Valor.

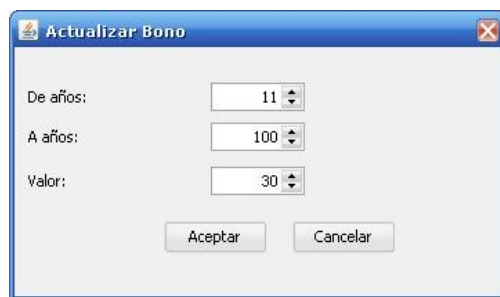


Figura N° 210: Pantalla Modificar Parámetro (Vacación)

PANTALLA MENSAJE PARAMETRO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del parámetro en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Parámetros (Vacación).

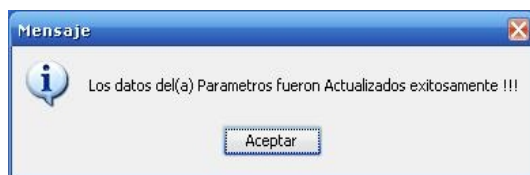


Figura N° 211: Pantalla Mensaje Parámetro Modificado

PANTALLA MENSAJE DE CONFIRMACION ELIMINAR PARAMETRO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Parámetro (Vacación), esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el parámetro seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al parámetro de la base de datos, caso contrario visualiza la pantalla Parámetro (Vacación).

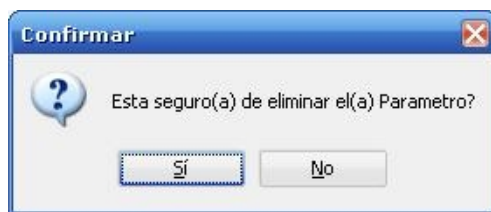


Figura N° 212: Pantalla Mensaje de Confirmación Eliminar Parámetro

PANTALLA MENSAJE PARAMETRO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Parámetro, haciendo click en la opción Aceptar visualiza la pantalla Parámetro (Vacación).

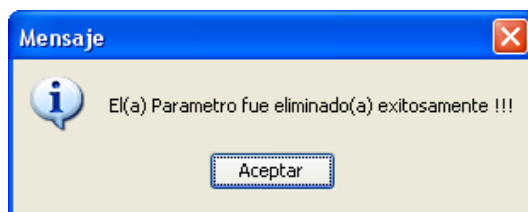


Figura N° 213: Pantalla Mensaje Parámetro Eliminado

II.1.4.3.4.7 MODULO II: PERSONAL

En este modulo se encuentran todos los procesos que puede realizar un usuario que tenga acceso al modulo.

PANTALLA PRINCIPAL MODULO PERSONAL

Esta pantalla muestra una vez verificados los datos en la base de datos del usuario que quiera ingresar al sistema y que quiera acceder al modulo personal, la pantalla visualiza el menú correspondiente al modulo.

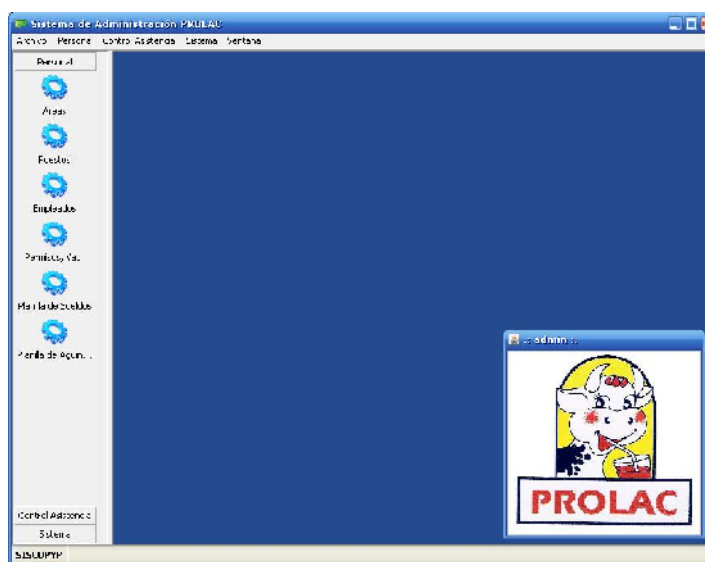


Figura N° 214: Pantalla Principal Modulo Personal

II.1.4.3.4.7.1 PANTALLA EMPLEADOS

En esta pantalla al hacer click en el enlace Empleados se visualiza una lista de los datos de los empleados que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar e Imprimir Credencial, también muestra un paginador que va visualizando los datos.



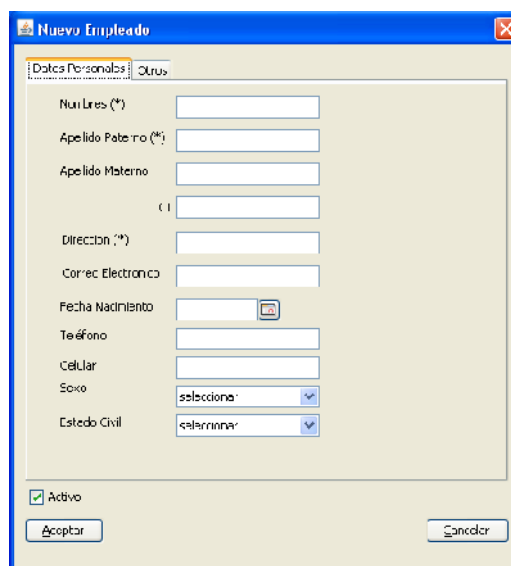
The screenshot shows a window titled 'Empleados' with a menu bar containing 'Nuevo', 'Editar', 'Eliminar', 'Administrar', 'Imprimir', and 'Imprimir Crear'. Below the menu is a table titled 'Lista de Empleados' with columns 'No.', 'CI', 'Nombres', and 'Puesto'. The table contains six rows of employee data.

No.	CI	Nombres	Puesto
1.15784910		Victor Cardoso	Gerente General
1.15784910		Victor Rodriguez Hernandez	Secretario I
1.15784910		Victor Rodriguez Hernandez	Secretario II
1.15784910		Sonia Perez Aguirre	Depto de Contabilidad y Finanzas
1.15784910		Cesar Mercedes Paz	Asistente de Finanzas
1.15784910		Katiza Sanchez Aguilar	Encargado de Mantenimiento

Figura N° 215: Pantalla Empleados

PANTALLA NUEVO EMPLEADO (DATOS PERSONALES)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Empleados, en esta se visualiza un formulario para introducir datos personales del empleado que queremos agregar, cuenta con campos como: Nombres, Apellido Paterno, Apellido Materno, CI, Dirección, Correo Electrónico, Fecha de Nacimiento, Teléfono, Celular, Sexo, Estado Civil.



The screenshot shows a window titled 'Nuevo Empleado' with a tabbed interface. The 'Datos Personales' tab is selected. The form contains the following fields and controls:

- Nombres (*)
- Apellido Paterno (*)
- Apellido Materno
- CI
- Dirección (*)
- Correo Electrónico
- Fecha Nacimiento
- Teléfono
- Celular
- Sexo (dropdown menu)
- Estado Civil (dropdown menu)
- ☒ Activo
- Aceptar button
- Cancelar button

Figura N° 216: Pantalla Nuevo Empleado (Datos Personales)

PANTALLA NUEVO EMPLEADO (OTROS DATOS)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Empleados, en esta se visualiza un formulario para introducir otros datos del empleado que queremos agregar, cuenta con campos como: Número, Clave, Área, Puesto y Fotografía.

Figura N° 217: Pantalla Nuevo Empleado (Otros Datos)

PANTALLA NUEVO EMPLEADO (CONTRATOS)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Empleados, en esta se visualiza un formulario para seleccionar fecha inicio y fecha fin de contrato, área, puesto e introducir observaciones.

No.	Fecha Ini.	Fecha Fin.	Estado	Fecha Fin. Final
1	08-01-2011	09-01-2012	Activo	

Figura N° 218: Pantalla Nuevo Empleado (Contratos)

PANTALLA MENSAJE CONTRATO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Empleados.

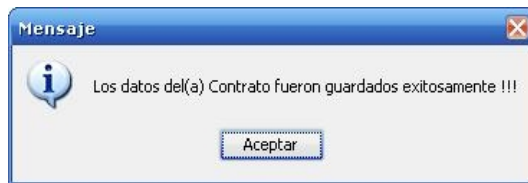


Figura N° 219: Pantalla Mensaje Contrato Adicionado

PANTALLA MENSAJE EMPLEADO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Empleados.

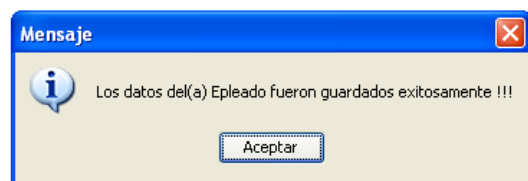


Figura N° 220: Pantalla Mensaje Empleado Adicionado

PANTALLA MODIFICAR EMPLEADO

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Empleados, en esta se visualiza un formulario con los datos actuales del empleado seleccionado, se tiene la opción de editar los campos que quiera el usuario, cuenta con campos como: Nombres, Apellido Paterno, Apellido Materno, CI, Dirección, Correo Electrónico, Fecha de Nacimiento, Teléfono, Celular, Sexo, Estado Civil, Numero, Clave, Puesto, Área y Fotografía.

Modificar Empleado

Datos Personales | Contratos | Anticipos | Vacaciones

Nombres (*) victor

Apellido Paterno (*) Cardozo

Apellido Materno

C.I. 3784910

Dirección (*) Juan XXIII

Correo Electrónico victor_c@hotmail.com

Fecha Nacimiento 26-mar-1982

Fecha Ingreso 26-mar-1982

Teléfono 321321

Celular 324124

Sexo masculino

Estado Civil casado

☒ Active

Aceptar Cancelar

Figura N° 221: Pantalla Modificar Empleado

PANTALLA MODIFICAR EMPLEADO (CONTRATOS)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Empleados y seleccionando la pestaña Contratos de la pantalla Modificar Empleado, en esta se visualiza un reporte del contrato actual del empleado seleccionado, además se tiene las opciones de nuevo, editar, eliminar, terminar y cancelar contratos.

Modificar Empleado

Datos Personales | Contratos | Anticipos | Vacaciones

Nuevo Cancelar Nuevo Editar Eliminar Actualizar

Contratos

No.	Fecha Ini	Fecha Fin	Area	Puesto	Sueldo	Estado	Fecha Fin Real
1	01-dic-2011	27-dic-2012	Secretaria	Secretaria 1	1200	Terminado	30-dic-2010
2	19-jul-2013	12-jul-2014	Administración y ...	Administrador de las 2500		Activo	

Aceptar Cancelar

Figura N° 222: Pantalla Modificar Empleado (Contratos)

PANTALLA MODIFICAR EMPLEADO (ADICIONAR CONTRATO)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Modificar Empleado (Contratos), en esta se visualiza un formulario para seleccionar e introducir los datos del contrato, cuenta con campos como: Fecha desde, Fecha hasta, Área, Puesto y Observaciones.

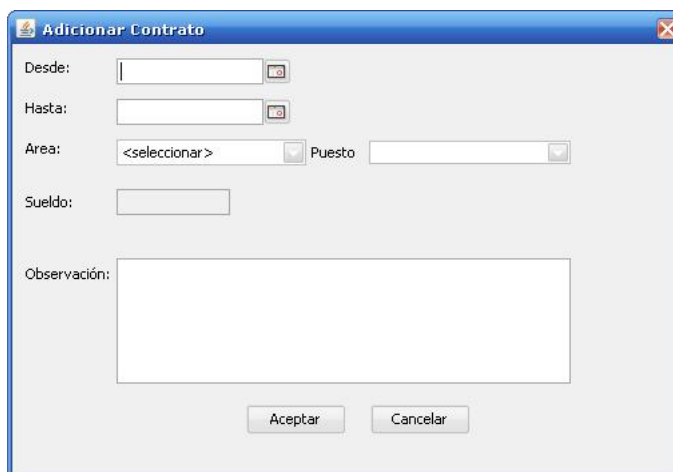


Figura N° 223: Pantalla Modificar Empleado (Adicionar Contrato)

PANTALLA MENSAJE CONTRATO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo contrato del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

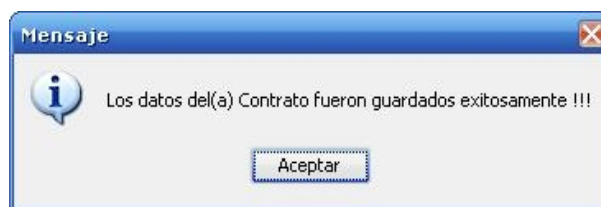


Figura N° 224: Pantalla Mensaje Contrato Adicionado

PANTALLA MODIFICAR EMPLEADO (ACTUALIZAR CONTRATO)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Modificar Empleado (Contratos), en esta se visualiza un formulario con datos

actuales, donde se tiene la opción de modificar los datos del contrato, cuenta con campos como: Fecha desde, Fecha hasta, Área, Puesto y Observaciones.

Figura N° 225: Pantalla Modificar Empleado (Actualizar Contrato)

PANTALLA MENSAJE CONTRATO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del contrato del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

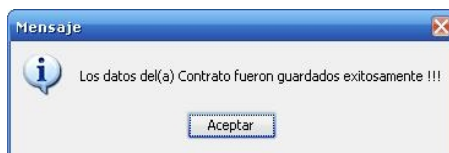


Figura N° 226: Pantalla Mensaje Contrato Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR CONTRATO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Modificar Empleado (Contratos), esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el contrato seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar el contrato de la base de datos, caso contrario visualiza la pantalla Modificar Empleado (Contratos).

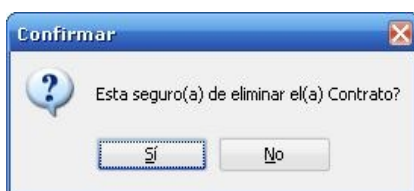


Figura N° 227: Pantalla Mensaje de Confirmación de Eliminar Contrato

PANTALLA MENSAJE CONTRATO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Contrato, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

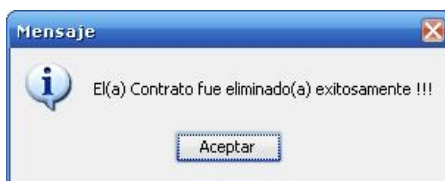


Figura N° 228: Pantalla Mensaje Contrato Eliminado

PANTALLA MODIFICAR EMPLEADO (TERMINAR CONTRATO)

Esta pantalla se muestra después de hacer click en el enlace Terminar de la pantalla Modificar Empleado (Contratos), en esta se visualiza un formulario para seleccionar la fecha de finalización de contrato, cuenta con campos como: Fecha finalización y Observaciones.



Figura N° 229: Pantalla Modificar Empleado (Terminar Contrato)

PANTALLA MENSAJE CONTRATO FINALIZADO

Esta pantalla se visualiza después de guardar los datos modificados del contrato del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

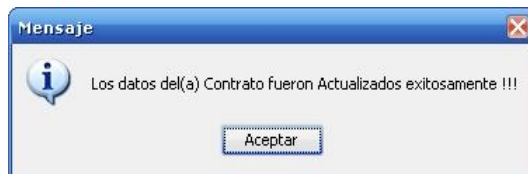


Figura N° 230: Pantalla Mensaje Contrato Finalizado

PANTALLA MENSAJE DE CONFIRMACION DE CANCELAR CONTRATO

Esta pantalla se muestra después de hacer click en el enlace Cancelar de la pantalla Modificar Empleado (Contratos), esta pantalla muestra un mensaje para que el usuario confirme si desea o no cancelar el contrato seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a cancelar el contrato, caso contrario visualiza la pantalla Modificar Empleado (Contratos).

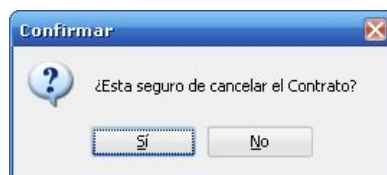


Figura N° 231: Pantalla Mensaje de Confirmación de Cancelar Contrato

PANTALLA MENSAJE CONTRATO CANCELADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Cancelar Contrato, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

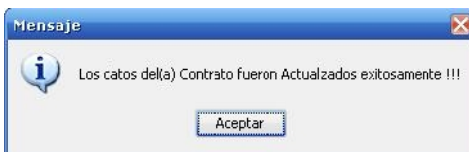


Figura N° 232: Pantalla Mensaje Contrato Cancelado

PANTALLA MODIFICAR EMPLEADO (ANTICIPOS)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Empleados y seleccionando la pestaña Anticipos de la pantalla Modificar Empleado, en esta se visualiza un reporte de los anticipos otorgados del empleado seleccionado, además se tiene las opciones de nuevo, editar y eliminar anticipos.

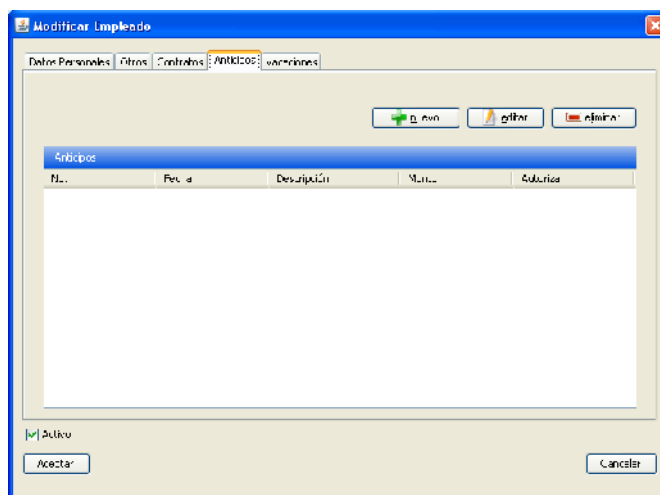


Figura N° 233: Pantalla Modificar Empleado (Anticipos)

PANTALLA MODIFICAR EMPLEADO (ADICIONAR ANTICIPO)

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Modificar Empleado (Anticipos), en esta se visualiza un formulario para seleccionar e introducir los datos del anticipo, cuenta con campos como: Fecha, Autoriza, Monto y Descripción.

Figura N° 234: Pantalla Modificar Empleado (Adicionar Anticipo)

PANTALLA MENSAJE ANTICIPO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo anticipo del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Anticipos).

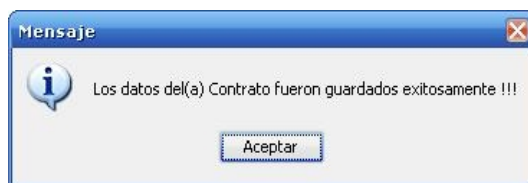


Figura N° 235: Pantalla Mensaje Anticipo Adicionado

PANTALLA MODIFICAR EMPLEADO (ACTUALIZAR ANTICIPO)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Modificar Empleado (Anticipos), en esta se visualiza un formulario con dato actuales, donde se tiene la opción de modificar los datos del anticipo, cuenta con campos como: Fecha, Autoriza, Monto y Descripción.

Figura N° 236: Pantalla Modificar Empleado (Actualizar Anticipo)

PANTALLA MENSAJE ANTICIPO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del contrato del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Contratos).

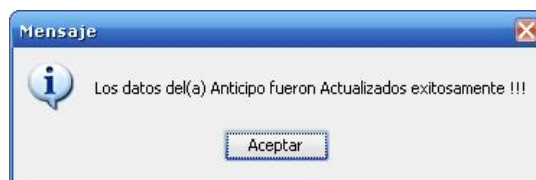


Figura N° 237: Pantalla Mensaje Anticipo Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR ANTICIPO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Modificar Empleado (Anticipos), esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el anticipo seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar el anticipo de la base de datos, caso contrario visualiza la pantalla Modificar Empleado (Anticipos).

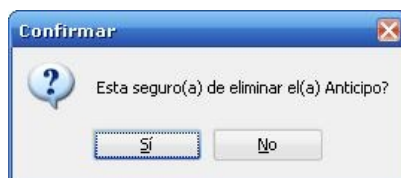


Figura N° 238: Pantalla Mensaje de Confirmación de Eliminar Anticipo

PANTALLA MENSAJE ANTICIPO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Anticipo, haciendo click en la opción Aceptar visualiza la pantalla Modificar Empleado (Anticipos).

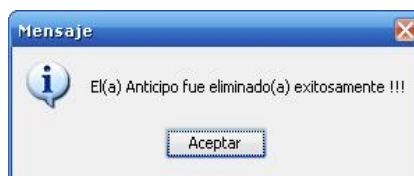


Figura N° 239: Pantalla Mensaje Anticipo Eliminado

PANTALLA MODIFICAR EMPLEADO (VACACIONES)

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Empleados y seleccionando la pestaña Vacaciones de la pantalla Modificar

Empleado, en esta se visualiza un reporte de las vacaciones otorgadas del empleado seleccionado.

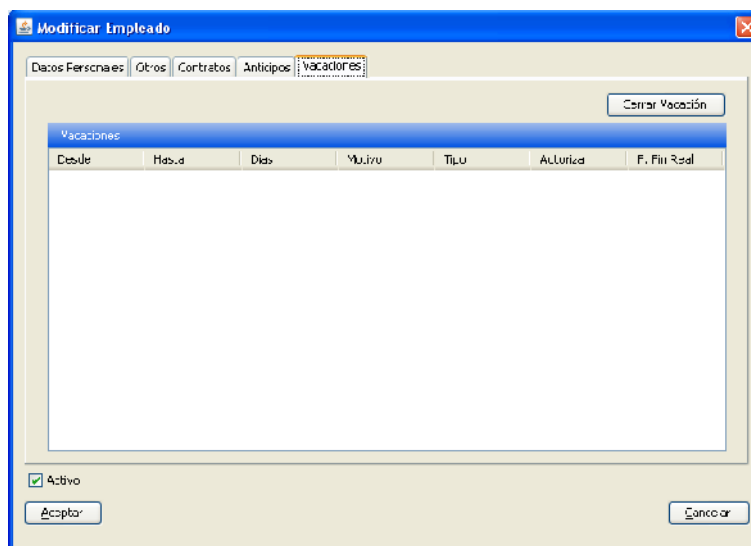


Figura N° 240: Pantalla Modificar Empleado (Vacaciones)

PANTALLA MENSAJE EMPLEADO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del empleado en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Empleados.

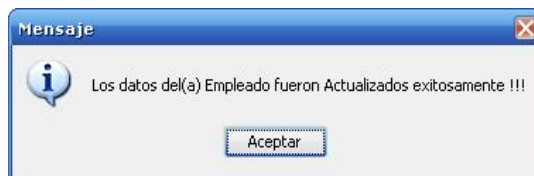


Figura N° 241: Pantalla Mensaje Empleado Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR EMPLEADO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Empleados, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el empleado seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al empleado de la base de datos, caso contrario visualiza la pantalla Empleados.

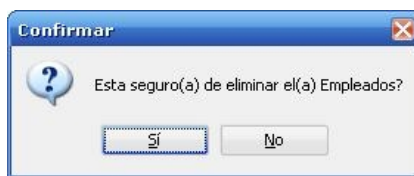


Figura N° 242: Pantalla Mensaje de Confirmación de Eliminar Empleado

PANTALLA MENSAJE EMPLEADO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Empleado, haciendo click en la opción Aceptar visualiza la pantalla Empleados.

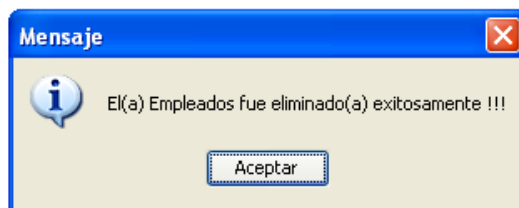


Figura N° 243: Pantalla Mensaje Empleado Eliminado

PANTALLA GENERAR CREDENCIAL

Esta pantalla se visualiza después de generar la credencial del empleado, primeramente debemos seleccionar un empleado de la pantalla Gestionar Empleados, luego hacemos click en el botón “Imprimir Credencial”, y se visualiza el credencial generado donde se puede imprimir.

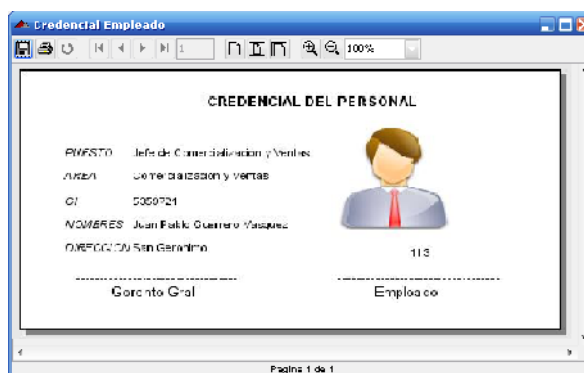


Figura N° 244: Pantalla Generar Credencial

II.1.4.3.4.7.2 PANTALLA AREAS

En esta pantalla al hacer click en el enlace Áreas se visualiza una lista de los datos de las áreas que se encuentran en la base de datos, en esta hay opciones como Nueva, Editar, Eliminar, Asociar Área-Puesto, Eliminar Área-Puesto, Filtrar e Imprimir, también muestra un paginador que va visualizando los datos.

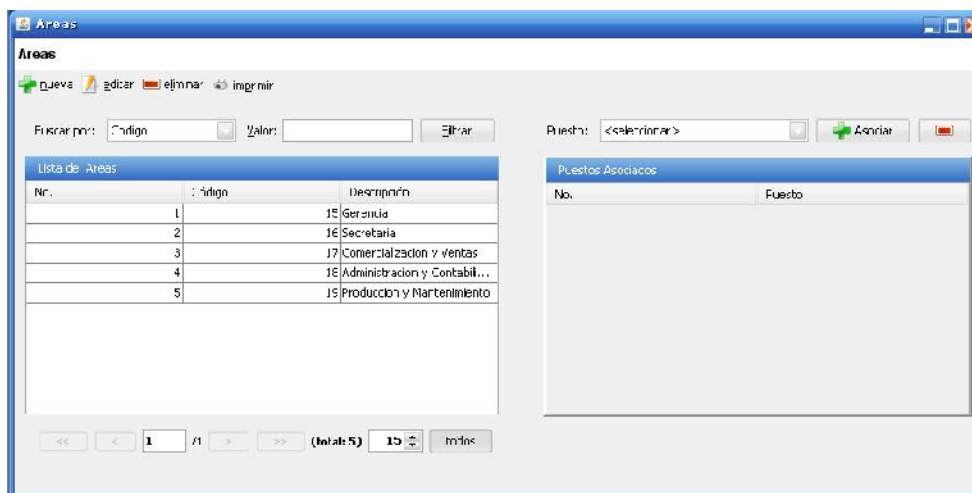


Figura N° 245: Pantalla Áreas

PANTALLA NUEVA AREA

Esta pantalla se muestra después de hacer click en el enlace Nueva de la pantalla Áreas, en esta se visualiza un formulario para introducir datos del área que se desea agregar, cuenta con el campo: Descripción.



Figura N° 246: Pantalla Nueva Área

PANTALLA MENSAJE AREA ADICIONADA

Esta pantalla se visualiza después de guardar los datos de la nueva área en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Áreas.



Figura N° 247: Pantalla Mensaje Área Adicionada

PANTALLA MODIFICAR AREA

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Áreas, en esta se visualiza un formulario con los datos actuales del área seleccionada, se tiene la opción de editar el campo: Descripción.

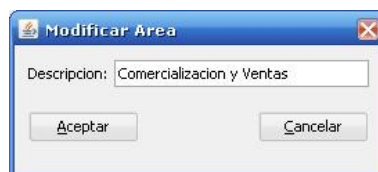


Figura N° 248: Pantalla Modificar Área

PANTALLA MENSAJE AREA MODIFICADA

Esta pantalla se visualiza después de guardar los datos modificados del área en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Áreas.

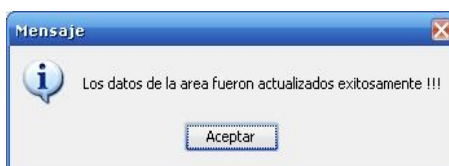


Figura N° 249: Pantalla Mensaje Área Modificada

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR AREA

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Áreas, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el área seleccionada, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al área de la base de datos, caso contrario visualiza la pantalla Áreas.

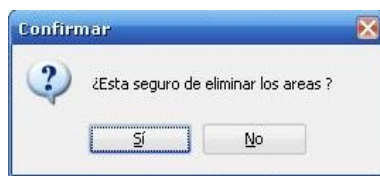


Figura N° 250: Pantalla Mensaje de Confirmación de Eliminar Área

PANTALLA MENSAJE AREA ELIMINADA

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Área, haciendo click en la opción Aceptar visualiza la pantalla Áreas.

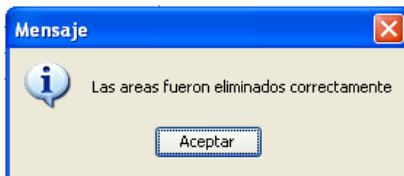


Figura N° 251: Pantalla de Mensaje Área Eliminada

II.1.4.3.4.4.3 PANTALLA PUESTOS

En esta pantalla al hacer click en el enlace Puestos se visualiza una lista de los datos de los puestos que se encuentran en la base de datos, en esta pantalla hay opciones como Nuevo, Editar, Eliminar, Filtrar e Imprimir, también muestra un paginador que va visualizando los datos.

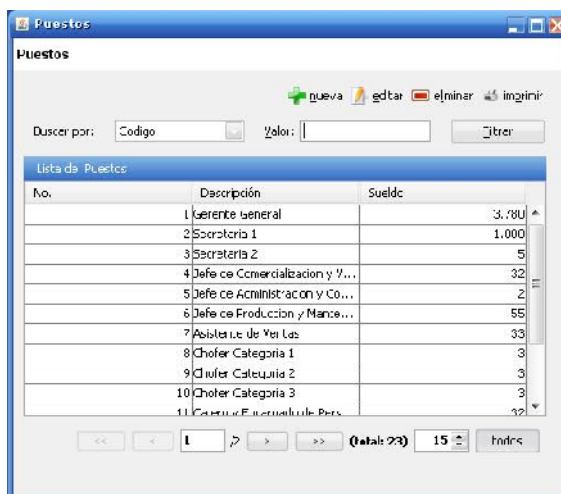


Figura N° 252: Pantalla Puestos

PANTALLA NUEVO PUESTO

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Puestos, en esta se visualiza un formulario para introducir datos del puesto que se desea agregar, cuenta con los campos: Descripción y Sueldo.



Figura N° 253: Pantalla Nuevo Puesto

PANTALLA MENSAJE PUESTO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo puesto en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Puestos.



Figura N° 254: Pantalla Mensaje Puesto Adicionado

PANTALLA MODIFICAR PUESTO

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Puestos, en esta se visualiza un formulario con los datos actuales del puesto seleccionado, se tiene la opción de editar los campos que desea el usuario, cuenta con los campos: Descripción y Sueldo.



Figura N° 255: Pantalla Modificar Puesto

PANTALLA MENSAJE PUESTO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del puesto en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Puestos.

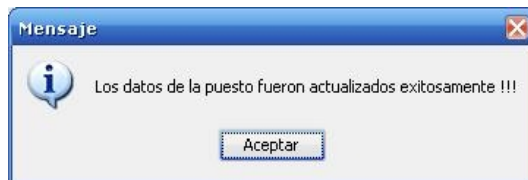


Figura N° 256: Pantalla Mensaje Puesto Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR PUESTO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Puestos, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el puesto seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al puesto de la base de datos, caso contrario visualiza la pantalla Puestos.

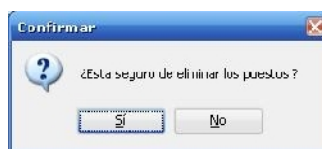


Figura N° 257: Pantalla Mensaje de Confirmación de Eliminar Puestos

PANTALLA MENSAJE PUESTO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Puesto, haciendo click en la opción Aceptar visualiza la pantalla Puestos.

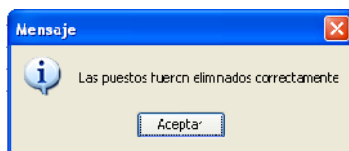


Figura N° 258: Pantalla de Mensaje Puesto Eliminado

II.1.4.3.4.4.4 PANTALLA PERMISOS, VACACIONES

En esta pantalla al hacer click en el enlace Permisos-Vacaciones se visualiza una lista de los datos de los empleados y los permisos que tienen, que se encuentran en la base de datos, además existe opciones como Nuevo, Editar, Eliminar, Filtrar e Imprimir, también muestra un paginador que va visualizando los datos.

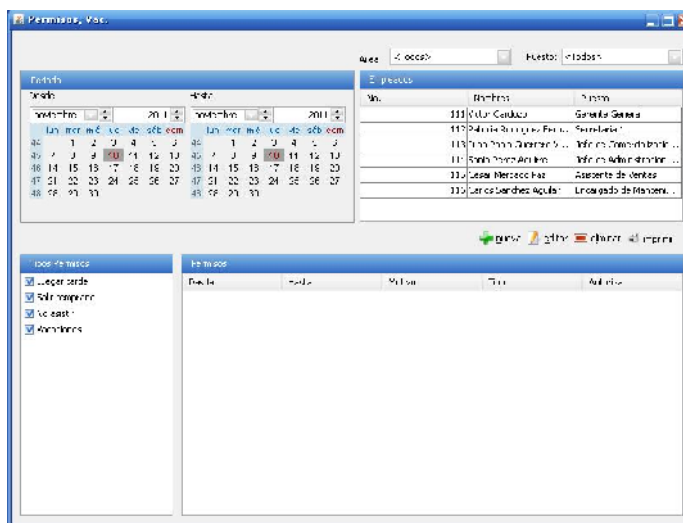


Figura N° 259: Pantalla Permisos-Vacaciones

PANTALLA NUEVO PERMISO

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Permisos-Vacaciones, en esta se visualiza un formulario para seleccionar e introducir datos del permiso que queremos agregar, cuenta con los campos: Fecha Desde, Fecha Hasta, Tipo, Autoriza y Motivo.



Figura N° 260: Pantalla Nuevo Permiso

PANTALLA MODIFICAR PERMISO

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Permisos-Vacaciones, en esta se visualiza un formulario con los datos actuales del permiso seleccionado, se tiene la opción de editar los campos que desea el usuario, cuenta con los campos: Fecha Desde, Fecha Hasta, Tipo, Autoriza y Motivo.

Figura N° 261: Pantalla Modificar Permiso

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR PERMISO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Permisos-Vacaciones, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el permiso seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al puesto de la base de datos, caso contrario visualiza la pantalla Permisos-Vacaciones.

Figura N° 262: Pantalla Mensaje de Confirmación de Eliminar Permiso

II.1.4.3.4.7.5 PANTALLA PLANILLA DE SUELDOS

En esta pantalla al hacer click en el enlace Planilla de Sueldos se visualiza una lista de los datos de las planillas de sueldo que se encuentran en la base de datos, en esta hay opciones como Nueva Planilla, Eliminar Planilla, Cargar Detalle e Imprimir.

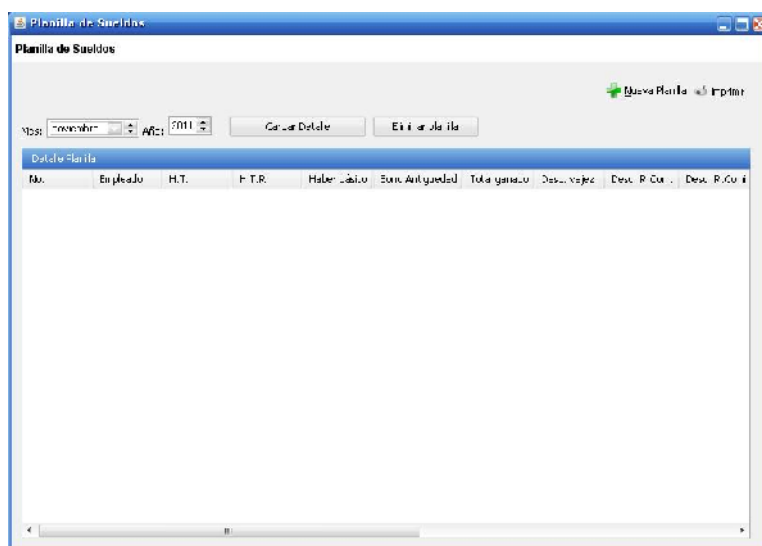


Figura N° 263: Pantalla Planilla de Sueldos

PANTALLA NUEVA PLANILLA DE SUELDOS

Esta pantalla se muestra después de hacer click en el enlace Nueva Planilla de la pantalla Planilla Sueldos, en esta se visualiza un formulario para seleccionar datos y generar la planilla de sueldo, cuenta con campos como: Mes, Año, Responsable.



Figura N° 264: Pantalla Nueva Planilla de Sueldos

PANTALLA MENSAJE PLANILLA DE SUELDO ADICIONADA

Esta pantalla se visualiza después de guardar los datos de la Planilla de Sueldo en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Planilla Sueldos.

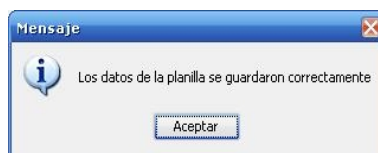


Figura N° 265: Pantalla de Mensaje Planilla Sueldo Adicionada

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR PLANILLA

Esta pantalla se muestra después de hacer click en el enlace Eliminar Planilla de la pantalla Planilla de Sueldos, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar la planilla seleccionada, al hacer click en la opción Si automáticamente el sistema pasa a eliminar la planilla de la base de datos, caso contrario visualiza la pantalla Planilla de Sueldos.

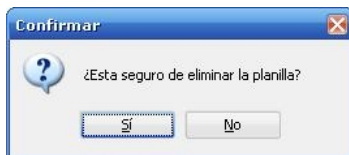


Figura N° 266: Pantalla Mensaje de Confirmación de Eliminar Planilla

PANTALLA MENSAJE PLANILLA ELIMINADA

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Planilla, haciendo click en la opción Aceptar visualiza la pantalla Planilla de Sueldos.

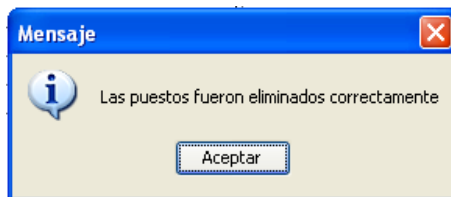


Figura N° 267 Pantalla Mensaje Planilla Eliminada

II.1.4.3.4.7.6 PANTALLA PLANILLA DE AGUINALDOS

En esta pantalla al hacer click en el enlace Planilla de Aguinaldos se visualiza una lista de los datos de las planillas de aguinaldo que se encuentran en la base de datos, en esta hay opciones como Nueva Planilla, Eliminar Planilla, Cargar Detalle e Imprimir.

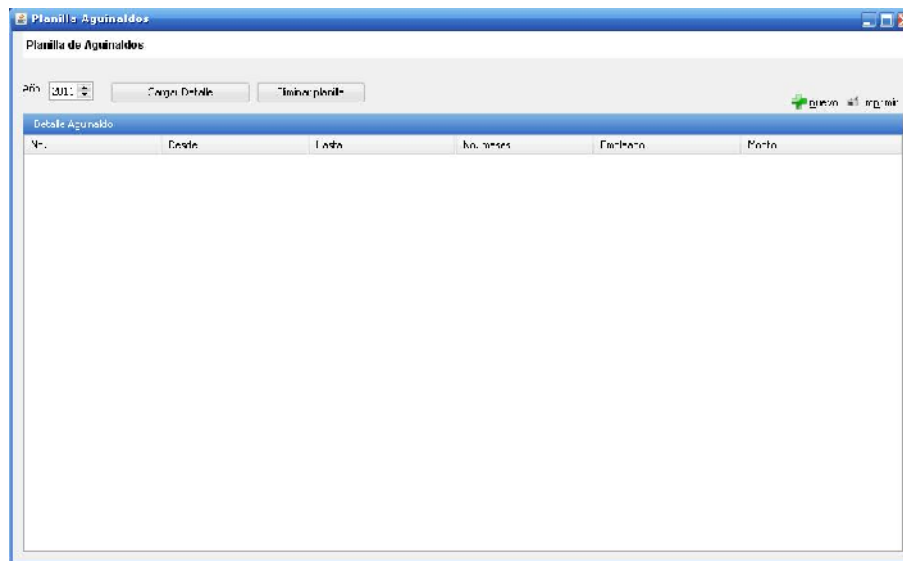


Figura N° 268: Pantalla Planilla de Aguinaldos

PANTALLA NUEVA PLANILLA DE AGUINALDO

Esta pantalla se muestra después de hacer click en el enlace Nueva Planilla de la pantalla Planilla Aguinaldos, en esta se visualiza un formulario para seleccionar datos y generar la planilla de aguinaldo, cuenta con campos como: Responsable, Año y Observación.



Figura N° 269: Pantalla Nueva Planilla de Aguinaldo

PANTALLA MENSAJE PLANILLA DE AGUINALDO ADICIONADA

Esta pantalla se visualiza después de guardar los datos de la Planilla de Aguinaldo en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Planilla Aguinaldos.

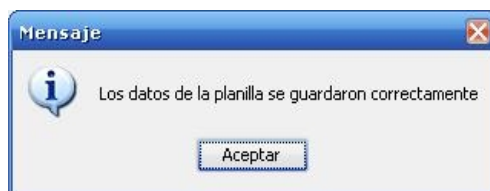


Figura N° 270: Pantalla Mensaje Planilla Aguinaldo Adicionada

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR PLANILLA

Esta pantalla se muestra después de hacer click en el enlace Eliminar Planilla de la pantalla Planilla de Aguinaldos, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar la planilla seleccionada, al hacer click en la opción Si automáticamente el sistema pasa a eliminar la planilla de la base de datos, caso contrario visualiza la pantalla Planilla de Aguinaldos.

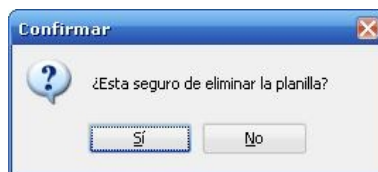


Figura N° 271: Pantalla Mensaje Confirmación de Eliminar Planilla

PANTALLA MENSAJE PLANILLA ELIMINADA

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Planilla, haciendo click en la opción Aceptar visualiza la pantalla Planilla de Aguinaldos.

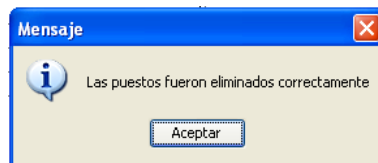


Figura N° 272: Pantalla Mensaje Planilla Eliminada

II.1.4.3.4.8 MODULO III: CONTROL ASISTENCIA

En este modulo se encuentran todos los procesos que puede realizar un usuario que tenga acceso al modulo.

PANTALLA PRINCIPAL MODULO CONTROL ASISTENCIA

Esta pantalla muestra una vez verificados los datos en la base de datos del usuario que quiera ingresar al sistema y que quiera acceder al modulo Control Asistencia, la pantalla visualiza el menú correspondiente al modulo.

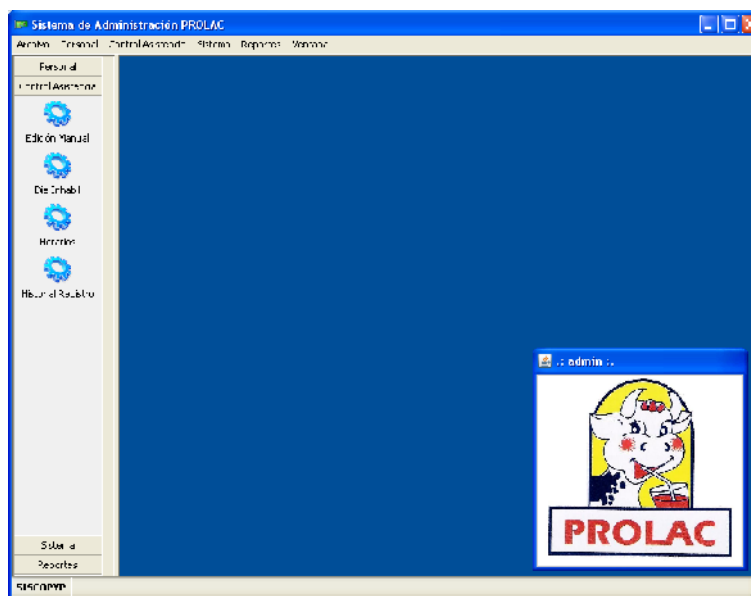


Figura N° 273: Pantalla Principal Modulo Control Asistencia

II.1.4.3.4.8.1 PANTALLA DIA INHABIL

En esta pantalla al hacer click en el enlace Día Inhabil se visualiza una lista de los datos de los días inhábiles que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar, Filtrar e Imprimir, también muestra un paginador que visualiza los datos.

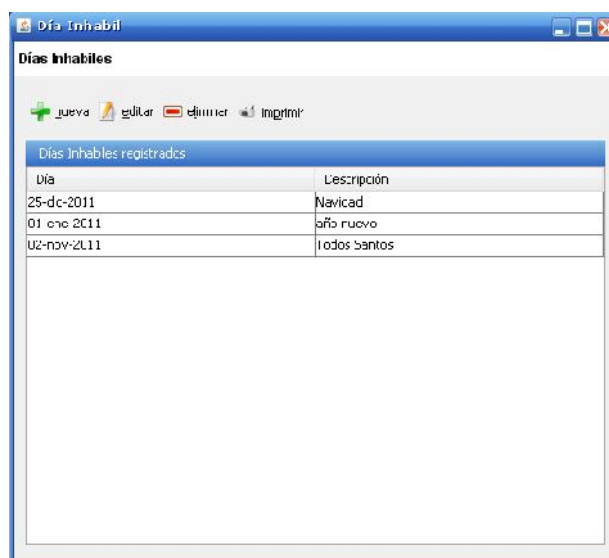


Figura N° 274: Pantalla Día Inhábil

PANTALLA NUEVO DIA INHABIL

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Día Inhábil, en esta se visualiza un formulario para seleccionar e introducir datos del día inhábil que queremos agregar, cuenta con campos como: Fecha y Descripción.



Figura N° 275: Pantalla Nuevo Día Inhábil

PANTALLA MENSAJE DIA INHABIL ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo día inhábil en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Día Inhábil.



Figura N° 276: Pantalla Mensaje Día Inhábil Adicionado

PANTALLA MODIFICAR DIA INHABIL

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Día Inhábil, en esta se visualiza un formulario con los datos actuales del día inhábil seleccionado, se tiene la opción de editar los campos que desea el usuario, cuenta con campos como: Fecha y Descripción.



Figura N° 277: Pantalla Modificar Día Inhábil

PANTALLA MENSAJE DIA INHABIL MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del día inhábil en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Día Inhábil.

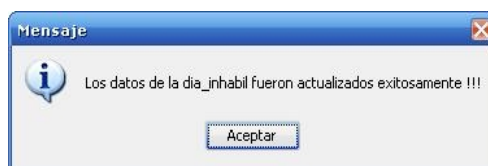


Figura N° 278: Pantalla Mensaje Día Inhábil Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR DIA INHABIL

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Día Inhábil, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el día inhábil seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar al día inhábil de la base de datos, caso contrario visualiza la pantalla Día Inhábil.

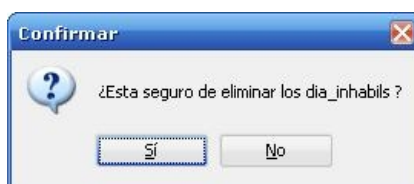


Figura N° 279: Pantalla Mensaje de Confirmación de Eliminar Día Inhábil

PANTALLA MENSAJE DIA INHABIL ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Día Inhábil, haciendo click en la opción Aceptar visualiza la pantalla Día Inhábil.

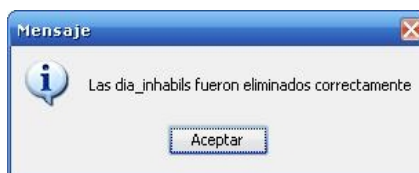


Figura N° 280: Pantalla de Mensaje Día Inhábil Eliminado

II.1.4.3.4.8.2 PANTALLA HORARIOS

En esta pantalla al hacer click en el enlace Horarios se visualiza una lista de los horarios que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar y Limpiar Selección.

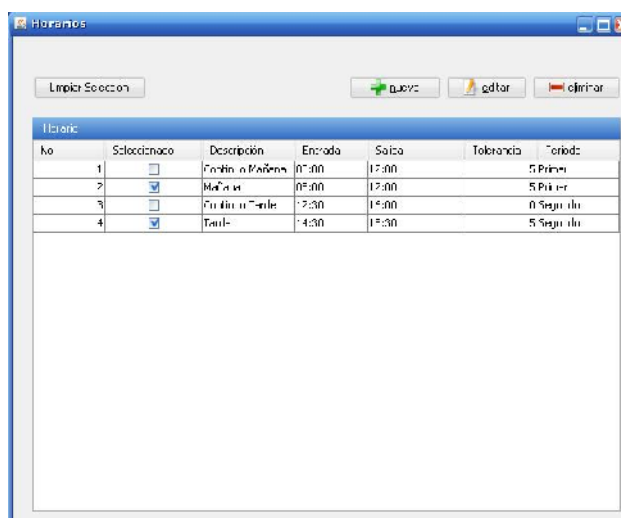


Figura N° 281: Pantalla Horarios

PANTALLA ADICIONAR HORARIO

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Horarios, se visualiza un formulario para seleccionar e introducir datos del horario que queremos agregar, cuenta con campos como: Descripción, Hora Entrada, Hora Salida, Tiempo de Tolerancia y Periodo.

Figura N° 282: Pantalla Adicionar Horario

PANTALLA MENSAJE HORARIO ADICIONADO

Esta pantalla se visualiza después de guardar los datos del nuevo horario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Horarios.

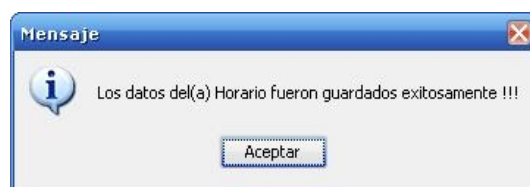


Figura N° 283: Pantalla Mensaje Horario Adicionado

PANTALLA MODIFICAR HORARIO

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Horarios, en esta se visualiza un formulario con los datos actuales del horario seleccionado, se tiene la opción de editar los campos que desea el usuario, cuenta con campos como: Descripción, Hora Entrada, Hora Salida, Tiempo de Tolerancia y Periodo.

Figura N° 284: Pantalla Modificar Horario

PANTALLA MENSAJE HORARIO MODIFICADO

Esta pantalla se visualiza después de guardar los datos modificados del horario en la base de datos, haciendo click en la opción Aceptar visualiza la pantalla Horarios.

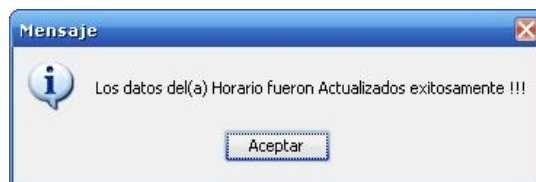


Figura N° 285: Pantalla Mensaje Horario Modificado

PANTALLA MENSAJE DE CONFIRMACION DE ELIMINAR HORARIO

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Horarios, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el horario seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar el horario de la base de datos, caso contrario visualiza la pantalla Horarios.

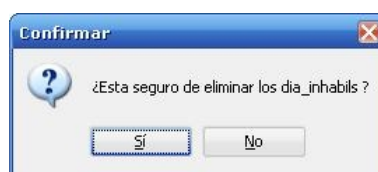


Figura N° 286: Pantalla Mensaje Confirmación de Eliminar Horario

PANTALLA MENSAJE HORARIO ELIMINADO

Esta pantalla se visualiza después de seleccionar la opción Si de la pantalla Mensaje de Confirmación de Eliminar Horario, haciendo click en la opción Aceptar visualiza la pantalla Horarios.

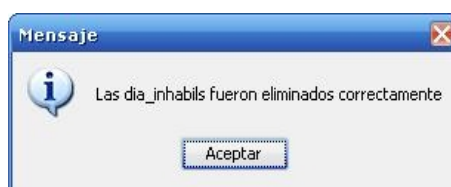


Figura N° 287: Pantalla Mensaje Horario Eliminado

II.1.4.3.4.8.3 PANTALLA EDICION MANUAL

En esta pantalla al hacer click en el enlace Edición Manual se visualiza una lista de los datos de los empleados y el registro de su asistencia que se encuentran en la base de datos, en esta hay opciones como Nuevo, Editar, Eliminar, Filtrar e Imprimir, también muestra un paginador que visualiza los datos.

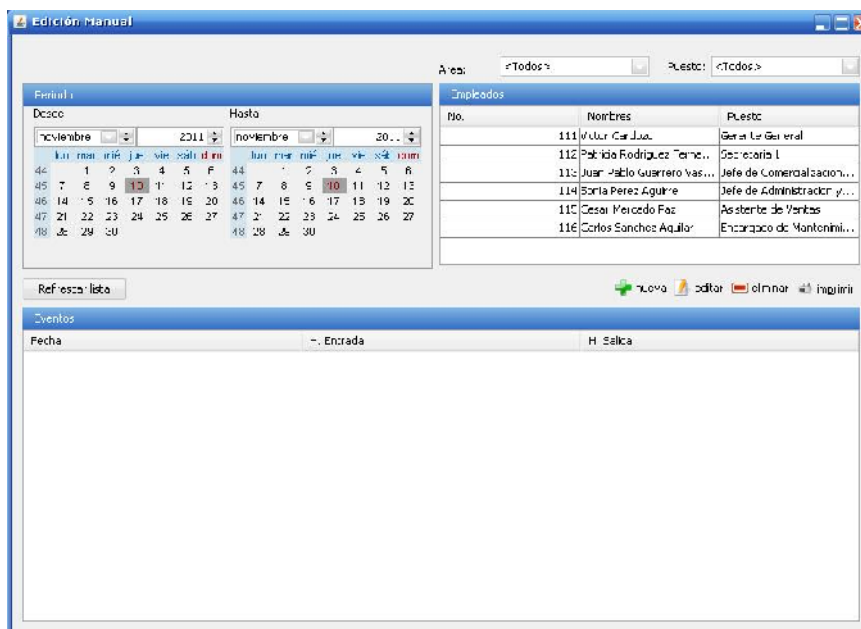


Figura N° 288: Pantalla Edición Manual

PANTALLA NUEVA EDICION MANUAL

Esta pantalla se muestra después de hacer click en el enlace Nuevo de la pantalla Edición Manual, en esta se visualiza un formulario para seleccionar e introducir datos de la edición manual que queremos agregar, cuenta con campos como: Fecha, Hora entrada y Hora Salida.



Figura N° 289: Pantalla Nueva Edición Manual

PANTALLA MODIFICAR EDICION MANUAL

Esta pantalla se muestra después de hacer click en el enlace Editar de la pantalla Edición Manual, en esta se visualiza un formulario con los datos actuales de la edición manual seleccionada, se tiene la opción de editar los campos que desea el usuario, cuenta con campos como: Fecha, Hora entrada y Hora Salida.



Figura N° 290: Pantalla Modificar Edición Manual

PANTALLA MENSAJE CONFIRMACION DE ELIMINAR EDICION MANUAL

Esta pantalla se muestra después de hacer click en el enlace Eliminar de la pantalla Edición Manual, esta pantalla muestra un mensaje para que el usuario confirme si desea o no eliminar el evento seleccionado, al hacer click en la opción Si automáticamente el sistema pasa a eliminar el evento de la base de datos, caso contrario visualiza la pantalla Edición Manual.

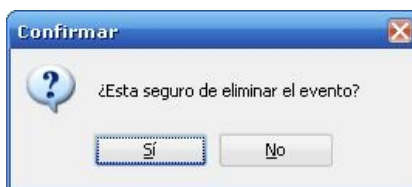


Figura N° 291: Pantalla Mensaje Confirmación de Eliminar Edición Manual

II.1.4.3.4.9 MODULO IV: REPORTE

En este modulo se encuentran todos los procesos que puede realizar un usuario que tenga acceso al modulo.

PANTALLA PRINCIPAL MODULO REPORTES

Esta pantalla muestra una vez verificados los datos en la base de datos del usuario que quiera ingresar al sistema y que quiera acceder al modulo reportes, la pantalla visualiza el menú correspondiente al modulo.

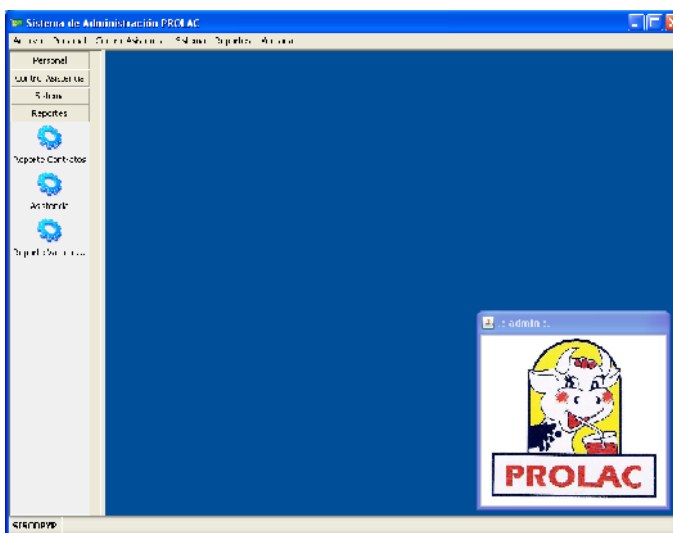


Figura N° 292: Pantalla Principal Modulo Reportes

II.1.4.3.4.9.1 PANTALLA GENERAR REPORTE DE CONTRATOS

Esta pantalla se muestra después de hacer click en el enlace Reporte Contratos, en esta se visualiza dos botones para seleccionar y generar el reporte del personal activo o inactivo de acuerdo a sus contratos.

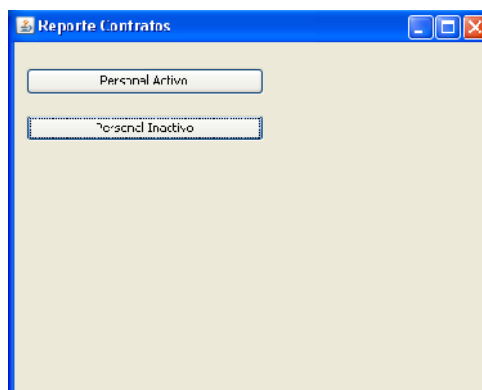


Figura N° 293: Pantalla Generar Reporte de Contratos

PANTALLA REPORTE DE PERSONAL ACTIVO/INACTIVO

Esta pantalla se muestra después de hacer click en el enlace Aceptar de la pantalla Generar Reporte de Asistencia, en esta se visualiza un reporte de datos de las horas trabajadas, de los retrasos, de las faltas y de las asistencias del empleado.

Nombre	CI	Desde	Hasta	Puesto	Sueldo
Area Administracion y Contabilidad					
Tchiquez Fernandez Patricia	5787463	19/07/2013	12/07/2014	Asistente de Ventas	2500.00
Total Administracion y Contabilidad:					1
Total					1

Figura N° 294: Pantalla Reporte de Personal Activo/Inactivo

II.1.4.3.4.9.2 PANTALLA GENERAR REPORTE DE ASISTENCIA

Esta pantalla se muestra después de hacer click en el enlace Reporte, en esta se visualiza un formulario para seleccionar datos para generar el reporte de asistencia, cuenta con campos como: Tipo de Reporte, Fecha Desde, Fecha Hasta, Área, Puesto y Seleccionar Empleados. En esta hay la opción de Generar Reporte.

Reportes de asistencia

Reporte: seleccione...

Desde: seleccione...

Reporte general de asistencia

altas

Retardos

Salidas antes de tiempo

44

45 7 8 9 10 11 12 13

46 14 15 16 17 18 19 20

47 21 22 23 24 25 26 27

48 28 29 30

Area: <Todos>

Puesto: <Todos>

No.	Nombres	Puesto
111	Victor Cardozo	Gerente General
112	Patricia Rodriguez Ferrandez	Secretaria 1
113	Juan Pabo Guerrero Vasquez	Jefe de Comercializacion y V...
114	Sonia Perez Aguirre	Jefe de Administracion y Con...
115	Cesar Mercado Paz	Asistente de Ventas
116	Carlos Sanchez Aguilar	Encargado de Mantenimiento

Crear reporte para

☐ Todos los empleados que aparecen en la lista superior

☒ Todos los empleados seleccionados

Aceptar Cancelar

Figura N° 295: Pantalla Generar Reporte de Asistencia

PANTALLA REPORTE DE ASISTENCIA

Esta pantalla se muestra después de hacer click en el enlace Aceptar de la pantalla Generar Reporte de Asistencia, en esta se visualiza un reporte de datos de las horas trabajadas, de los retrasos, de las faltas y de las asistencias del empleado.

Reporte Asistencia (del lun, 19-dic-2011 al lun, 19-dic-2011)

Empleado: Rodriguez Fernandez Patricia

Fecha	Entrada	Salida	Tiempo	Dinhs laborar	A favor	Retardo	Salida Pre.
lunes, 19 dic 2011		Falta		00:00			
Tiempo total laborado:		00:00	0.0%				

Tiempo	Sin tomar en cuenta faltas de asistencia	Tomando en cuenta faltas de asistencia
A laborar:	00:00	05:30
A Favor:	00:00	-06:30

Resumen del periodo

Días laborables asistidos: 0 de 1

Faltas de asistencia: 1

Detachados totales: 0

Salidas pendientes totales: 0

Firma del emisor: c

Figura N° 296: Pantalla Reporte de Asistencia

II.1.4.3.4.9.3 PANTALLA GENERAR REPORTE DE VACACIONES

Esta pantalla se muestra después de hacer click en el enlace Reporte Vacaciones, en esta se visualiza un botón para seleccionar y generar el reporte del personal en vacación.

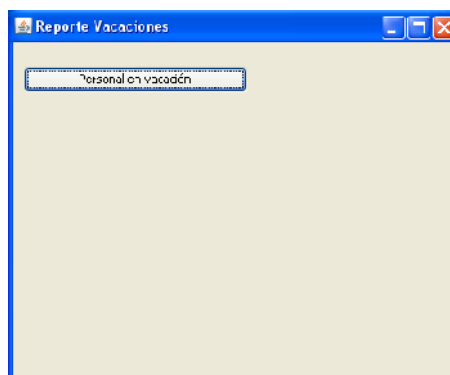


Figura N° 297: Pantalla Generar Reporte de Vacaciones

II.1.5 BASE DE DATOS



II.1.5.1 Modelado de Diagrama de Clases

II.1.5.1.1 Introducción

El Modelado de Diagrama de Clases es uno mas de los diagramas requeridos en la fase de Análisis/Diseño de la metodología que se esta implementando.

II.1.5.1.2 Propósito

- ✦ Comprender la estructura del sistema deseado para la organización.
- ✦ Identificar clases de análisis y diseño.

II.1.5.1.3 Alcance

- ✦ Describir las clases y objetos de diseño del sistema en su segunda iteración.
- ✦ Identificar y definir los objetos del sistema según los objetivos del sistema deseado aprobado por la organización.

II.1.5.2 Creación de la Base de Datos

```
CREATE DATABASE prolac
WITH OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'Spanish_Spain.1252'
LC_CTYPE = 'Spanish_Spain.1252'
CONNECTION LIMIT = -1;
```

Creación de las Tablas:

```
CREATE TABLE anticipo(
  id_anticipo serial NOT NULL,
  id_empleado integer,
  id_empleado_autoriza integer,
  fecha date,
  fecha_reg timestamp without time zone DEFAULT now(),
  monto double precision,
  descripcion character varying,
  id_usuario integer,
  CONSTRAINT anticipo_pkey PRIMARY KEY (id_anticipo),
  CONSTRAINT fk_anticipo_autoriza FOREIGN KEY (id_empleado_autoriza)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_anticipo_empleado FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_anticipo_usuario FOREIGN KEY (id_usuario)
    REFERENCES usuario (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

```
CREATE TABLE area(
  id      serial NOT NULL,
  descripcion varchar(50),
  CONSTRAINT area_pkey
    PRIMARY KEY (id)
) WITH (
  OIDS = FALSE
);
```

```
CREATETABLE area_puesto(
  id_area integer,
  id_puesto integer,
  id_area_puesto serial NOT NULL,
  CONSTRAINT area_puesto_pkey PRIMARY KEY (id_area_puesto),
  CONSTRAINT fk_area_puesto FOREIGN KEY (id_puesto)
    REFERENCES puesto (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_puesto_area FOREIGN KEY (id_area)
    REFERENCES area (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

```
CREATE TABLE bono_vacacion(
  id_bono_vacacion      integer      NOT      NULL      DEFAULT
  nextval('bono_antiguedad_id_bono_antiguedad_seq'::regclass),
  desde integer,
  hasta integer,
  valor double precision,
  tipo integer, -- 1=bono...
  id_usuario integer,
```

```

CONSTRAINT bono_antiguedad_pkey PRIMARY KEY (id_bono_vacacion),
CONSTRAINT fk_bono_usuario FOREIGN KEY (id_usuario)
    REFERENCES usuario (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE contrato(
    id_contrato serial NOT NULL,
    id_empleado integer,
    desde date,
    hasta date,
    observacion character varying,
    id_puesto integer,
    sueldo double precision,
    hasta_real date, -- representa la fecha fin del contrato real puede ser igual...
    estado integer DEFAULT 1, -- 1 = activo...
    id_area integer,
    CONSTRAINT contrato_pkey PRIMARY KEY (id_contrato),
    CONSTRAINT fk_contrato_area FOREIGN KEY (id_area)
        REFERENCES area (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk_contrato_empleado FOREIGN KEY (id_empleado)
        REFERENCES empleado (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk_contrato_puesto FOREIGN KEY (id_puesto)
        REFERENCES puesto (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE detalle_planilla_aguinaldo(
    id_detalle_planilla_aguinaldo serial NOT NULL,
    id_empleado integer,
    id_planilla_aguinaldo integer,
    monto double precision,
    meses_calculo integer,
    desde date,
    hasta date,
    CONSTRAINT detalle_planilla_aguinaldo_pkey PRIMARY KEY
(id_detalle_planilla_aguinaldo),
    CONSTRAINT fk_detalle_aguinaldo_empleado FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT fk_detalle_planilla FOREIGN KEY (id_planilla_aguinaldo)
    REFERENCES planilla_aguinaldo (id_planilla_aguinaldo) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE detalle_planilla_sueldo(
    id serial NOT NULL,
    id_empleado integer NOT NULL,
    horas_trabajadas double precision NOT NULL,
    bono_antiguedad double precision NOT NULL, -- 2 - 4 5%...
    total_ganado double precision NOT NULL,
    desc_ley_vejez double precision NOT NULL, -- 10%
    desc_ley_com double precision NOT NULL, -- 1.71 %
    desc_ley_comision double precision NOT NULL, -- 0.5 %
    total_desc_ley double precision NOT NULL,
    sueldo_netto double precision NOT NULL,
    otro_desc_rc_iva double precision NOT NULL,

```

```

otro_desc_anticipo double precision NOT NULL,
horas_trabajadas_real double precision NOT NULL,
otro_desc_asistencia double precision NOT NULL,
total_otros_desc double precision NOT NULL,
liquido_pagable double precision NOT NULL,
id_planilla integer NOT NULL,
id_puesto integer,
haber_basico double precision,
id_area integer,
CONSTRAINT detalle_planilla_sueldo_pkey PRIMARY KEY (id),
CONSTRAINT fk_detplasue_puesto FOREIGN KEY (id_puesto)
    REFERENCES puesto (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk_detplasuel_empleado FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk_detplasueldo_planilla FOREIGN KEY (id_planilla)
    REFERENCES planill_sueldo (id) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE dia_inhabil(
    id serial NOT NULL,
    fecha date,
    descripcion character varying(50),
    id_usuario integer,
    CONSTRAINT dia_inhabil_pkey PRIMARY KEY (id),
    CONSTRAINT fk_dia_usuario FOREIGN KEY (id_usuario)
        REFERENCES usuario (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION

```

);

```
CREATE TABLE empleado(
  id serial NOT NULL,
  nombre character varying(40) NOT NULL,
  ap character varying(30),
  am character varying(30),
  numero integer NOT NULL,
  foto character varying(20),
  direccion character varying(100),
  telefono integer,
  email character varying(30),
  celular integer,
  estado_civil character varying(20),
  fecha_nac date NOT NULL,
  sexo character varying(10) NOT NULL,
  estado character varying(20) NOT NULL,
  ci integer,
  clave character varying(50) NOT NULL,
  CONSTRAINT empleado_pkey PRIMARY KEY (id)
);
```

```
CREATE TABLE historial(
  id serial NOT NULL,
  id_empleado integer,
  hora timestamp without time zone,
  tipo integer,
  retardo_comida integer,
  retardo integer,
  CONSTRAINT historial_pkey PRIMARY KEY (id),
```

```

CONSTRAINT fk_historial_empleado FOREIGN KEY (id_empleado)
REFERENCES empleado (id) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE horarios(
  id_horario serial NOT NULL,
  entrada time without time zone NOT NULL,
  salida time without time zone NOT NULL,
  turno character varying,
  periodo integer,
  seleccionado boolean,
  tolerancia integer,
  id_usuario integer,
  CONSTRAINT horarios_pkey PRIMARY KEY (id_horario),
  CONSTRAINT fk_horarios_usuarios FOREIGN KEY (id_usuario)
  REFERENCES horarios (id_horario) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE modulo(
  idmod serial NOT NULL,
  nombre character varying(100) NOT NULL,
  codmod character varying(100),
  icono character varying(30),
  tipo_gui character varying(20),
  descripcion character varying(256),
  idmod_padre integer,
  CONSTRAINT modulo_pkey PRIMARY KEY (idmod),
  CONSTRAINT fk_modulo_modulo FOREIGN KEY (idmod_padre)

```

```
REFERENCES modulo (idmod) MATCH SIMPLE
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE movimiento(
  id serial NOT NULL,
  entrada timestamp without time zone,
  salida timestamp without time zone,
  id_empleado integer NOT NULL,
  id_horario integer,
  CONSTRAINT movimiento_pkey PRIMARY KEY (id),
  CONSTRAINT fk_mov_horario FOREIGN KEY (id_horario)
    REFERENCES horarios (id_horario) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT idempl FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

```
CREATE TABLE parametros(
  id_parametro serial NOT NULL,
  nombre character varying,
  valor double precision,
  id_usuario integer,
  CONSTRAINT parametros_pkey PRIMARY KEY (id_parametro),
  CONSTRAINT fk_parametros_usuario FOREIGN KEY (id_usuario)
    REFERENCES usuario (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);
```



```

CREATE TABLE permiso(
  id serial NOT NULL,
  id_emp_autoriza integer NOT NULL,
  id_empleado integer NOT NULL,
  fecha_ini date NOT NULL,
  fecha_fin date NOT NULL,
  motivo character varying(50) NOT NULL,
  vacacion character varying(20),
  llegar_tarde integer,
  salir_temprano integer,
  no_asistir_sin_sueldo integer,
  no_asistir_con_sueldo0 integer,
  sancion integer,
  vacaciones integer,
  incapacidad integer,
  fecha_fin_real date,
  dias integer,
  obs character varying,
  dias_real integer,
  CONSTRAINT permiso_pkey PRIMARY KEY (id),
  CONSTRAINT idempautoriza FOREIGN KEY (id_emp_autoriza)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT idemppermiso FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE planill_sueldo(
  id serial NOT NULL,
  mes integer NOT NULL,
  anio integer,
  id_usuario integer NOT NULL,
  id_empleado integer NOT NULL, -- responsable
  mes_str character varying(20) NOT NULL,
  CONSTRAINT pk_planill_sueldo PRIMARY KEY (id),
  CONSTRAINT fk_sueldo_empleado FOREIGN KEY (id_empleado)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_sueldo_usu FOREIGN KEY (id_usuario)
    REFERENCES usuario (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

```

CREATE TABLE planilla_aguinaldo(
  id_planilla_aguinaldo serial NOT NULL,
  anio integer,
  id_usuario integer NOT NULL,
  id_empleado_responsable integer NOT NULL,
  fecha_reg date NOT NULL DEFAULT ('now'::text)::date,
  obs character varying,
  CONSTRAINT pk_planilla_aguinaldo PRIMARY KEY (id_planilla_aguinaldo),
  CONSTRAINT fk_sueldo_empleado FOREIGN KEY (id_empleado_responsable)
    REFERENCES empleado (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_sueldo_usu FOREIGN KEY (id_usuario)
    REFERENCES usuario (id) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);

```

);

```
CREATE TABLE puesto(
    id serial NOT NULL,
    descripcion character varying(50),
    sueldo double precision,
    CONSTRAINT puesto_pkey PRIMARY KEY (id)
);
```

```
CREATE TABLE usuario(
    id serial NOT NULL,
    nombre character varying(10) NOT NULL,
    clave character varying(30) NOT NULL,
    CONSTRAINT usuario_pkey PRIMARY KEY (id)
);
```

```
CREATE TABLE usuario_modulo(
    id_usuario serial NOT NULL,
    idmod integer NOT NULL,
    CONSTRAINT usuario_modulo_pkey PRIMARY KEY (idmod, id_usuario),
    CONSTRAINT "Foreign_key01" FOREIGN KEY (id_usuario)
        REFERENCES usuario (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "Foreign_key02" FOREIGN KEY (idmod)
        REFERENCES modulo (idmod) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

II.1.5.3 Tablas: Descripción de Atributos.-

Tabla "ANTICIPO"

Atributo	Tipo	Descripción
ID_ANTICIPO	INTEGER	Not null clave primaria
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
ID_EMPLEADO_AUTORIZA	INTEGER	Not null clave foránea de la tabla empleado
FECHA	DATE	Fecha de registro
MONTO	DOUBLE PRECISION	Monto de anticipo
DESCRIPCION	VARCHAR(50)	Descripción del anticipo

Tabla N° 78: Descripción de Tabla Anticipo

Tabla "AREA"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
DESCRIPCION	VARCHAR(50)	Descripción o nombre de la área

Tabla N° 79: Descripción de Tabla Área

Tabla "AREA_PUESTO"

Atributo	Tipo	Descripción
ID_AREA_PUESTO	INTEGER	Not null clave primaria
ID_AREA	INTEGER	Not null clave foránea de la tabla Área

ID_ PUESTO	INTEGER	Not null clave foránea de la tabla Puesto

Tabla N° 80: Descripción de Tabla Área_puesto

Tabla "BONO_VACACION"

Atributo	Tipo	Descripción
ID_BONO_VACACION	INTEGER	Not null clave primaria
DESDE	INTEGER	Parámetro inicial
HASTA	INTEGER	Parámetro final
VALOR	DOUBLE PRECISION	Valor del parámetro
TIPO	INTEGER	Tipo

Tabla N° 81: Descripción de Tabla Bono_vacacion

Tabla "CONTRATO"

Atributo	Tipo	Descripción
ID_CONTRATO	INTEGER	Not null clave primaria
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla Empleado
DESDE	DATE	Fecha inicial del contrato
HASTA	DATE	Fecha final del contrato
OBSERVACION	VARCHAR	Descripción del contrato
ID_ PUESTO	INTEGER	Not null clave foránea de la tabla Puesto
SUELDO	DOUBLE PRECISION	Sueldo del empleado

ID_AREA	INTEGER	Not null clave foránea de la tabla Área
ESTADO	INTEGER	Estado del contrato

Tabla N° 82: Descripción de Tabla Contrato

Tabla "DETALLE_PLANILLA_SUELDO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
HORAS_TRABAJADAS	DOUBLE PRECISION	Horas de trabajo acumuladas del empleado
BONO_ANTIGUEDAD	DOUBLE PRECISION	Bono de acuerdo a su antigüedad que obtiene el empleado
TOTAL_GANADO	DOUBLE PRECISION	Monto que se paga al empleado
DESC_LEY_VEJEZ	DOUBLE PRECISION	Descuento que se realiza al empleado que es del 10%
DESC_LEY_COM	DOUBLE PRECISION	Descuento que se realiza al empleado que es del 1.71 %
DESC_LEY_COMISION	DOUBLE PRECISION	Descuento que se realiza al empleado que es del 0.5 %
TOTAL_DESC_LEY	DOUBLE PRECISION	Monto total de los

		descuentos
SUELDO_NETO	DOUBLE PRECISION	Monto para cancelar al empleado
OTRO_DESC_RC_IVA	DOUBLE PRECISION	Descuento que se realiza al empleado
OTRO_DESC_ANTICIPO	DOUBLE PRECISION	Descuento que se realiza al empleado
HORAS_TRABAJADAS_REAL	DOUBLE PRECISION	Horas de trabajo que se registran
OTRO_DESC_ASISTENCIA	DOUBLE PRECISION	Descuento que se realiza al empleado
TOTAL_OTROS_DESC	DOUBLE PRECISION	Monto total de otros descuentos
LIQUIDO_PAGABLE	DOUBLE PRECISION	Monto total a cancelar al empleado
ID_PLANILLA	INTEGER	Not null clave foránea de la tabla planill_sueldo
ID_PUESTO	INTEGER	Not null clave foránea de la tabla puesto
HABER_BASIC0	DOUBLE PRECISION	Sueldo base

Tabla N° 83: Descripción de Tabla Detalle_Planilla_Sueldo

Tabla "DETALLE_PLANILLA_AGUINALDO"

Atributo	Tipo	Descripción
ID_DETALLE_PLANILLA_AGUINALDO	INTEGER	Not null clave primaria
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
ID_PLANILLA_AGUINALDO	INTEGER	Not null clave foránea de la tabla Planilla Aguinaldo
MONTO	DOUBLE PRECISION	Monto del aguinaldo
MESES_CALCULO	INTEGER	Tiempo que trabaja el empleado
DESDE	DATE	Fecha inicio de la planilla
HASTA	DATE	Fecha fin de la planilla

Tabla N° 84: Descripción de Tabla Detalle_Planilla_Aguinaldo

Tabla "DIA_INHABIL"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
FECHA	DATE	Fecha del día inhábil
DESCRIPCION	VARCHAR(50)	Descripción o nombre del día inhábil

Tabla N° 85: Descripción de Tabla Dia_Inhabil

Tabla "EMPLEADO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
NOMBRE	VARCHAR(40)	Nombre del empleado
AP	VARCHAR(30)	Apellido paterno del empleado
AM	VARCHAR(30)	Apellido materno del empleado
NUMERO	INTEGER	Numero de registro del empleado
FOTO	VARCHAR(20)	Fotografía del empleado
DIRECCION	VARCHAR(100)	Domicilio del empleado
TELEFONO	INTEGER	Numero telefónico del empleado
EMAIL	VARCHAR(30)	Correo electrónico del empleado
CELULAR	INTEGER	Numero de celular del empleado
ESTADO_CIVIL	VARCHAR(20)	Estado civil del empleado
FECHA_NAC	DATE	Fecha de nacimiento del empleado
SEXO	VARCHAR(10)	Sexo del empleado
ESTADO	VARCHAR(20)	1 estado activo 0 estado inactivo

ID_PUESTO	INTEGER	Not null clave foránea de la tabla puesto
ID_AREA	INTEGER	Not null clave foránea de la tabla área
CI	INTEGER	C.I. del empleado
CLAVE	VARCHAR(50)	Clave para registrar la asistencia del empleado

Tabla N° 86: Descripción de Tabla Empleado

Tabla "HISTORIAL"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
HORA	TIMESTAMP	Hora de registro del empleado
TIPO	INTEGER	Tipo de registro
RETARDO_COMIDA	INTEGER	Retraso del empleado
RETARDO	INTEGER	Retraso del empleado

Tabla N° 87: Descripción de Tabla Historial

Tabla "HORARIO"

Atributo	Tipo	Descripción
ID_HORARIO	INTEGER	Not null clave primaria

ID_USUARIO	INTEGER	Not null clave foránea de la tabla usuario
ENTRADA	TIME	Hora de entrada al trabajo
SALIDA	TIME	Hora de salida del trabajo
TURNOS	VARCHAR	Periodo de Trabajo
PERIODO	INTEGER	Periodo de Trabajo
SELECCIONADO	BOOLEAN	Activo o inactivo
TOLERANCIA	TIME	Tiempo de tolerancia de ingreso

Tabla Nº 88: Descripción de Tabla Horario

Tabla "MODULO"

Atributo	Tipo	Descripción
IDMOD	INTEGER	Not null clave primaria
NOMBRE	VARCHAR(100)	Nombre del modulo
CODMOD	VARCHAR(100)	Código del modulo
ICONO	VARCHAR(30)	Icono del modulo
TIPO_GUI	VARCHAR(20)	Tipo del modulo
DESCRIPCION	VARCHAR(256)	Descripción del modulo
IDMOD_PADRE	INTEGER	Not null clave foránea de la tabla modulo

Tabla Nº 89: Descripción de Tabla Modulo

Tabla "MOVIMIENTO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
ENTRADA	TIMESTAMP	Registro de la hora de entrada del empleado
SALIDA	TIMESTAMP	Registro de la hora de salida del empleado
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado

Tabla N° 90: Descripción de Tabla Movimiento

Tabla "PARAMETRO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
NOMBRE	VARCHAR	Nombre del parámetro
VALOR	DOUBLE PRECISION	Valor del parámetro
ID_USUARIO	INTEGER	Not null clave foránea de la tabla usuario

Tabla N° 91: Descripción de Tabla Parámetro

Tabla "PERMISO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
ID_EMP_AUTORIZA	INTEGER	Not null clave foránea de la tabla empleado

ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
FECHA_INI	DATE	Fecha de inicio del permiso
FECHA_FIN	DATE	Fecha de fin del permiso
MOTIVO	VARCHAR(50)	Motivo o razón del permiso
VACACION	VARCHAR(20)	Si esta de vacación o no
LLEGAR_TARDE	INTEGER	1 Si permitido 0 No permitido
SALIR_TEMPRANO	INTEGER	1 Si permitido 0 No permitido
NO_ASISTIR_SIN_SUELDO	INTEGER	1 estado activo 0 estado inactivo
NO_ASISTIR_CON_SUELDO0	INTEGER	1 estado activo 0 estado inactivo
SANCION	INTEGER	1 estado activo 0 estado inactivo
VACACIONES	INTEGER	1 estado activo 0 estado inactivo
INCAPACIDAD	INTEGER	1 estado activo 0 estado inactivo

Tabla N° 92: Descripción de Tabla Permiso

Tabla "PLANILLA_SUELDO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
MES	INTEGER	Mes del año
ANIO	INTEGER	Numero del año
ID_USUARIO	INTEGER	Not null clave foránea de la tabla usuario
ID_EMPLEADO	INTEGER	Not null clave foránea de la tabla empleado
MES_STR	VARCHAR(20)	Mes generado

Tabla N° 93: Descripción de Tabla Planilla_Sueldo

Tabla "PLANILLA_AGUINALDO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
ANIO	INTEGER	Numero del año
ID_USUARIO	INTEGER	Not null clave foránea de la tabla usuario
ID_EMPLEADO_RESPONSABLE	INTEGER	Not null clave foránea de la tabla empleado
FECHA_REGISTRO	DATE	Fecha generado

Tabla N° 94: Descripción de Tabla Planilla_Aguinaldo

Tabla "PUESTO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
DESCRIPCION	VARCHAR(50)	Nombre o descripción del puesto
SUELDO	DOUBLE PRECISION	Sueldo del empleado

Tabla N° 95: Descripción de Tabla Puesto

Tabla "USUARIO"

Atributo	Tipo	Descripción
ID	INTEGER	Not null clave primaria
NOMBRE	VARCHAR(10)	Nombre del usuario
CLAVE	VARCHAR(30)	Clave del usuario

Tabla N° 96: Descripción de Tabla Usuario

Tabla "USUARIO_MODULO"

Atributo	Tipo	Descripción
ID_USUARIO	INTEGER	Not null clave foránea de la tabla usuario
IDMOD	INTEGER	Not null clave foránea de la tabla modulo

Tabla N° 97: Descripción de Tabla Usuario_Modulo

II.1.5.4 Modelado de Datos

II.1.5.4.1 Introducción

A continuación se describe la creación de la Base de Datos del software, apuntando su característica. Para expresar este modelo se utiliza un Diagrama de Clases y luego su correspondiente implementación en el lenguaje utilizado para el desarrollo del proyecto el cual es JAVA.

II.1.5.4.2 Especificación de Métodos

Clase Anticipo

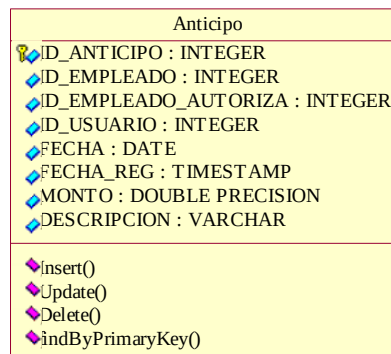


Figura N° 300: Especificación Método Anticipo

```

Public int insert(Anticipo anticipo ,Connection con)throws AnticipoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into ANTICIPO(
ID_EMPLEADO, ID_EMPLEADO_AUTORIZA, FECHA, FECHA_REG,
MONTO, DESCRIPCION, ID_USUAIRO) values (?, ?, ?, ?, ?, ?, ?)");
        ps.setInt(1,anticipo.getId_empleado());
        ps.setInt(2,anticipo.getId_empleado_autoriza());
        ps.setDate(3,anticipo.getFecha());
        ps.setTimestamp(4,anticipo.getFecha_reg());
        ps.setDouble(5,anticipo.getMonto());
    }
}
  
```



```

        ps.setString(6,anticipo.getDescripcion());
        ps.setInt(7,anticipo.getId_usuairo());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new AnticipoException(sqle);}
    catch(Exception e){throw new AnticipoException(e);}
}

```

Public int update(AnticipoPK anticipopk, Anticipo anticipo, Connection con)throws

AnticipoException{

```

    PreparedStatement ps = null;

```

```

    try

```

```

    {

```

```

        ps = con.prepareStatement("update ANTICIPO set
ID_EMPLEADO = ? , ID_EMPLEADO_AUTORIZA = ? , FECHA = ? ,
FECHA_REG = ? , MONTO = ? , DESCRIPCION = ? , ID_USUAIRO = ? where
id_anticipo = ?");

```

```

        ps.setInt(1,anticipo.getId_empleado());

```

```

        ps.setInt(2,anticipo.getId_empleado_autoriza());

```

```

        ps.setDate(3,anticipo.getFecha());

```

```

        ps.setTimestamp(4,anticipo.getFecha_reg());

```

```

        ps.setDouble(5,anticipo.getMonto());

```

```

        ps.setString(6,anticipo.getDescripcion());

```

```

        ps.setInt(7,anticipo.getId_usuairo());

```

```

        ps.setInt(8,anticipopk.getId_anticipo());

```

```

        return(ps.executeUpdate());

```

```

    }catch(SQLException sqle){throw new AnticipoException(sqle);}

```

```

    catch(Exception e){throw new AnticipoException(e);}

```

```

    }

```

Public int delete(AnticipoPK anticipopk , Connection con)throws

AnticipoException{

```

    PreparedStatement ps = null;

```

```

        try {
            ps = con.prepareStatement("delete from ANTICIPO where
id_anticipo = ?");
            ps.setInt(1, anticipopk.getId_anticipo());
            return(ps.executeUpdate());
        }catch(SQLException sqle) {throw new AnticipoException(sqle);}
        catch(Exception e) {throw new AnticipoException(e);}
    }

```

Public Anticipo findByPrimaryKey(int id_anticipo, Connection con) throws AnticipoException{

```

        PreparedStatement stmt = null;
        ResultSet rs = null;
        try {
            final String SQLSTATEMENT = "Select id_anticipo,
id_empleado, id_empleado_autoriza, fecha, fecha_reg, monto, descripcion,
id_usuario from anticipo where id_anticipo = ?";
            stmt=con.prepareStatement(SQLSTATEMENT);
            stmt.setInt(1, id_anticipo);
            rs = stmt.executeQuery();
            return fetchSingleResult(rs);
        }
        catch (SQLException sqle) {
            throw new AnticipoException(sqle);
        }
        catch(Exception e){throw new AnticipoException(e);}
        finally {}
    }

```

Clase Área

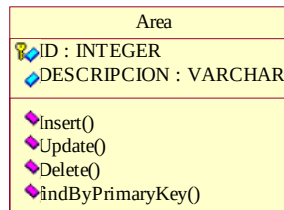


Figura N° 301: Especificación Método Área

```

public int insert(Area area ,Connection con)throws AreaException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into AREA( ID,
DESCRIPCION) values (?, ?)");
        ps.setInt(1,area.getId());
        ps.setString(2,area.getDescripcion());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new AreaException(sqle);}
    catch(Exception e){throw new AreaException(e);}
}

public int update(AreaPK areapk, Area area, Connection con)throws
AreaException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update AREA set DESCRIPCION
= ? where ID = ?");

        ps.setString(1,area.getDescripcion());
        ps.setInt(2,areapk.getId());
        return(ps.executeUpdate());
    }
}

```

```

        }catch(SQLException sqle){throw new AreaException(sqle);}
        catch(Exception e){throw new AreaException(e);}
    }

    public int delete(AreaPK areapk , Connection con)throws AreaException{
        PreparedStatement ps = null;
        try {
            ps = con.prepareStatement("delete from  AREA where ID =
?");

            ps.setInt(1, areapk.getId());
            return(ps.executeUpdate());
        }catch(SQLException sqle) {throw new AreaException(sqle);}
        catch(Exception e) {throw new AreaException(e);}
    }

    public Area findByPrimaryKey(int id, Connection con) throws AreaException{
        PreparedStatement stmt = null;
        ResultSet rs = null;
        try {
            final String SQLSTATEMENT = "Select ID, DESCRIPCION
from AREA where ID = ?";
            stmt=con.prepareStatement(SQLSTATEMENT);
            stmt.setInt(1, id);
            rs = stmt.executeQuery();
            return fetchSingleResult(rs);
        }
        catch (SQLException sqle) {
            throw new AreaException(sqle);
        }
        catch(Exception e){throw new AreaException(e);}
        finally {}
    }

```

Clase Área Puesto

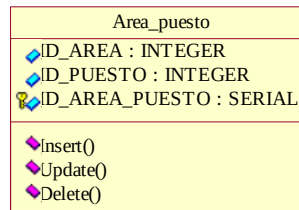


Figura N° 302: Especificación Método Área_Puesto

```

public int insert(Area_puesto area_puesto ,Connection con)throws
Area_puestoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into AREA_PUESTO(
ID_AREA, ID_PUESTO, ID_AREA_PUESTO) values (?, ?, ?)");
        ps.setInt(1,area_puesto.getId_area());
        ps.setInt(2,area_puesto.getId_puesto());
        ps.setInt(3,area_puesto.getId_area_puesto());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new Area_puestoException(sqle);}
    catch(Exception e){throw new Area_puestoException(e);}
}

public int update(Area_puestoPK area_puestopk, Area_puesto area_puesto,
Connection con)throws Area_puestoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update AREA_PUESTO set
ID_AREA = ? , ID_PUESTO = ? where id_area_puesto = ?");
        ps.setInt(1,area_puesto.getId_area());

```

```

        ps.setInt(2,area_puesto.getId_puesto());
        ps.setInt(3,area_puestopk.getId_area_puesto());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new Area_puestoException(sqle);}
    catch(Exception e){throw new Area_puestoException(e);}
}

public int delete(Area_puestoPK area_puestopk , Connection con)throws
Area_puestoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from AREA_PUESTO
where id_area_puesto = ?");
        ps.setInt(1, area_puestopk.getId_area_puesto());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new Area_puestoException(sqle);}
    catch(Exception e) {throw new Area_puestoException(e);}
}

```

Clase Bono_vacacion

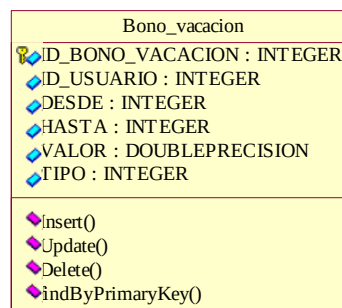


Figura N° 303: Especificación Método Bono_vacacion

```

public int insert(Bono_vacacion bono_vacacion ,Connection con)throws
Bono_vacacionException {

```

```

        PreparedStatement ps = null;
        try
        {
            ps = con.prepareStatement("insert into BONO_VACACION(
DESDE, HASTA, VALOR, TIPO, ID_USUARIO) values (?, ?, ?, ?, ?)");
            ps.setInt(1,bono_vacacion.getDesde());
            ps.setInt(2,bono_vacacion.getHasta());
            ps.setDouble(3,bono_vacacion.getValor());
            ps.setInt(4,bono_vacacion.getTipo());
            ps.setInt(5,bono_vacacion.getId_usuario());
            return(ps.executeUpdate());
        }catch(SQLException sqle){throw new
Bono_vacacionException(sqle);}
        catch(Exception e){throw new Bono_vacacionException(e);}
    }

    public int update(Bono_vacacionPK bono_vacacionpk, Bono_vacacion
bono_vacacion, Connection con)throws Bono_vacacionException{
        PreparedStatement ps = null;
        try
        {
            ps = con.prepareStatement("update BONO_VACACION set
DESDE = ? , HASTA = ? , VALOR = ? , TIPO = ? , ID_USUARIO = ? where
id_bono_vacacion = ?");

            ps.setInt(1,bono_vacacion.getDesde());
            ps.setInt(2,bono_vacacion.getHasta());
            ps.setDouble(3,bono_vacacion.getValor());
            ps.setInt(4,bono_vacacion.getTipo());
            ps.setInt(5,bono_vacacion.getId_usuario());
            ps.setInt(6,bono_vacacionpk.getId_bono_vacacion());
            return(ps.executeUpdate());
        }
    }

```

```

        }catch(SQLException sqle){throw new
Bono_vacacionException(sqle);}
        catch(Exception e){throw new Bono_vacacionException(e);}
    }
public int delete(Bono_vacacionPK bono_vacacionpk , Connection con)throws
Bono_vacacionException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from BONO_VACACION
where id_bono_vacacion = ?");
        ps.setInt(1, bono_vacacionpk.getId_bono_vacacion());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new
Bono_vacacionException(sqle);}
    catch(Exception e) {throw new Bono_vacacionException(e);}
}

public Bono_vacacion findByPrimaryKey(int id_bono_vacacion, Connection con)
throws Bono_vacacionException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select id_bono_vacacion,
desde, hasta, valor, tipo, id_usuario from bono_vacacion where id_bono_vacacion =
?";

        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id_bono_vacacion);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {

```



```

        throw new Bono_vacacionException(sql);
    }
    catch(Exception e){throw new Bono_vacacionException(e);}
    finally {}
}

```

Clase Contrato

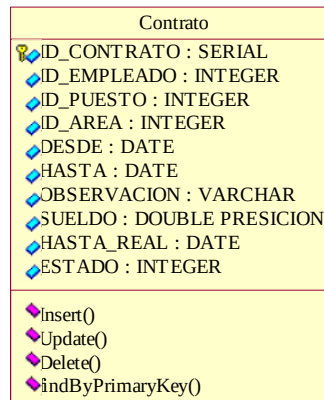


Figura N° 304: Especificación Método Contrato

```

public int insert(Contrato contrato ,Connection con)throws ContratoException {

```

```

    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into CONTRATO(
ID_EMPLEADO, DESDE, HASTA, OBSERVACION, ID_PUESTO, SUELDO,
HASTA_REAL, ESTADO, ID_AREA) values (?, ?, ?, ?, ?, ?, ?, ?, ?)");
        ps.setInt(1,contrato.getId_empleado());
        ps.setDate(2,contrato.getDesde());
        ps.setDate(3,contrato.getHasta());
        ps.setString(4,contrato.getObservacion());
        ps.setInt(5,contrato.getId_puesto());

```

```

        ps.setDouble(6,contrato.getSueldo());
        ps.setDate(7,contrato.getHasta_real());
        ps.setInt(8,contrato.getEstado());
        ps.setInt(9,contrato.getId_area());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ContratoException(sqle);}
    catch(Exception e){throw new ContratoException(e);}
}

public int update(ContratoPK contratopk, Contrato contrato, Connection con)throws
ContratoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update CONTRATO set
ID_EMPLEADO = ? , DESDE = ? , HASTA = ? , OBSERVACION = ? ,
ID_PUESTO = ? , SUELDO = ? , HASTA_REAL = ? , ESTADO = ? , ID_AREA =
? where id_contrato = ?");

        ps.setInt(1,contrato.getId_empleado());
        ps.setDate(2,contrato.getDesde());
        ps.setDate(3,contrato.getHasta());
        ps.setString(4,contrato.getObservacion());
        ps.setInt(5,contrato.getId_puesto());
        ps.setDouble(6,contrato.getSueldo());
        ps.setDate(7,contrato.getHasta_real());
        ps.setInt(8,contrato.getEstado());
        ps.setInt(9,contrato.getId_area());
        ps.setInt(10,contratopk.getId_contrato());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ContratoException(sqle);}
    catch(Exception e){throw new ContratoException(e);}
}

```

```

    }

    public int delete(ContratoPK contratopk , Connection con)throws
    ContratoException{
        PreparedStatement ps = null;
        try {
            ps = con.prepareStatement("delete from CONTRATO where
id_contrato = ?");
            ps.setInt(1, contratopk.getId_contrato());
            return(ps.executeUpdate());
        }catch(SQLException sqle) {throw new ContratoException(sqle);}
        catch(Exception e) {throw new ContratoException(e);}
    }

    public Contrato findByPrimaryKey(int id_contrato, Connection con) throws
    ContratoException{
        PreparedStatement stmt = null;
        ResultSet rs = null;
        try {
            final String SQLSTATEMENT = "Select id_contrato,
id_empleado, desde, hasta, observacion, id_puesto, sueldo, hasta_real, estado, id_area
from contrato where id_contrato = ?";
            stmt=con.prepareStatement(SQLSTATEMENT);
            stmt.setInt(1, id_contrato);
            rs = stmt.executeQuery();
            return fetchSingleResult(rs);
        }
        catch (SQLException sqle) {
            throw new ContratoException(sqle);
        }
        catch(Exception e){throw new ContratoException(e);}
        finally {}
    }

```

Clase Detalle Planilla Aguinaldo

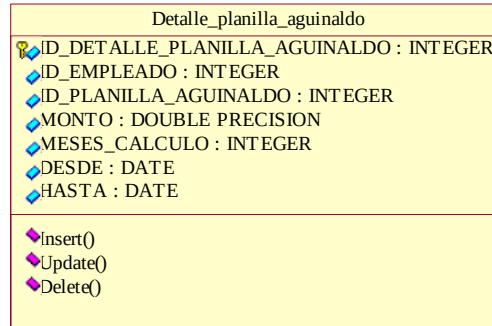


Figura N° 305: Especificación Método Detalle Planilla Aguinaldo

```

public int insert(Detalle_planilla_aguinaldo detalle_planilla_aguinaldo ,Connection
con)throws Detalle_planilla_aguinaldoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into
DETALLE_PLANILLA_AGUINALDO(
ID_EMPLEADO,
ID_PLANILLA_AGUINALDO, MONTO, MESES_CALCULO, DESDE, HASTA)
values (?, ?, ?, ?, ?, ?)");
        ps.setInt(1,detalle_planilla_aguinaldo.getId_empleado());
        ps.setInt(2,detalle_planilla_aguinaldo.getId_planilla_aguinaldo());
        ps.setDouble(3,detalle_planilla_aguinaldo.getMonto());
        ps.setInt(4,detalle_planilla_aguinaldo.getMeses_calculo());
        ps.setDate(5,detalle_planilla_aguinaldo.getDesde());
        ps.setDate(6,detalle_planilla_aguinaldo.getHasta());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Detalle_planilla_aguinaldoException(sqle);}
    catch(Exception e){throw new
Detalle_planilla_aguinaldoException(e);} }

```

```

public int update(Detalle_planilla_aguinaldoPK detalle_planilla_aguinaldopk,
Detalle_planilla_aguinaldo detalle_planilla_aguinaldo, Connection con)throws
Detalle_planilla_aguinaldoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update
DETALLE_PLANILLA_AGUINALDO set ID_EMPLEADO = ? ,
ID_PLANILLA_AGUINALDO = ? , MONTO = ? , MESES_CALCULO = ? ,
DESDE = ? , HASTA = ? where id_detalle_planilla_aguinaldo = ?");
        ps.setInt(1,detalle_planilla_aguinaldo.getId_empleado());
        ps.setInt(2,detalle_planilla_aguinaldo.getId_planilla_aguinaldo());
        ps.setDouble(3,detalle_planilla_aguinaldo.getMonto());
        ps.setInt(4,detalle_planilla_aguinaldo.getMeses_calculo());
        ps.setDate(5,detalle_planilla_aguinaldo.getDesde());
        ps.setDate(6,detalle_planilla_aguinaldo.getHasta());
        ps.setInt(7,detalle_planilla_aguinaldopk.getId_detalle_planilla_aguinaldo());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Detalle_planilla_aguinaldoException(sqle);}
    catch(Exception e){throw new
Detalle_planilla_aguinaldoException(e);}
}

public int delete(Detalle_planilla_aguinaldoPK detalle_planilla_aguinaldopk ,
Connection con)throws Detalle_planilla_aguinaldoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from
DETALLE_PLANILLA_AGUINALDO where id_detalle_planilla_aguinaldo = ?");

```

```

        ps.setInt(1,
detalle_planilla_aguinaldopk.getId_detalle_planilla_aguinaldo());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new
Detalle_planilla_aguinaldoException(sqle);}
    catch(Exception e) {throw new
Detalle_planilla_aguinaldoException(e);}
}

```

Clase Detalle Planilla Sueldo

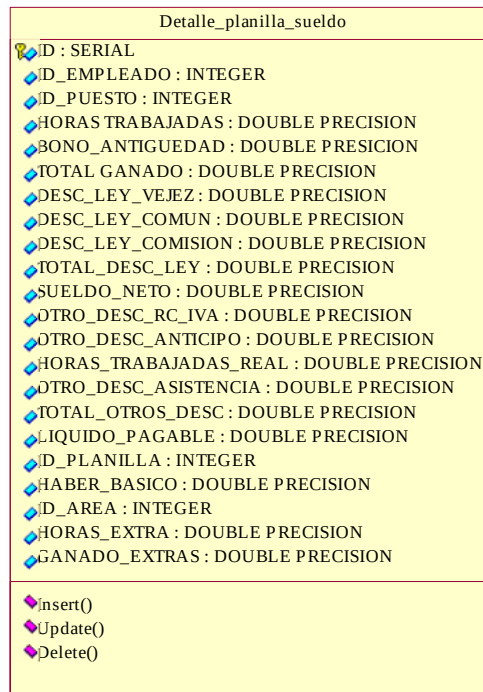


Figura N° 306: Especificación Método Detalle Planilla Sueldo

```

public int insert(Detalle_planilla_sueldo detalle_planilla_sueldo ,Connection
con)throws Detalle_planilla_sueldoException {
    PreparedStatement ps = null;
    try

```

```

{
    ps = con.prepareStatement("insert into
DETALLE_PLANILLA_SUELDO( ID_EMPLEADO, HORAS TRABAJADAS,
BONO_ANTIGUEDAD, TOTAL_GANADO, DESC_LEY_VEJEZ,
DESC_LEY_COM, DESC_LEY_COMISION, TOTAL_DESC_LEY,
SUELDO_NETO, OTRO_DESC_RC_IVA, OTRO_DESC_ANTICIPO,
HORAS TRABAJADAS_REAL, OTRO_DESC_ASISTENCIA,
TOTAL_OTROS_DESC, LIQUIDO_PAGABLE, ID_PLANILLA, ID_PUESTO,
HABER_BASICO, ID_AREA, HORAS_EXTRA, GANADO_EXTRAS) values (?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
    ps.setInt(1,detalle_planilla_sueldo.getId_empleado());
    ps.setDouble(2,detalle_planilla_sueldo.getHoras_trabajadas());
    ps.setDouble(3,detalle_planilla_sueldo.getBono_antiguedad());
    ps.setDouble(4,detalle_planilla_sueldo.getTotal_ganado());
    ps.setDouble(5,detalle_planilla_sueldo.getDesc_ley_vejez());
    ps.setDouble(6,detalle_planilla_sueldo.getDesc_ley_com());
    ps.setDouble(7,detalle_planilla_sueldo.getDesc_ley_comision());
    ps.setDouble(8,detalle_planilla_sueldo.getTotal_desc_ley());
    ps.setDouble(9,detalle_planilla_sueldo.getSueldo_neto());
    ps.setDouble(10,detalle_planilla_sueldo.getOtro_desc_rc_iva());
    ps.setDouble(11,detalle_planilla_sueldo.getOtro_desc_anticipo());
    ps.setDouble(12,detalle_planilla_sueldo.getHoras_trabajadas_real());
    ps.setDouble(13,detalle_planilla_sueldo.getOtro_desc_asistencia());
    ps.setDouble(14,detalle_planilla_sueldo.getTotal_otros_desc());
    ps.setDouble(15,detalle_planilla_sueldo.getLiquido_pagable());
    ps.setInt(16,detalle_planilla_sueldo.getId_planilla());
    ps.setInt(17,detalle_planilla_sueldo.getId_puesto());
    ps.setDouble(18,detalle_planilla_sueldo.getHaber_basico());
    ps.setInt(19,detalle_planilla_sueldo.getId_area());
    ps.setDouble(20,detalle_planilla_sueldo.getHoras_extra());

```

```

        ps.setDouble(21,detalle_planilla_sueldo.getGanado_extras());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Detalle_planilla_sueldoException(sqle);}
    catch(Exception e){throw new Detalle_planilla_sueldoException(e);}
}

public int update(Detalle_planilla_sueldoPK detalle_planilla_sueldopk,
Detalle_planilla_sueldo detalle_planilla_sueldo, Connection con)throws
Detalle_planilla_sueldoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update
DETALLE_PLANILLA_SUELDO set ID_EMPLEADO = ? ,
HORAS TRABAJADAS = ? , BONO_ANTIGUEDAD = ? , TOTAL_GANADO =
? , DESC_LEY_VEJEZ = ? , DESC_LEY_COM = ? , DESC_LEY_COMISION = ? ,
TOTAL_DESC_LEY = ? , SUELDO_NETO = ? , OTRO_DESC_RC_IVA = ? ,
OTRO_DESC_ANTICIPO = ? , HORAS TRABAJADAS_REAL = ? ,
OTRO_DESC_ASISTENCIA = ? , TOTAL_OTROS_DESC = ? ,
LIQUIDO_PAGABLE = ? , ID_PLANILLA = ? , ID_PUESTO = ? ,
HABER_BASIC0 = ? , ID_AREA = ? , HORAS_EXTRA = ? ,
GANADO_EXTRAS = ? where id = ?");
        ps.setInt(1,detalle_planilla_sueldo.getId_empleado());
        ps.setDouble(2,detalle_planilla_sueldo.getHoras_trabajadas());
        ps.setDouble(3,detalle_planilla_sueldo.getBono_antiguedad());
        ps.setDouble(4,detalle_planilla_sueldo.getTotal_ganado());
        ps.setDouble(5,detalle_planilla_sueldo.getDesc_ley_vejez());
        ps.setDouble(6,detalle_planilla_sueldo.getDesc_ley_com());
        ps.setDouble(7,detalle_planilla_sueldo.getDesc_ley_comision());
        ps.setDouble(8,detalle_planilla_sueldo.getTotal_desc_ley());
    }
}

```



```

ps.setDouble(9,detalle_planilla_sueldo.getSueldo_neto());
ps.setDouble(10,detalle_planilla_sueldo.getOtro_desc_rc_iva());
ps.setDouble(11,detalle_planilla_sueldo.getOtro_desc_anticipo());
ps.setDouble(12,detalle_planilla_sueldo.getHoras_trabajadas_real());
ps.setDouble(13,detalle_planilla_sueldo.getOtro_desc_asistencia());
ps.setDouble(14,detalle_planilla_sueldo.getTotal_otros_desc());
ps.setDouble(15,detalle_planilla_sueldo.getLiquido_pagable());
ps.setInt(16,detalle_planilla_sueldo.getId_planilla());
ps.setInt(17,detalle_planilla_sueldo.getId_puesto());
ps.setDouble(18,detalle_planilla_sueldo.getHaber_basico());
ps.setInt(19,detalle_planilla_sueldo.getId_area());
ps.setDouble(20,detalle_planilla_sueldo.getHoras_extra());
ps.setDouble(21,detalle_planilla_sueldo.getGanado_extras());
ps.setInt(22,detalle_planilla_sueldopk.getId());

return(ps.executeUpdate());
} catch(SQLException sqle){throw new
Detalle_planilla_sueldoException(sqle);}

catch(Exception e){throw new Detalle_planilla_sueldoException(e);}
}

public int delete(Detalle_planilla_sueldoPK detalle_planilla_sueldopk , Connection
con)throws Detalle_planilla_sueldoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from
DETALLE_PLANILLA_SUELDO where id = ?");
        ps.setInt(1, detalle_planilla_sueldopk.getId());
        return(ps.executeUpdate());
    } catch(SQLException sqle) {throw new
Detalle_planilla_sueldoException(sqle);}

    catch(Exception e) {throw new Detalle_planilla_sueldoException(e);} }

```

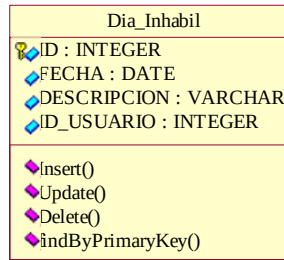
Clase Dia_inhabil

Figura N° 307: Especificación Método Dia_inhabil

```

public int insert(Dia_inhabil dia_inhabil ,Connection con)throws
Dia_inhabilException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into DIA_INHABIL( ID,
FECHA, DESCRIPCION) values (?, ?, ?)");
        ps.setInt(1,dia_inhabil.getId());
        ps.setDate(2,dia_inhabil.getFecha());
        ps.setString(3,dia_inhabil.getDescripcion());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new Dia_inhabilException(sqle);}
    catch(Exception e){throw new Dia_inhabilException(e);}
}

public int update(Dia_inhabilPK dia_inhabilpk, Dia_inhabil dia_inhabil, Connection
con)throws Dia_inhabilException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update DIA_INHABIL set FECHA
= ? , DESCRIPCION = ? where ID = ?");

```

```

        ps.setDate(1,dia_inhabil.getFecha());
        ps.setString(2,dia_inhabil.getDescripcion());
        ps.setInt(3,dia_inhabilpk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new Dia_inhabilException(sqle);}
    catch(Exception e){throw new Dia_inhabilException(e);}
}

public int delete(Dia_inhabilPK dia_inhabilpk , Connection con)throws
Dia_inhabilException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from DIA_INHABIL where ID = ?");
        ps.setInt(1, dia_inhabilpk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new Dia_inhabilException(sqle);}
    catch(Exception e) {throw new Dia_inhabilException(e);}
}

public Dia_inhabil findByPrimaryKey(int id, Connection con) throws
Dia_inhabilException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select ID, FECHA,
DESCRIPCION from DIA_INHABIL where ID = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {

```

```

        throw new Dia_inhabilException(sqle);
    }
    catch(Exception e){throw new Dia_inhabilException(e);}
    finally {}
}

```

Clase Empleado

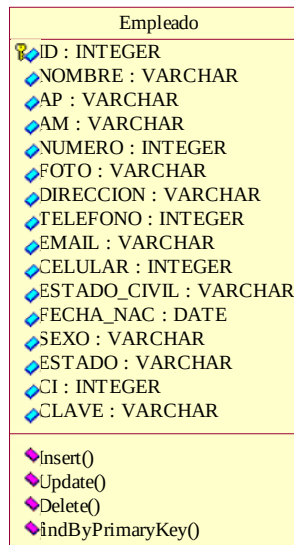


Figura N° 308: Especificación Método Empleado

```

public int insert(Empleado empleado ,Connection con)throws EmpleadoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into EMPLEADO(
        NOMBRE, AP, AM, NUMERO, FOTO, DIRECCION, TELEFONO, EMAIL,
        CELULAR, ESTADO_CIVIL, FECHA_NAC, SEXO, CI, CLAVE, ESTADO)
        values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

        ps.setString(1,empleado.getNombre());
        ps.setString(2,empleado.getAp());
    }
}

```

```

        ps.setString(3,empleado.getAm());
        ps.setInt(4,empleado.getNumero());
        ps.setString(5,empleado.getFoto());
        ps.setString(6,empleado.getDireccion());
        ps.setInt(7,empleado.getTelefono());
        ps.setString(8,empleado.getEmail());
        ps.setInt(9,empleado.getCelular());
        ps.setString(10,empleado.getEstado_civil());
        ps.setDate(11,empleado.getFecha_nac());
        ps.setString(12,empleado.getSexo());
        ps.setInt(13,empleado.getCi());
        ps.setString(14,empleado.getClave());
        ps.setInt(15,empleado.getEstado());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new EmpleadoException(sqle);}
    catch(Exception e){throw new EmpleadoException(e);}
}

public int update(EmpleadoPK empleadopk, Empleado empleado, Connection
con)throws EmpleadoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update EMPLEADO set
NOMBRE = ? , AP = ? , AM = ? , NUMERO = ? , FOTO = ? , DIRECCION = ? ,
TELEFONO = ? , EMAIL = ? , CELULAR = ? , ESTADO_CIVIL = ? ,
FECHA_NAC = ? , SEXO = ? , CI = ? , CLAVE = ? , ESTADO = ? where id = ?");
        ps.setString(1,empleado.getNombre());
        ps.setString(2,empleado.getAp());
        ps.setString(3,empleado.getAm());
        ps.setInt(4,empleado.getNumero());

```

```

        ps.setString(5,empleado.getFoto());
        ps.setString(6,empleado.getDireccion());
        ps.setInt(7,empleado.getTelefono());
        ps.setString(8,empleado.getEmail());
        ps.setInt(9,empleado.getCelular());
        ps.setString(10,empleado.getEstado_civil());
        ps.setDate(11,empleado.getFecha_nac());
        ps.setString(12,empleado.getSexo());
        ps.setInt(13,empleado.getCi());
        ps.setString(14,empleado.getClave());
        ps.setInt(15,empleado.getEstado());
        ps.setInt(16,empleadopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new EmpleadoException(sqle);}
    catch(Exception e){throw new EmpleadoException(e);}
}

public int delete(EmpleadoPK empleadopk , Connection con)throws
EmpleadoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from EMPLEADO where id = ?");
        ps.setInt(1, empleadopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new EmpleadoException(sqle);}
    catch(Exception e) {throw new EmpleadoException(e);}
}

public Empleado findByPrimaryKey(int id, Connection con) throws
EmpleadoException{
    PreparedStatement stmt = null;
    ResultSet rs = null;

```

```

try {
    final String SQLSTATEMENT = "Select id, nombre, ap, am,
numero, foto, direccion, telefono, email, celular, estado_civil, fecha_nac, sexo, ci,
clave, estado from empleado where id = ?";

    stmt=con.prepareStatement(SQLSTATEMENT);
    stmt.setInt(1, id);
    rs = stmt.executeQuery();
    return fetchSingleResult(rs);
}
catch (SQLException sqle) {
    throw new EmpleadoException(sqle);
}
catch(Exception e){throw new EmpleadoException(e);}
finally {}
}

```

Clase Historial

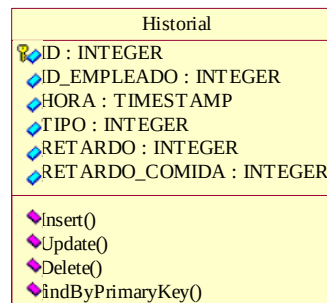


Figura N° 309: Especificación Método Historial

```

public int insert(Historial historial ,Connection con)throws HistorialException {
    PreparedStatement ps = null;
    try
    {

```

```

        ps = con.prepareStatement("insert into HISTORIAL( ID,
ID_EMPLEADO, HORA, TIPO, RETARDO_COMIDA, RETARDO) values (?, ?, ?,
?, ?, ?)");

```

```

        ps.setInt(1,historial.getId());
        ps.setInt(2,historial.getId_empleado());
        ps.setTimestamp(3,historial.getHora());
        ps.setInt(4,historial.getTipo());
        ps.setInt(5,historial.getRetardo_comida());
        ps.setInt(6,historial.getRetardo());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new HistorialException(sqle);}
    catch(Exception e){throw new HistorialException(e);}
}

```

public int update(HistorialPK historialpk, Historial historial, Connection con)throws HistorialException{

```

    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update HISTORIAL set
ID_EMPLEADO = ? , HORA = ? , TIPO = ? , RETARDO_COMIDA = ? ,
RETARDO = ? where ID = ?");

        ps.setInt(1,historial.getId_empleado());
        ps.setTimestamp(2,historial.getHora());
        ps.setInt(3,historial.getTipo());
        ps.setInt(4,historial.getRetardo_comida());
        ps.setInt(5,historial.getRetardo());
        ps.setInt(6,historialpk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new HistorialException(sqle);}
    catch(Exception e){throw new HistorialException(e);}    }

```



```

public int delete(HistorialPK historialpk , Connection con) throws
HistorialException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from HISTORIAL where ID = ?");
        ps.setInt(1, historialpk.getId());
        return(ps.executeUpdate());
    } catch(SQLException sqle) {throw new HistorialException(sqle);}
    catch(Exception e) {throw new HistorialException(e);}
}

public Historial findByPrimaryKey(int id, Connection con) throws
HistorialException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select ID, ID_EMPLEADO,
HORA, TIPO, RETARDO_COMIDA, RETARDO from HISTORIAL where ID =
?";

        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new HistorialException(sqle);
    }
    catch(Exception e){throw new HistorialException(e);}
    finally {}
}

```

Clase Horario

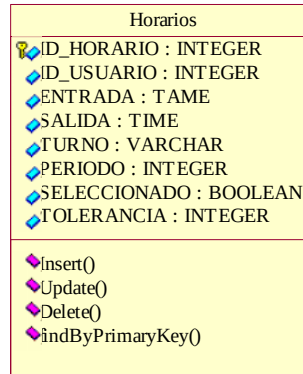


Figura N° 310: Especificación Método Horario

```

public int insert(Horarios horarios ,Connection con)throws HorariosException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into HORARIOS(
ENTRADA, SALIDA, TURNO, PERIODO, SELECCIONADO, TOLERANCIA,
ID_USUARIO) values (?, ?, ?, ?, ?, ?, ?, ?)");

        ps.setTime(1,horarios.getEntrada());
        ps.setTime(2,horarios.getSalida());
        ps.setString(3,horarios.getTurno());
        ps.setInt(4,horarios.getPeriodo());
        ps.setBoolean(5,horarios.getSeleccionado());
        ps.setInt(6,horarios.getTolerancia());
        ps.setInt(7,horarios.getId_usuario());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new HorariosException(sqle);}
    catch(Exception e){throw new HorariosException(e);}
}
  
```

```
public int update(HorariosPK horariospk, Horarios horarios, Connection con)throws
HorariosException{
```

```
    PreparedStatement ps = null;
```

```
    try
```

```
    {
```

```
        ps = con.prepareStatement("update HORARIOS set
ENTRADA = ? , SALIDA = ? , TURNO = ? , PERIODO = ? , SELECCIONADO = ?
, TOLERANCIA = ? , ID_USUARIO = ? where id_horario = ?");
```

```
        ps.setTime(1,horarios.getEntrada());
```

```
        ps.setTime(2,horarios.getSalida());
```

```
        ps.setString(3,horarios.getTurno());
```

```
        ps.setInt(4,horarios.getPeriodo());
```

```
        ps.setBoolean(5,horarios.getSeleccionado());
```

```
        ps.setInt(6,horarios.getTolerancia());
```

```
        ps.setInt(7,horarios.getId_usuario());
```

```
        ps.setInt(8,horariospk.getId_horario());
```

```
        return(ps.executeUpdate());
```

```
    }catch(SQLException sqle){throw new HorariosException(sqle);}
    catch(Exception e){throw new HorariosException(e);}
}
```

```
public int delete(HorariosPK horariospk , Connection con)throws
HorariosException{
```

```
    PreparedStatement ps = null;
```

```
    try {
```

```
        ps = con.prepareStatement("delete from HORARIOS where id_horario = ?");
```

```
        ps.setInt(1, horariospk.getId_horario());
```

```
        return(ps.executeUpdate());
```

```
    }catch(SQLException sqle) {throw new HorariosException(sqle);}
    catch(Exception e) {throw new HorariosException(e);}
}
```

```
}
```

```

public Horarios findByPrimaryKey(int id_horario, Connection con) throws
HorariosException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select id_horario, entrada,
salida, turno, periodo, seleccionado, tolerancia, id_usuario from horarios where
id_horario = ?";

        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id_horario);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new HorariosException(sqle);
    }
    catch(Exception e){throw new HorariosException(e);}
    finally {}
}

```

Clase Modulo

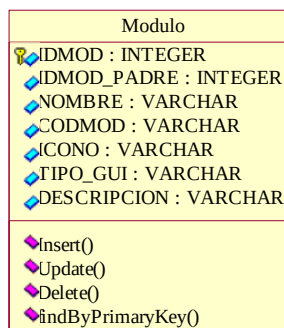


Figura N° 311: Especificación Método Modulo

```

public int insert(Modulo modulo ,Connection con)throws ModuloException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into MODULO( IDMOD,
NOMBRE, CODMOD, ICONO, TIPO_GUI, DESCRIPCION, IDMOD_PADRE)
values (?, ?, ?, ?, ?, ?, ?)");

        ps.setInt(1,module.getIdmod());
        ps.setString(2,module.getNombre());
        ps.setString(3,module.getCodmod());
        ps.setString(4,module.getIcono());
        ps.setString(5,module.getTipo_gui());
        ps.setString(6,module.getDescripcion());
        ps.setInt(7,module.getIdmod_padre());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ModuloException(sqle);}
    catch(Exception e){throw new ModuloException(e);}
}

```

```

public int update(ModuloPK modulopk, Modulo modulo, Connection con)throws
ModuloException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update MODULO set NOMBRE =
? , CODMOD = ? , ICONO = ? , TIPO_GUI = ? , DESCRIPCION = ? ,
IDMOD_PADRE = ? where IDMOD = ?");

        ps.setString(1,module.getNombre());
        ps.setString(2,module.getCodmod());
        ps.setString(3,module.getIcono());
        ps.setString(4,module.getTipo_gui());

```

```

        ps.setString(5,module.getDescripcion());
        ps.setInt(6,module.getIdmod_padre());
        ps.setInt(7,modulepk.getIdmod());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ModuloException(sqle);}
    catch(Exception e){throw new ModuloException(e);}
}

public int delete(ModuloPK modulepk , Connection con)throws ModuloException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from MODULO where IDMOD = ?");
        ps.setInt(1, modulepk.getIdmod());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new ModuloException(sqle);}
    catch(Exception e) {throw new ModuloException(e);}
}

public Modulo findByPrimaryKey(int idmod, Connection con) throws
ModuloException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select IDMOD, NOMBRE,
CODMOD, ICONO, TIPO_GUI, DESCRIPCION, IDMOD_PADRE from
MODULO where IDMOD = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, idmod);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {

```

```

        throw new ModuloException(sqle);
    }
    catch(Exception e){throw new ModuloException(e);}
    finally {}
}

```

Clase Movimiento

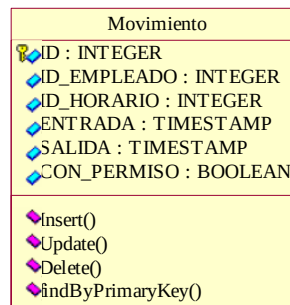


Figura N° 312: Especificación Método Movimiento

```

public int insert(Movimiento movimiento ,Connection con)throws
MovimientoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into MOVIMIENTO(ID,
ENTRADA, SALIDA, ID_EMPLEADO, ID_HORARIO, CON_PERMISO) values
(?, ?, ?, ?, ?, ?)");

        ps.setInt(1, movimiento.getId());
        ps.setTimestamp(2,movimiento.getEntrada());
        ps.setTimestamp(3,movimiento.getSalida());
        ps.setInt(4,movimiento.getId_empleado());
        ps.setInt(5,movimiento.getId_horario());
        if(movimiento.getId_horario() == 0)

```

```

        ps.setNull(5, java.sql.Types.NULL);
        ps.setBoolean(6,movimiento.getCon_permiso());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new MovimientoException(sqle);}
    catch(Exception e){throw new MovimientoException(e);}
}

public int update(MovimientoPK movimientopk, Movimiento movimiento,
Connection con)throws MovimientoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update MOVIMIENTO set
ENTRADA = ? , SALIDA = ? , ID_EMPLEADO = ? , ID_HORARIO = ? ,
CON_PERMISO = ? where id = ?");

        ps.setTimestamp(1,movimiento.getEntrada());
        ps.setTimestamp(2,movimiento.getSalida());
        ps.setInt(3,movimiento.getId_empleado());
        ps.setInt(4,movimiento.getId_horario());

        if(movimiento.getId_horario() == 0)
            ps.setNull(4, java.sql.Types.NULL);

        ps.setBoolean(5,movimiento.getCon_permiso());
        ps.setInt(6,movimientopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new MovimientoException(sqle);}
    catch(Exception e){throw new MovimientoException(e);}
}

public int delete(MovimientoPK movimientopk , Connection con)throws
MovimientoException{
    PreparedStatement ps = null;
    try {

```



```

        ps = con.prepareStatement("delete from MOVIMIENTO where id = ?");
        ps.setInt(1, movimientopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new MovimientoException(sqle);}
    catch(Exception e) {throw new MovimientoException(e);}
}

public Movimiento findByPrimaryKey(int id, Connection con) throws
MovimientoException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select id, entrada, salida,
id_empleado, id_horario, con_permiso from movimiento where id = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new MovimientoException(sqle);
    }
    catch(Exception e){throw new MovimientoException(e);}
    finally {}
}

```

Clase Parámetro

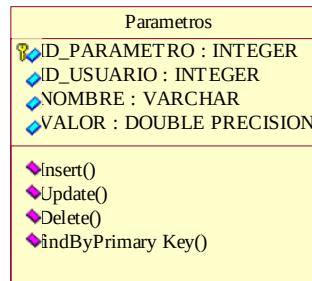


Figura N° 313: Especificación Método Parámetro

```

public int insert(Parametros parametros ,Connection con)throws
ParametrosException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into PARAMETROS(
NOMBRE, VALOR, ID_USUARIO) values (?, ?, ?)");
        ps.setString(1,parametros.getNombre());
        ps.setDouble(2,parametros.getValor());
        ps.setInt(3,parametros.getId_usuario());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ParametrosException(sqle);}
    catch(Exception e){throw new ParametrosException(e);}
}

public int update(ParametrosPK parametrospk, Parametros parametros, Connection
con)throws ParametrosException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update PARAMETROS set
NOMBRE = ? , VALOR = ? , ID_USUARIO = ? where id_parametro = ?");
  
```

```

        ps.setString(1,parametros.getNombre());
        ps.setDouble(2,parametros.getValor());
        ps.setInt(3,parametros.getId_usuario());
        ps.setInt(4,parametrospk.getId_parametro());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new ParametrosException(sqle);}
    catch(Exception e){throw new ParametrosException(e);}
}

public int delete(ParametrosPK parametrospk , Connection con)throws
ParametrosException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from PARAMETROS
where id_parametro = ?");
        ps.setInt(1, parametrospk.getId_parametro());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new ParametrosException(sqle);}
    catch(Exception e) {throw new ParametrosException(e);}
}

public Parametros findByPrimaryKey(int id_parametro, Connection con) throws
ParametrosException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select id_parametro,
nombre, valor, id_usuario from parametros where id_parametro = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id_parametro);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
}

```

```

    }
    catch (SQLException sqle) {
        throw new ParametrosException(sqle);
    }
    catch(Exception e){throw new ParametrosException(e);}
    finally {}
}

```

Clase Permiso

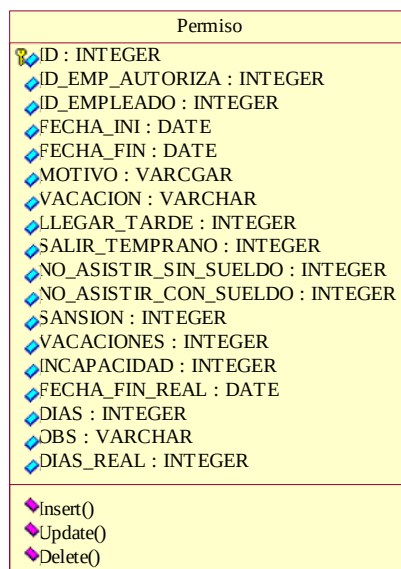


Figura N° 314: Especificación Método Permiso

```

public int insert(Permiso permiso ,Connection con)throws PermisoException {
    PreparedStatement ps = null;
    try
    {
        ps    =    con.prepareStatement("insert    into    PERMISO(
ID_EMP_AUTORIZA, ID_EMPLEADO, FECHA_INI, FECHA_FIN, MOTIVO,
VACACION,          LLEGAR_TARDE,          SALIR_TEMPRANO,

```

NO_ASISTIR_SIN_SUELDO, NO_ASISTIR_CON_SUELDO0, SANCION,
 VACACIONES, INCAPACIDAD, FECHA_FIN_REAL, DIAS, OBS, DIAS_REAL)
 values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?));

```

        ps.setInt(1,permiso.getId_emp_autoriza());
        ps.setInt(2,permiso.getId_empleado());
        ps.setDate(3,permiso.getFecha_ini());
        ps.setDate(4,permiso.getFecha_fin());
        ps.setString(5,permiso.getMotivo());
        ps.setString(6,permiso.getVacacion());
        ps.setInt(7,permiso.getLlegar_tarde());
        ps.setInt(8,permiso.getSalir_temprano());
        ps.setInt(9,permiso.getNo_asistir_sin_sueldo());
        ps.setInt(10,permiso.getNo_asistir_con_sueldo0());
        ps.setInt(11,permiso.getSancion());
        ps.setInt(12,permiso.getVacaciones());
        ps.setInt(13,permiso.getIncapacidad());
        ps.setDate(14,permiso.getFecha_fin_real());
        ps.setInt(15,permiso.getDias());
        ps.setString(16,permiso.getObs());
        ps.setInt(17,permiso.getDias_real());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new PermisoException(sqle);}
    catch(Exception e){throw new PermisoException(e);}
}

```

```

public int update(PermisoPK permisopk, Permiso permiso, Connection con)throws
PermisoException{
    PreparedStatement ps = null;
    try
    {

```

```

        ps = con.prepareStatement("update PERMISO set
ID_EMP_AUTORIZA = ? , ID_EMPLEADO = ? , FECHA_INI = ? , FECHA_FIN =
? , MOTIVO = ? , VACACION = ? , LLEGAR_TARDE = ? , SALIR_TEMPRANO
= ? , NO_ASISTIR_SIN_SUELDO = ? , NO_ASISTIR_CON_SUELDO0 = ? ,
SANCION = ? , VACACIONES = ? , INCAPACIDAD = ? , FECHA_FIN_REAL =
? , DIAS = ? , OBS = ? , DIAS_REAL = ? where id = ?");

```

```

        ps.setInt(1,permiso.getId_emp_autoriza());
        ps.setInt(2,permiso.getId_empleado());
        ps.setDate(3,permiso.getFecha_ini());
        ps.setDate(4,permiso.getFecha_fin());
        ps.setString(5,permiso.getMotivo());
        ps.setString(6,permiso.getVacacion());
        ps.setInt(7,permiso.getLlegar_tarde());
        ps.setInt(8,permiso.getSalir_temprano());
        ps.setInt(9,permiso.getNo_asistir_sin_sueldo());
        ps.setInt(10,permiso.getNo_asistir_con_sueldo0());
        ps.setInt(11,permiso.getSancion());
        ps.setInt(12,permiso.getVacaciones());
        ps.setInt(13,permiso.getIncapacidad());
        ps.setDate(14,permiso.getFecha_fin_real());
        ps.setInt(15,permiso.getDias());
        ps.setString(16,permiso.getObs());
        ps.setInt(17,permiso.getDias_real());
        ps.setInt(18,permisopk.getId());
        return(ps.executeUpdate());

```

```

    }catch(SQLException sqle){throw new PermisoException(sqle);}
    catch(Exception e){throw new PermisoException(e);}

```

```

    }

```

```

public int delete(PermisoPK permisopk , Connection con)throws PermisoException{
    PreparedStatement ps = null;

```

```

try {
    ps = con.prepareStatement("delete from PERMISO where id =
?");

    ps.setInt(1, permisopk.getId());
    return(ps.executeUpdate());
} catch(SQLException sqle) {throw new PermisoException(sqle);}
catch(Exception e) {throw new PermisoException(e);}
}

```

Clase Planilla_aguinaldo

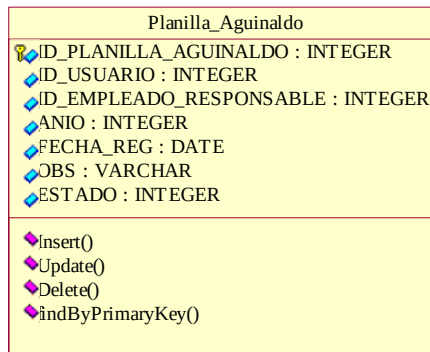


Figura N° 315: Especificación Método Planilla_Aguinaldo

```

public int insert(Planilla_aguinaldo planilla_aguinaldo ,Connection con)throws
Planilla_aguinaldoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into
PLANILLA_AGUINALDO(
ANIO, ID_USUARIO,
ID_EMPLEADO_RESPONSABLE, FECHA_REG, OBS, ESTADO) values (?, ?, ?,
?, ?, ?)");

        ps.setInt(1,planilla_aguinaldo.getAnio());
        ps.setInt(2,planilla_aguinaldo.getId_usuario());

```

```

        ps.setInt(3,planilla_aguinaldo.getId_empleado_responsable());
        ps.setDate(4,planilla_aguinaldo.getFecha_reg());
        ps.setString(5,planilla_aguinaldo.getObs());
        ps.setInt(6,planilla_aguinaldo.getEstado());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Planilla_aguinaldoException(sqle);}
    catch(Exception e){throw new Planilla_aguinaldoException(e);}
}

public int update(Planilla_aguinaldoPK planilla_aguinaldopk, Planilla_aguinaldo
planilla_aguinaldo, Connection con)throws Planilla_aguinaldoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update
PLANILLA_AGUINALDO set ANIO = ? , ID_USUARIO = ? ,
ID_EMPLEADO_RESPONSABLE = ? , FECHA_REG = ? , OBS = ? , ESTADO = ?
where id_planilla_aguinaldo = ?");
        ps.setInt(1,planilla_aguinaldo.getAnio());
        ps.setInt(2,planilla_aguinaldo.getId_usuario());
        ps.setInt(3,planilla_aguinaldo.getId_empleado_responsable());
        ps.setDate(4,planilla_aguinaldo.getFecha_reg());
        ps.setString(5,planilla_aguinaldo.getObs());
        ps.setInt(6,planilla_aguinaldo.getEstado());
        ps.setInt(7,planilla_aguinaldopk.getId_planilla_aguinaldo());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Planilla_aguinaldoException(sqle);}
    catch(Exception e){throw new Planilla_aguinaldoException(e);}
}

```



```

public int delete(Planilla_aguinaldoPK planilla_aguinaldopk , Connection
con)throws Planilla_aguinaldoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from
PLANILLA_AGUINALDO where id_planilla_aguinaldo = ?");
        ps.setInt(1, planilla_aguinaldopk.getId_planilla_aguinaldo());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new
Planilla_aguinaldoException(sqle);}
    catch(Exception e) {throw new Planilla_aguinaldoException(e);}
}

public Planilla_aguinaldo findByPrimaryKey(int id_planilla_aguinaldo,
Connection con) throws Planilla_aguinaldoException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select
id_planilla_aguinaldo, anio, id_usuario, id_empleado_responsable, fecha_reg, obs,
estado from planilla_aguinaldo where id_planilla_aguinaldo = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id_planilla_aguinaldo);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new Planilla_aguinaldoException(sqle);
    }
    catch(Exception e){throw new Planilla_aguinaldoException(e);}
    finally {}
}

```

Clase Planilla_sueldo

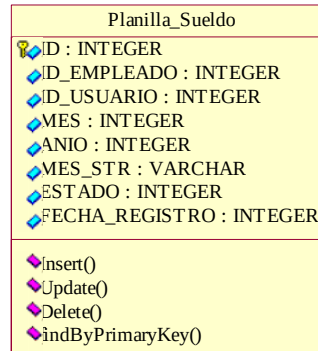


Figura N° 316: Especificación Método Planilla_Sueldo

```

public int insert(Planilla_sueldo planilla_sueldo, Connection con) throws
Planilla_sueldoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into PLANILLA_SUELDO(
MES, ANIO, ID_USUARIO, ID_EMPLEADO, MES_STR, ESTADO,
FECHA_REGISTRO) values (?, ?, ?, ?, ?, ?, ?)");

        ps.setInt(1, planilla_sueldo.getMes());
        ps.setInt(2, planilla_sueldo.getAnio());
        ps.setInt(3, planilla_sueldo.getId_usuario());
        ps.setInt(4, planilla_sueldo.getId_empleado());
        ps.setString(5, planilla_sueldo.getMes_str());
        ps.setInt(6, planilla_sueldo.getEstado());
        ps.setDate(7, planilla_sueldo.getFecha_registro());
        return(ps.executeUpdate());
    } catch(SQLException sqle){throw new
Planilla_sueldoException(sqle);}
    catch(Exception e){throw new Planilla_sueldoException(e);}
}
  
```

```

public int update(Planill_sueldoPK planill_sueldopk, Planill_sueldo planill_sueldo,
Connection con)throws Planill_sueldoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update PLANILL_SUELDO set
MES = ? , ANIO = ? , ID_USUARIO = ? , ID_EMPLEADO = ? , MES_STR = ? ,
ESTADO = ? , FECHA_REGISTRO = ? where id = ?");

        ps.setInt(1,planill_sueldo.getMes());
        ps.setInt(2,planill_sueldo.getAnio());
        ps.setInt(3,planill_sueldo.getId_usuario());
        ps.setInt(4,planill_sueldo.getId_empleado());
        ps.setString(5,planill_sueldo.getMes_str());
        ps.setInt(6,planill_sueldo.getEstado());
        ps.setDate(7,planill_sueldo.getFecha_registro());
        ps.setInt(8,planill_sueldopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new
Planill_sueldoException(sqle);}
    catch(Exception e){throw new Planill_sueldoException(e);}
}

public int delete(Planill_sueldoPK planill_sueldopk , Connection con)throws
Planill_sueldoException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from PLANILL_SUELDO
where id = ?");

        ps.setInt(1, planill_sueldopk.getId());
        return(ps.executeUpdate());
    }
}

```

```

        }catch(SQLException sqle) {throw new
Planill_sueldoException(sqle);}
        catch(Exception e) {throw new Planill_sueldoException(e);}
    }
    public Planill_sueldo findByPrimaryKey(int id, Connection con) throws
Planill_sueldoException{
        PreparedStatement stmt = null;
        ResultSet rs = null;
        try {
            final String SQLSTATEMENT = "Select id, mes, anio,
id_usuario, id_empleado, mes_str, estado, fecha_registro from planill_sueldo where
id = ?";

            stmt=con.prepareStatement(SQLSTATEMENT);
            stmt.setInt(1, id);
            rs = stmt.executeQuery();
            return fetchSingleResult(rs);
        }
        catch (SQLException sqle) {
            throw new Planill_sueldoException(sqle);
        }
        catch(Exception e){throw new Planill_sueldoException(e);}
        finally {}
    }
}

```

Clase Puesto

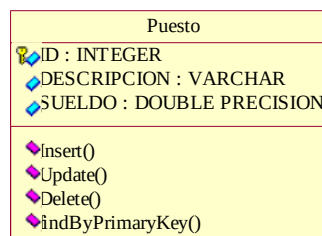


Figura N° 317: Especificación Método Puesto

```

public int insert(Puesto puesto ,Connection con)throws PuestoException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into PUESTO( ID,
DESCRIPCION, SUELDO) values (?, ?, ?)");
        ps.setInt(1,puesto.getId());
        ps.setString(2,puesto.getDescripcion());
        ps.setDouble(3,puesto.getSueldo());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new PuestoException(sqle);}
    catch(Exception e){throw new PuestoException(e);}
}

```

```

public int update(PuestoPK puestopk, Puesto puesto, Connection con)throws
PuestoException{
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("update PUESTO set
DESCRIPCION = ? , SUELDO = ? where ID = ?");
        ps.setString(1,puesto.getDescripcion());
        ps.setDouble(2,puesto.getSueldo());
        ps.setInt(3,puestopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new PuestoException(sqle);}
    catch(Exception e){throw new PuestoException(e);}
}

```

```

public int delete(PuestoPK puestopk , Connection con)throws PuestoException{
    PreparedStatement ps = null;
    try {

```

```

        ps = con.prepareStatement("delete from PUESTO where ID =
?");

        ps.setInt(1, puestopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new PuestoException(sqle);}
    catch(Exception e) {throw new PuestoException(e);}
}

public Puesto findByPrimaryKey(int id, Connection con) throws PuestoException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select ID, DESCRIPCION,
SUELDO from PUESTO where ID = ?";
        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new PuestoException(sqle);
    }
    catch(Exception e){throw new PuestoException(e);}
    finally {}
}

```

Clase Usuario

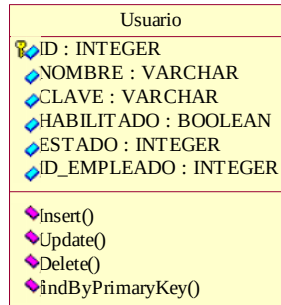


Figura N° 318: Especificación Método Usuario

```

public int insert(Usuario usuario ,Connection con)throws UsuarioException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into USUARIO( NOMBRE,
        CLAVE, HABILITADO, ESTADO, ID_EMPLEADO) values (?, ?, ?, ?, ?)");

        ps.setString(1,usuario.getNombre());
        ps.setString(2,usuario.getClave());
        ps.setBoolean(3,usuario.getHabilitado());
        ps.setInt(4,usuario.getEstado());
        ps.setInt(5,usuario.getId_empleado());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new UsuarioException(sqle);}
    catch(Exception e){throw new UsuarioException(e);}
}

public int update(UsuarioPK usuariopk, Usuario usuario, Connection con)throws
UsuarioException{
    PreparedStatement ps = null;
    try
    {

```

```

        ps = con.prepareStatement("update USUARIO set NOMBRE =
        ? , CLAVE = ? , HABILITADO = ? , ESTADO = ? , ID_EMPLEADO = ? where id
        = ?");

```

```

        ps.setString(1,usuario.getNombre());
        ps.setString(2,usuario.getClave());
        ps.setBoolean(3,usuario.getHabilitado());
        ps.setInt(4,usuario.getEstado());
        ps.setInt(5,usuario.getId_empleado());
        ps.setInt(6,usuariopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle){throw new UsuarioException(sqle);}
    catch(Exception e){throw new UsuarioException(e);}
}

```

```

public int delete(UsuarioPK usuariopk , Connection con)throws UsuarioException{
    PreparedStatement ps = null;
    try {
        ps = con.prepareStatement("delete from USUARIO where id =
        ?");

        ps.setInt(1, usuariopk.getId());
        return(ps.executeUpdate());
    }catch(SQLException sqle) {throw new UsuarioException(sqle);}
    catch(Exception e) {throw new UsuarioException(e);}
}

```

```

public Usuario findByPrimaryKey(int id, Connection con) throws
UsuarioException{
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        final String SQLSTATEMENT = "Select id, nombre, clave,
        habilitado, estado, id_empleado from usuario where id = ?";

```



```

        stmt=con.prepareStatement(SQLSTATEMENT);
        stmt.setInt(1, id);
        rs = stmt.executeQuery();
        return fetchSingleResult(rs);
    }
    catch (SQLException sqle) {
        throw new UsuarioException(sqle);
    }
    catch(Exception e){throw new UsuarioException(e);}
    finally {}
}

```

Clase Usuario_Modulo

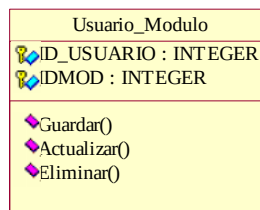


Figura N° 319: Especificación Método Usuario_Modulo

```

public int insert(Usuario_modulo usuario_modulo ,Connection con)throws
Usuario_moduloException {
    PreparedStatement ps = null;
    try
    {
        ps = con.prepareStatement("insert into USUARIO_MODULO(
ID_USUARIO, IDMOD) values (?, ?)");

        ps.setInt(1,usuario_modulo.getId_usaurio());
        ps.setInt(2,usuario_modulo.getIdmod());
        return(ps.executeUpdate());
    }
}

```

```

        }catch(SQLException sqle){throw new
Usuario_moduloException(sqle);}
        catch(Exception e){throw new Usuario_moduloException(e);}
    }
    public int update(Usuario_moduloPK usuario_modulopk, Usuario_modulo
usuario_modulo, Connection con)throws Usuario_moduloException{
        return -1;
    }
    public int delete(Usuario_moduloPK usuario_modulopk , Connection con)throws
Usuario_moduloException{
        PreparedStatement ps = null;
        try {
            ps = con.prepareStatement("delete from
USUARIO_MODULO where ID_USUARIO = ? AND IDMOD = ?");
            ps.setInt(1, usuario_modulopk.getIdmod());
            ps.setInt(2, usuario_modulopk.getId_usaurio());
            return(ps.executeUpdate());
        }catch(SQLException sqle) {throw new
Usuario_moduloException(sqle);}
        catch(Exception e) {throw new Usuario_moduloException(e);}
    }

```

II.1.5.5 Modelo de implementación

II.1.5.5.1 Introducción

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es solo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

II.1.5.5.2 Diagrama de Despliegue

Sistema y su Funcionamiento Interno

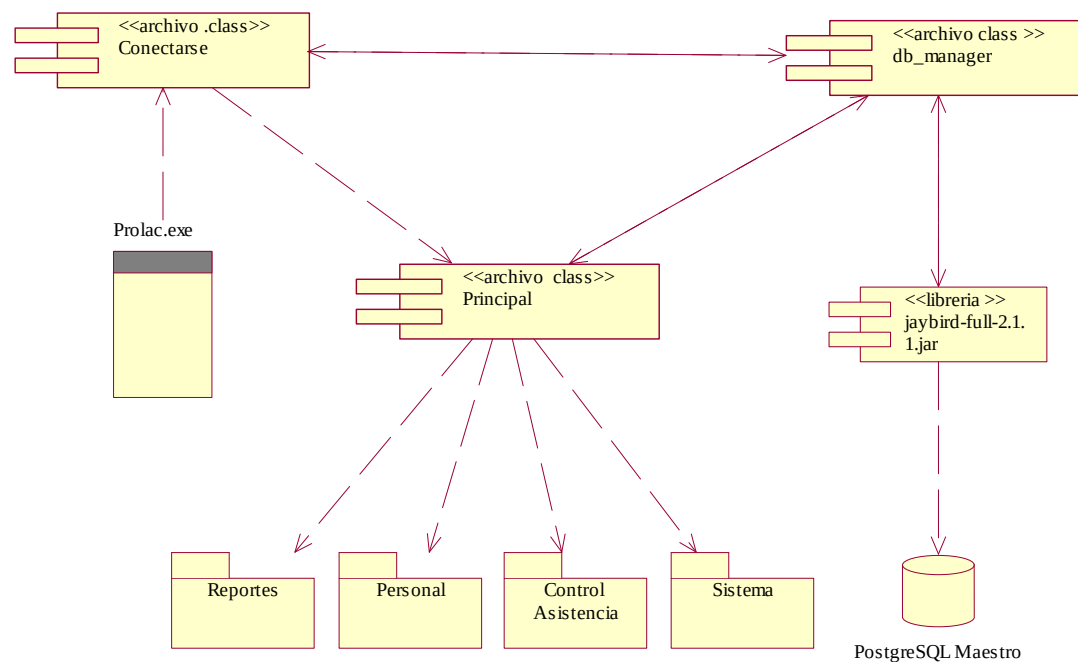


Figura N° 311: Diagrama de Despliegue

II.1.5.6 Casos de Prueba

II.1.5.6.1 Introducción

Actualmente se dispone de técnicas para estimación de proyectos que permiten la realización de evaluaciones más precisas que las obtenidas a través de métodos tradicionales (orientadas a calcular individualmente el esfuerzo correspondiente a cada una de las actividades del mismo).

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevara asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

II.1.5.6.2 Definición

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Todos los productos de Software son probados de dos formas:

- 1) Conociendo la función específica para la que fue diseñado el producto, se puede llevar a cabo pruebas que demuestren que cada función es completamente operativa, denominada **Prueba de Caja Negra**.
- 2) Conociendo el funcionamiento del producto, se puede realizar pruebas que aseguren que “todas las piezas encajan”, o sea que la especificación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada, esta prueba se denomina **Prueba de Caja Blanca**.

En nuestro caso utilizaremos la Prueba de Caja Negra para realizar pruebas a nuestro sistema.

II.1.5.6.3 Prueba de Caja Negra

La prueba de caja negra permite derivar conjunto de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa.

Los errores que se pretenden detectar mediante la prueba de caja negra son:

- Funciones Incorrecta o ausente.
- Errores de interfaz.
- Errores en las estructuras de datos.
- Errores de rendimientos.
- Errores de inicialización o terminación.
- Utilizamos el método de Participación Equivalente.

La prueba solo será aplicada a algunos de los formularios ya que en estos tenemos entrada y salida de datos.

II.1.5.6.4 Casos de Prueba

II.1.5.6.4.1 Método de Participación Equivalente

Entrada de datos: Registrar Control de Asistencia

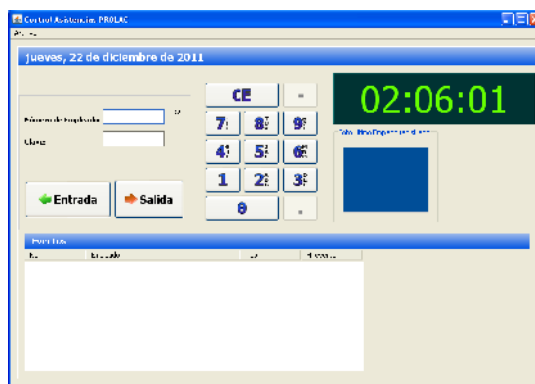


Figura N° 321: Formulario Registrar Control de Asistencia

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Numero	1.- Numérico	2.- Cualquier otra cosa
Longitud Numero	3.- $1 \leq \text{valor} \leq 6$	4.- >10 5.- <1

Tipo Clave	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Clave	8.- $1 \leq \text{valor} \leq 50$	9.- >50 10.- <1

Tabla N° 98: Condición de Entrada Registrar Control de Asistencia

Numero	Clave	Camino	Clase
"111"	"111"	1-3-6-8	Valida
"111"	" "	1-3-7-10	Invalida
" "	" "	2-5-7-10	Invalida

Tabla N° 99: Caminos de Condición de Entrada Registrar Control de Asistencia

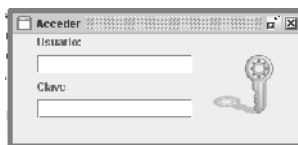
Entrada de datos: Ingresar al Sistema.

Figura N° 322 Formulario Acceso

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Usuario	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Usuario	2.- $1 \leq \text{valor} \leq 10$	4.- >10 5.- <1
Tipo Clave	6.- Alfanumérico	8.- Cualquier otra cosa
Longitud Clave	7.- $1 \leq \text{valor} \leq 30$	9.- >30 10.- <1

Tabla N° 100: Condición de Entrada Ingresar al Sistema

Entrada de datos: Nuevo Usuario

Figura N° 323: Formulario Nuevo Usuario

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Nombre	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Nombre	2.- $1 \leq \text{valor} \leq 10$	4.- >10 5.- <1
Tipo Clave	6.- Alfanumérico	8.- Cualquier otra cosa
Longitud Clave	7.- $1 \leq \text{valor} \leq 30$	9.- >30 10.- <1
Tipo Repetir Clave	11.- Alfanumérico	13.- Cualquier otra cosa
Longitud Repetir Clave	12.- $1 \leq \text{valor} \leq 30$	14.- >30 15.- <1

Tabla N° 101: Condición de Entrada Nuevo Usuario

Nombre	Clave	Repetir Clave	Camino	Clases
“Admin”	“123456”	“123456”	1-2-6-7-11-12	Validas
“Secretaria”	“111111”	“578975”	1-2-6-7-13-12	Invalidas
“ ”	“123”	“1234”	3-5-6-7-13-12	Invalidas

Tabla N° 102: Caminos de Condición de Entrada Nuevo Usuario

Entrada de datos: Editar Usuario

Figura N° 324: Formulario Editar Usuario

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Nombre	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Nombre	2.- $1 \leq \text{valor} \leq 10$	4.- >10 5.- <1
Tipo Clave	6.- Alfanumérico	8.- Cualquier otra cosa
Longitud Clave	7.- $1 \leq \text{valor} \leq 30$	9.- >30 10.- <1
Tipo Repetir Clave	11.- Alfanumérico	13.- Cualquier otra cosa
Longitud Repetir Clave	12.- $1 \leq \text{valor} \leq 30$	14.- >30 15.- <1

Tabla N° 103: Condición de Entrada Editar Usuario

Entrada de datos: Nuevo Empleado

Figura N° 325: Formulario Nuevo Empleado

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Nombre	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Nombre	3.- $1 \leq \text{valor} \leq 30$	4.- >30 5.- <1
Tipo Ap. Paterno	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Ap. Paterno	8.- $1 \leq \text{valor} \leq 30$	9.- >30 10.- <1
Tipo Ap. Materno	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Ap. Materno	13.- $1 \leq \text{valor} \leq 20$	14.- >20 15.- <1
Tipo C.I.	16.- Numérico	23.- Cualquier otra cosa
Longitud C.I.	17.- $1 \leq \text{valor} \leq 8$	18.- >8 19.- <1
Tipo Dirección	20.- Alfanumérico	22.- Cualquier otra cosa
Longitud Dirección	21.- $1 \leq \text{valor} \leq 100$	23.- >100 30.- <1
Tipo Correo Electrónico	24.- Alfanumérico	26.- Cualquier otra cosa
Longitud Correo Electrónico	25.- $0 \leq \text{valor} \leq 30$	27.- >30
Tipo Fecha Nacimiento	28.- Alfanumérico	29.- Cualquier otra cosa
Longitud Fecha Nacimiento	30.- $1 \leq \text{valor} \leq 20$	31.- >20 32.- <1
Tipo Fecha Ingreso	33.- Alfanumérico	34.- Cualquier otra cosa
Longitud Fecha Ingreso	35.- $1 \leq \text{valor} \leq 20$	36.- >20 37.- <1

Tipo Teléfono	38.- Numérico	39.- Cualquier otra cosa
Longitud Teléfono	40.- 0<= valor <=7	41.- >7
Tipo Celular	42.- Numérico	43.- Cualquier otra cosa
Longitud Celular	44.- 0<= valor <=8	45.- >8
Tipo Sexo	46.- Alfabético	47.- Cualquier otra cosa
Longitud Sexo	48.- 1<= valor <=10	49.- >10 50.- <1
Tipo Estado Civil	51.- Alfabético	52.- Cualquier otra cosa
Longitud Estado Civil	53.- 1<= valor <=20	54.- >20 55.- <1
Tipo Numero	56.- Numérico	57.- Cualquier otra cosa
Longitud Numero	58.- 1<= valor <=6	59.- >6 60.- <1
Tipo Clave	61.- Alfanumérico	62.- Cualquier otra cosa
Longitud Clave	63.- 1<= valor <=30	64.- >30 65.- <1
Tipo Repetir Clave	66.- Alfanumérico	67.- Cualquier otra cosa
Longitud Repetir Clave	68.- 1<= valor <=30	69.- >30 70.- <1
Tipo Puesto	71.- Alfanumérico	72.- Cualquier otra cosa
Longitud Puesto	73.- 1<= valor <=50	74.- >50 75.- <1
Tipo Área	76.- Alfanumérico	77.- Cualquier otra cosa
Longitud Área	78.- 1<= valor <=50	79.- >50 80.- <1

Tabla N° 104: Condición de Entrada Nuevo Empleado

Nombre	“Ana María”	“María Carmen”
Ap. Paterno	“Vargas ”	“Tapia ”
Ap. Materno	“López”	“Sánchez”
C.I.	“5787460”	“5698432”
Dirección	“C. La Paz”	“Av. Sucre”
Fecha Nacimiento	“13/03/1980”	“28/01/1982”
Fecha Ingreso	“01/01/2000”	“15/06/1998”
Teléfono	“66 - 40087”	
Celular	“72979768”	“Carmen”

Sexo	“Femenino”	“12345”
Estado Civil	“Soltera”	
Numero	“123”	“123”
Clave	“12345”	“12345”
Repetir Clave	“123456”	“123456”
Puesto	“Contador”	
Área	“Administración Finanzas”	
Correo Electrónico	“Ana_VLβ@hotmail.com”	
Camino	1-3-6-8-11-13-16-17-20- 21-24-25-28-30-33-35-38- 40-42-44-46-48-51-53-56- 58-61-63-66-68-71-73-76- 78.	1-3-6-8-11-13-16- 17-43-47-54-55- 74-75-79-80
Clases	Validas	Invalidas

Tabla N° 105: Caminos de Condición de Entrada Nuevo Empleado

Entrada de datos: Editar Empleado

The screenshot shows a Windows-style application window titled "Modificar Empleado". Inside, there's a form with the following elements:

- Fields for "Nombre", "Apellido", and "Fecha de Nacimiento".
- A "Foto" field with a small image placeholder.
- Fields for "Cargo" and "Area".
- Checkboxes for "Activo" and "Validado".
- Buttons for "Guardar" (Save) and "Cancelar" (Cancel).

Figura N° 326: Formulario Editar Empleado

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Nombre	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Nombre	3.- $1 \leq \text{valor} \leq 30$	4.- >30 5.- <1
Tipo Ap. Paterno	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Ap. Paterno	8.- $1 \leq \text{valor} \leq 30$	9.- >30 10.- <1
Tipo Ap. Materno	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Ap. Materno	13.- $1 \leq \text{valor} \leq 20$	14.- >20 15.- <1
Tipo C.I.	16.- Numérico	23.- Cualquier otra cosa
Longitud C.I.	17.- $1 \leq \text{valor} \leq 8$	18.- >8 19.- <1
Tipo Dirección	20.- Alfanumérico	22.- Cualquier otra cosa
Longitud Dirección	21.- $1 \leq \text{valor} \leq 100$	23.- >100 30.- <1
Tipo Correo Electrónico	24.- Alfanumérico	26.- Cualquier otra cosa
Longitud Correo Electrónico	25.- $1 \leq \text{valor} \leq 30$	27.- >30 28.- <1
Tipo Fecha Nacimiento	29.- Alfanumérico	30.- Cualquier otra cosa
Longitud Fecha Nacimiento	31.- $1 \leq \text{valor} \leq 20$	32.- >20 33.- <1
Tipo Fecha Ingreso	34.- Alfanumérico	35.- Cualquier otra cosa
Longitud Fecha Ingreso	36.- $1 \leq \text{valor} \leq 20$	37.- >20 38.- <1
Tipo Teléfono	39.- Numérico	40.- Cualquier otra cosa
Longitud Teléfono	41.- $1 \leq \text{valor} \leq 7$	42.- >7 43.- <1
Tipo Celular	44.- Numérico	45.- Cualquier otra cosa
Longitud Celular	46.- $1 \leq \text{valor} \leq 8$	47.- >8 48.- <1
Tipo Sexo	49.- Alfabético	50.- Cualquier otra cosa
Longitud Sexo	51.- $1 \leq \text{valor} \leq 10$	52.- >10 53.- <1
Tipo Estado Civil	54.- Alfabético	55.- Cualquier otra cosa
Longitud Estado Civil	56.- $1 \leq \text{valor} \leq 20$	57.- >20 58.- <1
Tipo Numero	55.- Numérico	56.- Cualquier otra cosa
Longitud Numero	57.- $1 \leq \text{valor} \leq 6$	58.- >6 59.- <1
Tipo Clave	60.- Alfanumérico	61.- Cualquier otra cosa
Longitud Clave	62.- $1 \leq \text{valor} \leq 30$	63.- >30 64.- <1

Tipo Repetir Clave	65.- Alfanumérico	66.- Cualquier otra cosa
Longitud Repetir Clave	67.- $1 \leq \text{valor} \leq 30$	68.- >30 69.- <1
Tipo Puesto	70.- Alfanumérico	71.- Cualquier otra cosa
Longitud Puesto	72.- $1 \leq \text{valor} \leq 50$	73.- >50 74.- <1
Tipo Área	75.- Alfanumérico	76.- Cualquier otra cosa
Longitud Área	77.- $1 \leq \text{valor} \leq 50$	78.- >50 79.- <1
Tipo Estado Civil	80.- Alfabético	81.- Cualquier otra cosa
Longitud Estado Civil	82.- $1 \leq \text{valor} \leq 20$	83.- >20 84.- <1

Tabla N° 106: Condición de Entrada Editar Empleado

Entrada de datos: Nuevo Área

Figura N° 327: Formulario Nuevo Área

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Descripción	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Descripción	2.- $1 \leq \text{valor} \leq 50$	4.- >50 5.- <1

Tabla N° 107: Condición de Entrada Nueva Área

Entrada de datos: Editar Área

Figura N° 328: Formulario Editar Área

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Descripción	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Descripción	2.- $1 \leq \text{valor} \leq 50$	4.- >50 5.- <1

Tabla N° 108: Condición de Entrada Editar Área

Entrada de datos: Nuevo Puesto

Figura N° 329: Formulario Nuevo Puesto

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Descripción	1.- Alfanumérico	3.- Cualquier otra cosa
Longitud Descripción	2.- $1 \leq \text{valor} \leq 50$	4.- >50 5.- <1

Tabla N° 109: Condición de Entrada Nuevo Puesto

Entrada de datos: Editar Puesto

Figura N° 330: Formulario Editar Puesto

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Descripción	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Descripción	3.- $1 \leq \text{valor} \leq 50$	4.- >50 5.- <1

Tabla N° 110: Condición de Entrada Editar Puesto

Entrada de datos: Nuevo Permiso

Figura N° 331: Formulario Nuevo Permiso

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Fecha	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Fecha	3.- $1 \leq \text{valor} \leq 20$	4.- >20 5.- <1
Tipo Tipo	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Tipo	8.- $1 \leq \text{valor} \leq 20$	9.- >20 10.- <1
Tipo Autoriza	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Autoriza	13.- $1 \leq \text{valor} \leq 50$	14.- >50 15.- <1
Tipo Motivo	16.- Alfanumérico	17.- Cualquier otra cosa
Longitud Motivo	18.- $1 \leq \text{valor} \leq 50$	19.- >50 20.- <1

Tabla N° 111: Condición de Entrada Nuevo Permiso

Fecha	Tipo	Autoriza	Motivo	Caminos	Clases
"24/09/2002"	"Llegar tarde"	"Víctor Cardozo"	"Capacitación"	1-3-6-8-11-13-16-18	Valida
"27/12/2008"	" "	" "	"Salud"	1-3-9-10-14-15-16-18	Invalida

Tabla N° 112: Caminos de Condición de Entrada Nuevo Permiso

Entrada de datos: Editar Permiso

Figura N° 332: Formulario Editar Permiso

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Fecha	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Fecha	3.- $1 \leq \text{valor} \leq 20$	4.- >20 5.- <1
Tipo Tipo	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Tipo	8.- $1 \leq \text{valor} \leq 20$	9.- >20 10.- <1
Tipo Autoriza	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Autoriza	13.- $1 \leq \text{valor} \leq 50$	14.- >50 15.- <1
Tipo Autoriza	16.- Alfanumérico	17.- Cualquier otra cosa
Longitud Autoriza	18.- $1 \leq \text{valor} \leq 50$	19.- >50 20.- <1
Tipo Motivo	21.- Alfanumérico	22.- Cualquier otra cosa
Longitud Motivo	23.- $1 \leq \text{valor} \leq 50$	24.- >50 25.- <1

Tabla N° 113: Condición de Entrada Editar Permiso

Entrada de datos: Nueva Planilla de Sueldo

Figura N° 333: Formulario Nueva Planilla de Sueldo

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Mes	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Mes	3.- $1 \leq \text{valor} \leq 20$	4.- >20 5.- <1
Tipo Año	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Año	8.- $1 \leq \text{valor} \leq 20$	9.- >20 10.- <1
Tipo Responsable	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Responsable	13.- $1 \leq \text{valor} \leq 50$	13.- >50 14.- <1

Tabla N° 114: Condición de Entrada Nueva Planilla de Sueldo

Mes	Año	Responsable	Caminos	Clases
Septiembre	2003	Víctor Cardozo G.	1-3-6-8-11-13	Valida
Agosto	1999		1-3-6-8-13-14	Invalida

Tabla N° 115: Caminos de Condición de Entrada Nueva Planilla de Sueldo

Entrada de datos: Nuevo Día Inhábil



Figura N° 334: Formulario Nuevo Día Inhábil

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Fecha	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Fecha	3.- $1 \leq \text{valor} \leq 20$	4.- >20 5.- <1
Tipo Descripción	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Descripción	8.- $1 \leq \text{valor} \leq 50$	9.- >50 10.- <1

Tabla N° 116: Condición de Entrada Nuevo Día Inhábil

Fecha	Descripción	Caminos	Clases
01/01/2000	“Año Nuevo”	1-3-6-8	Valida
25/12/2000	“ ”	1-3-9-10	Invalidad

Tabla N° 117: Caminos de Condición de Entrada Nuevo Día Inhábil

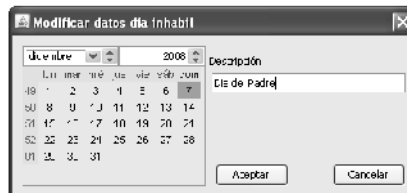
Entrada de datos: Editar Día Inhábil

Figura N° 335: Formulario Editar Día Inhábil

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Fecha	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Fecha	3.- $1 \leq \text{valor} \leq 20$	4.- >20 5.- <1
Tipo Descripción	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Descripción	8.- $1 \leq \text{valor} \leq 50$	9.- >50 10.- <1

Tabla N° 118: Condición de Entrada Editar Día Inhábil

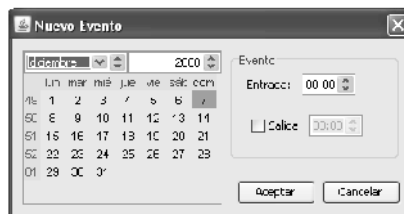
Entrada de datos: Nuevo Evento

Figura N° 336: Formulario Nuevo Evento

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Mes	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Mes	3.- $1 \leq \text{valor} \leq 10$	4.- >10 5.- <1
Tipo Año	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Año	8.- $1 \leq \text{valor} \leq 10$	9.- >10 10.- <1
Tipo Evento	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Evento	13.- $1 \leq \text{valor} \leq 10$	14.- >10 15.- <1

Tabla N° 119: Condición de Entrada Nuevo Evento

Mes	Año	Evento	Caminos	Clases
“Enero”	“2008”	“Entrada”	1-3-6-8-11-13	Valida
“Febrero”	“2007”		1-3-6-8-14-15	Invalida

Tabla N° 120: Caminos de Condición de Entrada Nuevo Evento

Entrada de datos: Modificar Evento

Figura N° 337: Formulario Modificar Evento

Condiciones externas	Clase de Equivalencia Validas	Clase de Equivalencia Invalidas
Tipo Mes	1.- Alfanumérico	2.- Cualquier otra cosa
Longitud Mes	3.- $1 \leq \text{valor} \leq 10$	4.- >10 5.- <1
Tipo Año	6.- Alfanumérico	7.- Cualquier otra cosa
Longitud Año	8.- $1 \leq \text{valor} \leq 10$	9.- >10 10.- <1

Tipo Evento	11.- Alfanumérico	12.- Cualquier otra cosa
Longitud Evento	13.- $1 \leq \text{valor} \leq 10$	14.- >10 15.- <1

Tabla N° 121: Condición de Entrada Modificar Evento

II.2 COMPONENTE 2: CAPACITACION Y SOCIALIZACION

II.2.1 Introducción

La importancia creciente que se está dando a invertir en Socialización y Capacitación requiere que el profesional este altamente preparado ante las demandas generadas por los acontecimientos cotidianos en el contexto global en el que se compete. En la era de la globalización, es un desafío para las empresas de países en vías de desarrollo es diseñar Socializaciones y Capacitaciones contingentes y técnicamente abordables, que permitan competir mano a mano con países desarrollados y así lograr el crecimiento y desarrollo que les permita una apertura al mundo en condiciones de equivalencia.

El origen de las demandas suelen ser diversas: nueva tecnología, presiones sindicales, gerentes operativos o los mismos trabajadores. Muchos exigen capacitarse en algunas materias que creen necesarias pero desconocen los verdaderos problemas de fondo o las soluciones que buscan.

Una de las claves a considerar es el entorno, comprender la situación en que actúan los trabajadores que van a ser capacitados y la utilización que podrán hacer del aprendizaje obtenido. Esto requiere que el analista o consultor considere el enfoque estratégico de la empresa y el carácter técnico. Cuando se puede definir de mejor modo los roles que cada uno de los protagonistas tendrá, sea relator, socializador, capacitado o analista. De esta manera, el análisis de las relaciones entre cada uno de ellos permitirá definir el campo de acción, las zonas de consenso y de conflicto, en las que se requiere de la negociación para que la acción de la socialización y capacitación pueda desarrollarse y alcanzar sus objetivos.

II.2.2.1 Introducción

El componente “Capacitación y Socialización”, ha optado una forma de llevar a cabo la realización de este componente:

- Mediante un curso de capacitación sobre el uso de las TIC y del Sistema, donde el personal y/o representantes de PROLAC serán capacitados.

II.2.2.2 Propósito

El propósito del componente “Capacitación y Socialización”, es socializar el sistema mediante capacitación al personal de PROLAC, en especial al área de gerencia que nos está brindando la información necesaria para el desarrollo de este proyecto.

II.2.2.3 Objetivos

- Dar a conocer al personal involucrado las funcionalidades y beneficios del Sistema desarrollado para el Control de Asistencia del personal.
- Capacitar al personal quienes nos cooperaron con la información necesaria para realización de este proyecto.

II.2.3 Capacitación al Personal de PROLAC

II.2.3.1 Técnicas y Métodos de Enseñanza

II.2.3.1.1 Principios Didácticos

Los métodos y Técnicas de la enseñanza, independencia de las teorías que los originen deben sujetarse a algunos principios comunes, teniendo en cuenta el desarrollo y madurez pedagógica alcanzada hasta el presente. Algunos de ellos son [3]:

- **Principio de Dirección:** Tornar claros y precisos los objetivos a alcanzar.
- **Principio de Marcha Propia y Continua:** Procura respetar las diferencias individuales, no exigiendo la misma realización de todos los educandos.
- **Principio de Adecuación:** Es necesario que las tareas y objetivos de la enseñanza sean acordes con la necesidades del educando.
- **Principio de Eficiencia:** El ideal, mínimo esfuerzo máxima eficiencia en el aprendizaje.
- **Principio de dificultad o Esfuerzo:** Es preciso tener cuidado de no colocar al educando ante situaciones de las que tenga posibilidades de salir bien, pues el fracaso continuado es peor veneno para la criatura humana.
- **Principio de Participación:** El educando es parte activa y dinámica del proceso.
- **Principio de Espontaneidad:** Cualquier proceso emprendido debe favorecer las manifestaciones naturales del educando.
- **Principio de Transparencia:** El conocimiento aprendido debe replicarse en otras situaciones de la vida diaria.
- **Principio de Evaluación:** Con un proceso continuo de evaluación, el docente podrá identificar a tiempo dificultades en el proceso de aprendizaje.
- **Principio de Reflexión:** Inducir al pensamiento reflexivo en el alumno como parte integral de actuar del ser humano.
- **Principio de Responsabilidad:** Encaminar todo el proceso de enseñanza de modo que el educando madure en cuanto a comportamiento responsable.

II.2.3.1.2 Métodos de Enseñanza

II.2.3.1.2.1 Métodos de Enseñanza Individualizada

Su objetivo máximo es ofrecer oportunidades de desenvolvimiento individual más eficiente, y llevar al educando a un completo desarrollo de sus posibilidades personales.

Método de Proyectos

Lleva al alumno a la realización efectiva de algo, es activo y lo lleva para que realice, actué, es en suma determinar una tarea y que el alumno la realice. Ofrece pasos para solucionar problemas con la solución para la realización que da experiencia al alumno.

- Solución de problemas por realización.
- Carácter general o global abarca conjunto de disciplinas.
- Restringido abarca una o dos disciplinas.
- Desenvuelve espíritu de iniciativa, responsabilidad, solidaridad y libertad.
- Cadena organizada de actividades para realizar algo.

II.2.3.2 Propuesta del Proyecto referente a la Capacitación

II.2.3.2.1 Descripción del Curso

Este curso de 2 días está diseñado para preparar a los participantes (al personal de PROLAC) en el conocimiento y utilización de las TIC, esto con el fin de que puedan manejar y administrar el sistema de escritorio desarrollado en el proyecto de manera individual y sin depender de algún otro experto.

Este curso está dirigido con preferencia y especial al área de gerencia, pero puede adaptarse de acuerdo a las circunstancias (falta de participación, petición de otras áreas interesados en participar) a la inclusión de otros participantes, además queda abierta la posibilidad de extender el curso de acuerdo a la respuesta.

II.2.3.2.2 Metas y Objetivos generales del Curso

Influenciar positivamente las actitudes de los participantes respecto a los beneficios de utilizar el Sistema de Escritorio desarrollado en el proyecto y el uso y manejo de las TIC como medio alternativo para incrementar su competitividad.

Proporcionar al participante los conocimientos y las habilidades necesarios para manejar y administrar correctamente el software de gestión.

II.2.3.2.3 Objetivos del Aprendizaje del participante

Después de asistir a los cursos de capacitación sobre las TIC y manejo del sistema de escritorio desarrollado en el proyecto, el participante estará en capacidad de administrar el sistema de gestión de su empresa.

II.2.3.2.4 Descripción de los Métodos de Capacitación

El proyecto plantea llevar a cabo para la capacitación una:

- **Metodología de Enseñanza Socializada**

El cual pretende ejecutar una metodología de enseñanza socializada, dirigida a los gerentes o encargados de las pequeñas y medianas empresas, con lo que se busca una interrogación social sin descuidar la individualización, aplicando métodos de asamblea y panel.

- **Técnica Expositiva**

Consiste en la exposición oral, por parte del capacitador, en la que se estimulara la participación de los microempresarios en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los participantes.

- **Técnica de la Experiencia**

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para realizar alguna acción.
- Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.

- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

Demostración: Pruebas, test, ensayos del Sistema.

Para la Capacitación: Se empleara Manuales de Usuario que contienen toda la información detallada sobre el manejo del sistema y se llevara un registro del personal que asiste a la capacitación.

II.2.3.2.5 Descripción de los materiales de Capacitación

Materiales Impresos: Guías de curso, manuales de usuario.

Materiales Audiovisuales: Equipo de computación, proyector, micrófono, parlantes.

Otros materiales: Pizarrón, marcadores.

II.2.3.2.6 Criterios para la selección de los participantes

Los participantes de este curso deben ser personal de PROLAC, no se exige a los participantes tener conocimiento acerca de computación.

II.2.3.2.7 Métodos para la Evaluación del Curso

Cuestionario Pre-Curso: Para evaluar los conocimientos pre curso acerca de computación.

Guías de Aprendizaje: Los participantes evalúan sus propias habilidades.

Verificación de lo Aprendido: El capacitador evalúa las habilidades de los participantes en el manejo del sistema de gestión y su administración.

Evaluación del Curso de Capacitación: Los participantes evalúan el curso.

II.2.3.2.8 Duración del Curso

El curso tendrá una duración de dos días, cada clase tendrá una duración aproximada de 2 horas por día.

II.2.3.2.9 Numero de Participantes

Preferiblemente se recomienda no sobrepasar la cantidad de 15 participantes en el curso, esto debido a la falta de instalaciones para acoger a más personas, al presupuesto del proyecto y para favorecer al propio ambiente de aprendizaje.

II.2.3.2.10 Fecha del Curso

De acuerdo al cronograma del proyecto, el curso se realizara a partir del 24 de noviembre hasta el 25 de noviembre del presente año, salvo algún contratiempo no planificado.

Una vez finalizada la capacitación se prosiguió con la entrega de certificados de capacitación a los presentes (ANEXO 1).

II.2.3.2.11 Organizador del Curso de Capacitación

El curso será organizado por el director del proyecto, bajo asesoramiento de los docentes de la materia de Taller III y del gerente de PROLAC.

II.2.3.2.12 Contenido del Curso sobre las TIC

- **Que son TIC**

Las Tecnologías de Información y Comunicación (TICs), se encargan del estudio, desarrollo, implementación, alimentación y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

- Hardware
- Software
- Servicios de TI
- Telecomunicaciones.

- **Globalización**

Es el proceso por el que la creciente comunicación e interdependencia entre los distintos países del mundo unifica mercados, sociedades y culturas, a través de una serie de transformaciones sociales, económicas y políticas que les dan un carácter global. Así, los modos de producción y de movimientos de capital se configuran a escala planetaria, mientras los gobiernos van perdiendo atribuciones ante lo que se ha denominado la sociedad en red.

En este contexto el desarrollo tecnológico, y refiriéndose en sí a la incorporación de las TIC en cada nación, hará que el desarrollo en esta llegue sin tardar demasiado, la competitividad de un país en un sistema globalizado depende en gran medida de la tecnología que adquiera, por eso es tan estrecha la relación entre TIC y globalización. Esperemos que nuestros países comiencen a incorporar las TIC en sus procesos productivos y así reducir la dependencia de potenciales mundiales.

- **Las TIC y la Competitividad**

La utilización de las TIC y el Comercio Electrónico por parte del sector empresarial, abre nuevas oportunidades de negocios que redundan en una serie de beneficios económicos y sociales, los cuales van desde el mejoramiento y la facilitación de la comunicación entre las empresas, hasta el manejo más eficiente de los recursos. En tal sentido, las tecnologías deben ser utilizadas por el sector en las diferentes etapas de la cadena de valor.

- Calidad
- Productividad
- Servicio
- Imagen

- **Software**

Las soluciones a implementar deben tener influencia sobre la productividad y la rentabilidad en muchos eslabones de la cadena productiva. La cartera de soluciones para una empresa debería estar formada al menos por:

- Datos históricos de la empresa para el análisis.
- Soluciones para gestionar la planificación, producción, comercialización, así como para el cumplimiento de las normas y estándares.
- Herramientas para la ingeniería de sistemas (Capacidad de análisis y evolución de los sistemas en el tiempo).
- Sistemas de Recursos Humanos.
- Controladores de Procesos productivos.
- Optimización de la vida útil de los procesos (Revisión de procesos)

Que debemos hacer cuando un cliente nos hace un pedido: analizar y evaluar.

- La disponibilidad de inventarios de materias primas.
- La capacidad para fabricar el producto solicitado y en el tiempo establecido.
- La disponibilidad de recursos de producción y de materiales para su fabricación.
- La adecuada logística para entregar el producto (ya sea propia o de terceros).
- La rentabilidad del producto.

El Software para la empresa es vital debido a:

- Necesidad de optimizar dinámicamente los procesos.
- Optimización de la cadena de suministro en la empresa y entre empresas.
- Establecer procesos comerciales en colaboración.
- Soluciones para gestionar la cadena de suministro.
- Conectividad comercial ilimitada.
- Modelización de procesos comerciales.
- Soluciones para mercados electrónicos privados o para la colaboración en el lado de las ventas.

II.2.4 Conclusiones

Una característica importante de los países con mayor grado de desarrollo empresarial, es contar con un alto nivel de competitividad y con una tecnología avanzada, el cual a su vez está asociado positivamente con el uso de las TIC's.

El uso de las TIC's por parte de las instituciones esta directamente asociado a la innovación, necesaria para incrementar su eficiencia y competitividad.

El uso de las computadoras está relacionado con la simplificación de labores administrativas y el mejor manejo de los procesos de control de asistencia del personal.

El grado de adopción de las TIC's por parte de las instituciones, tiende a reducir la brecha digital en los países en desarrollo.

El impacto de la adopción de TIC's en el desempeño de las instituciones públicas, se observa en la eficiencia media del trabajo, (más eficiencia con igual número de empleados):

El uso correcto de las TIC's en las instituciones, conlleva mejores controles administrativos y operativos, por lo tanto logramos mejores niveles de eficiencia, satisfacción del cliente y oportunidades de negocio.

Para la constancia de la ejecución de este componente se solicitó un certificado de conformidad con la capacitación y de entrega de certificados al Gerente General Sr. Víctor Cardozo el cual se hace constar que dichas actividades se llevaron a cabo por los encargados de las mismas (ANEXO 2).

II.2.5 Medios de Verificación

Los Medios de verificación irán en el ANEXO 1y 2.

CAPÍTULO III

CONCLUSIONES

Y

RECOMENDACIONES

III.1 Conclusiones

Durante la elaboración el presente trabajo se pudo observar que la información con la que trabaja PROLAC del departamento de Tarija es de importancia considerable, tomando en aspectos se concluye que la elaboración de un sistema de gestión para el Control de Asistencia del personal y generación de planillas y de sueldos salarios es muy útil, ya que traería beneficios al personal en general, desde ahorrar tiempo, esfuerzo, trabajo en el registro de los datos del personal, etc.

Se debe exponer que el éxito del proyecto permite llegar a la conclusión que la metodología RUP nos permite lograr mejoras al software, gracias a su proceso de iteración para alcanzar mayores resultados. La utilización del UML nos permite nos permite lograr un sistema confiable, seguro. La implementación de la Base de Datos en Postgres nos garantiza la seguridad de la información.

El sistema contara con información confiable, segura, oportuna en el momento en que los usuarios lo requieran, brindando información controladora y organizada.

III.2 Recomendaciones

Entre las recomendaciones que destacan se podrían mencionar las siguientes:

- Para el desarrollo de futuros proyectos con características similares a este, se recomienda hacer uso de las metodologías utilizadas como (UML), ya que permitieron llegar a la conclusión del Proyecto con alta tecnología de punta.
- Para optimiza el rendimiento del sistema, se recomienda a los administradores y/o técnicos implementar el sistema en equipos de última generación.
- Se recomienda a los administradores y/o usuarios consultar el manual de usuario y el manual de instalación para explotar al máximo el uso del sistema.
- Los componentes aquí implementados no resultan ser un estándar ya que se pueden adicionar otros que ayuden a obtener mejores resultados en futuros proyectos.
- Realizar cambios periódicos de claves de los usuarios del Sistema Informático tanto del Administrador como del personal (los usuarios y claves son generados por el Administrador).
- A partir de la implementación del proyecto se deben establecer normas estándares a ser usadas por futuras gestiones de PROLAC para la obtención y gestión de la información de la empresa.