

CAPÍTULO I

1. INTRODUCCIÓN

Para proponer un sistema de asistencia automatizado es necesario realizar una valoración previa referente al actual proceso de control de asistencia del personal en el Escuela Superior de Formación de Maestros “Juan Misael Saracho” de Canasmoro, que se lleva a cabo mediante un reloj tarjetero del cual se tienen tarjetas donde se registra la hora de entrada y salida de cada funcionario, de esa tarjeta se elaboran los reportes en forma manual. Mecanismo que presenta deficiencia en los recursos materiales que utiliza y en el registro de la tarjeta, además que la elaboración de los informes de asistencia esta sujeto a demoras y errores por la elaboración manual que se realiza.

La institución plantea la necesidad del desarrollo de un sistema de información que lleve en forma efectiva el control de asistencia del personal en la institución pública, pasando de un proceso manual a uno automatizado.

Este sistema automatizado permite realizar el riguroso control de entrada, salida, faltas, retrasos, permisos, también contará con la seguridad de acceso, lo cual admitirá solo al usuario encargado tener entrada al sistema, con el fin de evitar pérdidas de información a la institución. Será una herramienta confiable para el manejo y manipulación de datos evitando así pérdida de tiempo y de información.

Además de generar las planillas de asistencia, orientado a mejorar el proceso de elaboración de reportes de manera rápida y eficiente, creando un ambiente cómodo para el jefe de personal. La comunicación entre el usuario y el sistema propuesto es más confiable y amigable, facilitando así el efectivo funcionamiento del sistema de control de asistencia en la institución. Para tal efecto, el desarrollo del sistema debe pasar por etapas las cuales son muy importantes principalmente el análisis y diseño.

2. ANTECEDENTES

Actualmente el Escuela Superior de Formación de Maestros “Juan Misael Saracho” es una institución pública dependiente del Ministerio de Educación y Culturas, la cual presenta una estructura organizativa eficiente, (ver Anexo 1) parte de la definición de su misión y objetivos, los cuales a su vez se deben reflejar en una estructura, que permita que las funciones y responsabilidades de sus diferentes unidades viabilicen el cumplimiento y la medición de sus metas y propósitos.

El manejo de la información sobre el registro de asistencia del personal de la institución constituye una preocupación, ya que dicha información es manejada de forma manual, por lo tanto como consecuencia de esta situación a menudo se producen retardos y deficiencias en la elaboración de informes de asistencia del personal.

Son muchas las herramientas que, en la actualidad, facilitan al hombre el manejo del recurso informativo, así como el acceso a éste. Una de estas herramientas, que permite utilizar el recurso de la información de manera más eficiente, rápida y confiable, la constituyen las Computadoras, las cuales aparecen enmarcadas dentro del vertiginoso avance tecnológico que ha caracterizado a las últimas décadas del presente siglo.

Muchos países cuentan con equipos de última tecnología para el control de asistencia de los empleados, mediante tarjetas magnéticas, huella digital, identificadores, reconocimiento de voz, reconocimiento del iris del ojo, etc.

En Bolivia también se tuvieron grandes cambios tecnológicos en Instituciones públicas y privadas, en los Bancos y Universidades como la San Simón de Cochabamba, utilizando registro de huella digital, en las Cortes Departamentales, con sistemas Automatizados y en Tarija muchas de las instituciones públicas aún siguen trabajando de forma manual en sus registros de personal, utilizando planillas de

control de asistencia (firma en planillas), marcado de tarjetas con reloj, también con sistemas automatizados con características diferentes en como ser: La Universidad Autónoma “Juan Misael Saracho”, El Hospital Universitario “San Juan de Dios”, Bancos, Cooperativas etc.

Por este motivo se realiza el análisis y diseño de un sistema capaz de brindar información actualizada y confiable para el respectivo control de personal y para tener los reportes requeridos con eficiencia y rapidez necesaria.

3. PLANTEAMIENTO DEL PROBLEMA

La Escuela Superior de Formación de Maestros “Juan Misael Saracho” no cuenta con una mecanismo de control automatizado para el registro de entrada, salida y emisión de planillas de horas trabajadas, retrasos, ausencias, permisos para el personal de la institución.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Analizar y diseñar un mecanismo de control que permita automatizar el registro de asistencia de entrada, salida y emisión de reportes de personal, que trabaja en la institución, que permitirá obtener información actualizada, eficiente y oportuna sobre cada uno de los funcionarios de la Escuela Superior de Formación de Maestros “Juan Misael Saracho”.

4.2 OBJETIVOS ESPECÍFICOS

- Recolectar reportes manuales de la información a cerca de planillas, ingreso, salida, permisos, faltas, etc., del personal.

- Automatizar la operatividad de los trámites administrativos relacionados con licencias médicas, accidentes del trabajo, asistencia, permisos, vacaciones y otros trámites y/o certificaciones que requiera el personal.
- Aplicar el Lenguaje de Modelado Unificado UML para la modelación del análisis y diseño del sistema.
- Definir un diseño adecuado en cuanto se refiere a la posición de íconos, botones y el desplegado de la información que se ajuste al usuario, creando de esta manera un modelo de interfaz amigable para el usuario y el sistema.
- Diseñar un módulo que permita el almacenamiento de datos en una Base de Datos de forma segura para la gestión de:
 - Administración de Personal que trabaja (Mantener actualizada la documentación y los registros con todos los antecedentes y la información referida al personal).
 - Administración de departamentos ó áreas de trabajo.
 - Asignación horarios.

5. JUSTIFICACIÓN

5.1 JUSTIFICACIÓN SOCIAL

Los métodos tradicionales manuales para mantener la información del personal no son viables en una institución pública, la documentación que compete al personal de planta se encuentra en archivos manuales y el volumen es considerable, por lo cual generar cualquier tipo de información se hace difícil, pues el proceso manual implica horas de trabajo y está sujeto a errores.

La institución a través de su responsable de personal necesita gestionar fácilmente la información referente al personal con rapidez y agilidad a las diversas consultas y a optimizar la comunicación con los demás sectores de la institución generando información confiable, precisa y oportuna. La solución simple que facilita todas estas gestiones es el realizar el análisis y diseño de un sistema para la automatización del control del personal.

5.2 JUSTIFICACIÓN PRÁCTICA

Los procesos automatizados, son factores de vital importancia en la organización, en esta época se ha observado una gran innovación y desarrollo de tecnologías de la información que han permitido la evolución de computadores que son capaces de soportar sistemas de información a la sociedad con aportes a los distintos sectores económicos, sociales, políticos, educativos, entre otros.

Es necesario que la información sea procesada y almacenada de una forma más efectiva para agilizar los procesos de asistencia de personal, brindando información para la toma de decisiones en las actividades laborales de los Recursos Humanos de la Institución.

Con el desarrollo del sistema automatizado que abarca las necesidades y una mayor relación con los requerimientos del personal, proporcionando una mejor efectividad en el manejo del flujo y procesamiento de los datos. Al establecer este sistema poseerá un impacto positivo en la organización, la cual proporciona a la misma, una información confiable, agilizando y facilitando el trabajo en el proceso de registro de asistencia de personal, todo el personal se siente partidario por hacer realidad un proyecto que traería un soporte técnico confiable y acorde a los avances de la actualidad tecnológica de la información.

5.3 JUSTIFICACIÓN ACADÉMICA

El desarrollo de este sistema permitirá mostrar a la institución del estado, y a otras, como a la misma Universidad Autónoma “Juan Misael Saracho”, que en sus aulas preparan profesionales competitivos con capacidad de generar mejoras técnicas en las instituciones que así lo requieran.

Los conocimientos que imparte la Carrera de Ingeniería Informática a lo largo de años de estudio, permite estos trabajos (Proyecto de Grado) de la mejor manera posible y así demostrar que la Universidad genera profesionales capaces y aptos para enfrentar cualquier desafío aplicando las metodologías y técnicas adquiridas.

5.4 JUSTIFICACIÓN ECONÓMICA

Hoy en día ante la complejidad de los negocios, es necesaria una visión global y estratégica, que permita estar en contacto directo con la situación económica de la empresa o institución, mediante sistemas que puedan emitir datos para la toma de decisiones, de modo de no incurrir en fallos y lo más importante tener la información de personal de la institución en línea para cualquier momento.

5.5 JUSTIFICACIÓN TECNOLÓGICA

El sistema propuesto será una herramienta que facilitará y agilizará en gran medida el trabajo que desempeña el encargado de Personal de la Escuela Superior de Formación de Maestros “Juan Misael Saracho”, puesto que permitirá realizar funciones laborales de una forma más organizada y eficiente, debido a que la institución cuenta y dispone de una gran variedad de herramientas, medios y tecnología adecuada que serán de ayuda a la hora de concretizar dicho sistema.

El sistema reducirá al mínimo los problemas generados y detectados manualmente y se aprovechará los beneficios que este otorga en el procesamiento de la información en forma rápida y confiable.

6. ALCANCES Y LIMITACIONES

6.1 ALCANCE DEL TRABAJO

- Diseñar una base de datos para registrar a todo el personal que trabaja en la institución, el sistema permitirá el registro de datos de manera más organizada, facilitando información mucho más rápida y en cualquier momento.
- El sistema cuenta con las funciones de imprimir varios tipos de reportes en cualquier momento ya sea por pantalla, impresora o ambas.

- Cuenta con claves internas donde el usuario solo tiene que introducir su número de código, y luego validar su huella digital, adicionalmente el sistema estará preparado para bloquear un número desconocido y nombre de usuario desconocido, evitando el uso del sistema por usuarios indebidos.
- Los datos serán almacenados en el equipo local y adicionalmente en un disco espejo ubicado en un servidor, al cual sólo tendrá acceso personal autorizado.
- Copias de respaldo de la base de datos para resguardar la información contenida en esta.

6.2 LIMITACIONES DEL TRABAJO

El presente trabajo no abarcará los siguientes puntos:

- Personal de Seguridad.
- No contempla la parte de sueldo y salarios.
- El presente trabajo es un producto terminado que solo contempla las etapas, requerimientos, análisis y diseño, ya que son instrumentos necesarios para el desarrollo e implementación del sistema posteriormente.

7. PLANIFICACIÓN DEL TRABAJO

7.1 ACTIVIDADES A DESARROLLAR

- Elaboración de entrevistas y recopilación de la información.
- Determinación de requerimientos de la información.
- Análisis del sistema.
- Diseño del sistema.
- Diseño de la Base de Datos.
- Diseño de planillas de reportes.
- Diseño de pantallas y navegación (interfaz gráfica de usuario).
- Documentación y presentación del trabajo.
- Presentación y defensa.

CAPITULO II

MARCO TEÓRICO

2.1 ANTECEDENTES

El objeto de estudio es la Escuela Superior de Formación de Maestros “Juan Misael Saracho” de Tarija, ubicado en la provincia Méndez concretamente en la comunidad de Canasmoro, el cual cuenta con un organigrama funcional (Ver Anexo 1), es una institución pública dependiente del Ministerio de Educación y Culturas, la cual es encargada de formar maestros de los niveles inicial, primaria y secundaria, en las asignaturas de: Profesor de Inicial, Primer y Segundo Ciclos del Nivel Primario, Matemáticas de 3er. Ciclo, Lenguaje de 3er. Ciclo, Música, Artes Plásticas, Sociales, Matemáticas, Comunicación y Lenguaje, Biología, Física-Química de Secundaria.

La escuela Normal “Juan Misael Saracho” de Canasmoro fue fundada el 25 de abril de 1938 en la localidad de San Lorenzo capital de la primera sección de la provincia Méndez durante el gobierno del Tte. Cnl. Germán Busch y ministro de instrucción el Dr. Bernardo Navajas.

Tres años más tarde, se trasladó al lugar denominado “La Pampa” del cantón de Canasmoro donde funciona actualmente. Su principal objetivo fue el de formar maestros para el área rural en el ciclo básico.

En la década de los 80 se incursionó en la especialización de docentes bivalentes para el ciclo intermedio en las asignaturas de Matemáticas, Lenguaje, Ciencias Naturales y Estudios Sociales.

Posteriormente mediante la ley de Reforma Educativa promulgada el 7 de julio de 1994, se tomo previsiones a través del Art. 15 para la transformación de las escuelas Normales urbanas y rurales en Institutos Normales Superiores en Educación.

Como efecto de las reformas estructurales ejercitadas en nuestro país, el INS “Juan Misael Saracho” mediante contrato suscrito en fecha 12 de agosto de 1999 entre el

Ministerio de Educación Cultura y Deportes y representantes de la Universidad Católica Boliviana, pasa a ser administrado académicamente por esta institución educativa privada.

Finalmente después de varias jornadas de lucha, se logra que el INS “Juan Misael Saracho”, vuelva a depender del Estado Boliviano a través de la administración del Ministerio de Educación y Culturas a partir de la gestión 2006.

Actualmente mediante el Proyecto de Ley Avelino Siñani-Elizardo Pérez en el gobierno del presidente Evo Morales Ayma los institutos normales Superiores pasan a ser denominados Escuelas Superiores de Formación de Maestros.

Por lo tanto, se constituye esta institución como la entidad encargada de formar los maestros del futuro con idoneidad y capacidad que tengan por finalidad servir a la sociedad y así elevar la calidad de vida de sus estudiantes y forjar un mejor mañana para Bolivia.

Las políticas y estrategias destinadas al logro de la misión y los objetivos, están dirigidas al cumplimiento de las funciones y competencias de cada funcionario de la institución. Los funcionarios son el recurso más importante que puede tener una empresa o Institución, sin embargo llevar un adecuado registro de la asistencia de cada uno de ellos se puede volver una tarea pesada cuando no se cuenta con las herramientas o equipos adecuados para conocer la correcta información.

La manera de cómo se realiza el registro de asistencia es a través de tarjetas de cartón marcadas por un reloj, esta información más tarde es vaciada a planillas de Excel con todos los datos de asistencia de cada una de las personas que trabajan en la Institución.

Esto ocasiona un retraso en el envío de la información al Ministerio de Educación y Culturas en la ciudad de La Paz, es allí donde se hacen los descuentos de las faltas, retrasos, etc., para la cancelación del salario mensual, además la entrega de reportes

de asistencia, retrasos, faltas, vacaciones, permisos a los directores cuando lo requieran ocasiona errores en la captura y llenado de datos, además consume tiempo pues se realiza el reporte al finalizar el mes o cuando de forma repentina es solicitado de urgencia.

A continuación se hace una breve descripción de los procesos de control de asistencia y horarios que se realizan en la Escuela Superior de Formación de Maestros “Juan Misael Saracho” de Canasmoro.

- Todo el personal debe asistir puntualmente a sus labores diarias y encontrarse listo en su puesto de trabajo, a la hora establecida para el inicio de la jornada laboral.
- El personal debe suministrar a la Institución los datos fidedignos para su ficha personal. Las actualizaciones en la información de su folder personal de cada funcionario generalmente se la realiza a petición de Dirección Administrativa o en su caso cuando hay una nueva contratación o concurso de institucionalización en la institución.
- Toda contratación se realiza en coordinación con Gestión Docente dependiente del Ministerio de Educación, que son los que dan el visto bueno, solo en caso de que existiera alguna afección o se haya producido la renuncia de algún funcionario.
- Como la institución depende del Ministerio de Educación generalmente la finalización de un contrato de trabajo se realiza cuando hay un examen de competencia ó la renuncia del funcionario.
- El personal a quien se le solicite marcado de entrada y salida, debe cumplir con dicho requisito en los registros reglamentarios, donde además constan los retrasos, permisos y retiros antes de hora de salida.
- Los retrasos y las faltas al trabajo deben ser justificados por escrito y están sujetos al reglamento interno de la institución.
- Los trámites personales deben hacerse en horas fuera de oficina. De necesitar hacerlos en horas de trabajo, se requiere autorización de Dirección

Administrativa. Estos permisos pueden otorgarse, 2 medios días, hasta acumular un máximo de 8 hrs. durante el mes de trabajo del solicitante.

- Si los trámites requieren de más del equivalente a una jornada diaria de trabajo del solicitante, el tiempo utilizado se descuenta de sus días de vacación.
- Los permisos por enfermedad deben ser respaldados por el certificado médico extendido por la Caja Nacional de Salud.
- Las faltas al trabajo debido a duelos se someten a lo reglamentado en la ley: “tres días de licencia con remuneración completa al trabajador, en caso de fallecimiento de su cónyuge o de sus parientes dentro del segundo grado de consanguinidad o afinidad”.
- Las vacaciones deberán ser gozadas y no pueden sustituirse por pagos adicionales, salvaguardando así la necesidad de todos al descanso y optimizando los fondos de la Institución.
- Las personas que no toman sus vacaciones, sólo pueden acumularlas por 2 años, según reglamento interno de la Institución. La utilización de este tiempo de vacaciones acumulado debe ser planificado de acuerdo con los fines institucionales y con la debida autorización del jefe de Personal, del solicitante y de la Dirección Administrativa.
- El horario fijo asignado debe cumplirse íntegramente en el día, teniendo 45 minutos al mes para cubrir llegadas tarde.
- El marcado registrado en el reloj se realiza todos los días hábiles, las marcas son ingresadas en las planillas de Excel con los datos personales de cada una de las personas, en donde se realizan todas las gestiones referentes al control de horario: licencias, faltas, permisos, retrasos, etc. desde donde se realizan todos los informes mensuales.

2.2 METODOLOGÍA

2.2.1 DEFINICIÓN DE SOFTWARE

Software, también conocido como programático y aplicación informática, es la parte lógica del ordenador, esto es, el conjunto de programas que puede ejecutar el

hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

2.2.2 INGENIERÍA DE SOFTWARE¹

La ingeniería del software se define como la disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos de software que son desarrollados y modificados en tiempo y dentro de un presupuesto definido. La ingeniería de software difiere de la programación tradicional en que se utiliza técnicas de ingeniería de especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecido para el proyecto; además esta ingeniería se preocupa por aspectos administrativos que quedan fuera del dominio normal de la programación.

La ingeniería del software se base en las áreas de la ciencia de la computación, y de la administrativa, la económica, las habilidades de comunicación y el enfoque de la ingeniería de la solución de problemas, los objetivos principales de la ingeniería de software son mejorar la calidad de los productos de software e incrementar la productividad y satisfacción en el trabajo de los ingenieros de software.

La ingeniería del software es una disciplina que integra una serie de pasos que abarca métodos, metodologías, herramientas y procedimientos para el desarrollo del software con la intención de mejorar el proceso de control y supervisión de los proyectos desarrollados por un grupo de personas, estos pasos se denominan paradigmas.

La elección de paradigmas para la ingeniería del software se lleva acabo de acuerdo a la naturaleza del proyecto, la aplicación, los métodos, las herramientas a utilizar y los controles a implementar.

En el caso del trabajo de grado se especifica la teoría consultada para su desarrollo de la siguiente manera:

¹ Presuman Roger S.. “Ingeniería del Software”.

Etapa de Requerimientos

- En esta etapa se concluye con la recopilación de toda la información necesaria para el análisis y posterior diseño, a través de Entrevistas-observación, formularios, cuestionarios, planillas recolectadas.

Etapa de Análisis

- En esta etapa se consultó RUP Proceso Racional Unificado y Lenguaje de Modelado Unificado UML, dado que se ha visto por conveniente aplicarlo para representar el modelado de casos de uso, diagramas de clase, actividad, secuencia y paquetes.
- También se aplicó Paradigma Orientado a Objetos POO para la determinación de los tipos de objetos que son modelados según su comportamiento en el sistema.

Etapa de Diseño

- En esta etapa se consultó Proceso Racional Unificado RUP y Lenguaje de Modelado Unificado UML para el modelado de la Base de datos, el diseño navegacional dentro del sistema.
- Para el diseño de la Pantallas se aplicó la metodología RUP para la descripción misma de cada pantalla presentada por el sistema, con la descripción de: Funcionalidad, opciones de pantalla (campos), validación de datos y tablas usadas.
- El diseño del modelo entidad-relación se realizó en la herramienta ER/estudio Embarcadero.

2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE

2.3.1 PROCESO RACIONAL UNIFICADO “RUP”

Rational Unified Process RUP es una metodología de Ingeniería de Software que reúne las mejores características en el desarrollo del mismo. Emplea UML para desarrollar sistemas de software orientado a objetos. Es una forma disciplinaria de

asignar tareas y responsabilidades en una empresa de desarrollo (quien hace que, cuando y como). RUP es una guía de cómo usar UML de la forma más efectiva y a la vez UML es la base del modelamiento visual de RUP.

2.3.2 PRINCIPIOS DE RUP

RUP² pretende implementar las mejores prácticas actuales en ingeniería de software:

- Desarrollo interactivo del software
- Administración de requisitos
- Uso de arquitecturas basadas en componentes
- Modelamiento visual del software
- Verificación de la calidad del software
- Control de cambio

2.3.3 CARACTERÍSTICAS DE LA METODOLOGÍA RUP³

a) GUIADO/MANEJADO POR CASOS DE USO

La razón de ser de un sistema, software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

b) CENTRADO EN ARQUITECTURA

La arquitectura involucra los elementos más significativos del sistema y esta influencia entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

²Introducción a la Metodología “RUP”

³Introducción a la Metodología “RUP”

c) ITERATIVO E INCREMENTAL

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

d) DESARROLLO BASADO EN COMPONENTES

La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtiene o que se desarrollen y maduren sus componentes.

e) UTILIZACIÓN DE UN ÚNICO LENGUAJE DE MODELAMIENTO

UML es adoptado como único lenguaje de modelamiento para el desarrollo de todos los modelos.

- **UML**, El lenguaje unificado de modelado es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. UML capta información sobre la estructura estática y el comportamiento dinámico de un sistema.

f) PROCESO INTEGRADO

Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo para construir los diferentes modelos e integrar el trabajo a través de todos los modelos.

2.3.3.1 FASES E ITERACIONES

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases:

- Inicio
- Elaboración
- Construcción
- Transición

Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones. Las fases de iniciación y elaboración incluyen las actividades de diseño del ciclo de vida del desarrollo; la construcción y la transición constituyen su producción.

Dentro de cada fase hay varias iteraciones. Una iteración representa un ciclo de desarrollo completo, desde la captura de requisitos en el análisis hasta la implementación y pruebas, que produce como resultado la entrega al cliente o la salida al mercado de un proyecto ejecutable.

Cada fase e iteración se centra en disminuir algún riesgo y concluye con un hito bien definido. La revisión de hitos desde el momento adecuado para evaluar como se están satisfaciendo los objetivos y si el proyecto necesita ser reestructurado de alguna forma para continuar.

La figura muestra las fases de desarrollo⁴ y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cual de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

⁴ The Rational Unified Process: An introduction, Philippe Kruchten , Addison Wesley, 2000

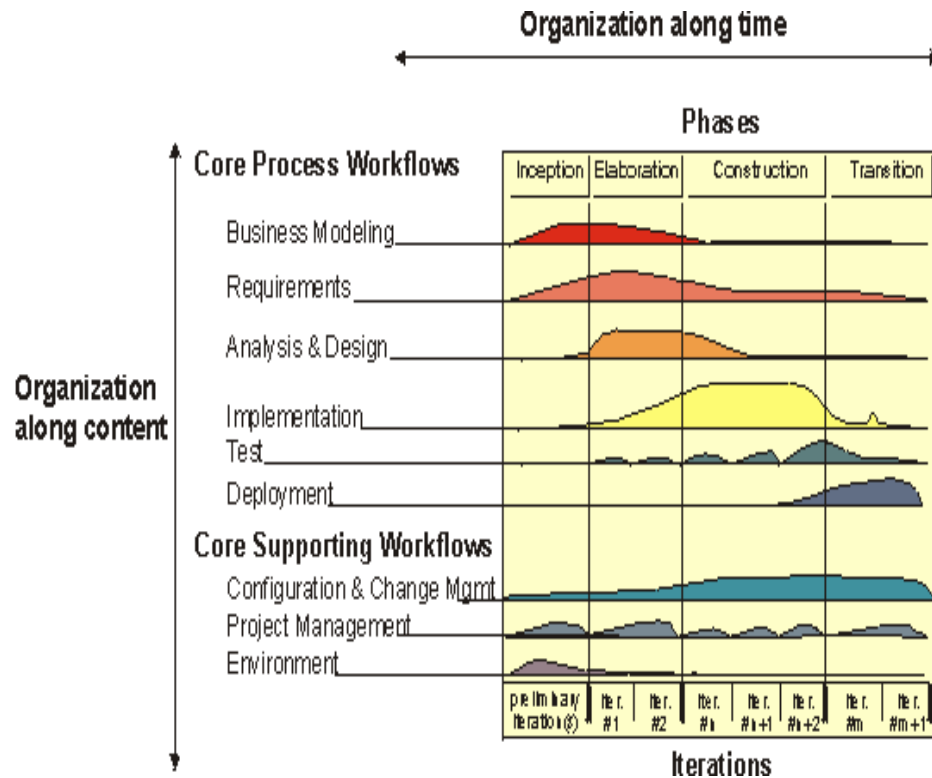


Figura 1. Fases e iteraciones de RUP

2.4 LENGUAJE DE MODELADO UNIFICADO (UML)

2.4.1 DEFINICIÓN

UML=Lenguaje de modelo unificado

UML (Unified Modeling Lenguaje) es un lenguaje estándar que sirve para escribir los planos del software, es decir, UML⁵ es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

UML permite interpretar sistemas mediante gráficos y texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo, ya que pueden ser interpretados por personas que no participaron en su diseño, por ser un lenguaje que permite especificar modelos concretos, no ambiguos y completos.

⁵ El Lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson. Addison Wesley Iberoamericana, 1999.

Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre tales sistemas como también para la representación conceptual y física de sistemas, UML está basado en un pequeño número de conceptos que fácilmente pueden aprender y aplicar la mayoría de los desarrolladores de mecanismos orientados a objetos.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación así como las relaciones entre objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos.

UML dispone de diferentes diagramas para representar gráficamente el sistema, las cuales intentan resaltar determinadas características que componen el software.

La arquitectura del UML define cuatro capas, definida a fin de cumplir con la especificación Meta Object Facility del OMG:

- Meta-metamodelo; define el lenguaje para especificar metamodelos.
- Metamodelo: define el lenguaje para especificar modelos.
- Modelo: define el lenguaje para describir un dominio de información.
- Objetos de usuario; define un dominio de información específico.

DIAGRAMAS UML

Un diagrama es una representación gráfica de una colección de elementos del modelado de sistemas, a menudo dibujada como un grafo convexo de arcos (relaciones) y vértices (otros elementos del modelo). Un diagrama no es un elemento semántico, un diagrama muestra representaciones de elementos semánticos del modelo, pero su significado no se ve afectado por la forma en que son representados.

Un diagrama está contenido dentro de un paquete. La mayoría de los diagramas UML y algunos símbolos complejos son grafos que contienen formas conectadas por rutas. La notación de UML está pensada para ser dibujada en superficies bidimensionales. Algunas formas bidimensionales son proyecciones de formas tridimensionales tales como cubos, pero todavía se representan como íconos en una superficie bidimensional.

Los diagramas que componen UML son los siguientes:

DIAGRAMAS DE CASO DE USO

Es una técnica para capturar información de cómo un sistema trabaja o cómo se espera que trabaje. Debe señalarse que no es específicamente una técnica orientada a objetos.

Un **Diagrama de Casos de Uso**⁶ muestra la relación entre los actores y los casos de uso del sistema. No permite modelar el sistema a su entorno o sea la comunicación del usuario y el sistema.

Los casos de uso no forman parte del **paradigma OO**, pero se usa para el modelado de especificaciones de requisitos.

Un diagrama de casos de uso es una representación gráfica de parte o el total de los actores y casos de usos del sistema, incluyendo sus interacciones. Todo el sistema tiene como mínimo un diagrama principal de casos de uso, que es una representación gráfica del entorno del sistema (Actores) y su funcionalidad principal (Casos de Uso).

Relaciones en un Diagrama de Casos de Uso

- **Comunicación:** Relación entre un actor y un caso de uso, denota la participación el actor en el caso de uso determinado.

⁶ El Lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson. Addison Wesley Iberoamericana, 1999.

- **Inclusión (use):** una instancia del caso de uso u origen, incluye también el comportamiento descrito por el caso de uso destino. <<include>> reemplazo al denominado <<uses>>.
- **Extensión:** El caso de uso origen extiende el comportamiento del, caso de uso destino <<extend>>.
- **Herencia:** El caso de uso origen hereda la especificación del caso de uso destino y posiblemente la modifica y/o amplía.

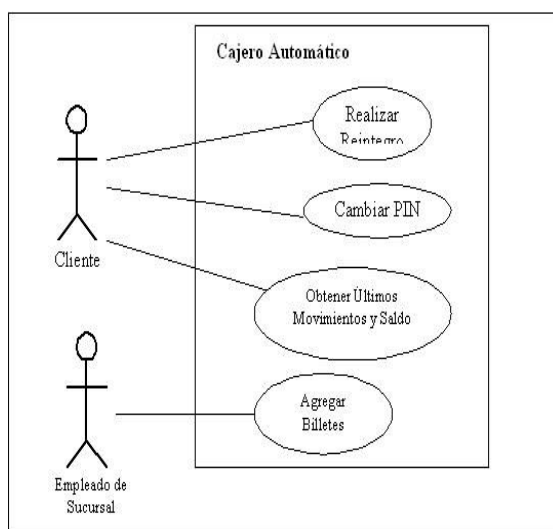


Figura 2. Casos de Uso

DIAGRAMAS DE CLASES

Una clase es la unidad básica que encapsula toda la información de un objeto. A través de ella podemos modelar el sistema de estudio.

El diagrama de clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones.

El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

MODELADO DE CLASES DE UN ESQUEMA LÓGICO DE BASE DE DATOS

Muchos de los sistemas que se modelan tendrán objetos persistentes, lo que significa que estos objetos podrán ser almacenados en una base de datos con el fin de poderlos recuperar posteriormente.

Los diagramas de clase UML son el súper conjunto de los diagramas E-R, una herramienta de modelado para el diseño lógico de base de datos utilizada con mucha frecuencia.

Mientras los diagramas E-R clásicos se centran solo en los datos, los diagramas de clases van un paso más allá permitiendo el modelado del comportamiento. En la base de datos física, estas operaciones lógicas normales se convierten en disparadores entre (triggers), consultas o procedimientos almacenados.

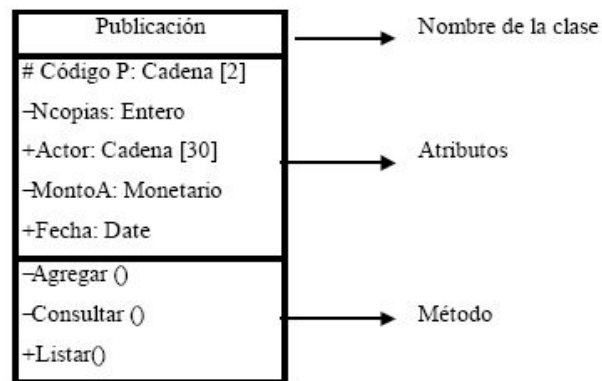


Figura 3. Clase

DIAGRAMA DE SECUENCIA

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores

participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

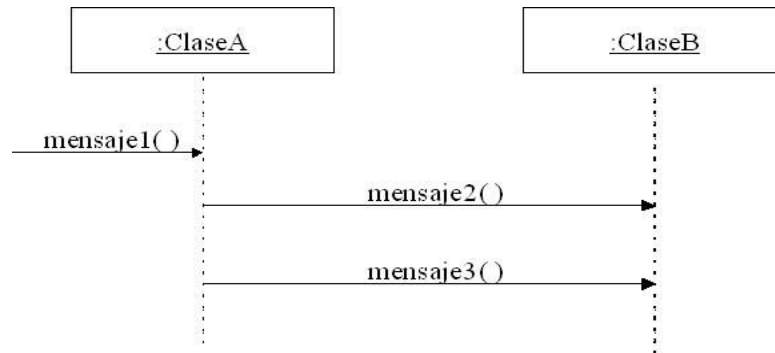


Figura 4. Diagrama de secuencia

Línea de Vida de un Objeto

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denoten la ejecución de métodos. El rectángulo de encabezado contiene el nombre del objeto y el de su clase en un formato nombre Objeto: nombre clase.

Activación

Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por si mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto.

Mensaje

El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.

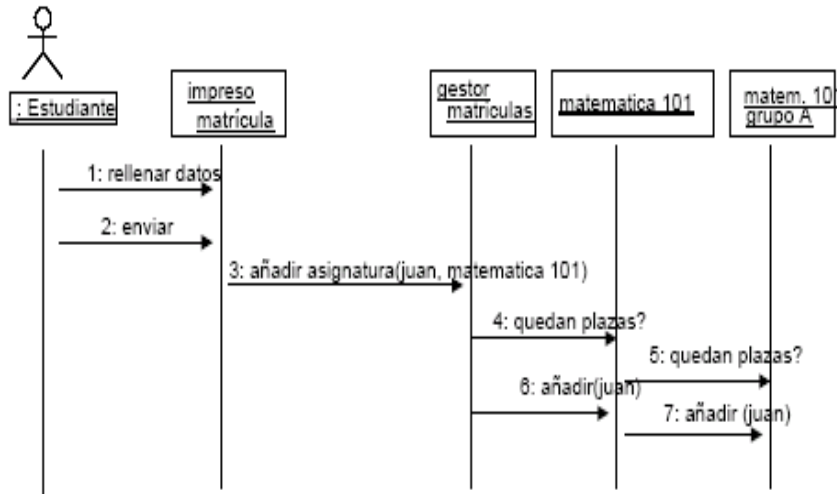


Figura 5. Ejemplo de diagrama de secuencia

Estereotipos

Los estereotipos son el mecanismo de extensibilidad incorporado más utilizado dentro de UML. Un estereotipo representa una distinción de uso. Puede ser aplicado a cualquier elemento de modelado, incluyendo clases, paquetes, relaciones de herencia, etc. Por ejemplo, una clase con estereotipo 'actor' es una clase usada como un agente externo en el modelado de negocio. Una clase patrón es modelada como una clase con estereotipo parametrizado, lo que significa que puede contener parámetros.

DIAGRAMA DE ACTIVIDADES

Los diagramas de actividades sirven fundamentalmente para modelar el flujo de control entre actividades.

El diagrama de actividades sirve para representar el sistema desde otra perspectiva, gráficamente un diagrama de actividades será un conjunto de arcos y nodos. Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo. Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases. Colaborando todas ellas para conseguir un mismo fin.

Contenido del diagrama de actividades

Básicamente un diagrama de actividades contiene:

- Estados de actividad.
- Estados de acción.
- Transiciones.
- Objetos.

Estados de actividad y estados de acción

La representación de ambos es un rectángulo en cuyo interior se representa bien una actividad o bien una acción.

Transiciones

Las transiciones reflejan el paso de un estado a otro, bien sea de actividad o de acción. Esta transición se produce como resultado de la finalización del estado del que parte el arco dirigido que marca la transición. Como todo flujo de control debe empezar y terminar en algún momento.

Dediciones

Se representa mediante una transición múltiple que sale de un estado, donde cada camino tiene un label distinto. Se representa mediante un diamante al cual llega la transición de estado inicial y del cual salen las múltiples transiciones de los estados finales.

2.4.2 OBJETIVOS DE UML

- UML es un lenguaje modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en

construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

2.4.3 PROCESO UNIFICADO (UP)

El proceso unificado utiliza el lenguaje de modelado UML, para preparar todos los esquemas de un sistema software.

Un proceso de desarrollo de software es el conjunto de actividades necesario para transformar los requisitos en un sistema software. Sin embargo, el Proceso Unificado no es más que un simple proceso; es un marco de trabajo genérico que puede especificarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitudes y diferentes tamaños de proyectos.

El desarrollo de un producto de software comercial supone un gran esfuerzo que puede durar entre varios meses, hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada. Es por esto que son mini-proyectos.

El proceso unificado se repite a lo largo de una serie de ciclo que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes. Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo a su vez, se divide en cuatro fases: inicio, elaboración, construcción y transición.

A través de una secuencia de modelos los implicados visualizan lo que esta sucediendo en esas fases, los desarrolladores pueden descomponer adicionalmente el trabajo. Al final se obtiene un conjunto de datos a partir del seguimiento del tiempo y esfuerzo consumido en cada fase. Estos datos son importantes en el control del progreso contrastado con las planificaciones.

2.5 ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS⁷

La modelación y Diseño Orientado a Objetos es una forma de pensar a cerca de los problemas utilizando modelos organizados sobre conceptos del mundo real. La construcción básica del modelo es el objeto, el cual combina toda la estructura de datos, como el comportamiento en una unidad única.

En primera instancia, se concibe el modelo de análisis para estudiar los aspectos abstractos esenciales del dominio de la aplicación, sin tomar en cuenta algún tipo de implementación eventual. Este modelo contiene objetos que se encuentran en el dominio de la aplicación, incluyendo la descripción de las propiedades de los objetos y de su comportamiento.

Posteriormente se toma las decisiones de diseño y sus detalles se incorporan al modelo para describir y optimizar la implementación. Los objetos del dominio de la aplicación forman la organización del modelo de diseño, pero son implementados en términos de objetos de dominio de la computadora. Finalmente, el modelo de diseño es implementado en un lenguaje de programación, base de datos o hardware.

⁷ Michel Blaha, James Rumbaugh, “Modelo y Diseño Orientado a Objetos”.

Los objetos de dominio de aplicación así como los objetos de dominio de la computación pueden ser dominados, diseñados e implementados utilizando los mismos conceptos y notación orientados a objetos. Una notación semejante es utilizada desde el análisis hasta el diseño y la implementación, de forma tal que la información adicionada en una etapa del desarrollo no se tendrá que traducir o perder en la siguiente etapa.

Las características principales del Diseño Orientado a Objetos, contempla:

- ❖ Los objetos son abstracciones del mundo real o entidades del sistema que se administran entre ellas mismas.
- ❖ Los objetos son independientes y encapsulan el estado y la representación de información.
- ❖ La funcionalidad del sistema se expresa en términos de servicios de los objetos.
- ❖ Las áreas de datos compartidas son eliminadas. Los objetos se comunican mediante paso de parámetros.
- ❖ Los objetos pueden estar distribuidos y pueden ejecutarse en forma secuencial o en paralelo.

Las ventajas del Diseño Orientado a Objetos, radican esencialmente en:

- ❖ Fácil de mantener, los objetos representan entidades auto-contenidas.
- ❖ Los objetos son componentes reutilizables.
- ❖ Para algunos sistemas, puede haber un mapeo obvio entre las entidades del mundo real y los objetos del sistema.

La **programación orientada a objetos** o **POO (OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas de ordenador. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

Los objetos son entidades que combinan *estado (atributo)*, *comportamiento (método)* e *identidad*:

- El *estado* está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- El *comportamiento* está definido por los procedimientos o métodos con que puede operar dicho objeto, es decir, qué operaciones se pueden realizar con él.
- La *identidad* es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).

La programación orientada a objetos, es necesaria en la vida de un programador ya que con esta visualizamos las actividades de cada una de esas. Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento.

Los **métodos (comportamiento)** y **atributos (estado)** están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a alguno de ellos. Hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro. De esta manera se estaría realizando una **programación estructurada camuflada** en un lenguaje de programación orientado a objetos.

Conceptos fundamentales

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

- **Clase**: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- **Herencia**: (por ejemplo, herencia de la clase C a la clase D) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de OOP.
- **Objeto**: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.
- **Método**: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- **Evento**: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir

como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

- **Mensaje**: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- **Propiedad o atributo**: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.
- **Estado interno**: es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.
- **Componentes de un objeto**: atributos, identidad, relaciones y métodos.
- **Identificación de un objeto**: un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

En comparación con un lenguaje imperativo, una "variable", no es más que un contenedor interno del atributo del objeto o de un estado interno, así como la "función" es un procedimiento interno del método del objeto.

Características de la POO

Existe un acuerdo acerca de qué características contempla la "orientación a objetos", las características siguientes son las más importantes:

- **Abstracción**: Denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.

- **Encapsulamiento**: Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- **Principio de ocultación**: Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una *interfaz* a otros objetos que específica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo**: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.
- **Herencia**: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Cuando un objeto hereda de más de una clase se dice que hay *herencia múltiple*.

2.6 HERRAMIENTAS Y TECNOLOGÍAS

Para el desarrollo del sistema se toma en cuenta las siguientes herramientas y tecnologías.

2.6.1 ENTERPRISE ARCHITECT (EA)

Enterprise Architect provee modelado del ciclo de vida completo para:

- Sistemas de negocio e IT
- Ingeniería de software y sistemas
- Desarrollo en tiempo real y embebido

Con capacidades de gestión de requisitos, Enterprise Architect lo ayuda a trazar especificaciones de alto nivel a modelos de análisis, diseño, implementación, pruebas y mantenimiento, usando UML, SysML, BPMN y otros estándares abiertos para modelado.

Enterprise Architect es una herramienta gráfica multiusuario diseñada para ayudarle a su equipo a construir sistemas robustos y mantenibles. Y usando facilidades de incorporadas de reportes y documentación, de alta calidad.

Velocidad, estabilidad y desempeño

Rapidez, Enterprise Architect tiene un excelente desempeño y rapidez al cargar modelos grandes en segundos.

Con un repositorio de alto desempeño, Enterprise Architect facilita que los grandes equipos compartan la misma visión de la empresa.

Con capacidades de control de versiones bien integradas, Enterprise Architect también permite equipos distribuidos globalmente para colaborar eficientemente en proyectos compartidos.

Trazabilidad de punta a punta

Enterprise Architect provee trazabilidad completa desde los modelos de requisitos, análisis y diseño, hasta la implementación y despliegue.

La verificación y validación efectiva y el análisis del impacto inmediato son posibles a través del ciclo de vida completo, usando capacidades tales como la matriz de relaciones y la vista de jerarquía de Enterprise Architect.

2.6.2 LA CAPA DE PRESENTACIÓN : EL PATRÓN MVP

El patrón Modelo Vista Presentador (MVP) es un patrón derivado del patrón Modelo Vista Controlador (MVC) que nos ayuda a ofrecer una clara separación entre la vista, el modelo y el controlador.

La clave del patrón MVP es una estricta regulación de la interacción entre la vista y el controlador, aunque en el patrón MVP, al controlador se le conoce como presentador.

La vista y el modelo de datos son claramente separados a través de un contrato (interfaz) que expone la vista y al cual el presentador accede de modo polimórfico.

Resumiendo, podemos decir que el patrón MVP es una mejora del patrón MVC basado en tres características:

- La vista no conoce el modelo.
- El presentador es independiente de la tecnología de interfaz de usuario.
- La vista y el presentador es testeable puesto que esta basada en un contrato.

Mediante el uso de un contrato en la vista, podemos testear la propia vista y el presentador de modo independiente :

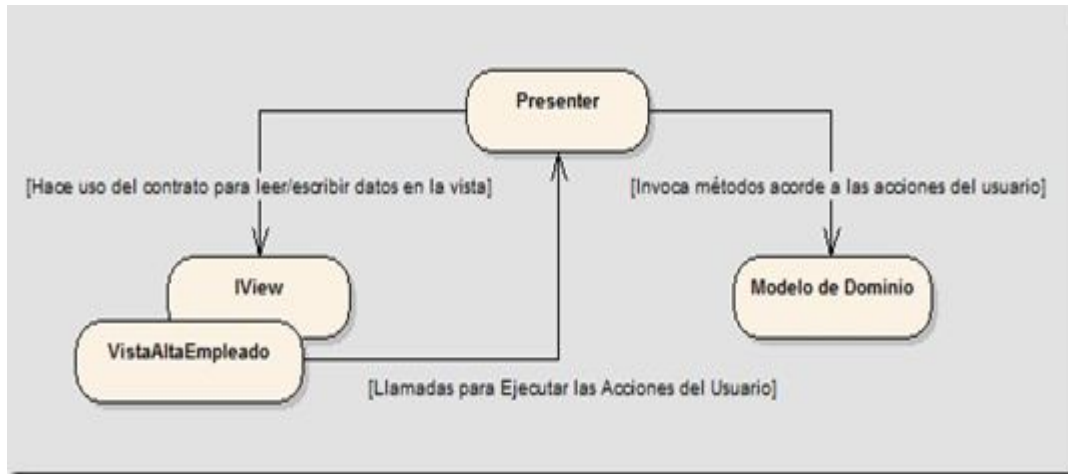


Figura 6. Vista de presentador de modo independiente

En aplicaciones web ASP.NET, las vistas se representan a través de formularios web, y en aplicaciones Windows a través de formularios Windows.

En la implementación de la vista, tenemos dos opciones :

1. Una vista pasiva
2. Una vista activa

Si optamos por una vista pasiva, obtendremos un mayor grado de testeabilidad por que la lógica representada por el contrato se reducirá al mínimo, sin embargo aumentaremos la complejidad en el presentador.

Si optamos por una vista activa que contiene lógica como el enlace y el formateo de datos, necesitaremos implementar una funcionalidad extra que convertirá la vista en menos testeable.

El presentador es el componente encargado de “presentar” las acciones del usuario al sistema y obtener la respuesta del mismo.

El presentador se ubica entre la vista y el modelo, y actúa de la siguiente manera:

1. Recibe datos de la vista

2. Convierte los datos de la vista en acciones que se ejecutan contra el sistema
3. Con la respuesta del sistema, actualiza los datos de la vista.

Las acciones del usuario que requieran navegación a otra vista, deberán ser delegadas en el presentador, el cual será responsable de aplicar la redirección a la vista correspondiente.

El patrón MVP, es un patrón que no puede ser implementado sin haber hecho un análisis previo que determine que miembros deben formar parte del contrato de cada vista.

Una vez que tenemos claro el contrato que debe implementar cada vista, podemos desarrollar la lógica de presentación de forma paralela al diseño gráfico.

La mejor forma de crear una clase presentador es con los siguientes miembros:

- Un método de inicialización
- Un método de actualización de la vista
- Un conjunto de métodos que ejecutan métodos de la capa de servicios / negocio.

2.6.3 ER/STUDIO EMBARCADERO

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejas.

Funcionalidades:

- ❖ Capacidad fuerte en el diseño lógico.
- ❖ Sincronización bidireccional de los diseños lógico y físico.

- ❖ Construcción automática de Base de Datos.
- ❖ Reingeniería inversa de Base de Datos.
- ❖ Documentación basada en HTML.
- ❖ Un Repositorio para el modelado.

ER/Studio Enterprise Portal provee los siguientes beneficios:

- ❖ Una interfaz de maquinaria de búsqueda que resulta inmediatamente familiar a los usuarios finales de negocio.
- ❖ Reporteo de autoservicio que incrementa el conocimiento y productividad del trabajador con documentación baja demanda.
- ❖ Distribución vía Web que permite estandarización global y la utilización de activos de datos a cualquier organización.
- ❖ Instalación e implementación simples que permite la comunicación con los metadatos en minutos.

Numero de Base de Datos que Soporta ER/Studio

ER/Studio soporta el muy popular SQL y Bases de Datos de escritorio, incluyendo:

- Oracle 7.3® , 8.x & 9i

Oracle Database ofrece una solución optimizada de base de datos para aquellas implementaciones que requieren escalabilidad, confiabilidad y alto desempeño empresarial.

La base de datos Oracle se integra por completo con las características avanzadas del sistema operativo Windows y el hardware subyacente, como el soporte NUMA y de Paginación. Oracle ofrece desempeño de primer nivel a través del soporte de memorias extendidas, archivos binarios y de gran volumen, y *grid computing*.

La base de datos Oracle está certificada para funcionar en sistemas operativos Windows de 32 bits y 64 bits.

La base de datos Oracle de 32 bits es soportada por el sistema Windows de 32 bits con hardware estándar de x86, con inclusión de Windows Vista. En el sistema Windows de 64 bits, Oracle de 64 bits se ejecuta en los sistemas operativos Windows x64 (AMD64/EM64T) y Windows Itanium. El hardware de 64 bits ofrece mayor escalabilidad y desempeño respecto de los sistemas de 32 bits.

- **Sybase® System 11.9.2, 12.x & 12.5**

Sybase® es una poderosa herramienta basada en una tecnología orientada a alinear el negocio y la Tecnología de Información (IT); es una solución de modelado y diseño empresarial que colabora en la implementación efectiva de la arquitectura empresarial y brinda técnicas poderosas de análisis y diseño durante todo el ciclo de vida de desarrollo del proyecto con gestión de metadatos, funciones de análisis de impacto y verdadero repositorio empresarial.

Sybase® adicionalmente ofrece técnicas avanzadas para mejorar la calidad, integridad y funcionamiento de los sistemas DW, cumple con las necesidades de DW a través de técnicas como el manejo de requerimientos, modelado de procesos, modelado conceptual, lógico y físico de datos, modelado multidimensional con jerarquías, ETL, EII, y mapeo de replicación de fuentes externas.

Ofrece diagramas multidimensionales que documentan el ambiente DW con objetos multidimensionales enlazados a tablas físicas y con la posibilidad de mostrar jerarquías, jerarquías virtuales y mucho más. Una definición de DW tampoco podría estar completa sin la definición de las transformaciones de la fuente de datos que será guardada en ella. El Liquidity Model de PowerDesigner (ILM), documenta todos los aspectos en el movimiento de información. Fuentes de datos inicial y destino, múltiples transformaciones, publicaciones y suscripciones sirven para documentar cualquier ETL, EII o proceso de replicación usado en DW y Datamarts.

- **Microsoft® SQL Server 6.5, 7 & 2000**

Microsoft SQL Server constituye un lanzamiento determinante para los productos de bases de datos de Microsoft. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

Objetivos del Diseño de SQL Server

Los clientes están buscando soluciones para sus problemas de negocios. La mayoría de las "soluciones" de bases de datos solamente traen múltiples niveles de costos y complejidad. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

SQL Server disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo. También libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática.

Mejoras en programabilidad y lenguaje.

- Triggers, procedimientos almacenados (autoexec), disparador de eventos antes y después de conexiones.

- Procedimientos almacenados extendidos (funciones definidas por el usuario) utilizando C/C++.
- Cursores basados en el motor con scrolling hacia adelante y atrás; posicionamiento absoluto y relativo.
- Sentencias DLL permitidas dentro de transacciones.
- Transacciones distribuidas dentro de interfaces DB-Library, ODBC, Transact-SQL, XA y OLE Transaction.
- Procedimientos almacenados OLE Automation.

- **Informix® OnLine and SE**

Es una base de datos transaccional para las prácticas actuales del ebusiness, que proporciona la infraestructura de datos crítica necesaria para gestionar su negocio.

Características y Beneficios

- Alto desempeño y facilidad de administración.
- Bajo costo
- Uso eficiente de los recursos de cómputo.
- Alta accesibilidad y sencillez para su manejo
- Capacidad de gestión, ampliación y rendimiento.
- Compatibilidad con otras aplicaciones.
- Actualizaciones a nuevas versiones y otras plataformas (servicios Web y Linux).
- Maneja grandes volúmenes de datos complejos a velocidades altas.

CAPÍTULO III

ANÁLISIS

3.1 MARCO DEL PROBLEMA: OBJETIVO DE ESTUDIO

El objetivo principal de la Escuela Superior de Formación de Maestros “Juan Misael Saracho” de Canasmoro, es de realizar una mejor administración de sus recursos humanos.

3.1.1 MODELO DE ACTIVIDADES DEL SISTEMA DE INFORMACIÓN DE RECURSOS HUMANOS

Un sistema de información, es, por definición, un Sistema mediante el cual los datos se obtienen, procesan y se transforman en información, de manera esquematizada y ordenada, para que sirva de ayuda en el proceso de toma de decisiones. El sistema de información recibe entradas que son procesadas y transformadas en salidas bajo la forma de informes, documentos, índices, listados, medidas estadísticas de posición o de tendencias, etc., el objetivo final de un sistema de información es la de suministrar a las jefaturas información acerca del personal.

3.1.2 JEFATURA DE PERSONAL

Tanto el jefe de personal y los empleados tienen que consultar el manual de funciones y el reglamento interno de la institución para tomar decisiones y acciones para realizar eficientemente y efectivamente sus tareas cotidianas, al no tenerlos a disponibilidad provocan retrasos en la toma de decisiones del mismo.

3.1.2.1 DIRECTOR ADMINISTRATIVO

El/la director /a Administrativo/a es, el que se encarga de mandar a jefatura de personal registros de órdenes, capacitaciones del personal, seguros, permisos, retiros e incentivos, al no llevar un control adecuado de estos registros son factibles a modificaciones y pérdidas.

3.1.2.2 JEFE DE PERSONAL

Función principal

Organizar y coordinar actividades de administración de personal y labores de transcripción y archivo.

Funciones secundarias:

- Organizar, dirigir, controlar a todo el personal de la institución y aplicar las disposiciones legales inherentes a la función.
- Tener a su cargo los fóldeos del personal de la institución que deben estar permanentemente actualizados, debiendo incluir todos los registros necesarios del personal.
- Programar el rol anual de vacaciones del personal, debiendo tomar las previsiones necesarias para suplir las funciones del empleado que hago uso de este beneficio.
- Organizar y supervisar las acciones tendientes a regularizar las vacaciones, baja médica, permisos y otros.
- Apoyar en las tareas de difusión, revisión y actualización de manuales, estatutos y reglamentos internos de la Institución.

- Controlar la asistencia, entradas y salidas del personal, mediante las tarjetas de asistencia y las papeletas de salida que deben ser debidamente autorizadas por Dirección Administrativa.
- Revisar y elaborar en forma mensual las planillas de días trabajados por el personal.
- Revisar las tarjetas de asistencia en forma diaria y en caso de inasistencia sellar la tarjeta indicando el motivo de la inasistencia a su fuente de trabajo.
- Controlar la asistencia del personal e informar a la Dirección Administrativa sobre faltas, licencias u otras causales de inasistencia o atraso.
- Realizar toda tarea que sea necesaria para el buen funcionamiento de la institución.
- Mantener informado a sus superiores sobre la ausencia de personal previa verificación de la papeleta de salida.
- Revisar las tarjetas de asistencia en forma mensual, para determinar los días trabajados, retrasos y faltas.

3.2 VISTA DE CASOS DE USO

ACTORES INVOLUCRADOS

Un actor es un usuario del sistema que necesita o usa alguno de los casos de uso. Un usuario puede jugar más de un rol. Un solo actor puede actuar en muchos casos de uso; recíprocamente, un caso de uso puede tener varios actores. Los actores no necesitan ser humanos pueden ser sistemas externos que necesitan alguna información del sistema actual.

Los actores indicados en la etapa de análisis de requerimientos del sistema realizado en la Escuela Superior de Formación de Maestros “Juan Misael Saracho” son:

1. Administrador del sistema
2. Funcionarios

Administrador del Sistema.- Es la persona encargada de administrar los usuarios y roles, los privilegios de acceso al sistema, administrar su propia clave y además añadir, modificar, habilitar, controlar la asistencia, elaborar planillas de asistencia, registrar datos de personas nuevas en la institución, elaborar informe mensual de asistencia, recibir y registrar solicitudes de permisos e inasistencia.

Funcionarios.- Son las personas que se encargan de asistir diariamente a su fuente laboral, dando a conocer su asistencia, marcando su ingreso y salida.

3.3 GENERALIDADES

De la jornada de trabajo: Los horarios de ingreso y salida son para docentes 7:30 a.m. a 15:30 p.m., funcionarios administrativos 8:00 a.m. a 16:00 p.m. y personal de servicio 6:00 a.m. a 18:00 p.m.

El funcionario esta obligado a concurrir a la hora exacta al lugar de sus funciones, conforme a horarios establecidos, y en este lapso no podrán abandonar sus labores sin autorización escrita del inmediato superior, previo llenado de papeleta de salida.

En los horarios establecidos de entrada se reconoce una tolerancia de 30 minutos, los funcionarios estarán sujetos a la sanción correspondiente con la siguiente relación de descuento de sus haberes básicos:

- De 0 a 30' cero descuento
- De 30' a 45' medio día de descuento
- De 46' a 60' un día de descuento
- De 61' a 90' dos días de descuento
- De 90' adelante 3 días de descuento

Aguinaldo

El aguinaldo es la gratificación que se otorga a favor de los funcionarios con motivo de fiestas de fin de año, este aguinaldo esta sujeto a las leyes de trabajo y del funcionario público y reglamentos del magisterio.

El monto del aguinaldo es equivalente a un mes de sueldo por año completo de servicio en el periodo respectivo, los trabajadores que no hubieran cumplido un año continuo de servicio en el periodo correspondiente, percibirán el aguinaldo por duodécimas de acuerdo a los meses trabajados.

Vacación

Todos los funcionarios tienen el derecho irrenunciable y de uso obligatorio a su descanso anual. La vacación tiene el fin fundamental de garantizar la conservación de la salud física y mental de todos los empleados, como requisito indispensable para lograr eficiencia en el cumplimiento de sus deberes y responsabilidades.

Las vacaciones serán usadas de acuerdo al cronograma anual de vacaciones elaboradas por Dirección Administrativa, detallando el mes y los días de los cuales cada uno de los funcionarios tomará la vacación respectiva, coordinando el normal desarrollo de las actividades de la empresa; conforme a la siguiente escala:

- De 1 año y un día, hasta 3 años de trabajo 15 días hábiles
- De 3 años y un día, hasta 5 años 20 días hábiles
- De 5 años y un día en adelante 30 días hábiles

Si el funcionario por razones de extrema necesidad solicita hacer uso de sus vacaciones en tiempo distinto al programado, la institución podrá atender favorablemente dicha petición siempre que su ausencia no altere la normalidad de los servicios de la sección donde cumple funciones.

Antes de hacer uso de la vacación cada funcionario, está obligado a dejar sus trabajos al día, en completo orden y a satisfacción del inmediato superior.

De las licencias

El objeto de la institución es establecer los principios que debe seguirse en materia de licencias por ausencia del personal de trabajo, las licencias se clasifican en:

- **Vacación anual;** es el descanso periódico con goce de haber al que tienen derecho todos los funcionarios por el tiempo de servicio en la institución.

La vacación anual será concedida a petición del funcionario mediante solicitud escrita, previa firma de autorización de Dirección Administrativa, sin embargo ambos tienen la facultad de conceder o postergar dicha vacación, en cuanto a la fecha solicitada, sujeta a las necesidades y normal desenvolvimiento del trabajo.

- **Enfermedad o accidente;** la licencia por enfermedad o accidente significa que el permiso al que tienen derecho los empleados cuando se hallan incapacitados, para realizar un trabajo.

La ausencia motivada por este concepto debe ser comunicada por el funcionario dentro del primer día de inasistencia al trabajo.

El periodo de licencia por enfermedad o accidente se computará desde la fecha del parte de baja hasta la fecha de parte del día.

- **Natalidad;** las licencias por natalidad y lactancia se otorgarán de conformidad a las disposiciones de Código de Seguridad Social y disposiciones afines.

El periodo de licencia por natalidad se computará desde la fecha del parte de baja médica, hasta la fecha de parte del alta.

La licencia por lactancia será de una hora diaria previo acuerdo en el jefe inmediato superior y no será acumulada ni compensable.

- **Comisión**; la licencia por comisión oficial, es aquella que la institución otorga al funcionario para que realice diligencias para la institución fuera de sus instalaciones.

La licencia en comisión por capacitación significa el permiso que la empresa concede a un funcionario por razones de estudio o de práctica de especialización.

- **Licencia por otros motivos**; significa el permiso con o sin remuneración otorgado por circunstancias excepcionales que debe ser aprobado por Dirección Administrativa.

Será autorizado por periodos determinados debidamente justificados, en caso de que la ausencia del funcionario exceda el tiempo autorizado, la falta se considera abandono de trabajo.

En casos excepcionales, la institución podrá conceder licencia a los funcionarios, siempre que existan razones justificadas y debidamente comprobadas. La licencia podrá ser otorgada:

- Con goce de haberes
- Con cargo a vacación

La licencia con goce de haberes será concedida en los siguientes casos:

- a) Por fallecimiento de padres, hermanos, cónyuges, hijos, nietos, suegros por dos días.
- b) Por contraer nupcias, tres días.
- c) Por asistencia a cursos o capacitación aprobados por la institución, por el lapso de duración del curso.
- d) Por cumpleaños del funcionario ½ día.

Las licencias con goce de haberes, serán autorizadas por la institución previa justificación y comprobación de las causas registradas en la solicitud de licencia.

Para el desarrollo del sistema se tiene las siguientes vistas:

- Vista de casos de uso
- Vista de iteración
- Vista estática

3.4 DIAGRAMAS DE CASOS DE USO

3.4.1 Diagrama de casos de uso general.

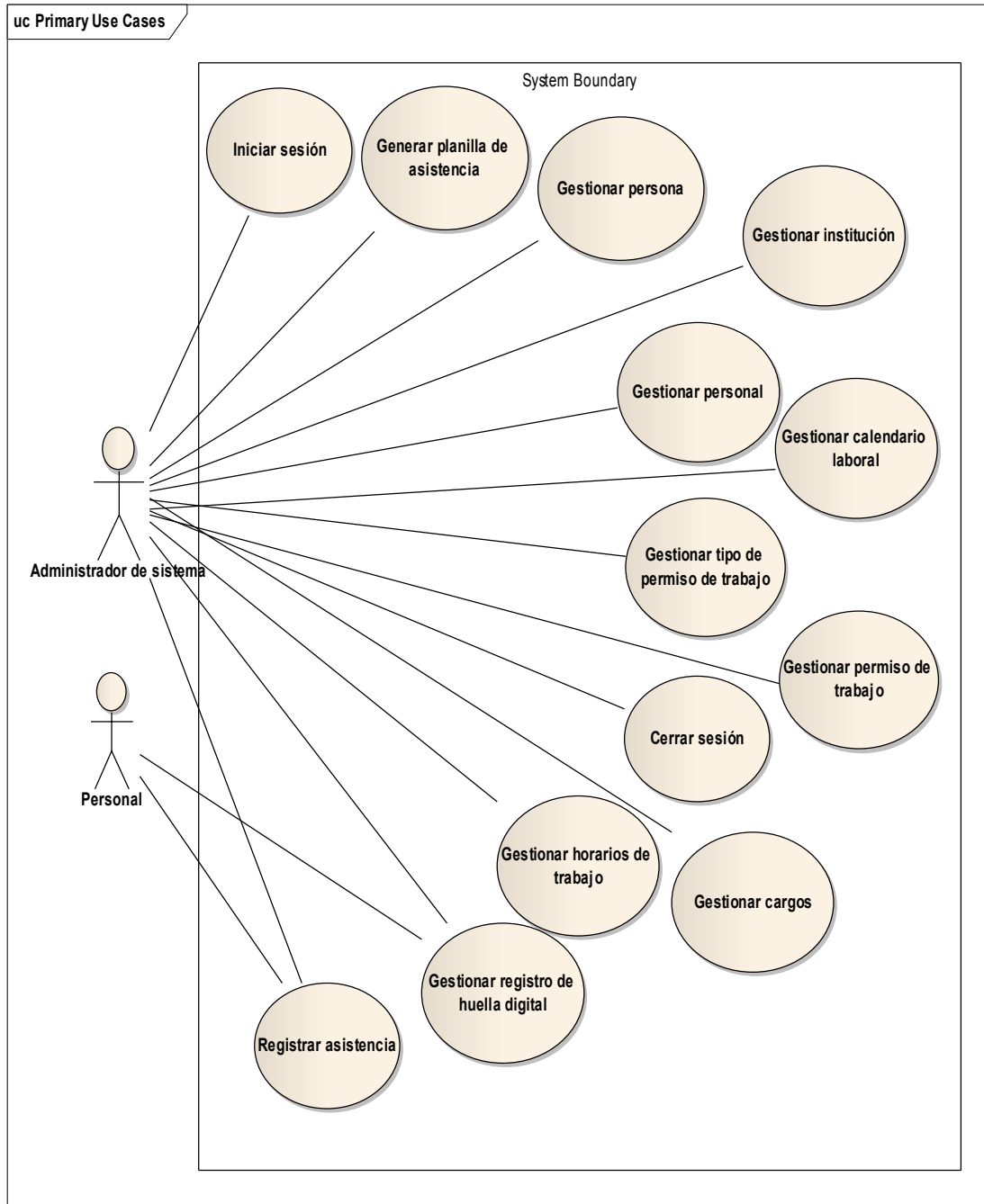


Figura 7. Diagrama de Casos de uso: general

I. MÓDULO DE SISTEMA

3.4.2 Diagrama de Casos de Uso: Iniciar sesión

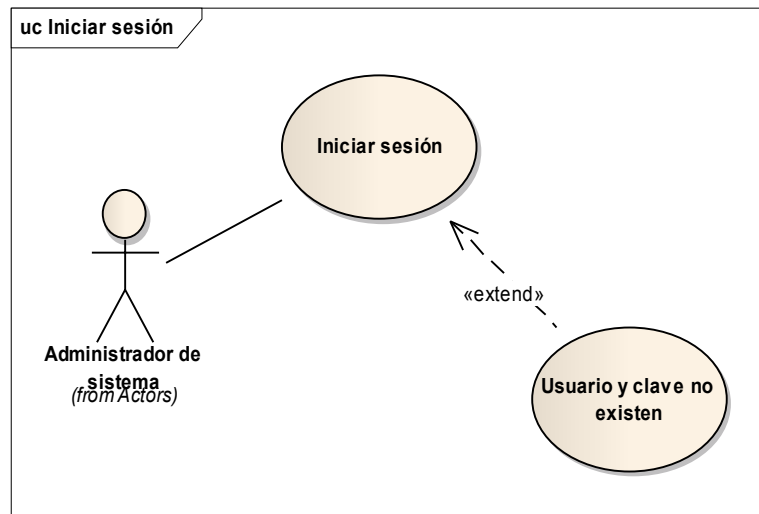


Figura 8. Diagrama de Casos de Uso: Iniciar sesión

II. MODULO DE ASISTENCIA

3.4.3 Diagrama de Casos de Uso: Registrar asistencia por huella digital

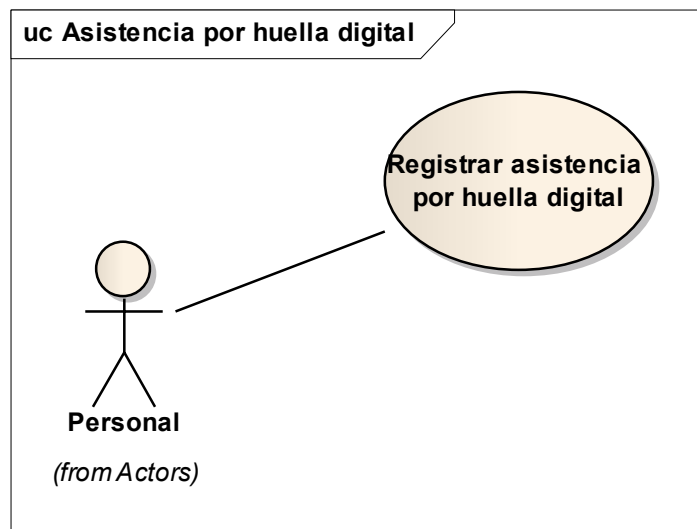


Figura 9. Diagrama de Casos de Uso Registrar asistencia por huella digital

3.4.4 Diagrama de Casos de Uso: Registrar asistencia por código de personal

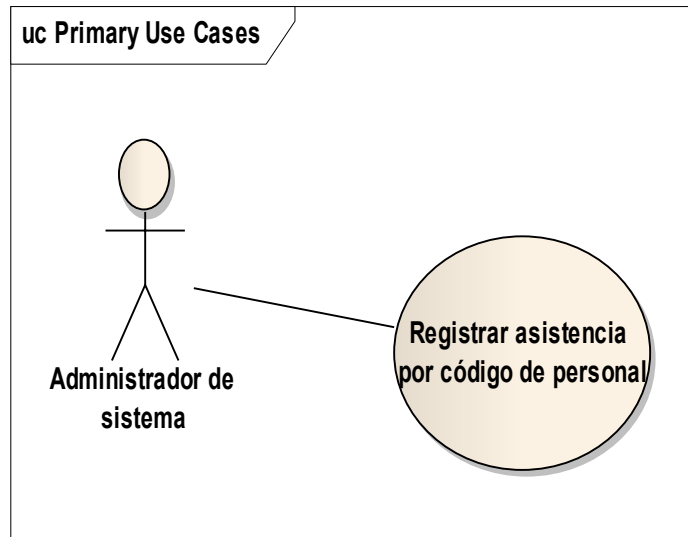


Figura 10. Diagrama de Casos de Uso Registrar asistencia por código de personal

3.4.5 Diagrama de Casos de Uso: Gestionar calendario laboral

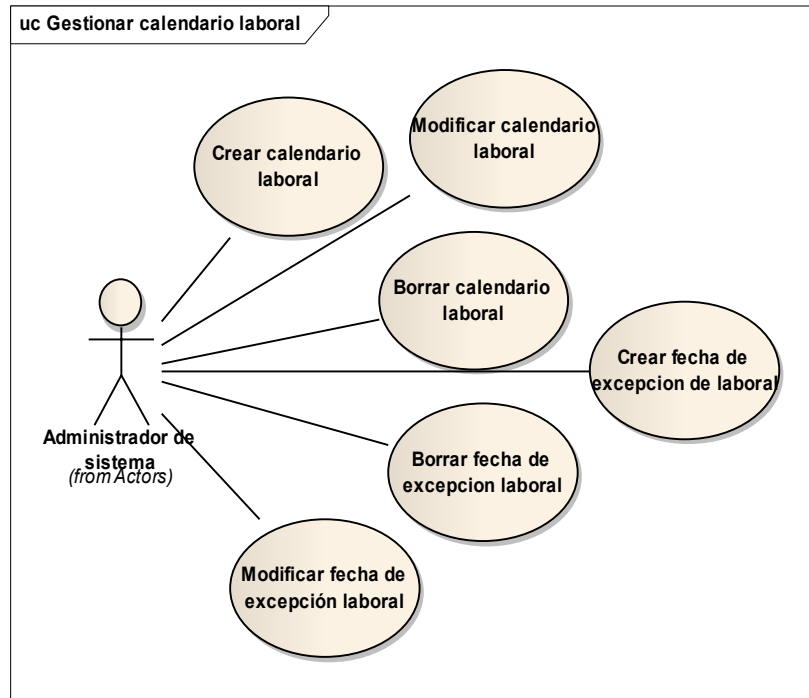


Figura 11. Diagrama de Casos de Uso: Gestionar calendario laboral

3.4.6 Diagrama de Casos de Uso: Gestionar permiso a personal

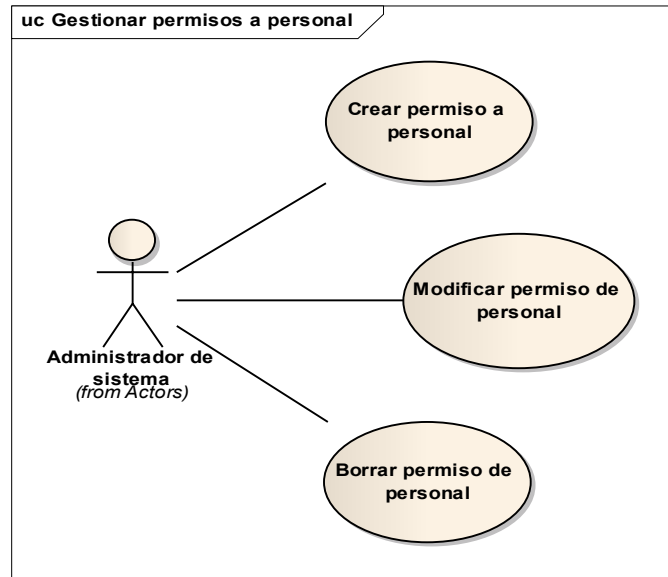


Figura 12. Diagrama de Casos de Uso: Gestionar permiso a personal

3.4.7 Diagrama de Casos de Uso: Gestionar tipo de permiso a personal

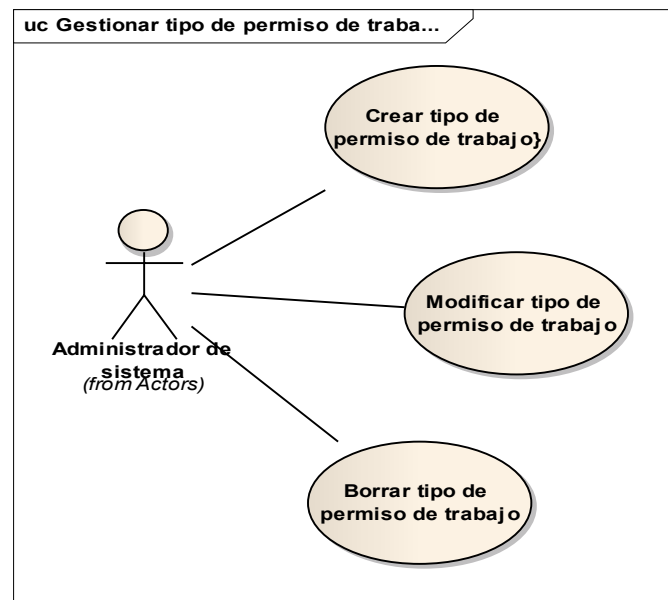


Figura 13. Diagrama de Casos de Uso: Gestionar tipo de permiso a personal

3.4.8 Diagrama de Casos de Uso: Gestionar horario de trabajo

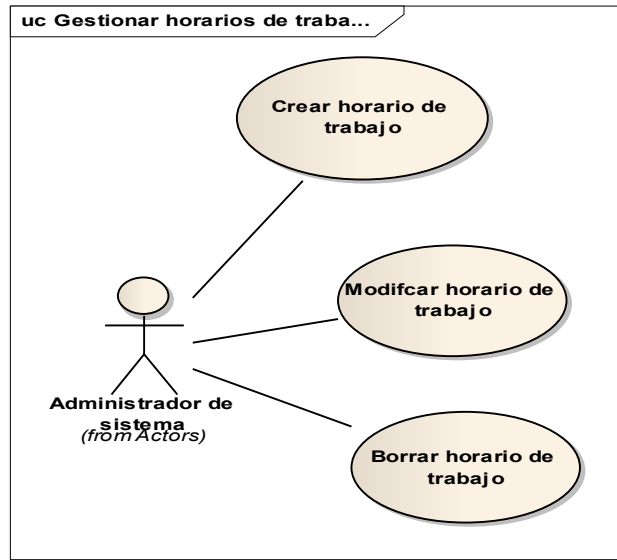


Figura 14. Diagrama de Casos de Uso: Gestionar horario de trabajo

3.4.9 Diagrama de Casos de Uso: Gestionar registro de personal

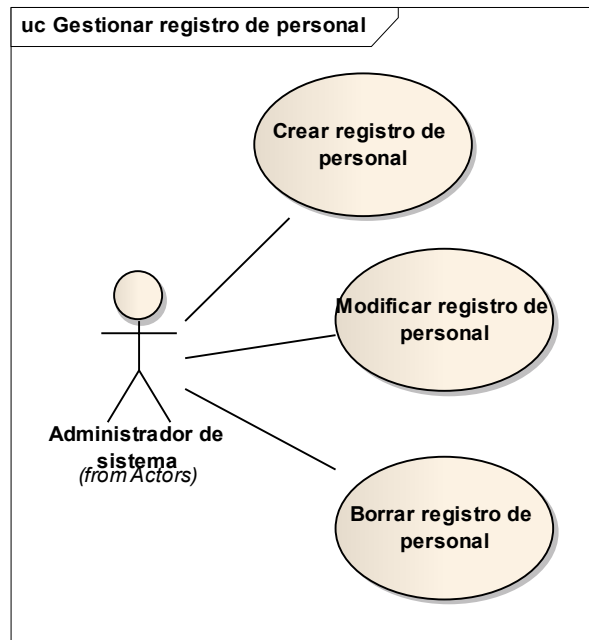


Figura 15. Diagrama de Casos de Uso: Gestionar registro de personal

3.4.10 Diagrama de Casos de Uso: Gestionar huella digital

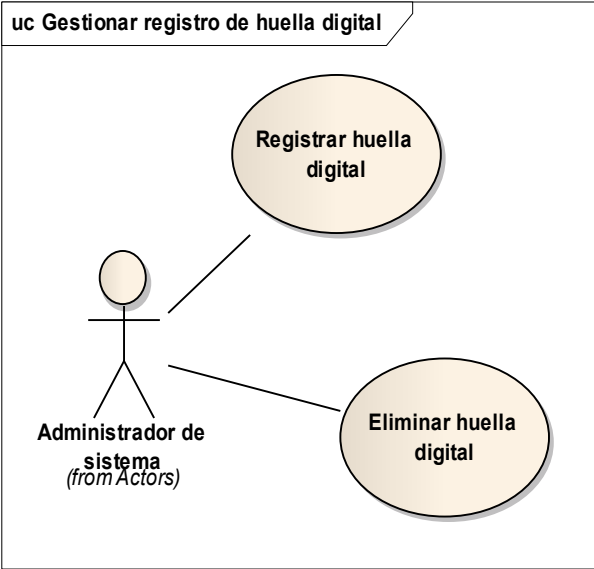


Figura 16. Diagrama de Casos de Uso: Gestionar huella digital

3.4.11 Diagrama de Casos de Uso: Gestionar cargo

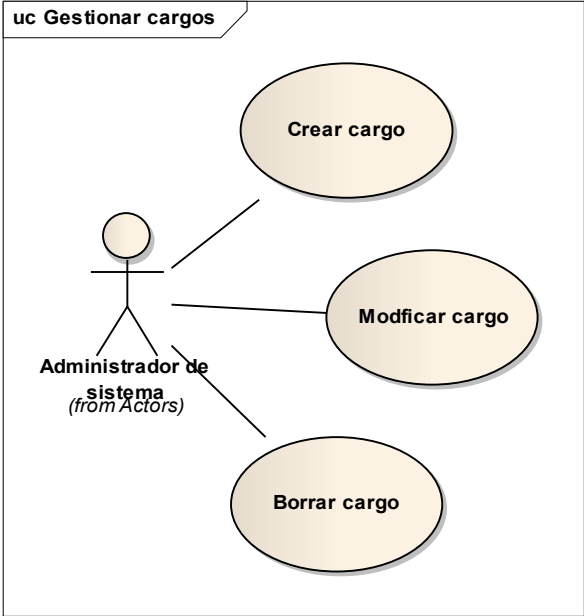


Figura 17. Diagrama de Casos de Uso: Gestionar cargo

III. MODULO GENERAL

3.4.12 Diagrama de Casos de Uso: Gestionar persona

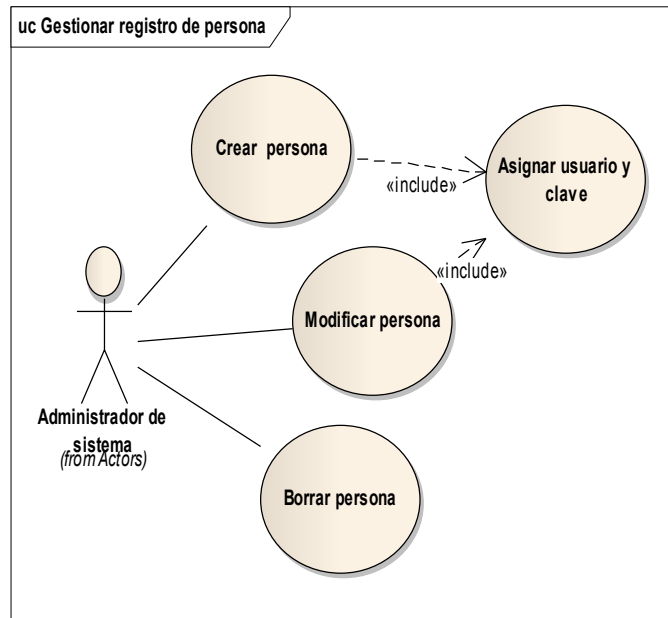


Figura 18. Diagrama de Casos de Uso: Gestionar persona

IV. MODULO DE REPORTES

3.4.13 Diagrama de Casos de Uso: Generar reporte de asistencia

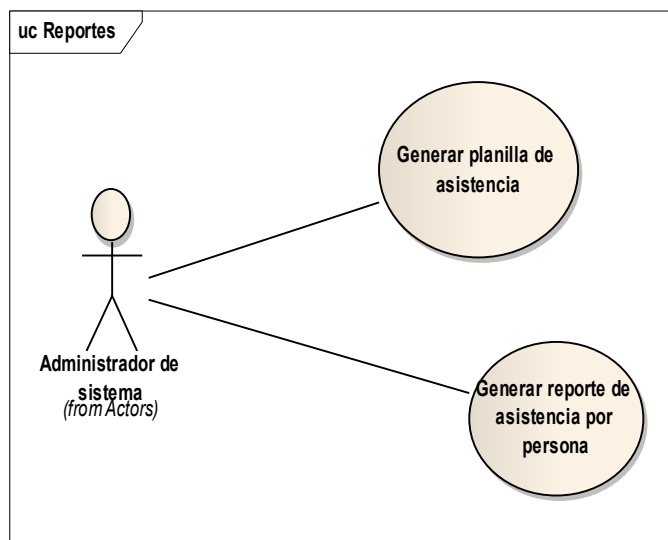


Figura 19. Diagrama de Casos de Uso: Generar reporte de asistencia

3.5 DESCRIPCIÓN DETALLADA DE LOS CASOS DE USO

I. MÓDULO DEL SISTEMA

Caso de Uso: Iniciar sesión	
<i>Alcance</i>	Registro de asistencia.
<i>Nivel</i>	Ingreso del usuario al sistema
<i>Actor primario</i>	Usuario del sistema
<i>Interesados e intereses</i>	Administrador de sistema.
<i>Pre-condiciones</i>	1.- El sistema debe tener sesión cerrada
<i>Pos-condiciones</i>	1.- El usuario de sistema se autentica y obtiene permisos.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	Eventos del sistema
<p>1.-El usuario pide al sistema iniciar sesión.</p> <p>3.- El usuario ingresa los datos y pide ingresar al sistema.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema muestra la interfaz para el ingreso de datos para autenticación.</p> <p>4.- El sistema valida y autentica al usuario con los datos ingresados, habilita las opciones de sistema que le corresponden y luego envía un mensaje de confirmación al usuario.</p>

Extensiones (Flujo alternativo).

*.- El sistema falla en cualquier momento.

1. El sistema informa al usuario del error y vuelve al estado inicial.
2. El usuario recibe el mensaje de error.

4.a.- El usuario y clave no se encuentra registrado

1.a.- El sistema informa al usuario que su usuario y clave ingresado no se encuentran registrado.

Vuelve al paso 3

1.b.-Es la tercera vez que falla en la validación y autenticación.

1.-El sistema se bloquea por 3 min.

Vuelve al paso 3.

Requerimientos especiales. (Son requerimientos no funcionales relacionados con en el sistema)

- El inicio de sesión podría realizar mediante lector de huella digital.

Tecnología y lista de variación de datos. (Variaciones de métodos de entrada y salida, y formatos de los datos.)

- El inicio de sesión puede realizarse mediante el ingreso del usuario y clave o el ingreso de huella digital.

Frecuencia de ocurrencia.

- Este caso de uso está fuera de los elementales para el negocio.

Varios. (Abierto a varias cuestiones relacionadas con el caso de uso.)

- ¿Existirán más usuarios a futuro?

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Iniciar sesión
Precondiciones:	<ul style="list-style-type: none"> • El sistema ha sido iniciado. • No existe sesión abierta
Pos condiciones:	<ul style="list-style-type: none"> - El sistema creo la interfaz de inicio de sesión - El sistema mostro la interfaz de inicio de sesión al usuario en el panel central del sistema.

Operación:	iniciarSesion ()
Referencia cruzada:	Caso de uso: Iniciar sesión
Precondiciones:	<ul style="list-style-type: none"> • No existe sesión abierta
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha verificado la existencia del usuario y clave ingresado. - El sistema ha obtenido los módulos para los que se encuentra habilitado el usuario. - El usuario tiene habilitado todas las opciones que tiene permitida.

Caso de Uso: Cerrar sesión	
<i>Alcance</i>	Registro de asistencia.
<i>Nivel</i>	Ingreso del usuario al sistema
<i>Actor primario</i>	Usuario del sistema
<i>Interesados e intereses</i>	Administrador de sistema.

Pre-condiciones	1.- El sistema debe tener sesión cerrada
Pos-condiciones	1.- El usuario de sistema cierra sesión y se desactivan las opciones de menú habilitadas.
Camino Básico:	
Eventos del actor	Eventos del sistema
1.-El usuario pide al sistema cerrar sesión. 3.-El usuario recibe el mensaje de confirmación.	2.- El sistema verifica si existen operaciones pendientes, deshabilita los accesos de opciones.
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>2.a.- Existen operaciones pendientes.</p> <p>1.a.- El usuario elige cancelar el cierre de sesión</p> <p>1.- El sistema cancela el cierre de sesión.</p> <p>1.b.- El usuario elige continuar.</p> <p><i>Continúa el caso de uso.</i></p>	

Contrato de operaciones

Operación:	cerrarSesion ()
Referencia cruzada:	Caso de uso: Cerrar sesión
Precondiciones:	<ul style="list-style-type: none"> • Existe sesión abierta
Pos condiciones:	- El sistema ha deshabilitado las opciones de menú.

Caso de Uso: Asignar usuario y clave	
<i>Alcance</i>	Control de ingreso al sistema.
<i>Nivel</i>	Módulo de sistema
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema Desea cuidar la seguridad de sus datos y puede asignar usuarios al sistema.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Se asignó usuario y clave a una persona.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.- El usuario pide asignar usuario y clave a una persona.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar el usuario y clave.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p>	

1.- El sistema cancela la operación.

1.a.- Existen datos por guardar.

1.- El sistema pide confirmación al usuario del sistema.

2.- El usuario confirma la cancelación de la operación.

3.- El sistema cancela la operación.

Vuelve al paso 3

2.- El registro de Usuario no existe.

1.- El sistema crea un nuevo objeto de Usuario.

Sigue paso 2.

4.- Existen atributos del objeto Usuario vacíos.

1.- El sistema envía un mensaje de error al usuario.

Vuelve paso 3.

II. MÓDULO DE ASISTENCIA

Caso de Uso: Registrar asistencia por huella digital	
<i>Alcance</i>	Todo el sistema para el control de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Personal
<i>Interesados e intereses</i>	Personal: las personas que trabajan en la institución desean que se registre su asistencia de manera rápida y sin errores. Administrador del sistema Desear tener el registro de asistencia de manera rápida.

Pre-condiciones	<p>1.- Haber sido habilitado para el registro de asistencia</p> <p>2.- El sistema se encuentra en estado de espera, para el registro de asistencia.</p>
Post-condiciones	1.- El usuario ha registrado su asistencia.
Camino Básico	
Eventos del actor	Eventos del sistema
<p>1.- El usuario ingresa su identificador o código de personal.</p> <p>3.- El usuario ingresa la huella digital.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide al usuario ingresar la huella digital.</p> <p>4.- El sistema validó la huella digital y envía un mensaje de confirmación del registro al usuario y vuelve al estado de espera de petición.</p>
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>2.- El código de personal es incorrecto.</p> <p>1.- El sistema envía un mensaje de información.</p> <p>2.- El usuario recibe el mensaje del error.</p> <p><i>Vuelve al paso 1</i></p>	

4.-La huella digital no corresponde al identificador ingresado por el usuario.

1.- El sistema envía un mensaje informativo al usuario, pidiéndole que vuelva a intentar el ingreso de la huella.

Vuelve al paso 3. Como máximo 3 intentos, luego vuelve al estado inicial

Requerimientos especiales. (Son requerimientos no funcionales relacionados con en el sistema)

1. El registro de asistencia, cuando es realizado mediante huella digital debe ser realizado en una media de dos intentos por personas, para lo cual el lector de huella debe ser de alta calidad.

Contratos de operación

Operación:	buscarIdentificador(id:Integer)
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none">• Sistema en espera de registro de asistencia.
Pos condiciones:	<ul style="list-style-type: none">- Se ha revisado los registros de personal y ubicado a la al identificador.

Operación:	validarHuellaDigital(id:Integer)
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none">• Sistema ha encontrado el registro del personal.
Pos condiciones:	<ul style="list-style-type: none">- Revisó las muestras de huellas digitales que tiene registrado para el personal que ingresó su huella digital- Ha validado la huella digital.- Creó un registro de Asistencia para el personal que ingresó su huella digital.- Guardó el registro de asistencia.

Caso de Uso: Registrar asistencia por código de personal.	
Alcance	Todo el sistema para el control de asistencia.
Nivel	Módulo de asistencia
Actor primario	Personal
Interesados e intereses	Personal: las personas que trabajan en la institución desean que se registre su asistencia de manera rápida y sin errores. Administrador del sistema Desear tener el registro de asistencia de manera rápida.
Pre-condiciones	1.- Haber sido habilitado para el registro de asistencia
Post-condiciones	1.- El usuario ha registrado su asistencia.
Camino Básico:	
Eventos del actor	Eventos del sistema
1.-El usuario pide al sistema registrar su asistencia. 3.- El usuario ingresa los datos. 5.- El usuario recibe el mensaje de confirmación.	2.- El sistema pide el identificador del personal que desea registrar su asistencia. 4.- El sistema valida los datos, registra la asistencia y envía un mensaje de confirmación al usuario.
Extensiones (Flujo alternativo).	
*.- El sistema falla en cualquier momento. 1. El sistema informa al usuario del error.	

<p>2. El usuario recibe el mensaje de error.</p> <p>3.- Vuelve al estado inicial</p> <p>4.- El código del usuario no se encuentra en registros</p> <p>1.- El sistema envía un mensaje de información.</p> <p>2.- El usuario recibe el mensaje de información.</p> <p><i>Vuelve al paso 3</i></p>
--

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de registro de asistencia. - El sistema ha cargado la interfaz con los datos necesarios.

Caso de Uso: Crear calendario laboral.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia.
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	<p>Administrador del sistema</p> <p>Desea poder registrar días feriados, días no laborales y cualquier otro día excepcional.</p>

Pre-condiciones	1.- Haber sido autenticado por el sistema
Post-condiciones	1.- Día de excepción registrado en el sistema.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario crear un calendario laboral.</p> <p>3.- El usuario ingresa los datos y pide guardar los mismos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos.</p> <p>4.- El sistema valida los datos, guarda los datos y envía un mensaje al usuario.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.a.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p>3.b.- Pide editar fechas de excepción laboral: incluye Gestionar fecha de excepción laboral.</p> <p><i>Vuelve al paso 3</i></p>	

Contrato de operaciones

Operación:	crearCalendarioLaboral()
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Post condiciones:	<ul style="list-style-type: none"> - Se ha creado un objeto de Calendario laboral - El sistema está en espera de que el usuario pide guardar los datos.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear calendario laboral
Precondiciones:	<ul style="list-style-type: none"> • Existe una excepción laboral en espera para ingreso de datos.
Post condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de calendario laboral. - Se ha guardado la excepción laboral.

Caso de Uso: Modificar calendario laboral.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Personal
Interesados e intereses	Administrador del sistema: desea poder modificar los datos de un calendario laboral.
Pre-condiciones	1.- El usuario se encuentra autenticado ante el sistema.
Post-condiciones	1.- El sistema ha modificado un objeto de calendario laboral.

Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide modificar un calendario laboral.</p> <p>3.- El usuario modifica los datos del calendario laboral y pide al sistema guardar los datos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz con los datos para la modificación de un calendario laboral.</p> <p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <ol style="list-style-type: none"> 1. El sistema informa al usuario del error y vuelve al estado inicial. 2. El usuario recibe el mensaje de error. <p>*.b.- El usuario cancela la operación</p> <ol style="list-style-type: none"> 1.- El sistema cancela la operación. <ol style="list-style-type: none"> 1.a.- Existen datos por guardar. <ol style="list-style-type: none"> 1.- El sistema pide confirmación al usuario del sistema. 2.- El usuario confirma la cancelación de la operación. 3.- El sistema cancela la operación. 3.- El nombre de referencia para el calendario laboral se encuentra registrado. <ol style="list-style-type: none"> 1.- El sistema cancela la modificación e informa al usuario. 	

2.- El usuario recibe el informe.

Vuelve al paso 3

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar calendario laboral.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de calendario laboral.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarCalendarioLaboral(index)
Referencia cruzada:	Caso de uso: Modificar calendario laboral
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- Se ha indicado un objeto de Calendario laboral a modificar.- El sistema está en espera de que el usuario pida guardar los datos.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear calendario laboral
Pre-condiciones:	<ul style="list-style-type: none">• Existe una excepción laboral en espera para ingreso de datos.
Pos-condiciones:	<ul style="list-style-type: none">- Se ha ingresado los datos para el objeto de calendario laboral.- Se ha guardado el calendario laboral.

Caso de Uso: Borrar calendario laboral.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Administrador de sistema.
Interesados e intereses	Administrador del sistema: desea borrar el calendario laboral.
Pre-condiciones	1.- Haber sido autenticado por el sistema.
Post-condiciones	1.- El usuario ha registrado el calendario laboral.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide eliminar una excepción al calendario.</p> <p>3.- El usuario confirma la eliminación del registro.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide confirmación al usuario.</p> <p>4.- El sistema cambia de estado al registro de fecha de excepción a inactivo.</p>
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.-El usuario cancela la eliminación del registro.</p> <p>1.- El sistema vuelve a estado inicio.</p>	

Contrato de operaciones

Operación:	borrarCalendarioLaboral()
Referencia cruzada:	Caso de uso: Borrar calendario laboral.
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos del calendario laboral.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardarCalendarioLaboral()
Referencia cruzada:	Caso de uso: Borrar calendario laboral
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de la excepción laboral
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha cambiado de estado al objeto de calendario laboral a inactivo. - Se han guardado los cambios.

Caso de Uso: Crear fecha de excepción laboral.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Administrador
Interesados e intereses	Administrador del sistema Desea poder registrar excepciones.
Pre condiciones	1.- Haber sido autenticado por el sistema

Post condiciones	1.- Día de excepción registrado en el sistema.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario pide registrar una fecha de excepción de trabajo.</p> <p>3.- El selecciona una fecha y la marca como fecha de excepción.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema muestra la interfaz para ingresar los datos.</p> <p>4.- El sistema valida, guarda la fecha de excepción y envía un mensaje.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>*.b.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p>2.- El sistema cancela la operación.</p> <p>4.- Los datos ingresados son incorrectos</p> <p>1.- El sistema envía un mensaje de información.</p>	

2.- El usuario recibe el mensaje de información.

Vuelve al paso 3

Contrato de operaciones

Operación:	motrarInterfaz()
Referencia cruzada:	Caso de uso: Crear fecha de excepciones de calendario.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz para crear fecha de excepción laboral.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearExcepcionLaboral()
Referencia cruzada:	Caso de uso: Crear fecha de excepciones.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- Se ha creado un objeto de fecha de excepción laboral.- El sistema está en estado de espera hasta que el usuario ingrese los datos para la fecha de excepción laboral.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear excepción laboral
Precondiciones:	<ul style="list-style-type: none">• Existe una fecha de excepción laboral en espera para ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none">- Se ha ingresado los datos para el objeto de fecha

	<p>de excepción laboral.</p> <ul style="list-style-type: none"> - Se ha guardado la excepción laboral.
--	---

Caso de Uso: Borrar fecha de excepción laboral.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: desea borrar una fecha de excepción laboral.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema.
<i>Post-condiciones</i>	1.- El usuario ha borrado una fecha de excepción laboral.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.-El usuario pide eliminar una fecha de excepción al calendario.</p> <p>3.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema borra el registro de fecha de excepción.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar fecha de excepción laboral.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para gestionar las fechas de excepción laboral. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarExcepcionLaboral()
Referencia cruzada:	Caso de uso: Borrar fecha de excepción laboral.
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de la excepción laboral
Pos condiciones:	<ul style="list-style-type: none"> - El sistema elimina el registro de fecha de excepción laboral para el calendario.

Caso de Uso: Crear permiso de trabajo.	
Alcance:	El control de asistencia con la generación de planillas de asistencia.
Nivel:	Módulo de asistencia
Actor primario:	Administrador
Interesados e intereses	<p>Administrador del sistema</p> <p>Desea poder registrar permiso del personal.</p> <p>Personal</p> <p>Desear tener el registro de los permisos laborales obtenidos</p>

	en cualquier momento.	
Pre condiciones:	1.- Haber sido autenticado por el sistema	
Post condiciones:	1.- Permiso laboral registrado.	
Camino Básico:		
	Eventos del actor	Eventos del sistema
	<p>1.- El usuario pide registrar un permiso de trabajo.</p> <p>3.- El usuario ingresa los datos para el permiso de trabajo y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos del personal.</p> <p>4.- El sistema valida los datos y guarda los datos recibidos y envía un mensaje de confirmación.</p>
Extensiones (Flujo alternativo).		
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p>		

<p>3.- El sistema cancela la operación.</p> <p>4.- La fecha de finalización para el permiso de trabajo es anterior a la fecha actual.</p> <p>1.- El sistema envía un mensaje de error al usuario.</p> <p>2.- El usuario recibe el informe.</p> <p>3.- El sistema vuelve a presentar al usuario para corrección.</p> <p><i>Vuelve al paso 3</i></p>
--

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Post condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de registro de permiso. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearPermisoTrabajo()
Referencia cruzada:	Caso de uso: Crear permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Post condiciones:	<ul style="list-style-type: none"> - Se ha creado un objeto de permiso de trabajo. - El sistema está en estado de espera hasta que el usuario ingrese los datos para el permiso de trabajo

Operación:	guardar()
-------------------	------------------

Referencia cruzada:	Caso de uso: Gestionar permiso de trabajo
Precondiciones:	<ul style="list-style-type: none"> • Existe un objeto de permiso laboral para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de permiso de trabajo. - Se ha guardado el permiso de trabajo.

Caso de Uso: Modificar permiso de trabajo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Personal
<i>Interesados e intereses</i>	Administrador del sistema: desea modificar el permiso de trabajo en casos especiales.
<i>Pre-condiciones</i>	1.- El usuario ha sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Se ha modificado el permiso de trabajo.
Camino Básico:	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.-El usuario pide modificar un permiso de trabajo.</p> <p>3.- El usuario modifica los datos y pide al sistema guardar los cambios.</p> <p>5.- El usuario recibe el mensaje de</p>	<p>2.- El sistema muestra los datos del objeto al usuario para que los modifique.</p> <p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>

confirmación.	
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.- El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2.- El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p>4.- La fecha de finalización para el permiso de trabajo es anterior a la fecha actual.</p> <p>1.- El sistema envía un mensaje de error al usuario.</p> <p>2.- El usuario recibe el informe.</p> <p>3.- El sistema vuelve a presentar al usuario para corrección.</p> <p><i>Vuelve al paso 3</i></p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de registro de asistencia. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarPermisoTrabajo()
Referencia cruzada:	Caso de uso: Modificar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha obtenido y presentado todos los datos a modificar por parte del usuario. - El usuario ha recibido los datos para modificarlos.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Modificar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos del permiso de trabajo
Pos condiciones:	<ul style="list-style-type: none"> - El usuario a modificado los datos del permiso de trabajo - El sistema ha validado los datos ingresados. - Se han guardado los cambios.

Caso de Uso: Borrar permiso de trabajo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar registro de permisos de trabajos.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema.

Post-condiciones	1.- Se ha borrado el permiso de trabajo.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide eliminar un permiso de trabajo.</p> <p>3.- El usuario confirma la eliminación del objeto</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide confirmación al usuario.</p> <p>4.- El sistema cambia de estado al objeto a inactivo.</p>
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.-El usuario cancela la eliminación del registro.</p> <p>1.- El sistema vuelve a estado inicio.</p> <p>4.- La fecha de inicio del permiso de trabajo es anterior a la actual.</p> <p>1.- El sistema informa al usuario que el permiso de trabajo ya se encuentra en curso y cancela el borrado.</p> <p>2.- El usuario recibe el mensaje.</p> <p><i>Vuelve al paso 3</i></p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz para la gestión de los permisos de trabajo. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarPermisoTrabajo()
Referencia cruzada:	Caso de uso: Borrar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha recibido la petición de borrar un permiso de trabajo - El sistema ha validado la eliminación del registro. - El sistema ha pedido confirmación de la acción al usuario. - El sistema cambiado de estado al registro de permiso de trabajo a inactivo.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos del permiso de trabajo.
Pos condiciones:	<ul style="list-style-type: none"> - Se han guardado los cambios.

Caso de Uso: Crear tipo de permiso de trabajo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia.
<i>Actor primario</i>	Administrador.
<i>Interesados e intereses</i>	Administrador del sistema. Desea poder administrar los tipos de permisos al personal.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema.
<i>Post-condiciones</i>	1.- Registro de tipo de permiso de trabajo registrado.
<i>Camino Básico</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
1.- El usuario pide registrar tipo de permiso de trabajo. 3.- El usuario ingresa los datos y pide guardarlos. 5.- El usuario recibe el mensaje de confirmación.	2.- El sistema presenta la interfaz para ingresar los datos del tipo de permiso. 4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.
<i>Extensiones (Flujo alternativo).</i>	
*.a.- El sistema falla en cualquier momento. <ol style="list-style-type: none"> 1. El sistema informa al usuario del error y vuelve al estado inicial. 2. El usuario recibe el mensaje de error. 3.- El usuario cancela la operación. 	

1.- El sistema cancela la operación.

1.a.- Existen datos por guardar.

1.- El sistema pide confirmación al usuario del sistema.

2.- El usuario confirma la cancelación de la operación.

3.- El sistema cancela la operación.

Vuelve al paso 3

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear tipo de permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de tipo de permiso- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearTipoPermisoTrabajo()
Referencia cruzada:	Caso de uso: Gestionar tipo permiso de trabajo
Pre condiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Post condiciones:	<ul style="list-style-type: none">- Se ha creado un objeto de Tipo de permiso de trabajo.- El sistema está en estado de espera hasta que el usuario ingrese los datos para el tipo de permiso de trabajo.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Gestionar tipo de permiso de trabajo
Precondiciones:	<ul style="list-style-type: none"> • Existe un objeto de tipo de permiso laboral para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de permiso de trabajo. - Los datos del tipo de permiso de trabajo han sido validado. - Se ha guardado el tipo de permiso de trabajo.

Caso de Uso: Modificar el tipo de permiso de trabajo.	
Alcance:	El control de asistencia con la generación de planillas de asistencia.
Nivel:	Módulo de asistencia
Actor primario:	Personal
Interesados e intereses	Administrador del sistema: desea modificar el tipo de permiso de trabajo.
Pre-condiciones:	1.- El usuario ha sido autenticado por el sistema
Post-condiciones:	1.- Se ha modificado el tipo de permiso de trabajo.
Camino Básico:	
Eventos del actor	Eventos del sistema
1.-El usuario pide modificar el tipo de permiso de trabajo.	2.- El sistema muestra los datos del objeto al usuario para que los modifique.

<p>3.- El usuario modifica los datos y pide al sistema guardar los cambios.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>
--	--

Extensiones (Flujo alternativo).

- *.- El sistema falla en cualquier momento.
 - 1.1. El sistema informa al usuario del error y vuelve al estado inicial.
 - 1.2. El usuario recibe el mensaje de error.
- 3.- El usuario cancela la operación
 - 1.a.- El sistema cancela la operación.
 - 1.b.- Existen datos por guardar.
 - 1.- El sistema pide confirmación al usuario del sistema.
 - 2.- El usuario confirma la cancelación de la operación.
 - 3.- El sistema cancela la operación.

Vuelve al paso 3

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de tipo de permiso de trabajo.

	<ul style="list-style-type: none"> - El sistema ha cargado la interfaz con los datos necesarios.
--	---

Operación:	modificarTipoPermisoTrabajo()
Referencia cruzada:	Caso de uso: Gestionar tipo permiso de trabajo
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha obtenido y presentado todos los datos a modificar por parte del usuario. - El usuario ha recibido los datos para modificarlos.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Modificar tipo de permiso de trabajo
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de tipo de permiso de trabajo
Pos condiciones:	<ul style="list-style-type: none"> - El usuario a modificado los datos del tipo de permiso de trabajo - El sistema ha validado los datos ingresados. - Se han guardado los cambios.

Caso de Uso: Borrar tipo de permiso de trabajo.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Administrador de sistema.

<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar registro de tipo de permiso de trabajo.
<i>Pre-condiciones</i>	1.- El usuario se ha autenticado ante el sistema
<i>Post-condiciones</i>	1.- El usuario borra el tipo de permiso de trabajo.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.-El usuario pide eliminar un tipo de permiso de trabajo.</p> <p>3.- El usuario confirma la eliminación del objeto.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide confirmación al usuario.</p> <p>4.- El sistema cambia de estado al objeto a inactivo y envía un mensaje al usuario.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.-El usuario cancela la eliminación del registro.</p> <p>1.- El sistema vuelve a estado inicio.</p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar asistencia

Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de registro de asistencia. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarTipoPermisoTrabajo()
Referencia cruzada:	Caso de uso: Borrar tipo de permiso de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha recibido la petición de borrar un tipo de permiso de trabajo. - El sistema ha validado la eliminación del registro. - El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar de tipo de permiso de trabajo
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de la excepción laboral
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha cambiado el estado al registro tipo de permiso de trabajo. - Se han guardado los cambios.

Caso de Uso: Crear horario de trabajo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema Desea crear horarios de trabajo cuando lo vea necesario.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Calendario laboral.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.- El usuario pide registrar un horario de trabajo.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para crear un horario de trabajo.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p>	

3.- El usuario cancela la operación

1.- El sistema cancela la operación.

1.a.- Existen datos por guardar.

1.- El sistema pide confirmación al usuario del sistema.

2.- El usuario confirma la cancelación de la operación.

3.- El sistema cancela la operación.

4.a.- Existe un horario para ese cargo

1.- El sistema manda un mensaje pidiendo confirmación para reemplazar el horario para ese cargo.

1.a.- El usuario pide reemplazar el horario de trabajo.

1.- El sistema cambia a estado inactivo al horario para ese cargo.

Continúa paso 4.

2.- El usuario cancela la operación.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear horario de trabajo
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestión para los horarios de trabajo.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearHorarioTrabajo()
Referencia cruzada:	Caso de uso: Crear horario de trabajo

Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha creado un objeto de Horario Trabajo - El sistema está en estado de espera hasta que el usuario ingrese los datos para el objeto de Horario de Trabajo.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear horario de trabajo
Precondiciones:	<ul style="list-style-type: none"> • Existe un objeto de permiso laboral para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de Horario de Trabajo. - Se ha creado la relación entre cargo y horario. - Se ha guardado el Horario de Trabajo.

Caso de Uso: Modificar Horario de Trabajo.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Administrador de Personal
Interesados e intereses	Administrador del sistema: desea modificar Horario de Trabajo
Pre-condiciones	1.- El usuario ha sido autenticado por el sistema
Post-condiciones	1.- Se ha modificado el horario de trabajo.

Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide modificar un Horario de Trabajo.</p> <p>3.- El usuario modifica los datos y pide al sistema guardar los cambios.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema muestra los datos del objeto al usuario para que los modifique.</p> <p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.b.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p>4.a.- Existe un horario para ese cargo</p> <p>1.- El sistema manda un mensaje pidiendo confirmación para reemplazar el horario para ese cargo.</p> <p>1.a.- El usuario pide reemplazar el horario de trabajo.</p> <p>1.- El sistema cambia a estado inactivo al horario para ese cargo.</p>	

Continúa paso 4.

2.- El usuario cancela la operación.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar Horario de Trabajo.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestión para los horarios de trabajo.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarHorarioTrabajo()
Referencia cruzada:	Caso de uso: Modificar Horario de Trabajo.
Precondiciones:	<ul style="list-style-type: none">• El usuario se ha autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha obtenido y presentado todos los datos a modificar por parte del usuario.- El usuario ha recibido los datos para modificarlos.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Modificar Horario de Trabajo.
Precondiciones:	<ul style="list-style-type: none">• Se han modificado datos del horario de trabajo
Pos condiciones:	<ul style="list-style-type: none">- El usuario modificó los datos del Horario de Trabajo.- El sistema ha validado los datos ingresados.

	- Se han guardado los cambios.
--	--------------------------------

Caso de Uso: Borrar horario de trabajo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar registro de horario.
<i>Pre-condiciones</i>	1.- Usuario ha sido autenticado por el sistema.
<i>Post-condiciones</i>	1.- El usuario ha cambiado de estado al horario de trabajo.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
1.-El usuario pide eliminar un Horario de Trabajo. 3.- El usuario confirma la eliminación del objeto. 5.- El usuario recibe el mensaje de confirmación.	2.- El sistema pide confirmación al usuario. 4.- El sistema cambia de estado al objeto a inactivo.
<i>Extensiones (Flujo alternativo).</i>	
*.- El sistema falla en cualquier momento. 1.1. El sistema informa al usuario del error y vuelve al estado inicial.	

1.2. El usuario recibe el mensaje de error.
3.-El usuario cancela la eliminación del registro.
1.- El sistema vuelve a estado inicio.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar horario de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de gestión de horarios de trabajo. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarHorarioTrabajo ()
Referencia cruzada:	Caso de uso: Borrar horario de trabajo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha recibido la petición de borrar un Horario de Trabajo. - El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar horario de trabajo
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha pedido borrar un horario de trabajo.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema cambiado de estado al registro de

	<p>horario de trabajo.</p> <ul style="list-style-type: none"> - Se han guardado los cambios.
--	---

Caso de Uso: Crear registro de personal.	
<i>Alcance</i>	Todo el módulo de sistema.
<i>Nivel</i>	Módulo de recursos humanos
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema Desea poder administrar los tipos de permisos al personal.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Registró a una persona en el sistema.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.- El usuario pide realizar el registro a personal.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos del registro de personal.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>

Extensiones (Flujo alternativo).

- *.a.- El sistema falla en cualquier momento.
 - 1. El sistema informa al usuario del error y vuelve al estado inicial.
 - 2. El usuario recibe el mensaje de error.
- 3.- El usuario cancela la operación.
 - 1.- El sistema cancela la operación.
 - 1.a.- Existen datos por guardar.
 - 1.- El sistema pide confirmación al usuario del sistema.
 - 2.- El usuario confirma la cancelación de la operación.
 - 3.- El sistema cancela la operación.
- 4.- La persona ya tiene un cargo registrado y activo.
 - 1.- El sistema envía un mensaje de error.
 - 2.- El usuario recibe el informe.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear registro de personal.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de registro de personal.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearRegistroPersonal()
Referencia cruzada:	Caso de uso: Crear registro de personal.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha creado un objeto de personal. - El sistema está en estado de espera hasta que el usuario ingrese los datos para el registro de personal.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear registro de personal.
Precondiciones:	<ul style="list-style-type: none"> • Existe un objeto de persona laboral para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de personal. - Se ha guardado el registro de personal.

<i>Caso de Uso: Modificar registro de personal.</i>	
<i>Alcance</i>	Todo el módulo de sistema.
<i>Nivel</i>	Módulo de recursos humanos.
<i>Actor primario</i>	Personal
<i>Interesados e intereses</i>	Administrador del sistema: desea modificar el registro de personal
<i>Pre-condiciones</i>	1.- El usuario ha sido autenticado por el sistema.
<i>Post-condiciones</i>	1.- Se ha modificado el registro de personal.

Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide modificar el registro de personal.</p> <p>3.- El usuario modifica los datos y pide al sistema guardar los cambios.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema muestra los datos del objeto al usuario para que los modifique.</p> <p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>
Extensiones (Flujo alternativo).	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen cambios en datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p>4.- La persona ya tiene un cargo registrado y activo.</p> <p>1.- El sistema envía un mensaje de error.</p> <p>2.- El usuario recibe el informe.</p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar registro de personal.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestión de personal.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarRegistroPersonal()
Referencia cruzada:	Caso de uso: Modificar registro de personal.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha obtenido y presentado todos los datos a modificar por parte del usuario.- El usuario ha recibido los datos para modificarlos.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Modificar registro de personal.
Precondiciones:	<ul style="list-style-type: none">• Se han modificado los datos de registro de personal.
Pos condiciones:	<ul style="list-style-type: none">- El usuario ha modificado los datos del registro de personal.- El sistema ha validado los datos ingresados.- Se han guardado los cambios.

Caso de Uso: Borrar registro de personal.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar registro de personal.
<i>Pre-condiciones</i>	1.- El usuario se ha autenticado ante el sistema
<i>Post-condiciones</i>	1.- El usuario borra el registro de personal.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.-El usuario pide eliminar un registro de personal.</p> <p>3.- El usuario confirma la eliminación del objeto.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide confirmación al usuario.</p> <p>4.- El sistema cambia de estado al objeto a inactivo.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.-El usuario cancela la eliminación del registro.</p> <p>1.- El sistema vuelve a estado inicio.</p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar registro de personal.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz gestión de personal.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarRegistroPersonal()
Referencia cruzada:	Caso de uso: Borrar registro de personal.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha recibido la petición de borrar un registro de personal.- El sistema ha validado la eliminación del objeto.- El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar el registro de personal
Precondiciones:	<ul style="list-style-type: none">• Se han modificado los datos del registro de personal
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha cambiado el estado al registro de personal.- Se han guardado los cambios.

Caso de Uso: Registrar huella digital.	
Alcance	El control de asistencia con la generación de planillas de asistencia.
Nivel	Módulo de asistencia
Actor primario	Administrador
Interesados e intereses	Administrador del sistema Desea registrar la huella digital de una persona.
Pre-condiciones	1.- Haber sido autenticado por el sistema
Post-condiciones	1.- Se ha registrado la huella digital.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario pide registrar huella digital.</p> <p>3.-El usuario selecciona el dedo del que desea registrar su huella digital.</p> <p>5.-El usuario ingresa su huella digital</p> <p>7.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para el registro de huella digital.</p> <p>4.- El sistema pide ingresar las muestras de la huella digital.</p> <p>6.- El sistema válida la calidad de la muestra tomada de la huella digital y guarda el registro.</p> <p>Se toman tres muestras por dedo</p>

Extensiones (Flujo alternativo).

- *.a.- El sistema falla en cualquier momento.
 - 1. El sistema informa al usuario del error y vuelve al estado inicial.
 - 2. El usuario recibe el mensaje de error.
- 3.- El usuario cancela la operación
 - 1.- El sistema cancela la operación.
 - 1.a.- Existen datos por guardar.
 - 1.- El sistema pide confirmación al usuario del sistema.
 - 2.- El usuario confirma la cancelación de la operación.
 - 3.- El sistema cancela la operación.
- 4.- Existen muestras para el dedo seleccionado.
 - 1.- El sistema pide confirmación para borrar las muestras anteriores.
 - 2.- El usuario confirma la eliminación de las muestras de huella digital.
 - 2.a.- El usuario cancela la eliminación de las muestras de huella digital.
 - 1.- El sistema cancela la operación y vuelve al estado inicial.
 - 3.- El sistema elimina las muestras.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar huella digital.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de registro de huella digital.

	<ul style="list-style-type: none"> - El sistema ha cargado la interfaz con los datos necesarios.
--	---

Operación:	crearRegistroHuellaDigital()
Referencia cruzada:	Caso de uso: Registrar huella digital.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha creado un objeto de Huella digital - El sistema está en estado de espera hasta que el usuario ingrese la muestra de huella digital.

Operación:	validaHuellaDigital()
Referencia cruzada:	Caso de uso: Registrar huella digital
Precondiciones:	<ul style="list-style-type: none"> • Se ha ingresado una muestra de huella digital
Pos condiciones:	<ul style="list-style-type: none"> - La muestra de huella digital validada se encuentra en un rango aceptable para guardarla.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Registrar huella digital.
Precondiciones:	<ul style="list-style-type: none"> • Se ha ingresado al objeto persona las tres muestras de huella digital.
Pos condiciones:	<ul style="list-style-type: none"> - Se han tomado 3 muestra de la huella digital por dedo elegido. - Se ha guardado las muestras de huella digital.

Caso de Uso: Crear cargo.	
<i>Alcance</i>	Todo el módulo de RR. HH.
<i>Nivel</i>	Módulo de recursos humanos.
<i>Actor primario</i>	Administrador.
<i>Interesados e intereses</i>	Administrador del sistema. Desea poder administrar los tipos de permisos al personal.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema.
<i>Post-condiciones</i>	1.- Registró a un cargo.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.- El usuario pide registrar un cargo en el sistema.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos del cargo.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p>	

1.a.- Existen datos por guardar.

1.- El sistema pide confirmación al usuario del sistema.

2.- El usuario confirma la cancelación de la operación.

3.- El sistema cancela la operación.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear cargo.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestión de cargo.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearCargo()
Referencia cruzada:	Caso de uso: Crear cargo.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- Se ha creado un objeto de cargo.- El sistema está en estado de espera hasta que el usuario ingrese los datos para el registro de cargo.

Operación:	guardar()
Referencia cruzada:	Caso de uso: Crear cargo.
Precondiciones:	<ul style="list-style-type: none">• Existe un objeto de cargo para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none">- Se ha ingresado los datos para el objeto de cargo.

	<ul style="list-style-type: none"> - Se ha creado la relación ClasifAsistencia. - Se ha guardado el cargo.
--	--

Caso de Uso: Modificar cargo.	
Alcance	Todo el módulo de sistema.
Nivel	Módulo de recursos humanos.
Actor primario	Personal
Interesados e intereses	Administrador del sistema: desea modificar el registro de personal
Pre-condiciones	1.- El usuario ha sido autenticado por el sistema
Post- condiciones	1.- Se ha modificado el cargo.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.-El usuario pide modificar el cargo.</p> <p>3.- El usuario modifica los datos y pide al sistema guardar los cambios.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema muestra los datos del objeto al usuario para que los modifique.</p> <p>4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.</p>
Extensiones (Flujo alternativo).	

*.- El sistema falla en cualquier momento.

1. El sistema informa al usuario del error y vuelve al estado inicial.
2. El usuario recibe el mensaje de error.
- 3.- El usuario cancela la operación
 - 1.- El sistema cancela la operación.
 - 1.a.- Existen cambios en datos por guardar.
 - 1.- El sistema pide confirmación al usuario del sistema.
 - 2.- El usuario confirma la cancelación de la operación.
 - 3.- El sistema cancela la operación.

Vuelve al paso 3

- 4.- El nombre del cargo ya se encuentra registrado.
 - 1.- El sistema informa al usuario.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar cargo
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestión para cargos- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarCargo()
Referencia cruzada:	Caso de uso: Modificar cargo
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha obtenido y presentado todos los datos a modificar por parte del usuario. - El usuario ha recibido los datos para modificarlos.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Modificar cargo
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de cargo.
Pos condiciones:	<ul style="list-style-type: none"> - El usuario ha modificado los datos del cargo. - El sistema ha validado los datos ingresados. - Se han guardado los cambios.

Caso de Uso: Borrar cargo.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar cargo
<i>Pre-condiciones</i>	1.- El usuario se ha autenticado ante el sistema
<i>Post-condiciones</i>	1.- El usuario borra el registro de cargo.

<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.-El usuario pide eliminar un registro de cargo.</p> <p>3.- El usuario confirma la eliminación del objeto.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema pide confirmación al usuario.</p> <p>4.- El sistema cambia de estado al objeto a inactivo.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.- El sistema falla en cualquier momento.</p> <p>1.1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>1.2. El usuario recibe el mensaje de error.</p> <p>3.-El usuario cancela la eliminación del registro.</p> <p>1.- El sistema vuelve a estado inicio.</p>	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar cargo.
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de gestión de cargo. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarCargo()
Referencia cruzada:	Caso de uso: Borrar cargo.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha recibido la petición de borrar cargo - El sistema ha validado la eliminación del objeto. - El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar el cargo
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos del cargo
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha cambiado el estado al cargo a inactivo. - Se han guardado los cambios.

III. MODULO GENERAL

Caso de Uso: Crear persona.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de general
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema Desea poder administrar los registros de persona.

Pre-condiciones	1.- Haber sido autenticado por el sistema
Post-condiciones	1.- Se registró a la persona.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario pide registrar a una persona.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos de la persona.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación.</p> <p>1.- El sistema cancela la operación.</p> <p>1.a.- Existen datos por guardar.</p> <p>1.- El sistema pide confirmación al usuario del sistema.</p> <p>2.- El usuario confirma la cancelación de la operación.</p> <p>3.- El sistema cancela la operación.</p> <p><i>Vuelve al paso 3</i></p> <p>4.a.- El nombre y el apellido coincide con un registro existente.</p>	

1.- El sistema informa al usuario.

1.a.- El usuario pide continuar con el registro

Continúa paso 4.

1.b.- El usuario cancela el registro.

El sistema vuelve a la pantalla con los datos del usuario.

4.b.- El documento de identidad ya se encuentra registrado.

1.- El sistema envía al usuario un mensaje de error.

2.- El usuario acepta el mensaje.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Crear persona.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de gestionar personas.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	crearPersona()
Referencia cruzada:	Caso de uso: Crear persona.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- Se ha creado un objeto de Persona.- El sistema está en estado de espera hasta que el usuario ingrese los datos para la persona.

Operación:	guardarDatosPersona()
Referencia cruzada:	Caso de uso: Crear persona.
Precondiciones:	<ul style="list-style-type: none"> • Existe un objeto persona para el ingreso de datos.
Pos condiciones:	<ul style="list-style-type: none"> - Se ha ingresado los datos para el objeto de persona. - Los datos han sido validados exitosamente - Se ha guardado la persona.

<i>Caso de Uso: Modificar persona.</i>	
<i>Alcance</i>	Todo el sistema.
<i>Nivel</i>	Módulo de general
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: desea modificar los datos de una persona.
<i>Pre-condiciones</i>	1.- El usuario ha sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Se ha modificado la persona.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
1.-El usuario pide modificar la persona.	2.- El sistema muestra los datos del objeto al usuario para que los modifique.
3.- El usuario modifica los datos y pide	

al sistema guardar los cambios.	4.- El sistema guarda las modificaciones y envía un mensaje de confirmación al usuario.
5.- El usuario recibe el mensaje de confirmación.	

Extensiones (Flujo alternativo).

- *.- El sistema falla en cualquier momento.
 - 1.1. El sistema informa al usuario del error y vuelve al estado inicial.
 - 1.2. El usuario recibe el mensaje de error.
- 3.- El usuario cancela la operación
 - 1.- El sistema cancela la operación.
 - 1.a.- Existen datos por guardar.
 - 1.- El sistema pide confirmación al usuario del sistema.
 - 2.- El usuario confirma la cancelación de la operación.
 - 3.- El sistema cancela la operación.
 - Vuelve al paso 3*
- 4.a.- El nombre y el apellido coincide con un registro existente.
 - 1.- El sistema informa al usuario.
 - 1.- El usuario pide continuar con el registro
 - 1.b.- El usuario cancela el registro.
 - 1.-El sistema vuelve a la pantalla con los datos del usuario.
 - Continúa paso 4.
- 4.b.- El documento de identidad ya se encuentra registrado.
 - 1.- El sistema envía al usuario un mensaje de error.
 - 2.- El usuario acepta el mensaje.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Modificar persona.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz para gestionar registro de persona.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	modificarPersona(int)
Referencia cruzada:	Caso de uso: Modificar persona.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema.
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha obtenido y presentado todos los datos de persona a modificarse por parte del usuario.- El usuario ha recibido los datos para modificarlos.

Operación:	guardar ModificacionPersona()
Referencia cruzada:	Caso de uso: Gestionar persona
Precondiciones:	<ul style="list-style-type: none">• Se han modificado los datos de persona.
Pos condiciones:	<ul style="list-style-type: none">- El usuario ha modificado los datos de la persona.- El sistema ha validado los datos ingresados.- Se han guardado los cambios.

Caso de Uso: Borrar persona.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo general
<i>Actor primario</i>	Administrador de sistema.
<i>Interesados e intereses</i>	Administrador del sistema: poder eliminar a una persona.
<i>Pre-condiciones</i>	1.- El usuario se ha autenticado ante el sistema
<i>Post-condiciones</i>	1.- El usuario a cambiado a estado inactivo a la persona.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
1.-El usuario pide eliminar una persona. 3.- El usuario confirma la eliminación del objeto. 5.- El usuario recibe el mensaje de confirmación.	2.- El sistema pide confirmación al usuario. 4.- El sistema cambia de estado al objeto a inactivo.
<i>Extensiones (Flujo alternativo).</i>	
*.- El sistema falla en cualquier momento. <ol style="list-style-type: none"> 1. El sistema informa al usuario del error y vuelve al estado inicial. 2. El usuario recibe el mensaje de error. 3.-El usuario cancela la eliminación del registro. <ol style="list-style-type: none"> 1.- El sistema vuelve a estado inicio. 	

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Borrar persona.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz de registro de asistencia.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	borrarPersona()
Referencia cruzada:	Caso de uso: Borrar persona.
Precondiciones:	<ul style="list-style-type: none">• El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha recibido la petición de borrar a una persona.- El sistema ha validado el borrado del objeto.- El sistema ha pedido confirmación de la acción al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Borrar persona.
Precondiciones:	<ul style="list-style-type: none">• Se han modificado los datos de la persona.
Pos condiciones:	<ul style="list-style-type: none">- El sistema ha cambiado el estado a la persona.- Se han guardado los cambios.

Caso de Uso: Gestionar institución.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo general
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema Desea poder administrar los datos de la institución.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema
<i>Post-condiciones</i>	1.- Se administró los datos de la institución.
<i>Camino Básico:</i>	
<i>Eventos del actor</i>	<i>Eventos del sistema</i>
<p>1.- El usuario pide registrar a un registro de institución.</p> <p>3.- El usuario ingresa los datos y pide guardarlos.</p> <p>5.- El usuario recibe el mensaje de confirmación.</p>	<p>2.- El sistema presenta la interfaz para ingresar los datos de la institución.</p> <p>4.- El sistema valida los datos, guarda los datos recibidos y envía un mensaje de confirmación.</p>
<i>Extensiones (Flujo alternativo).</i>	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p>	

3.- El usuario cancela la operación

1.a.- El sistema cancela la operación.

1.b.- Existen datos por guardar.

1.- El sistema pide confirmación al usuario del sistema.

2.- El usuario confirma la cancelación de la operación.

3.- El sistema cancela la operación.

Vuelve al paso 3

2.- El registro de la institución no existe.

1.- El sistema crea un nuevo objeto de Institución.

Sigue paso 2.

4.- Existen atributos del objeto Institución vacíos.

1.- El sistema envía un mensaje de error al usuario.

Vuelve paso 3.

Contrato de operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Gestionar institución.
Precondiciones:	<ul style="list-style-type: none">• Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none">- El sistema recibió petición para cargar la interfaz para gestionar institución.- El sistema ha cargado la interfaz con los datos necesarios.

Operación:	gestionarInstitucion()
Referencia cruzada:	Caso de uso: Gestionar institución.
Precondiciones:	<ul style="list-style-type: none"> • El usuario ha sido autenticado por el sistema
Pos condiciones:	<ul style="list-style-type: none"> - El sistema ha recibido la petición de gestionar los datos de la institución. - El sistema ha obtenidos los datos de la institución. - El sistema ha presentado los datos de la institución al usuario.

Operación:	guardar ()
Referencia cruzada:	Caso de uso: Gestionar institución.
Precondiciones:	<ul style="list-style-type: none"> • Se han modificado los datos de la institución.
Pos condiciones:	<ul style="list-style-type: none"> - Se validaron los datos de entrada. - Se han guardado los cambios.

IV. MODULO DE REPORTE

Caso de Uso: Generar reporte de asistencia.	
<i>Alcance</i>	El control de asistencia con la generación de planillas de asistencia.
<i>Nivel</i>	Módulo de asistencia
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema: desea tener los reportes de asistencia en diferentes fechas.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema

Post-condiciones	1.- Se generó la reporte de asistencia.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario pide mostrar reporte de asistencia.</p> <p>3.- El usuario ingresa los datos y pide generar los reportes.</p> <p>5.- El usuario recibe el reporte.</p>	<p>2.- El sistema presenta la interfaz para ingresar los parámetros del reporte.</p> <p>4.- El sistema valida los datos, y genera el reportes.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p> <p>3.- El usuario cancela la operación</p> <p>1.- El sistema cancela la operación.</p> <p><i>Vuelve al paso 1</i></p>	

Contrato de Operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.

Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de reportes de asistencia. - El sistema ha cargado la interfaz con los datos necesarios.
-------------------------	---

Operación:	generarReporteAsistencia()
Referencia cruzada:	Caso de uso: Generar reporte de asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - Se revisó la lista de cargos y a partir de esa se recorrió cada uno de ellos para generar la planilla de asistencia del personal. - Se contrastó la asistencia con el calendario laboral, la clasificación de asistencia y el horario de trabajo para cada cargo. - En base a las fechas pedidas se generó el reporte de planilla de asistencia mostrándole al usuario.

Caso de Uso: Generar reporte de asistencia por persona.	
<i>Alcance</i>	El control de asistencia con la generación de reporte de asistencia por personas.
<i>Nivel</i>	Módulo de reportes
<i>Actor primario</i>	Administrador
<i>Interesados e intereses</i>	Administrador del sistema: desea tener los reportes de asistencia entre diferentes fechas.
<i>Pre-condiciones</i>	1.- Haber sido autenticado por el sistema

Post-condiciones	1.- Se generó el reporte de asistencia.
Camino Básico:	
Eventos del actor	Eventos del sistema
<p>1.- El usuario pide mostrar reporte de asistencia por persona.</p> <p>3.- El usuario ingresa los datos y pide generar los reportes.</p> <p>5.- El usuario recibe el reporte.</p>	<p>2.- El sistema presenta la interfaz para ingresar los parámetros del reporte.</p> <p>4.- El sistema valida los datos, y genera el reportes.</p>
Extensiones (Flujo alternativo).	
<p>*.a.- El sistema falla en cualquier momento.</p> <p>1. El sistema informa al usuario del error y vuelve al estado inicial.</p> <p>2. El usuario recibe el mensaje de error.</p>	

Contrato de Operaciones

Operación:	mostrarInterfaz()
Referencia cruzada:	Caso de uso: Registrar asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - El sistema recibió petición para cargar la interfaz de reportes de asistencia por persona. - El sistema ha cargado la interfaz con los datos necesarios.

Operación:	generarReporteAsistenciaPersona()
Referencia cruzada:	Caso de uso: Generar reporte de asistencia
Precondiciones:	<ul style="list-style-type: none"> • Sistema iniciado y en estado de espera de peticiones.
Pos condiciones:	<ul style="list-style-type: none"> - Se revisó el cargo del personal para el cual se va a generar el reporte. - Se contrastó la asistencia con el calendario laboral, la clasificación de asistencia y el horario de trabajo que tienen relación con su cargo. - En base a las fechas pedidas se generó el reporte de asistencia para el personal pedido.

3.6 DIAGRAMA DE CLASES

3.6.1 Diagrama de Clases General

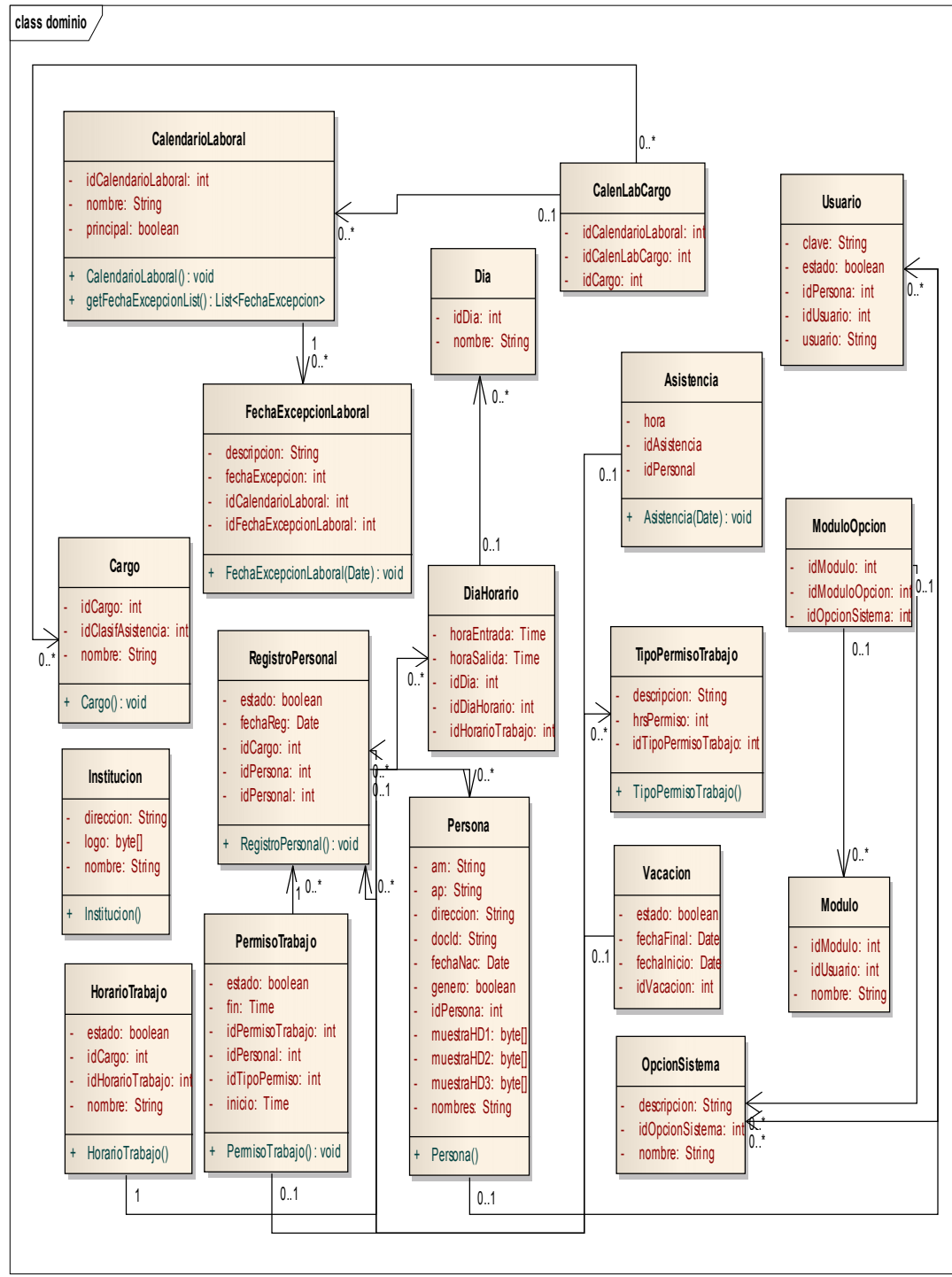


Figura 20. Diagrama de Clases General

3.7 VISTA DE ITERACIÓN

3.7.1 DIAGRAMA DE SECUENCIA

I. MODULO DEL SISTEMA

Caso de Uso: Iniciar Sesión

Autor: Administrador del Sistema

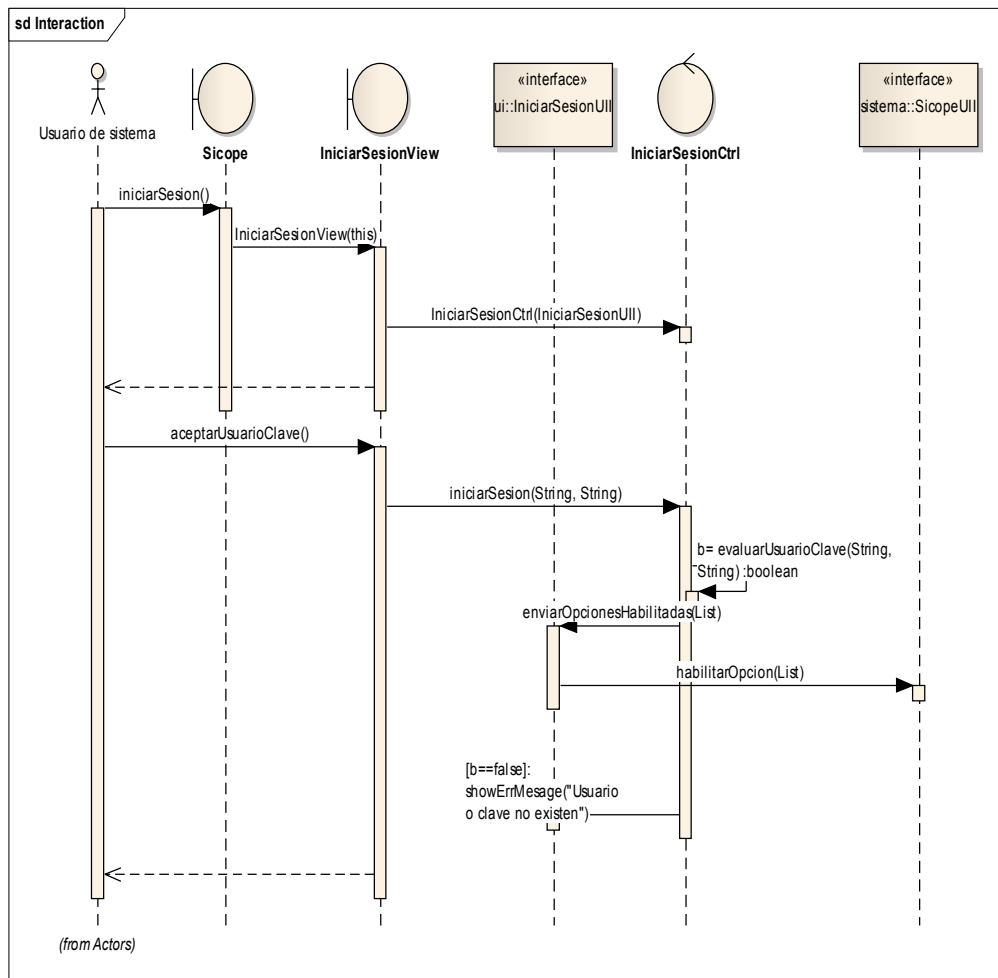


Figura 21. Diagrama de secuencia: Iniciar Sesión

Caso de Uso: Cerrar Sesión

Autor: Administrador del Sistema

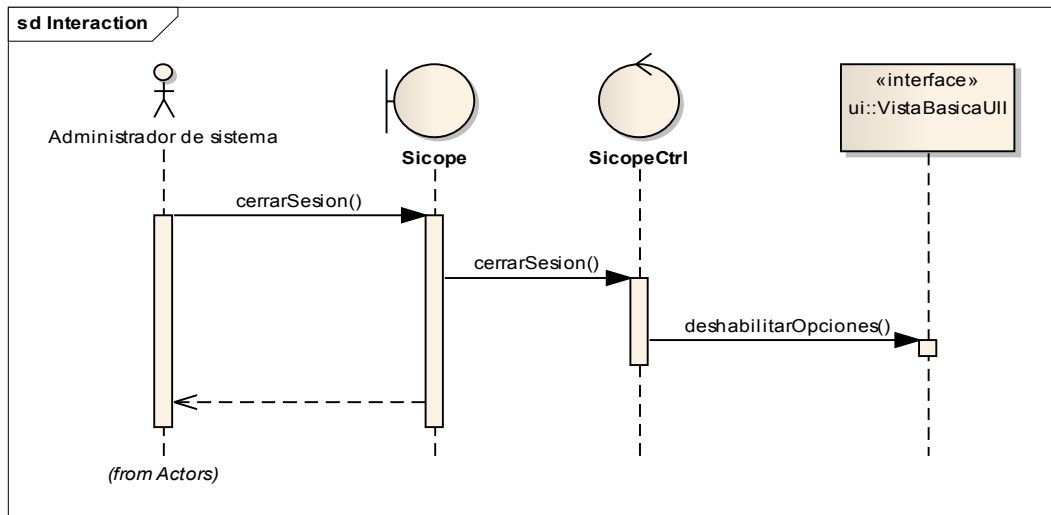


Figura 22. Diagrama de secuencia: Cerrar Sesión

Caso de Uso: Asignar Usuario y Clave

Autor: Administrador del Sistema

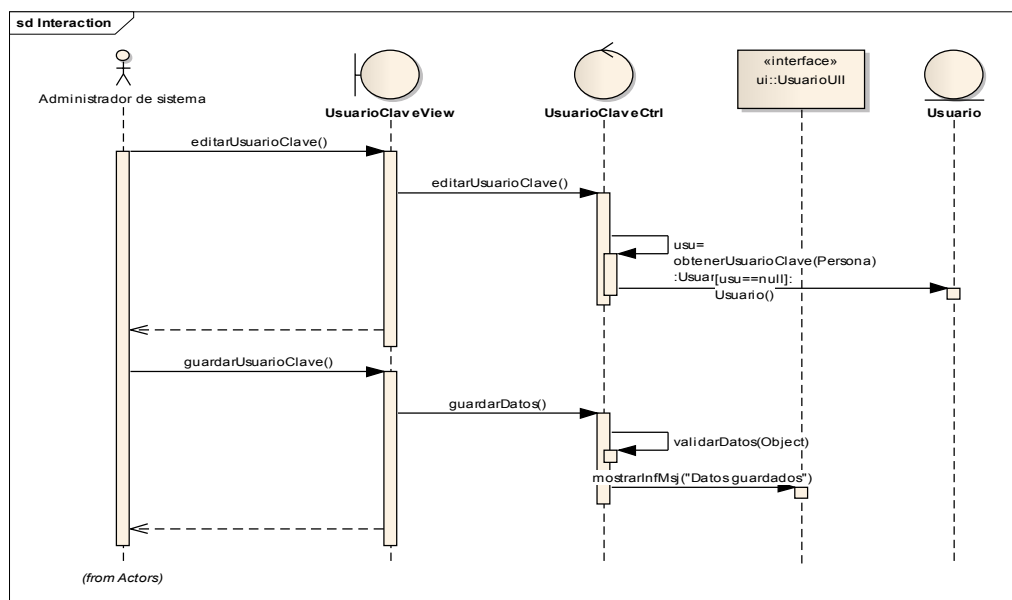


Figura 23. Diagrama de secuencia: Asignar Usuario y Clave

II. MÓDULO DE ASISTENCIA

Caso de Uso: Registrar Asistencia por Huella Digital

Autor: Administrador del Sistema, Funcionarios

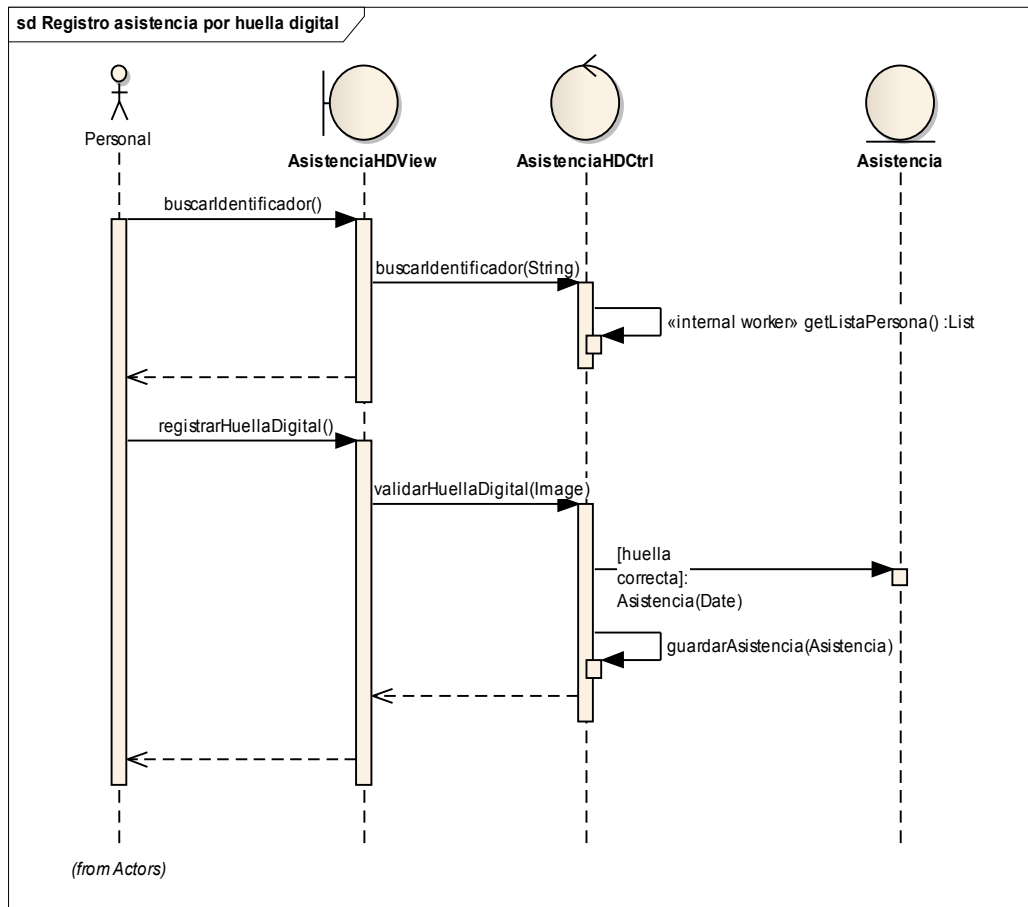


Figura 24. Diagrama de secuencia: Registrar Asistencia por Huella Digital

Caso de Uso: Registrar Asistencia por Código de Persona

Autor: Administrador del Sistema, Funcionarios

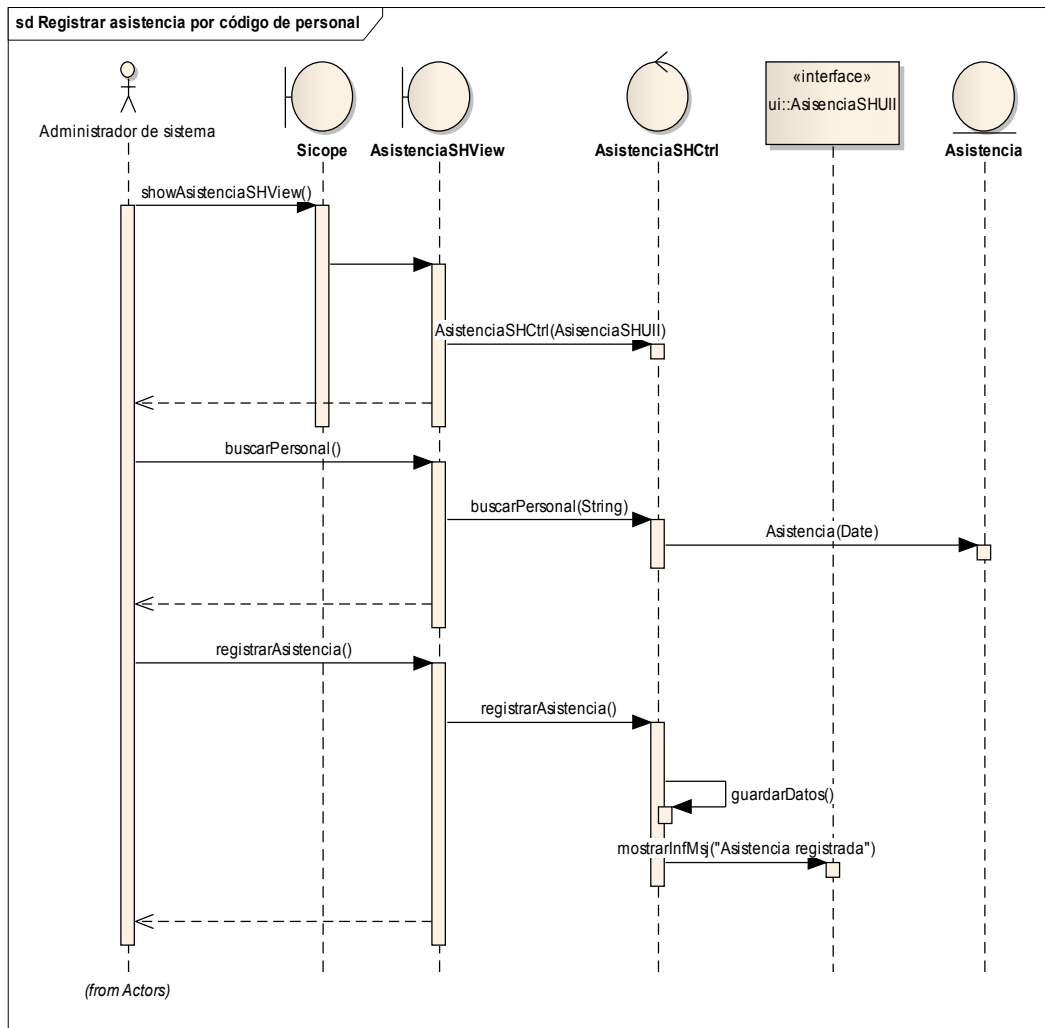


Figura 25. Diagrama de secuencia: Registrar Asistencia por Código de Persona

Caso de Uso: Gestionar Calendario Laboral

Autor: Administrador del Sistema

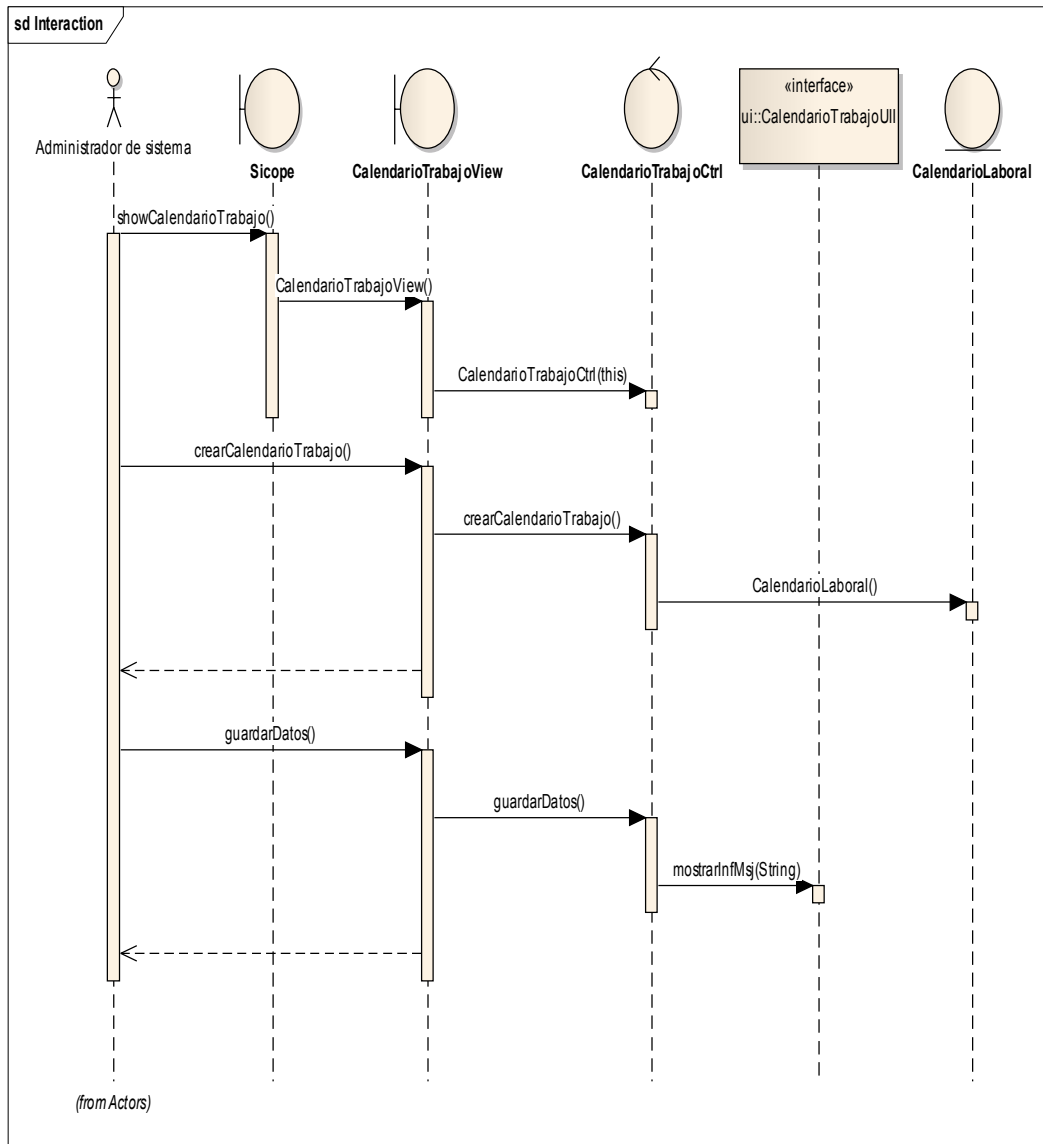


Figura 26. Diagrama de secuencia: Gestionar Calendario Laboral

Caso de Uso: Modificar Calendario Laboral

Autor: Administrador del Sistema

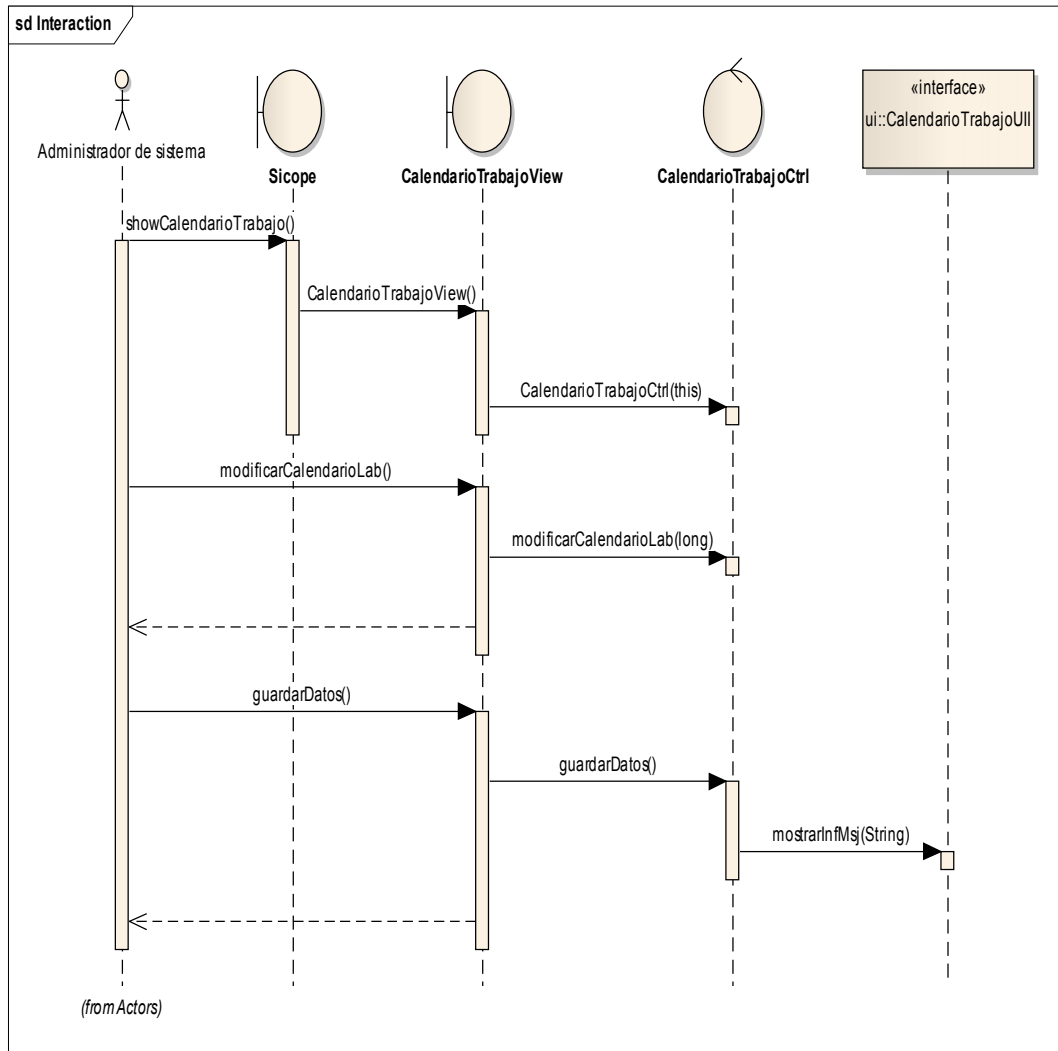


Figura 27. Diagrama de secuencia: Modificar Calendario Laboral

Caso de Uso: Borrar Calendario Laboral

Autor: Administrador del Sistema

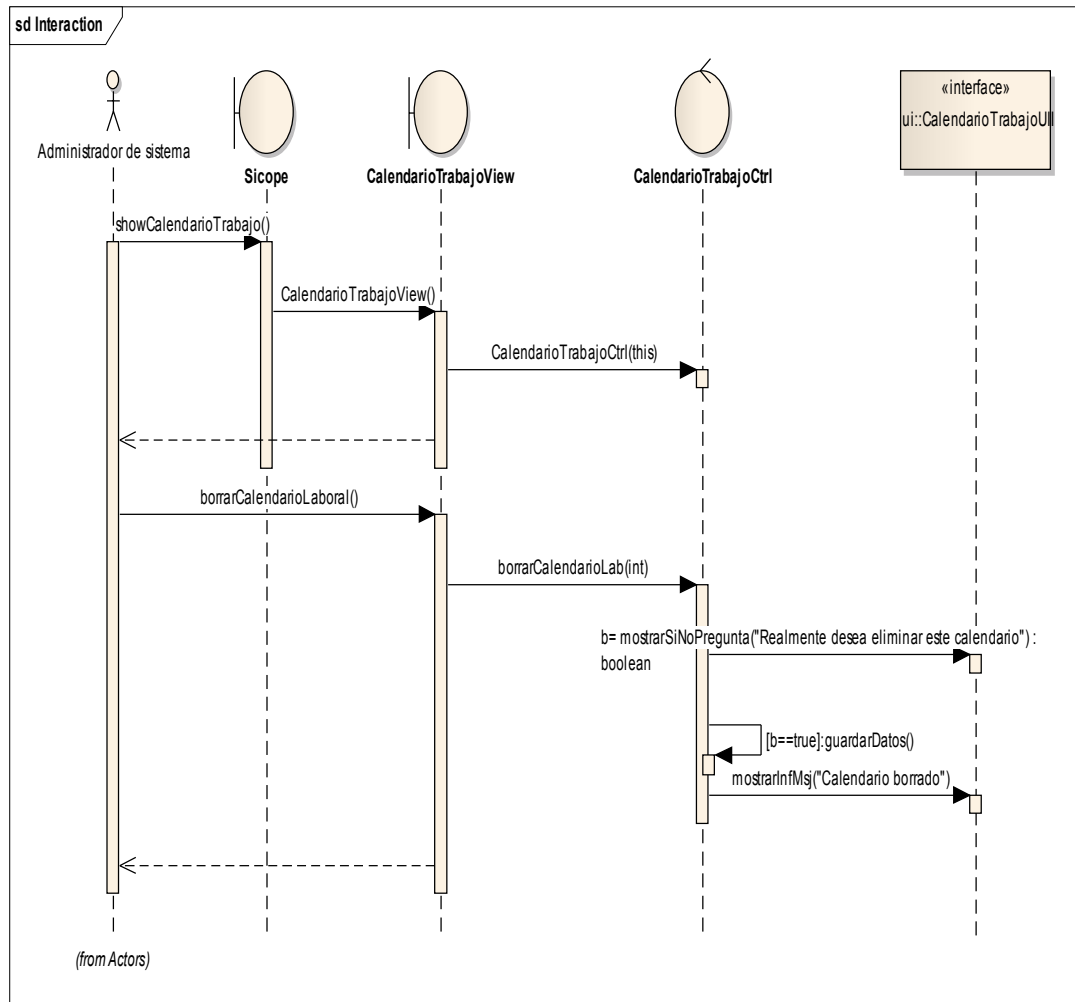


Figura 28. Diagrama de secuencia: Borrar Calendario Laboral

Caso de Uso: Crear Fecha de Excepción Laboral

Autor: Administrador del Sistema

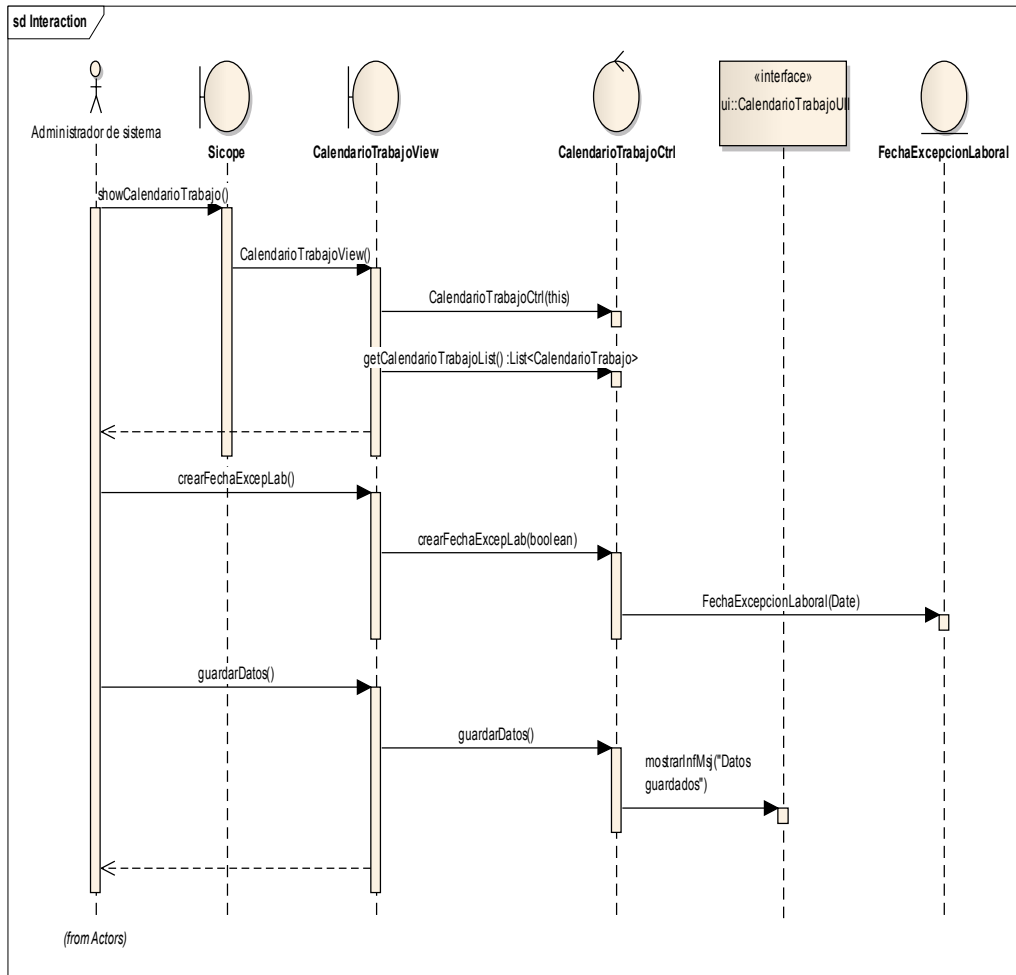


Figura 29. Diagrama de secuencia: Crear Fecha de Excepción Laboral

Caso de Uso: Borrar Fecha de Excepción Laboral

Autor: Administrador del Sistema

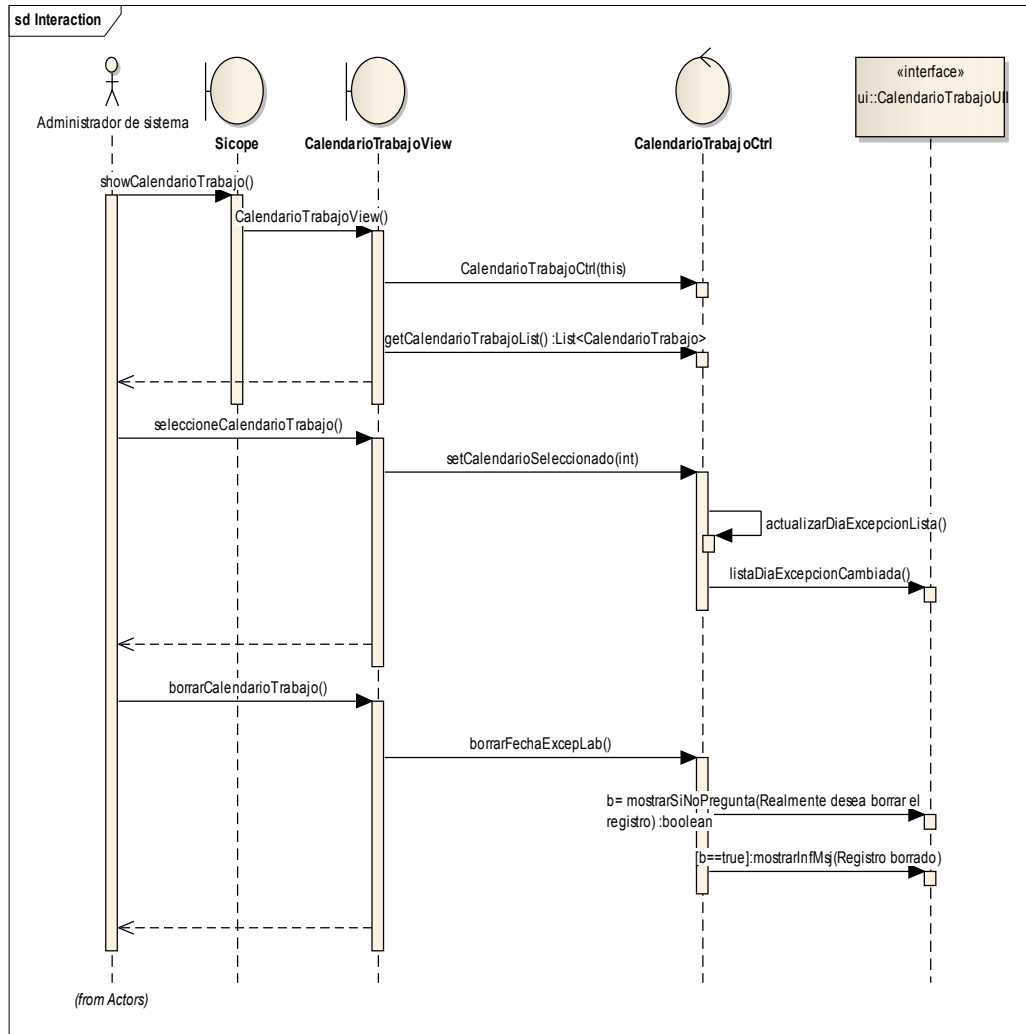


Figura 30. Diagrama de secuencia: Borrar Fecha de Excepción Laboral

Caso de Uso: Crear Permiso de Trabajo

Autor: Administrador del Sistema

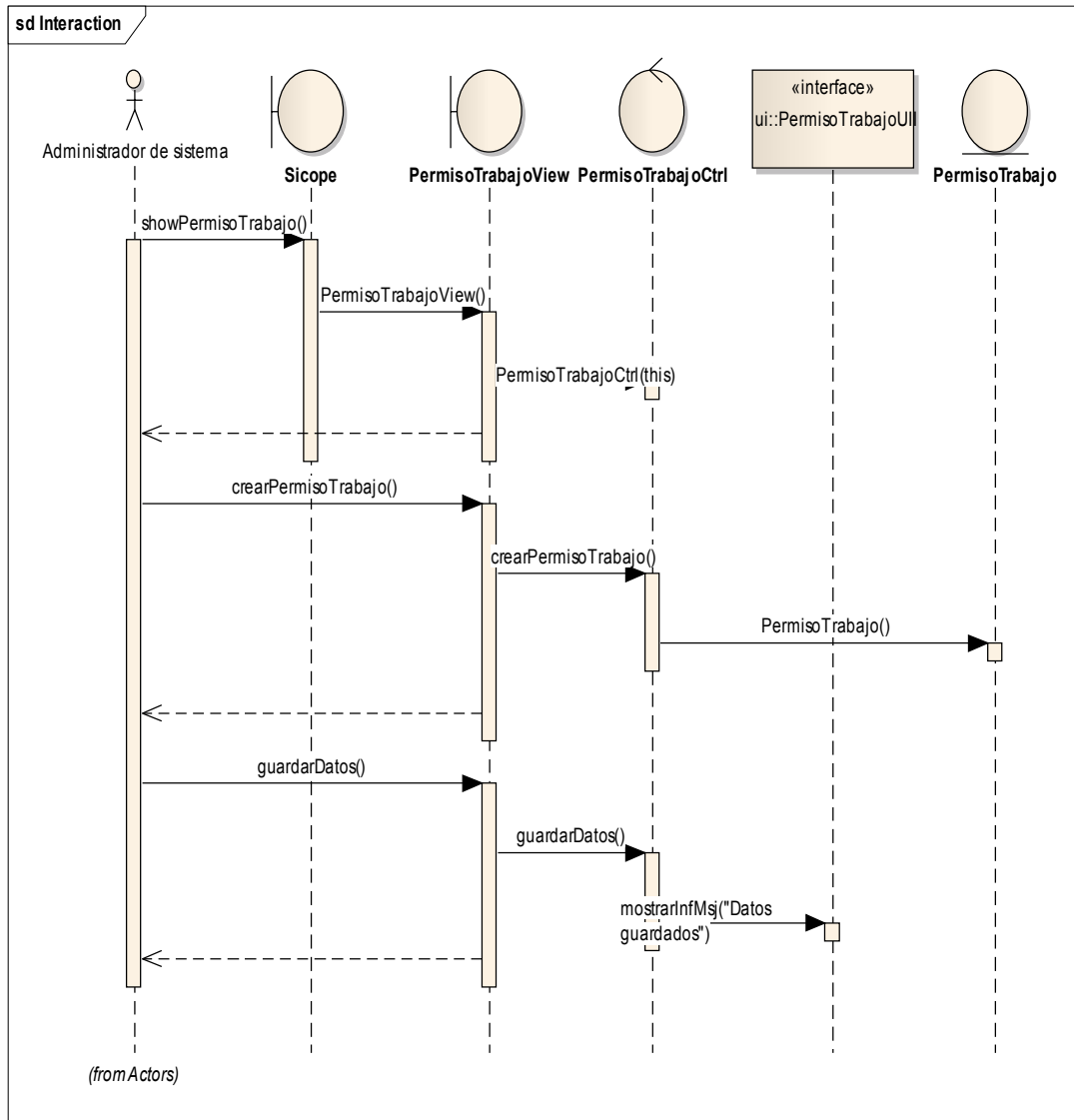


Figura 31. Diagrama de secuencia: Crear Permiso de Trabajo

Caso de Uso: Modificar Permiso de Trabajo

Autor: Administrador del Sistema

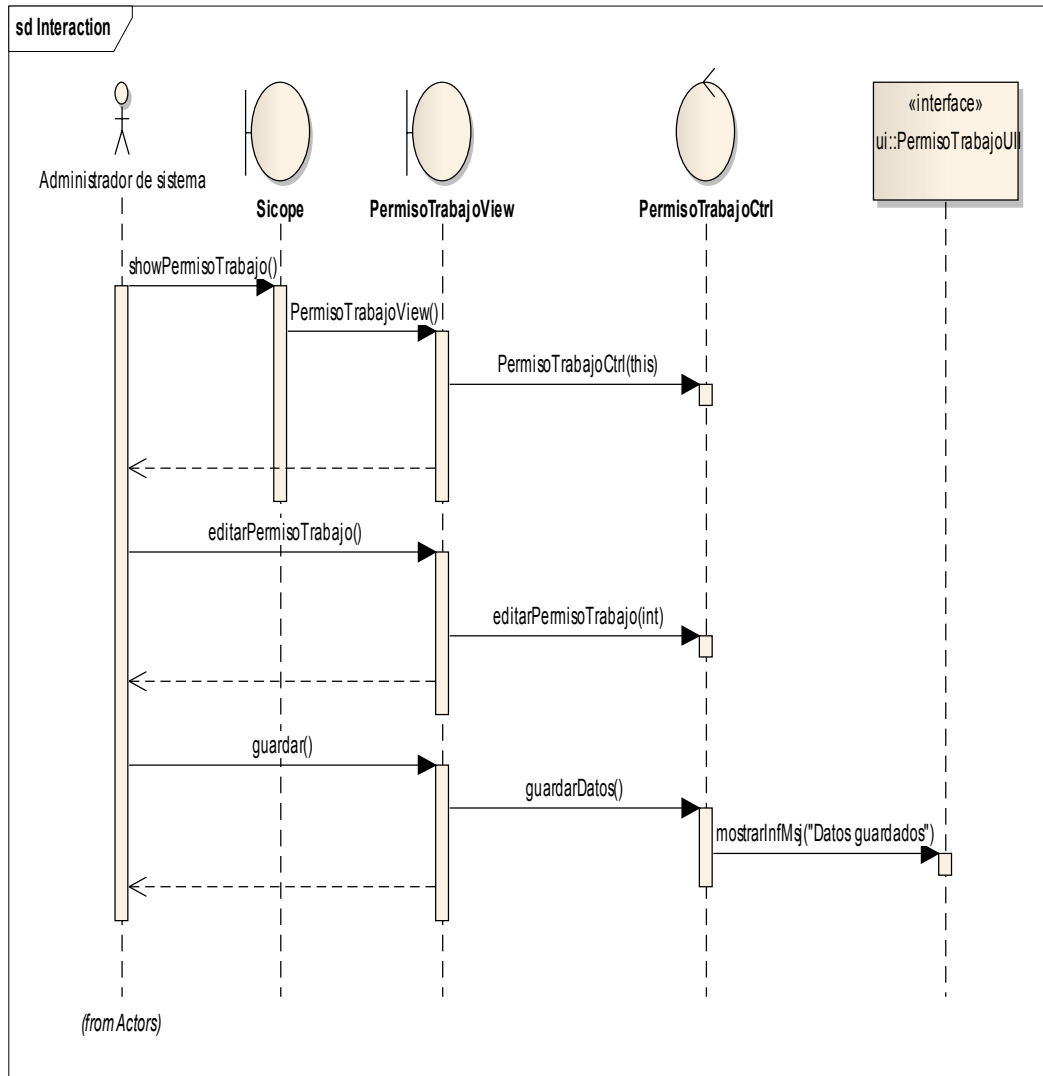


Figura 32. Diagrama de secuencia: Modificar Permiso de Trabajo

Caso de Uso: Borrar Permiso de Trabajo

Autor: Administrador del Sistema

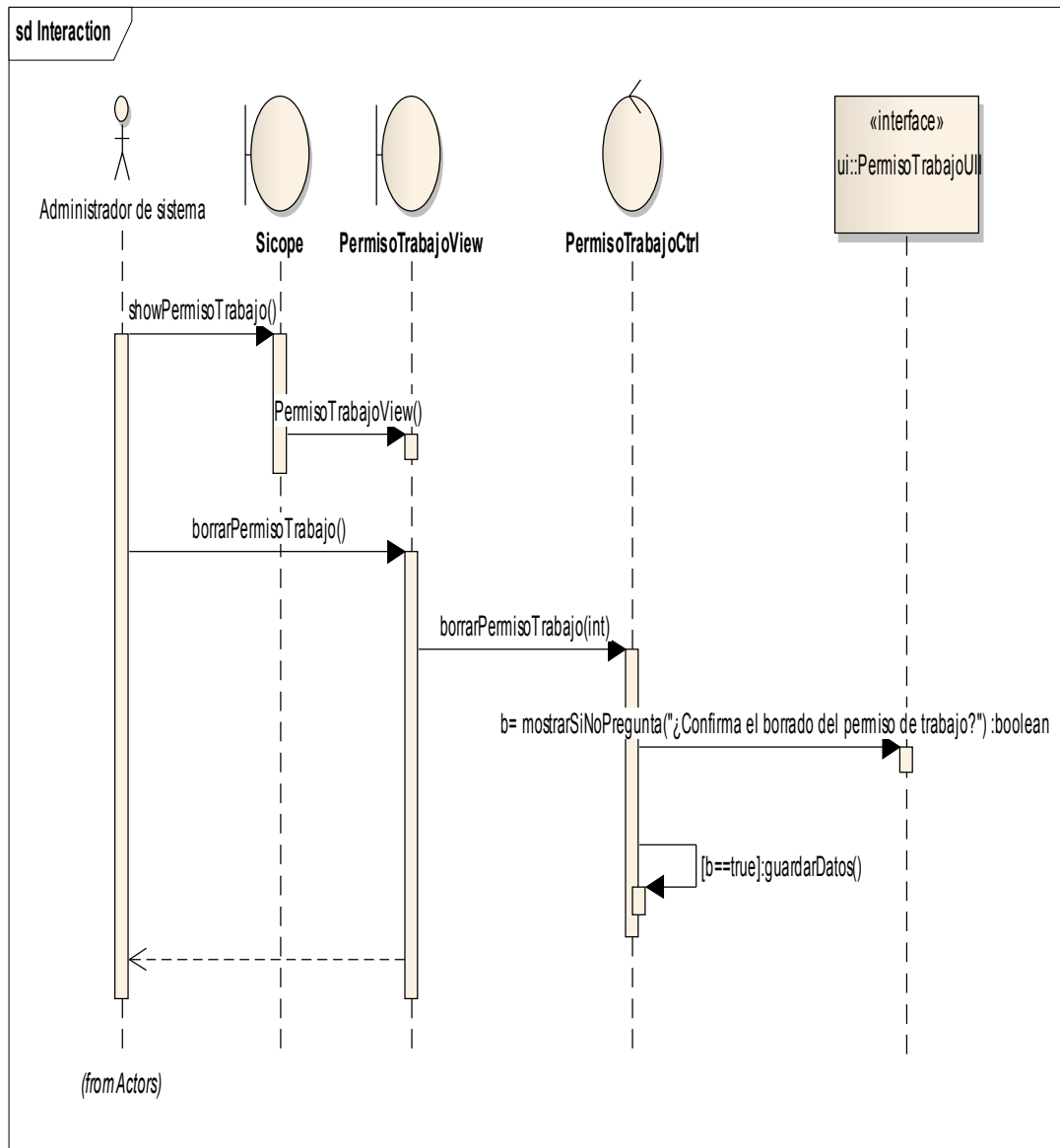


Figura 33. Diagrama de secuencia: Borrar Permiso de Trabajo

Caso de Uso: Crear Tipo de Permiso de Trabajo

Autor: Administrador del Sistema

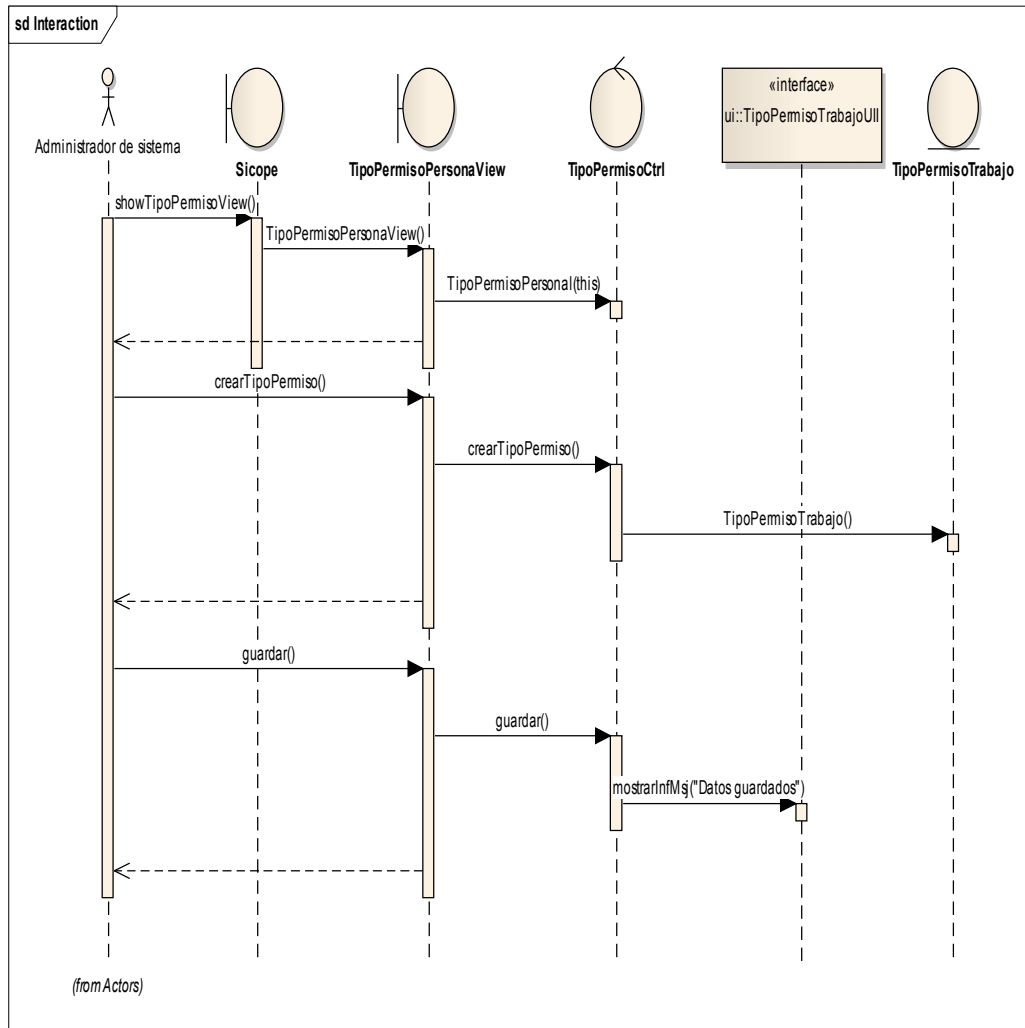


Figura 34. Diagrama de secuencia: Crear Tipo de Permiso de Trabajo

Caso de Uso: Modificar Tipo de Permiso de Trabajo

Autor: Administrador del Sistema

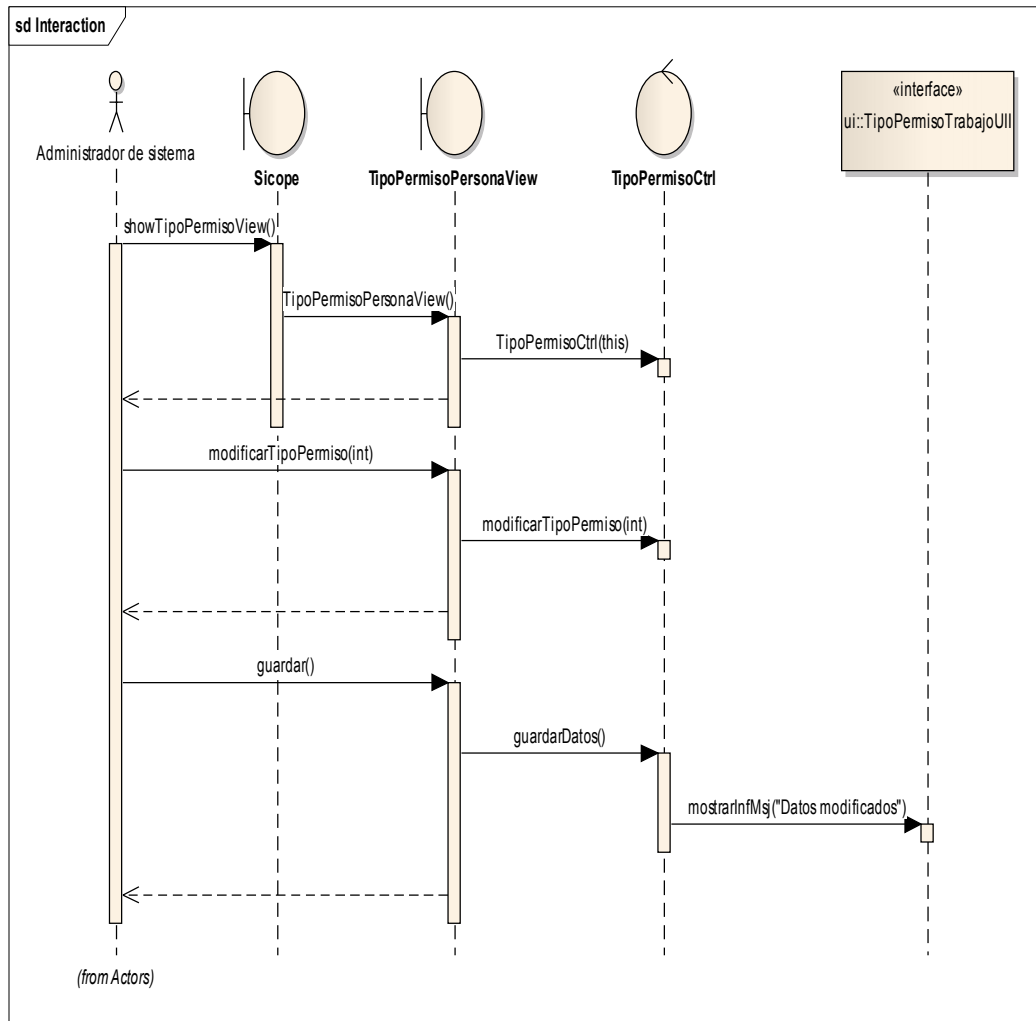


Figura 35. Diagrama de secuencia: Modificar Tipo de Permiso de Trabajo

Caso de Uso: Borrar Tipo de Permiso de Trabajo

Autor: Administrador del Sistema

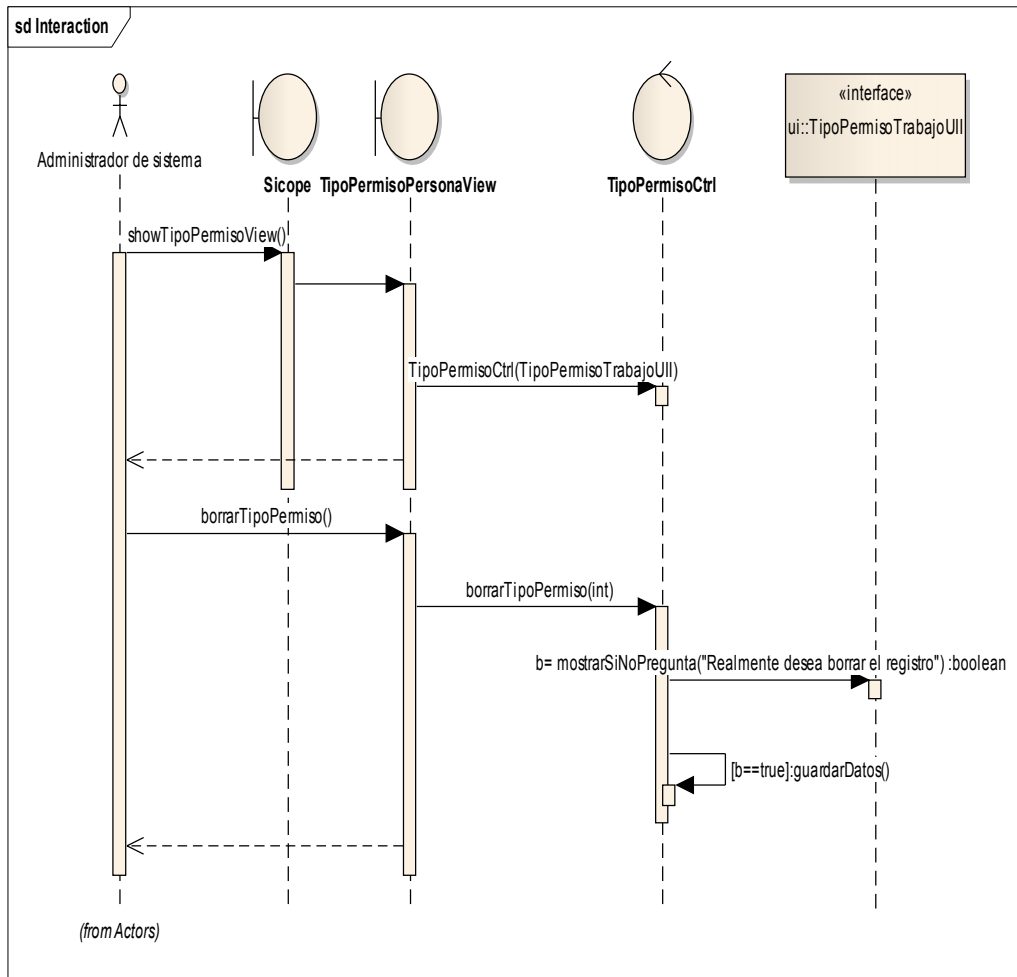


Figura 36. Diagrama de secuencia: Borrar Tipo de Permiso de Trabajo

Caso de Uso: Crear Horario de Trabajo

Autor: Administrador del Sistema

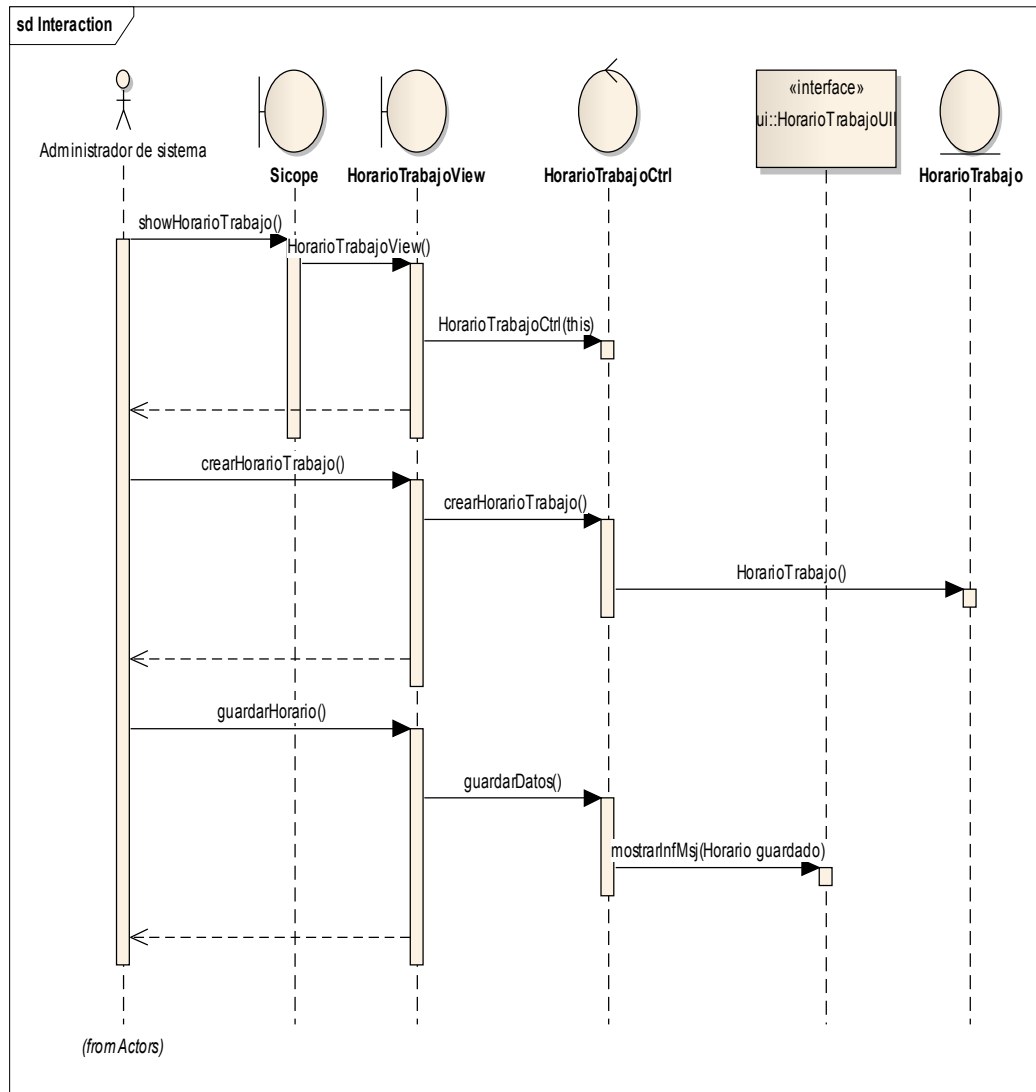


Figura 37. Diagrama de secuencia: Crear Horario de Trabajo

Caso de Uso: Modificar Horario de Trabajo

Autor: Administrador del Sistema

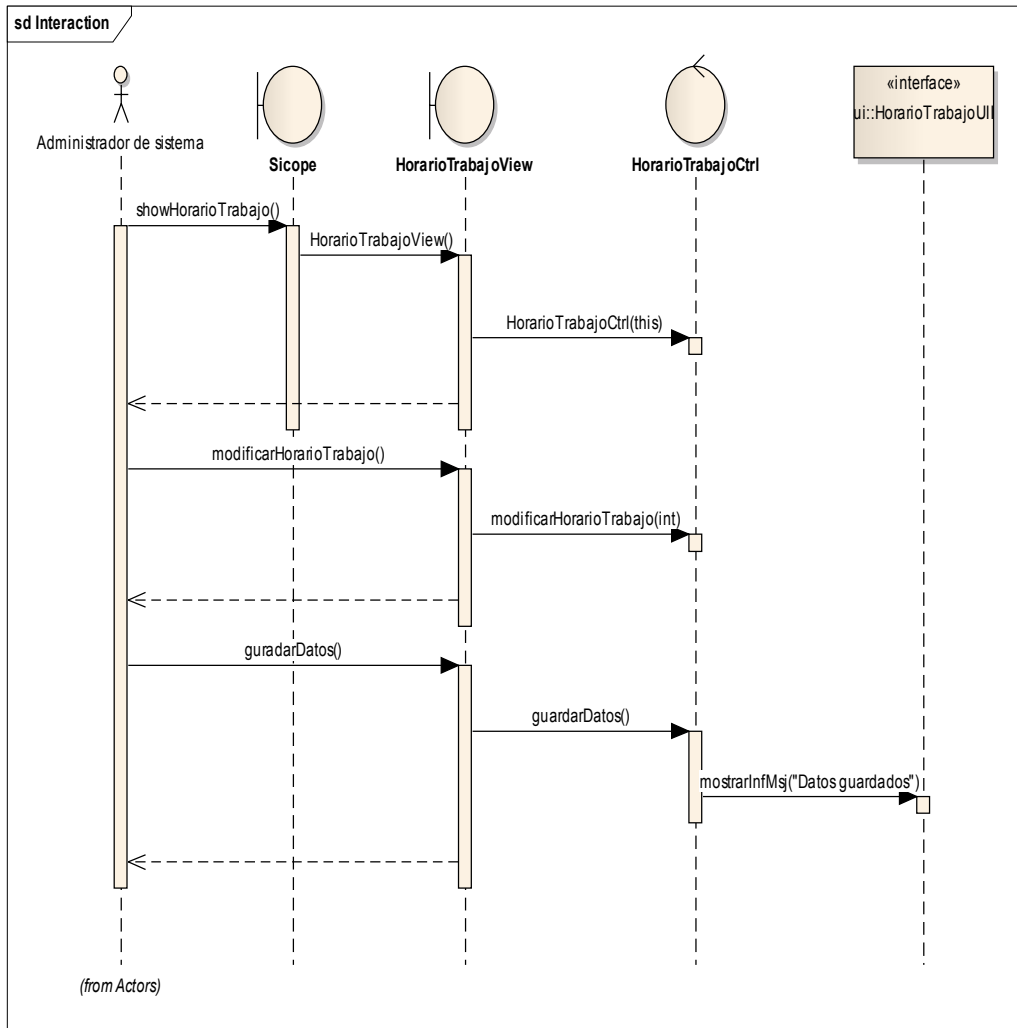


Figura 38. Diagrama de secuencia: Modificar Horario de Trabajo

Caso de Uso: Borrar Horario de Trabajo

Autor: Administrador del Sistema

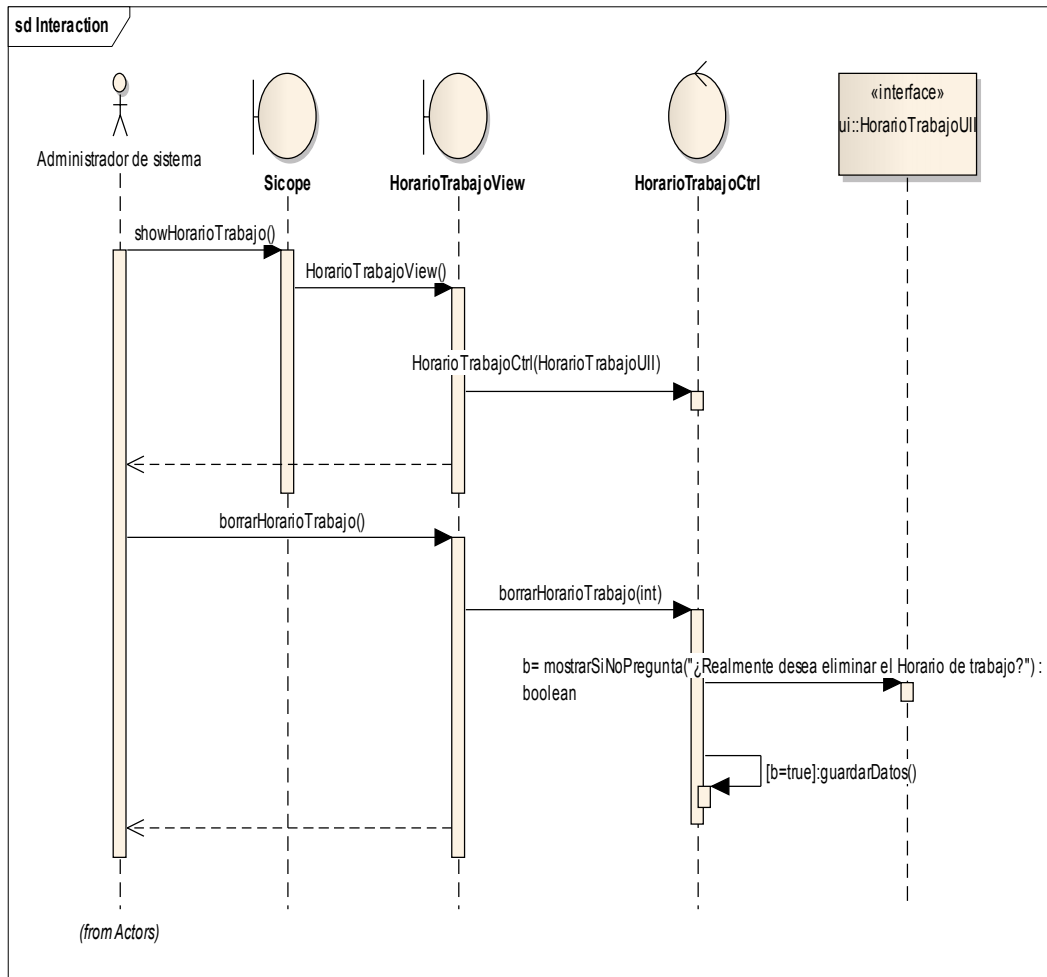


Figura 39. Diagrama de secuencia: Borrar Horario de Trabajo

Caso de Uso: Crear Registro de Personal

Autor: Administrador del Sistema

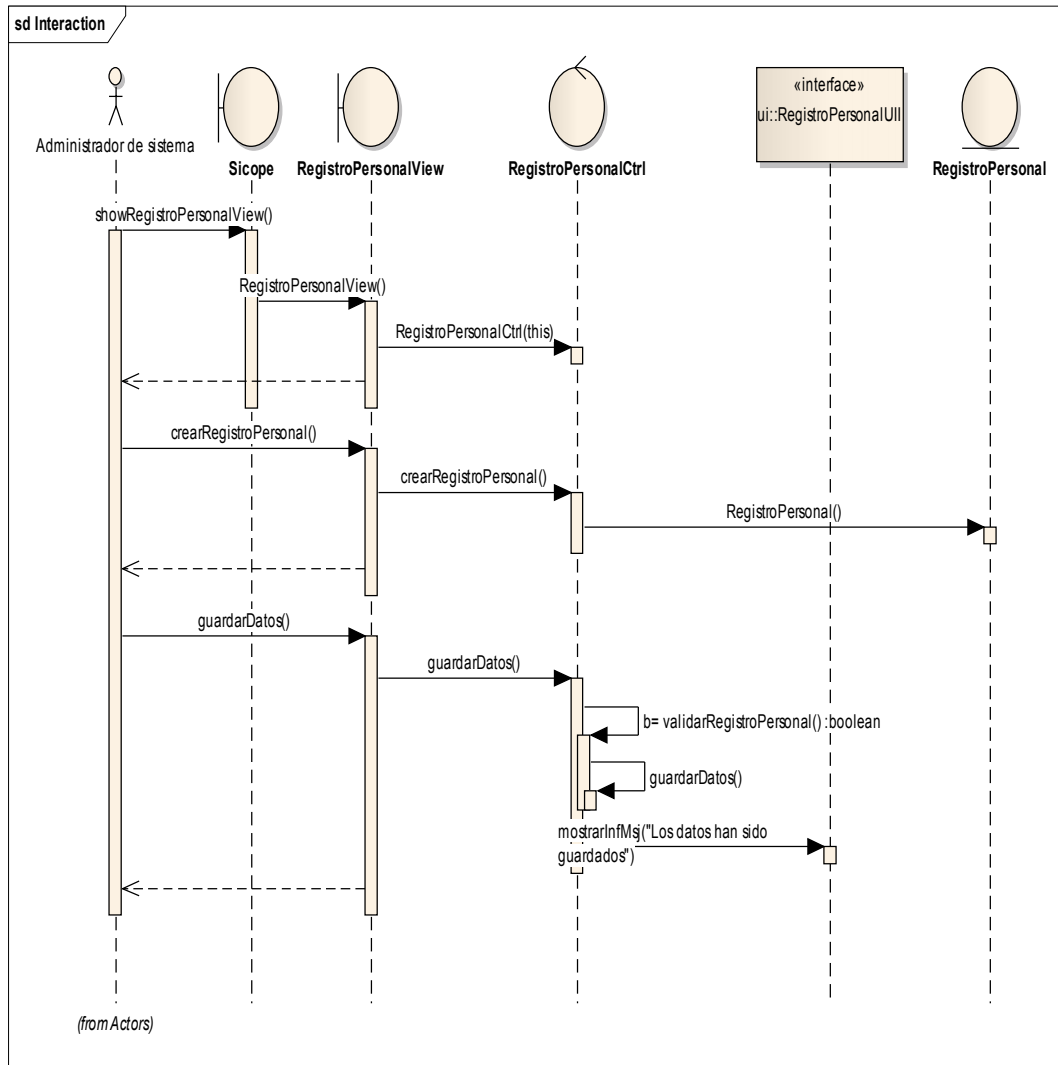


Figura 40. Diagrama de secuencia: Crear Registro de Personal

Caso de Uso: Modificar Registro de Personal

Autor: Administrador del Sistema

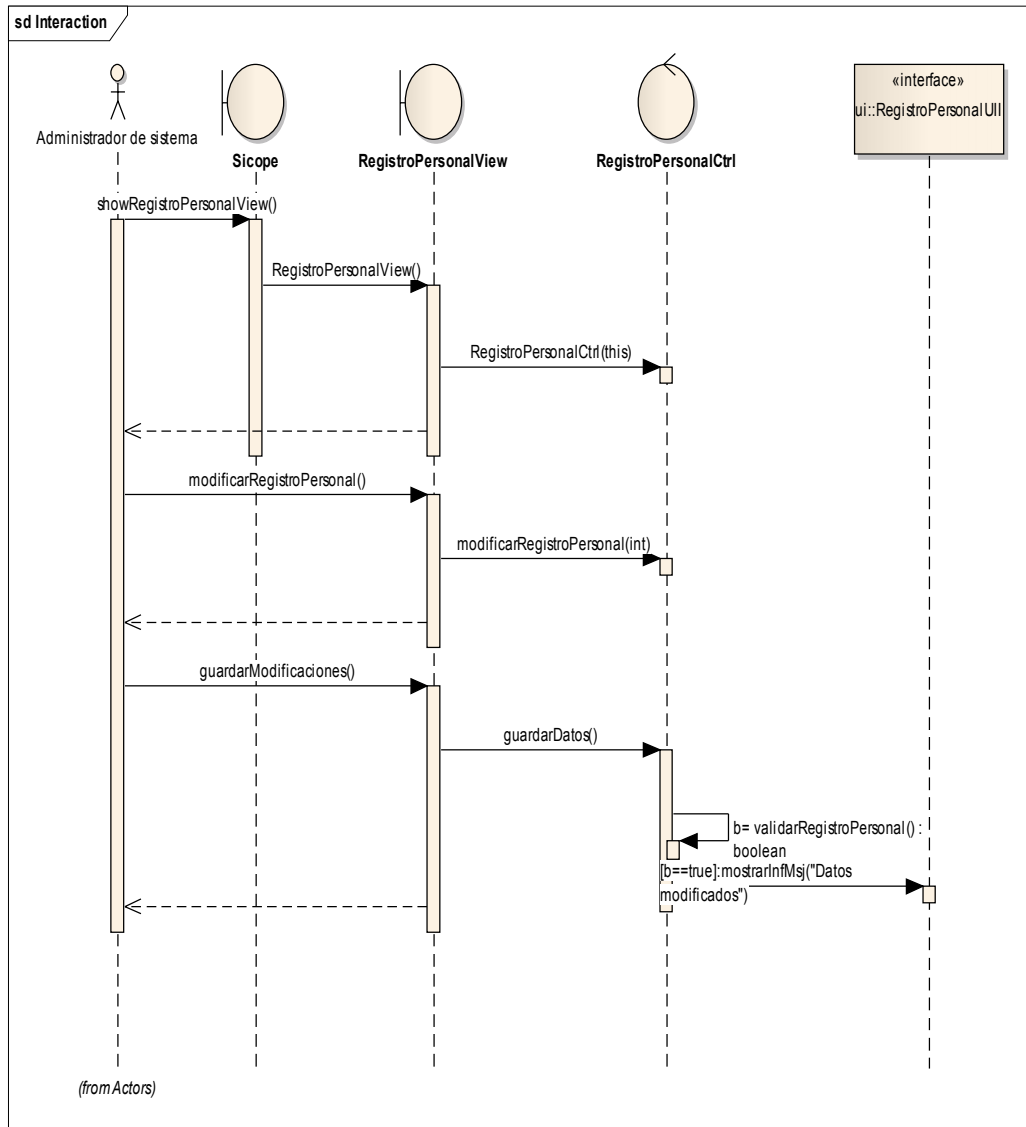


Figura 41. Diagrama de secuencia: Modificar Registro de Personal

Caso de Uso: Borrar Registro de Personal

Autor: Administrador del Sistema

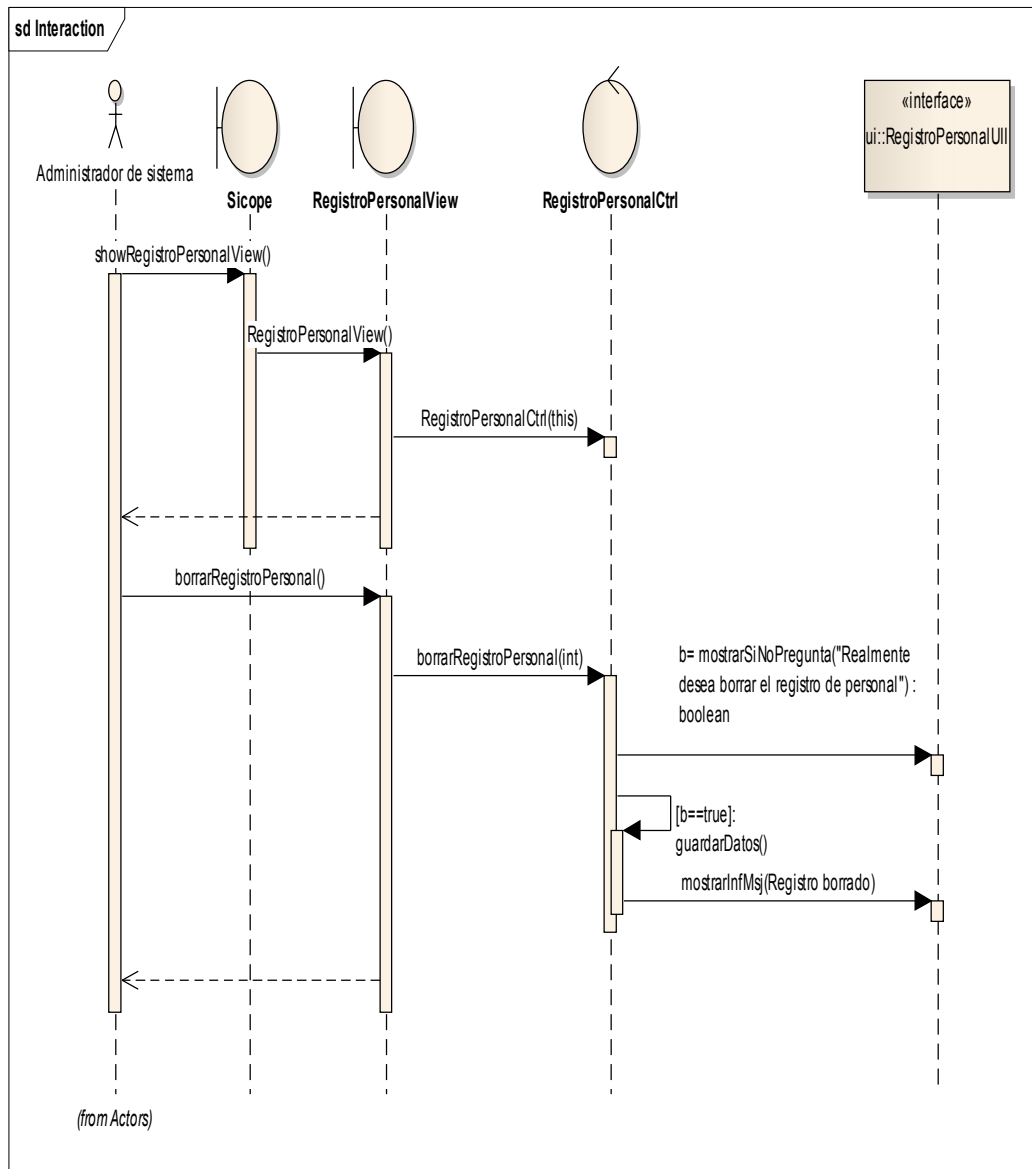


Figura 42. Diagrama de secuencia: Borrar Registro de Personal

Caso de Uso: Registrar Huella Digital

Autor: Administrador del Sistema, Funcionarios

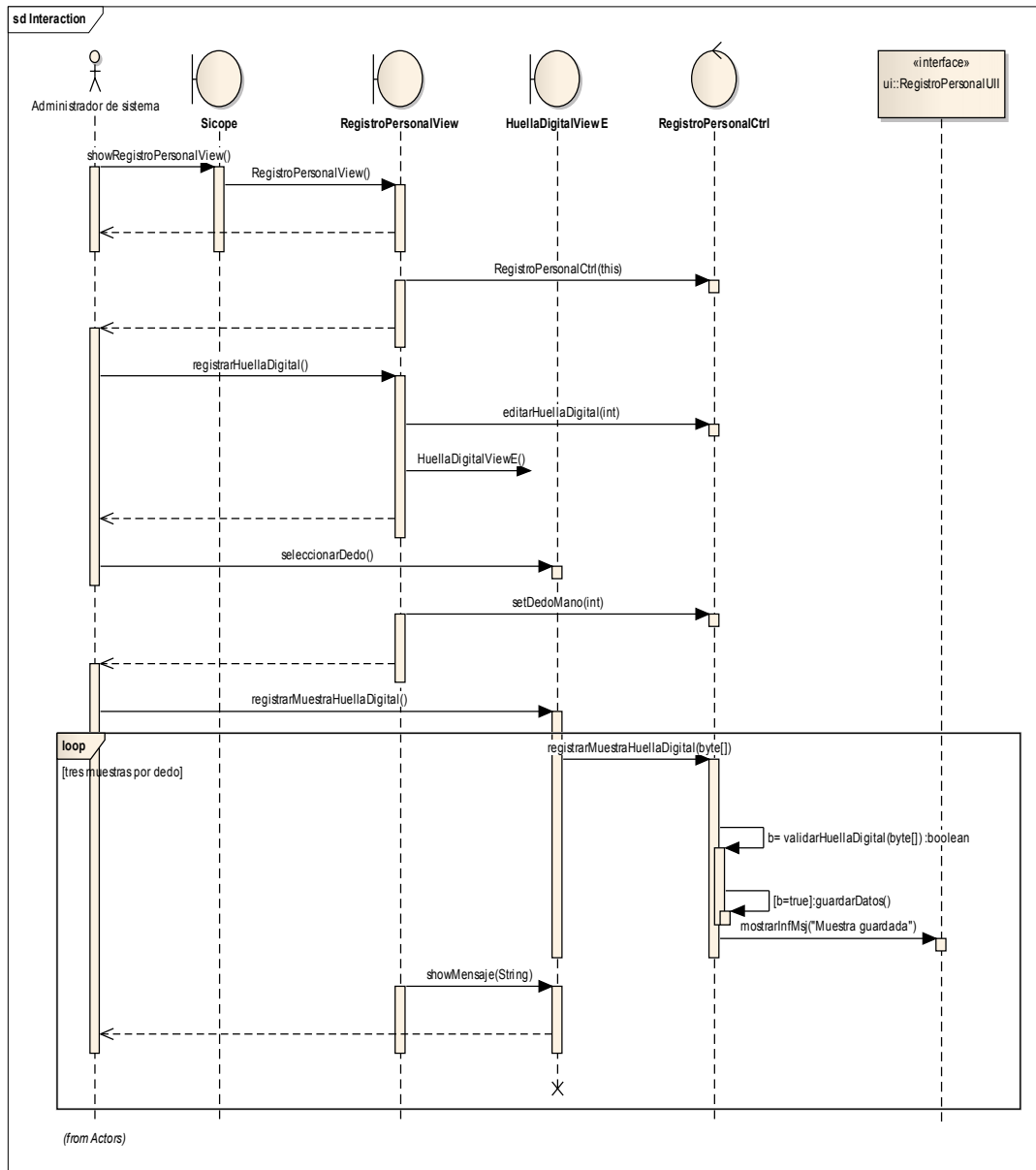


Figura 43. Diagrama de secuencia: Registrar Huella Digital

Caso de Uso: Crear Cargo

Autor: Administrador del Sistema

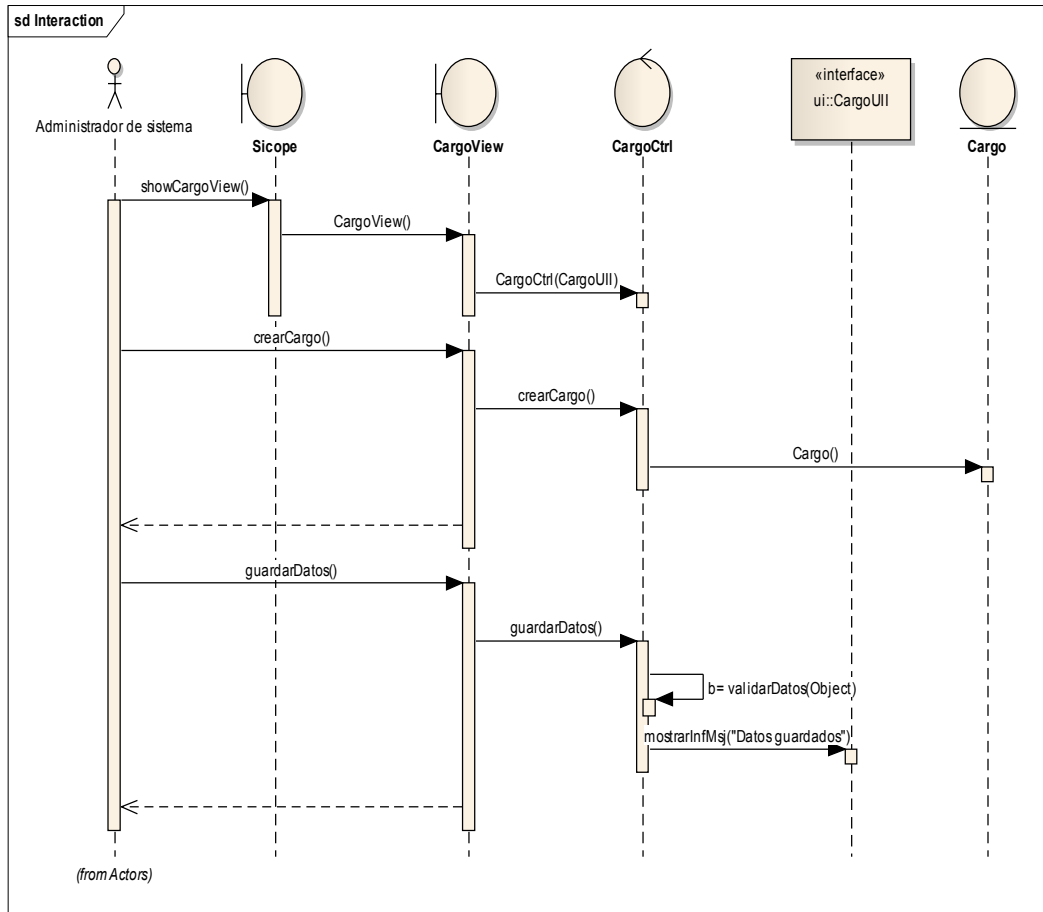


Figura 44. Diagrama de secuencia: Crear Cargo

Caso de Uso: Modificar Cargo

Autor: Administrador del Sistema

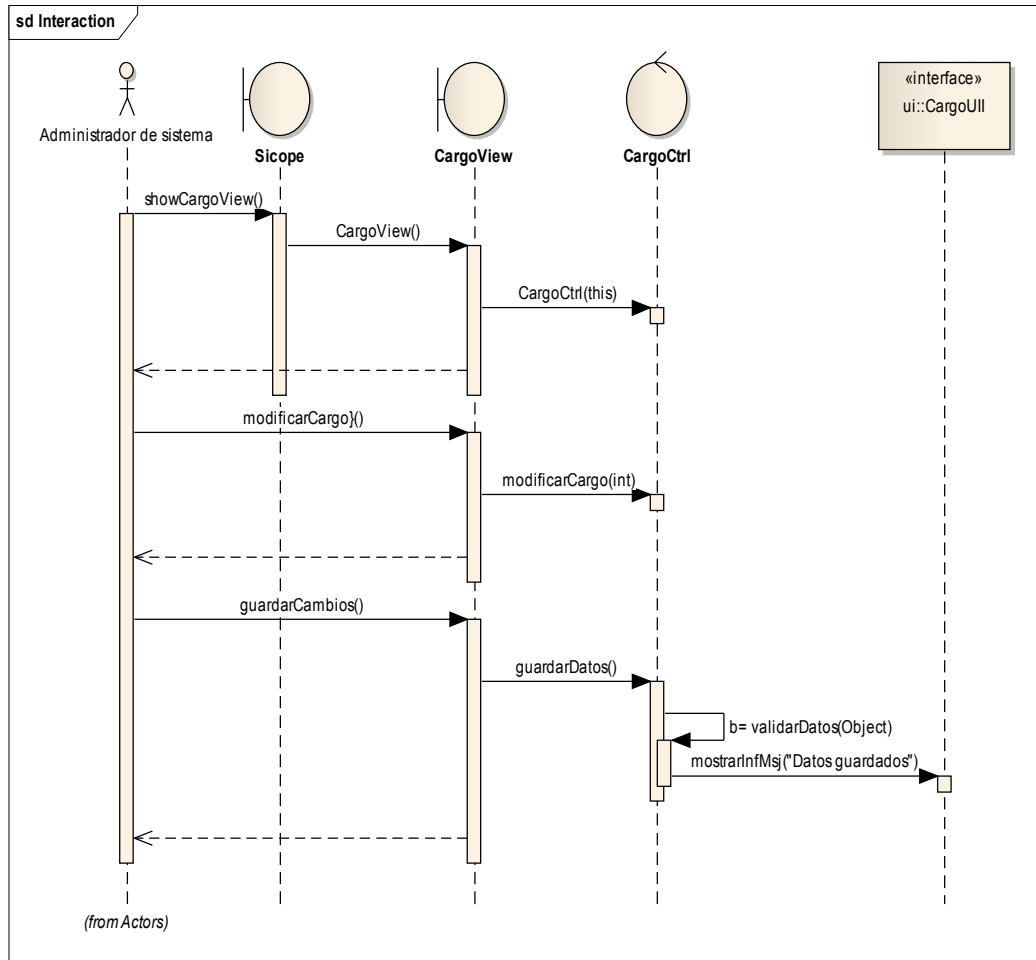


Figura 45. Diagrama de secuencia: Modificar Cargo

Caso de Uso: Borrar Cargo

Autor: Administrador del Sistema

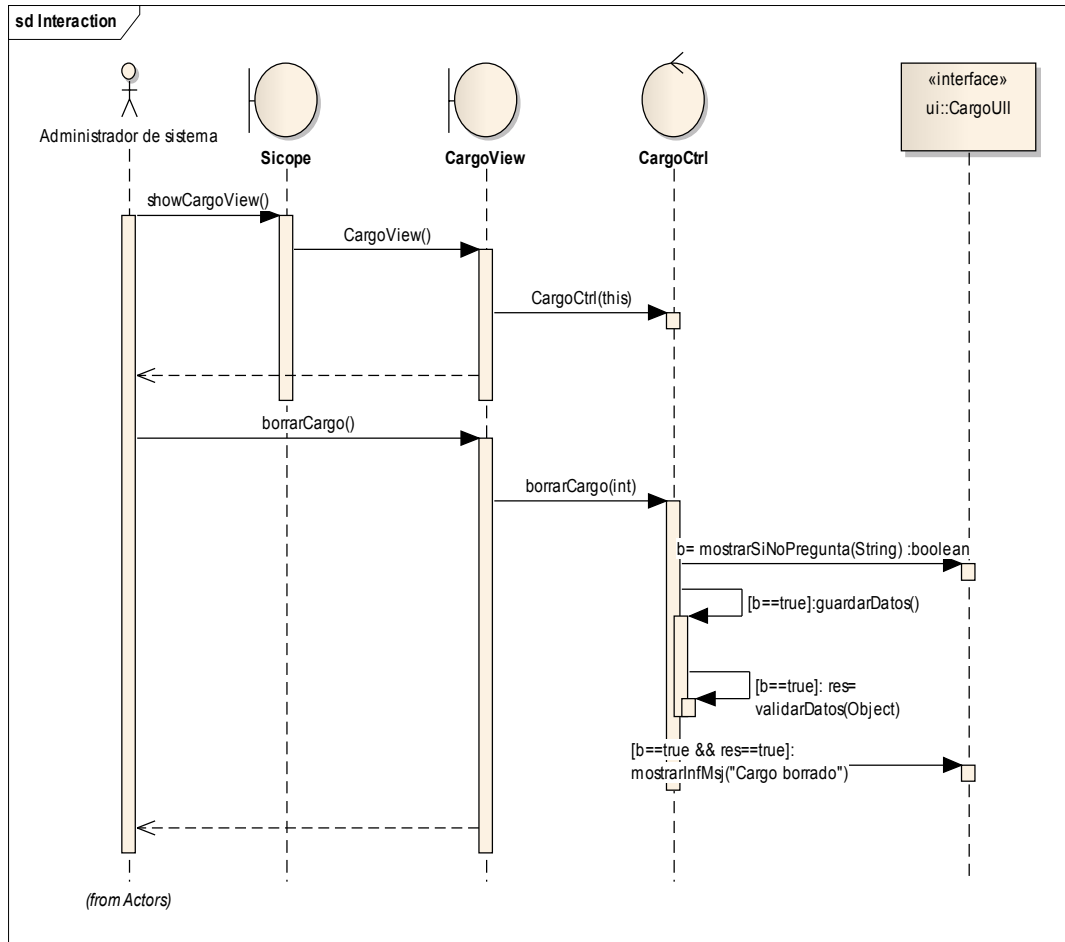


Figura 46. Diagrama de secuencia: Borrar Cargo

III. MÓDULO GENERAL

Caso de Uso: Crear Persona

Autor: Administrador del Sistema

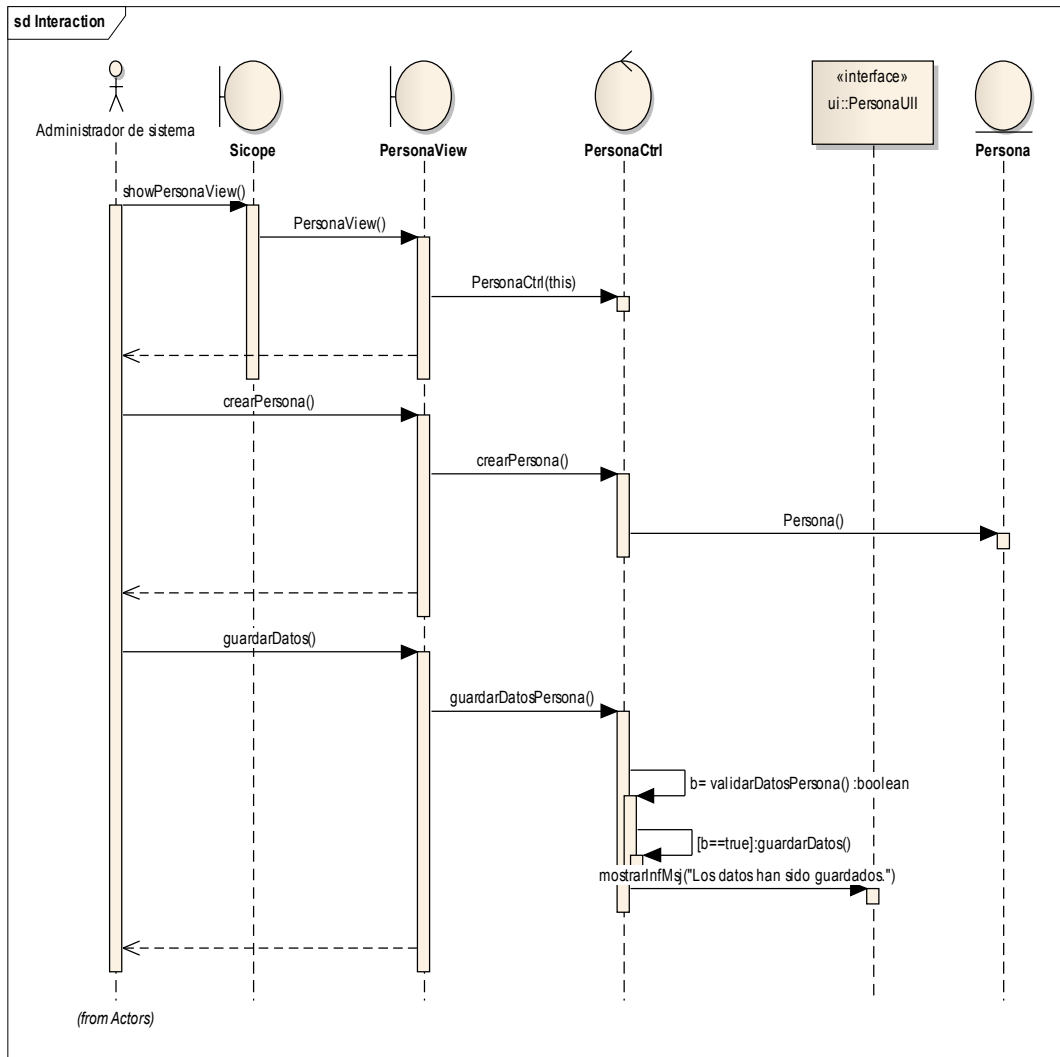


Figura 47. Diagrama de secuencia: Crear Persona

Caso de Uso: Modificar Persona

Autor: Administrador del Sistema

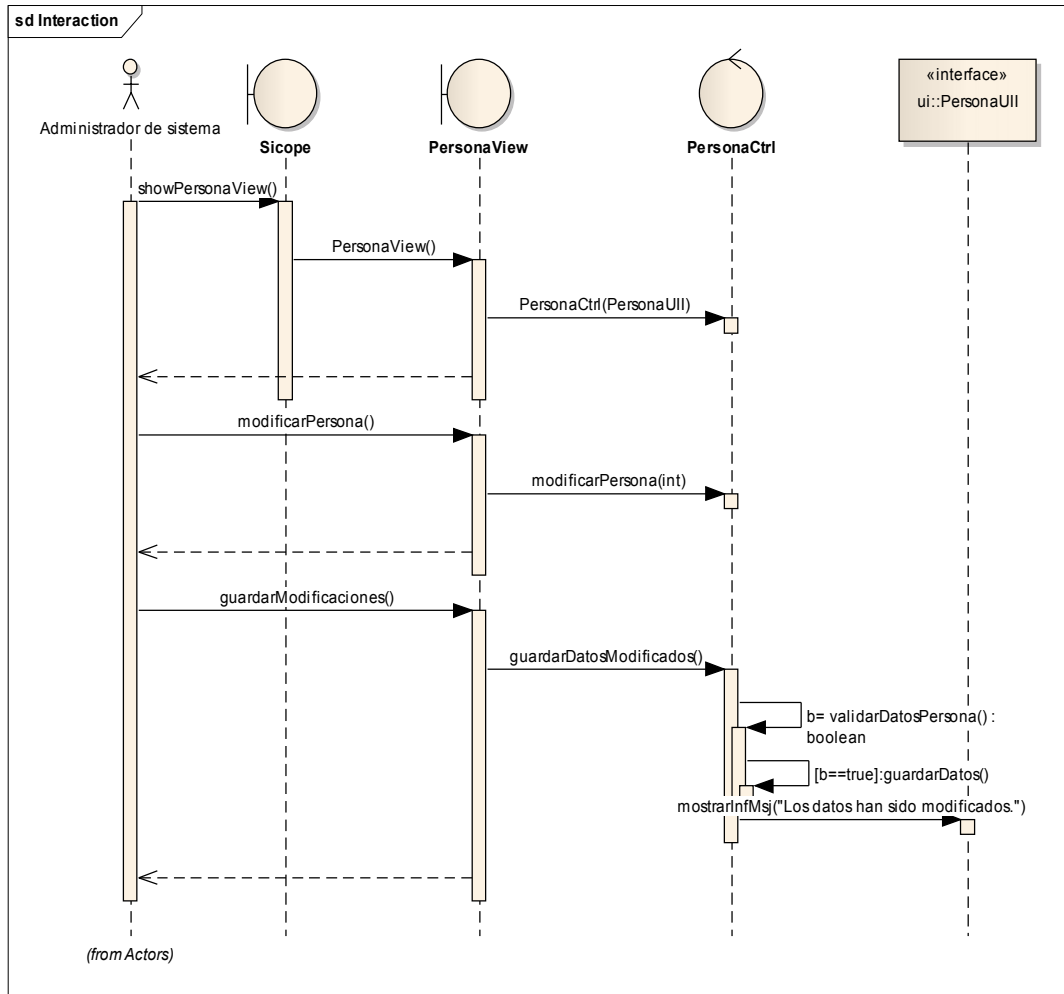


Figura 48. Diagrama de secuencia: Modificar Persona

Caso de Uso: Borrar Persona

Autor: Administrador del Sistema

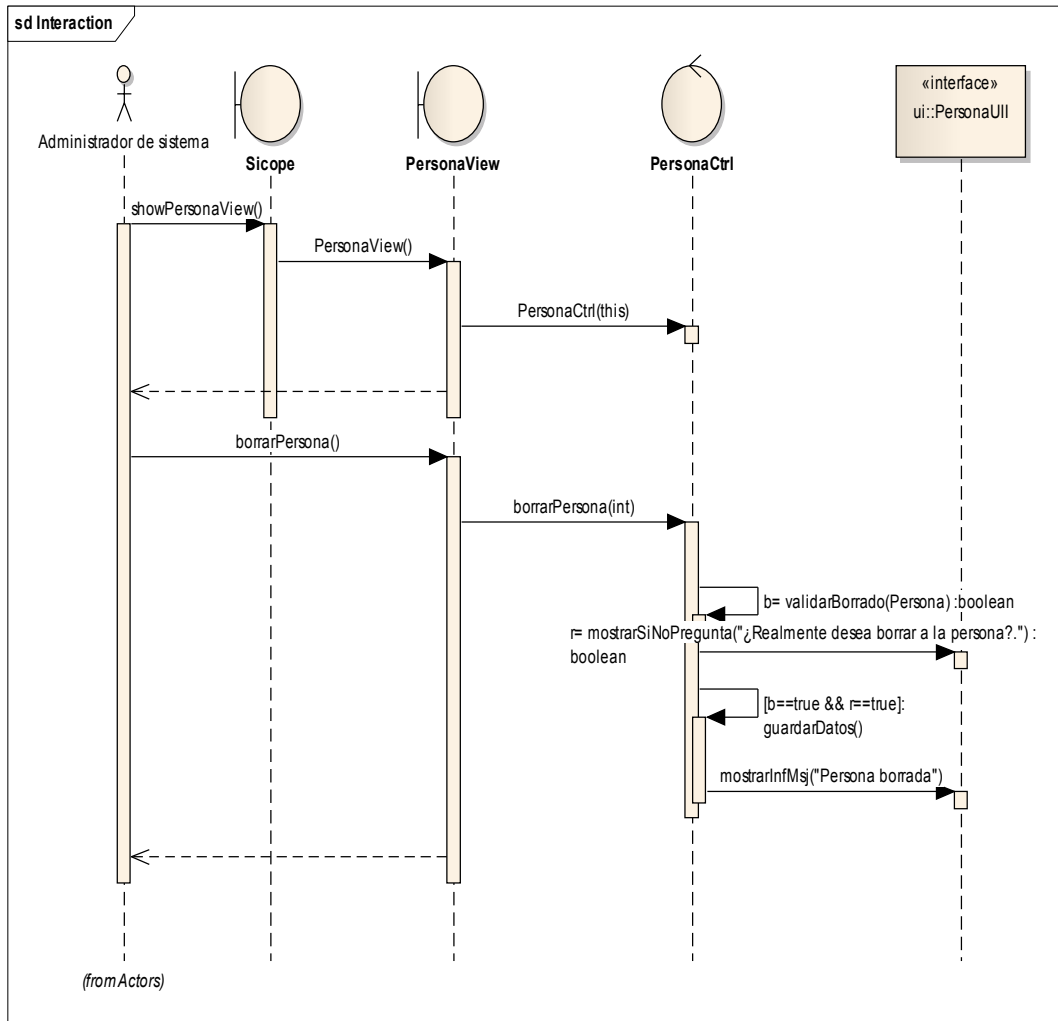


Figura 49. Diagrama de secuencia: Borrar Persona

Caso de Uso: Gestionar Institución

Autor: Administrador del Sistema

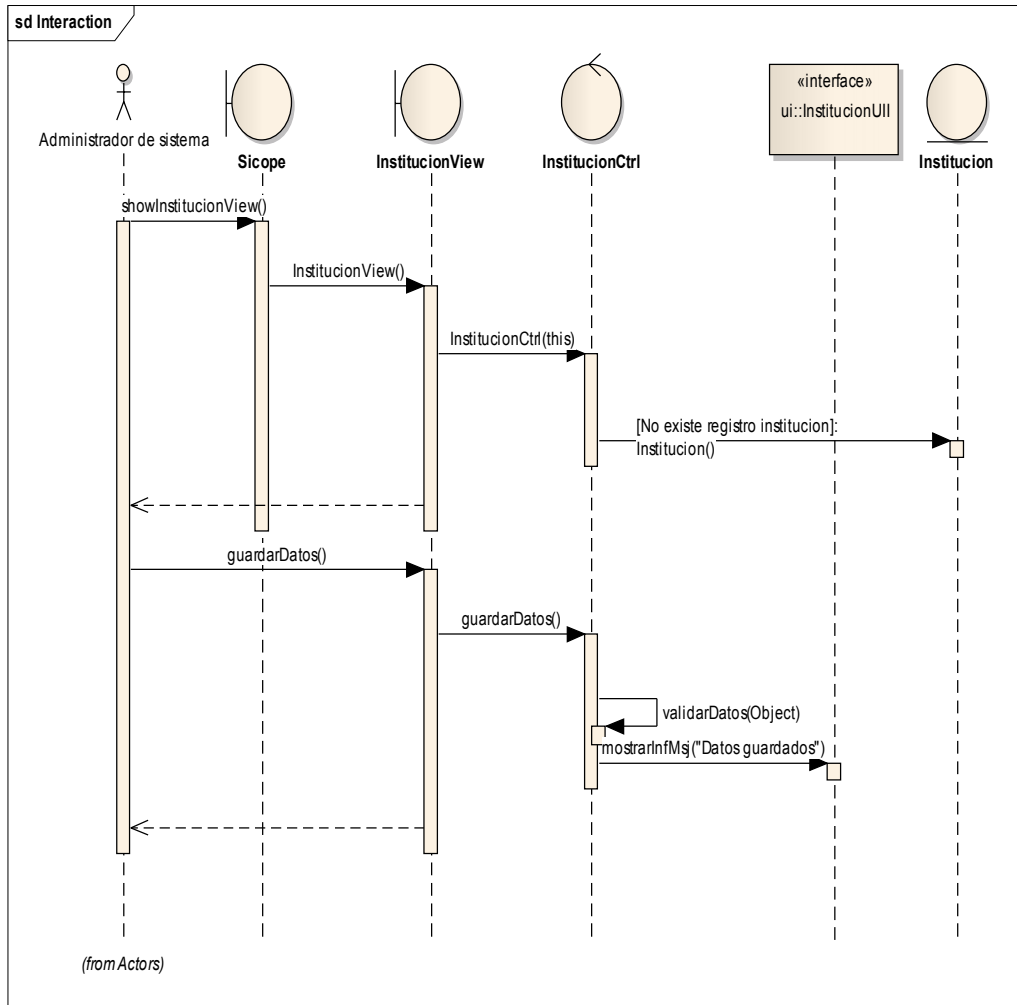


Figura 50. Diagrama de secuencia: Gestionar Institución

IV. MÓDULO DE REPORTES

Caso de Uso: Generar Reporte de Asistencia

Autor: Administrador del Sistema

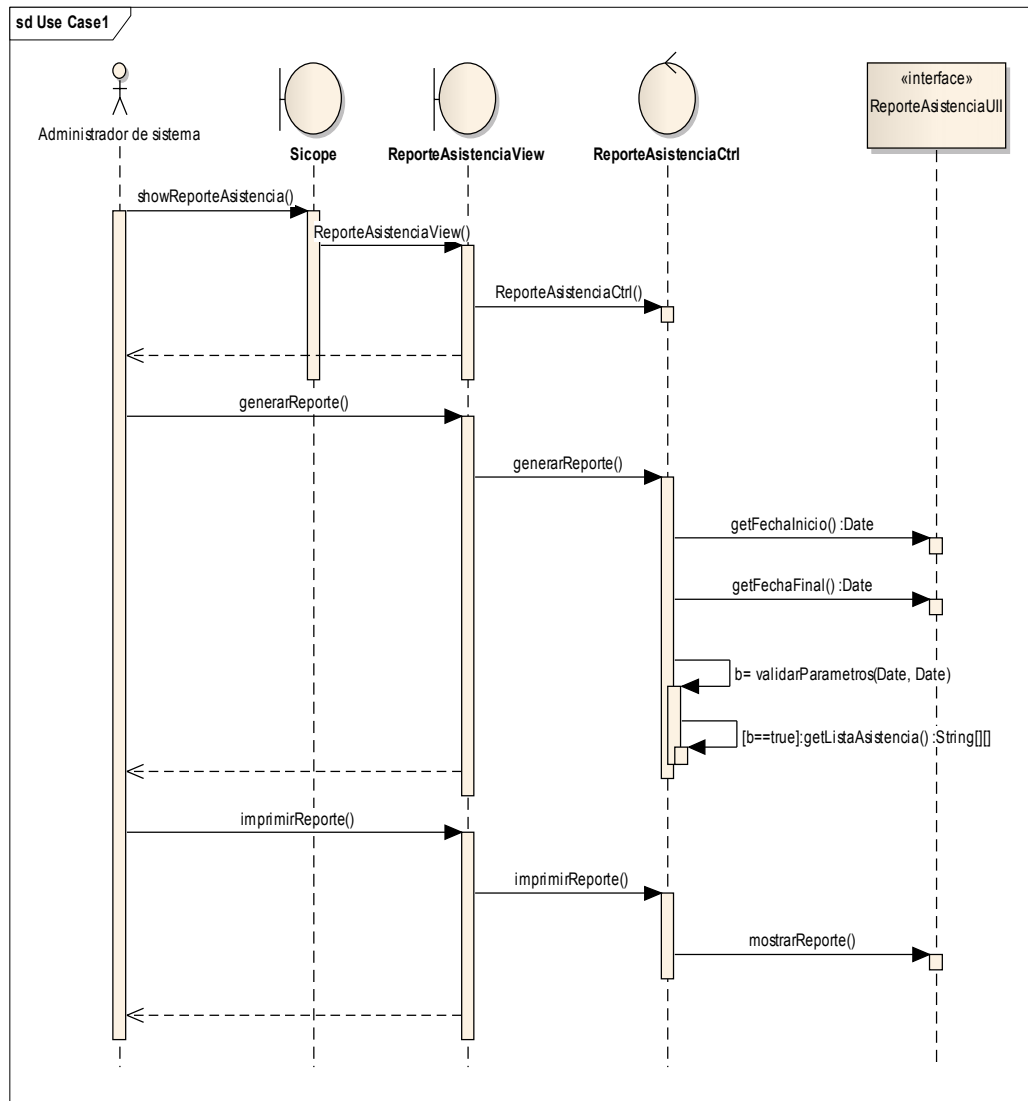


Figura 51. Diagrama de secuencia: Generar Reporte de Asistencia

Caso de Uso: Generar Reporte de Asistencia por Persona

Autor: Administrador del Sistema

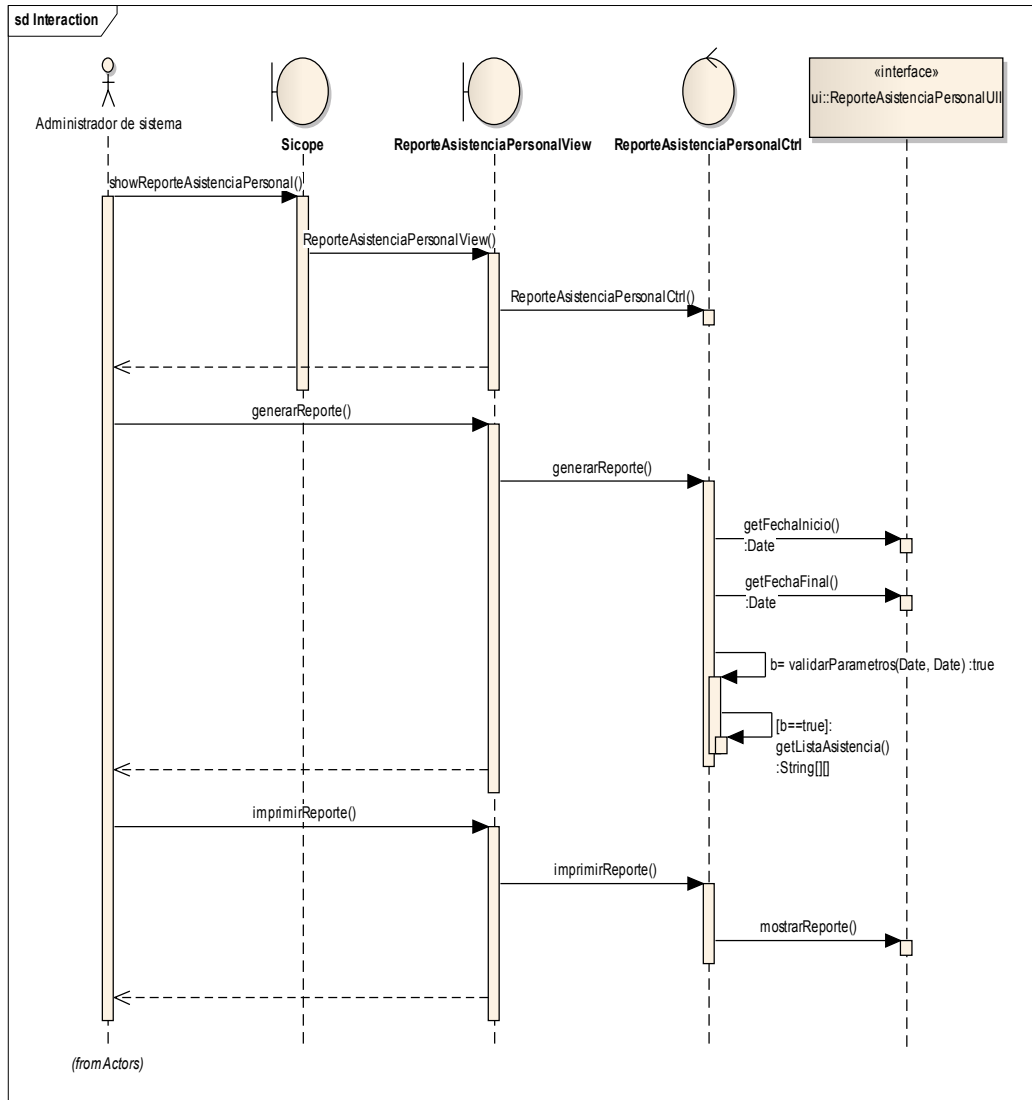


Figura 52. Diagrama de secuencia: Generar Reporte de Asistencia por Persona

3.8 VISTA ESTÁTICA

3.7.1 DIAGRAMA DE PAQUETES

Arquitectura Modelo-Vista-Presentador

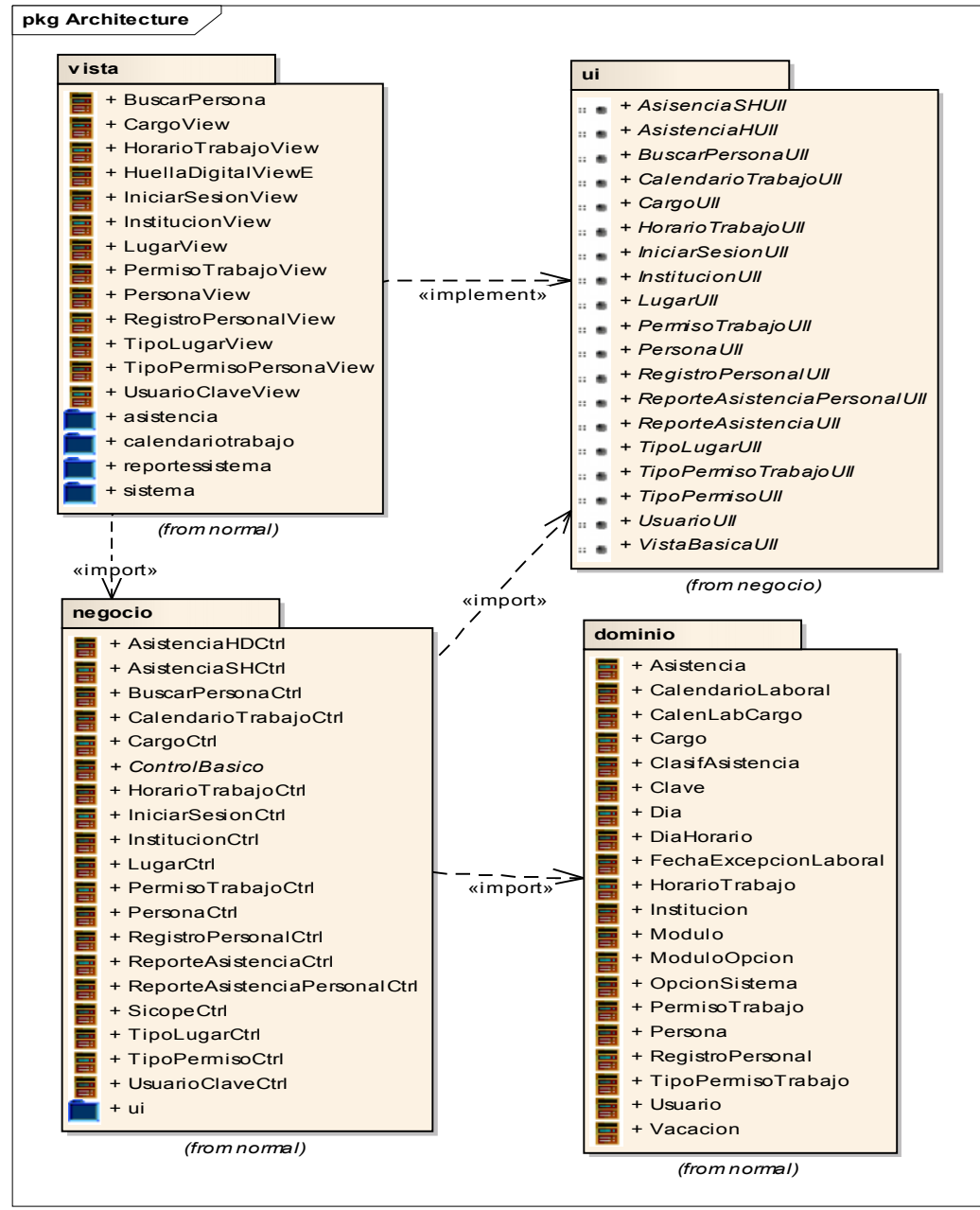
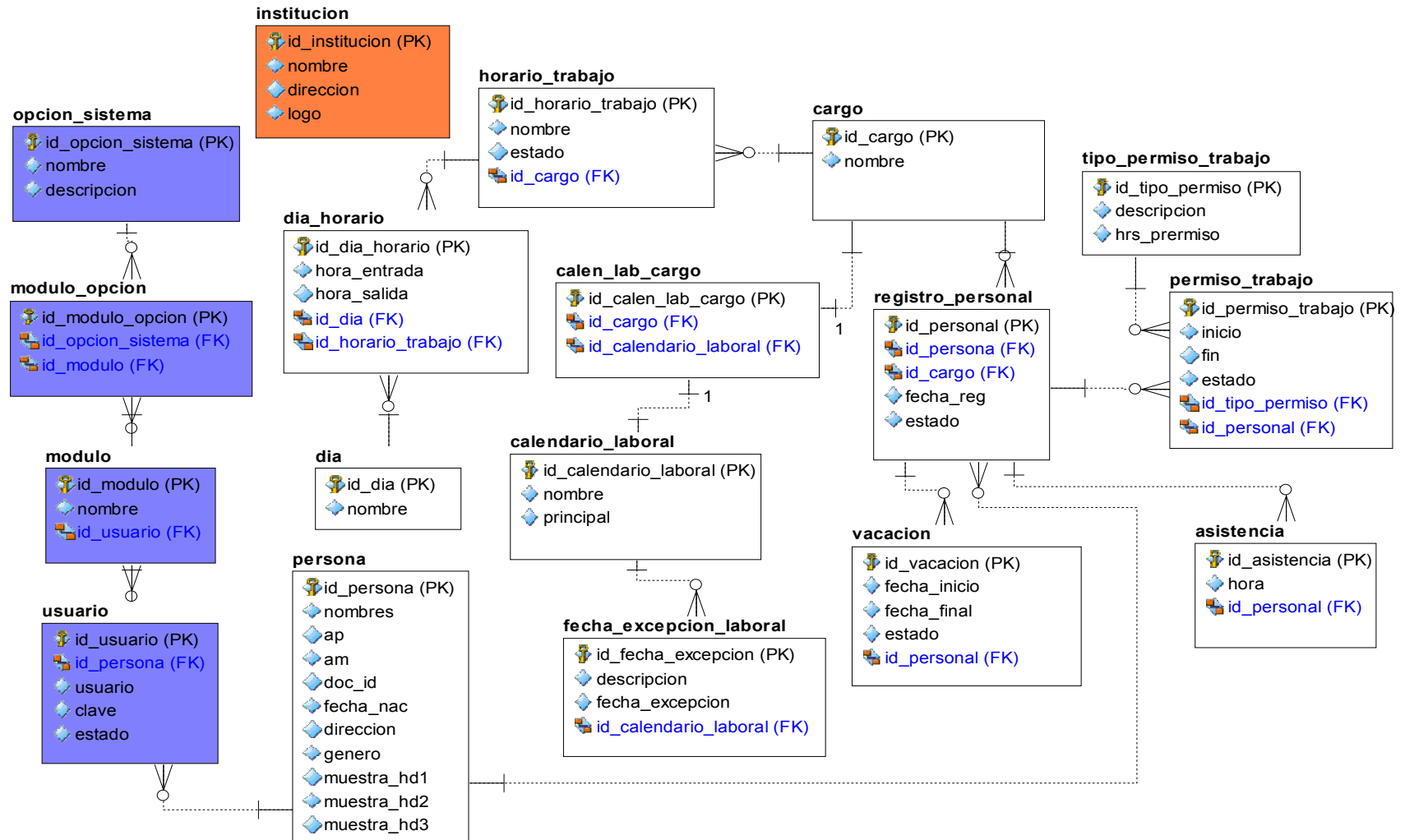


Figura 53. Diagrama de paquetes

4. DISEÑO DE LA BASE DE DATOS

4.1 MODELO ENTIDAD RELACIÓN



4.2 DESCRIPCIÓN DE LAS TABLAS DE BASE DE DATOS DEL SISTEMA

Tabla No. 1: asistencia

Atributos	Tipo de dato	Nulos	Clave primaria
id_asistencia	Integer	No	Si
hora	Date time	No	No
id_personal	Integer	No	No

Tabla No. 2: calen_lab_cargo

Atributos	Tipo de dato	Nulos	Clave primaria
id_calen_lab_cargo	Integer	No	Si
id_cargo	Integer	No	No
id_calendario_laboral	Integer	No	No

Tabla No. 3: calendario_laboral

Atributos	Tipo de dato	Nulos	Clave primaria
id_calendario_laboral	Integer	No	Si
nombre	Varchar(50)	No	No
principal	Bool	No	No

Tabla No. 4: cargo

Atributos	Tipo de dato	Nulos	Clave primaria
id_cargo	Integer	No	Si
nombre	Varchar(50)	No	No
principal	Bool	No	No

Tabla No. 5: día

Atributos	Tipo de dato	Nulos	Clave primaria
Id_dia	Integer	No	Si
Nombre	Varchar(50)	No	Si

Tabla No. 6: dia_horario

Atributos	Tipo de dato	Nulos	Clave primaria
id_dia_horario	Integer	No	Si
hora_entrada	Timestamp	No	No
hora_salida	Timestamp	No	No
id_dia	Integer	No	No
id_horario_trabajo	Integer	No	No

Tabla No. 7: fecha_excepcion_laboral

Atributos	Tipo de dato	Nulos	Clave primaria
id_fecha_excepcion	Integer	No	Si
descripcion	Varchar(100)	No	No
fecha_excepcion	integer	No	No
id_calendario_laboral	Integer	No	No
id_horario_trabajo	Integer	No	No

Tabla No. 8: horario_trabajo

Atributos	Tipo de dato	Nulos	Clave primaria
id_horario_trabajo	Integer	No	Si
nombre	Varchar(50)	No	No
estado	Integer	No	No
id_cargo	Integer	No	No

Tabla No. 9: modulo

Atributos	Tipo de dato	Nulos	Clave primaria
id_modulo	Integer	No	Si
nombre	Varchar(50)	No	No
id_usuario	Integer	No	No

Tabla No. 10: modulo_opcion

Atributos	Tipo de dato	Nulos	Clave primaria
id_modulo_opcion	Integer	No	Si
id_opcion_sistema	Integer	No	No
id_modulo	Integer	No	No

Tabla No. 11: opcion_sistema

Atributos	Tipo de dato	Nulos	Clave primaria
id_opcion_sistema	Integer	No	Si
nombre	Varchar(50)	No	No
descripcion	Varchar(100)	No	No

Tabla No. 12: permiso_trabajo

Atributos	Tipo de dato	Nulos	Clave primaria
id_permiso_trabajo	Integer	No	Si
inicio	Time	No	No
Fin	Time	No	No
estado	Bool	No	No
id_tipo_permiso	Integer	No	No
id_personal	Integer	No	No

Tabla No. 13: persona

Atributos	Tipo de dato	Nulos	Clave primaria
id_persona	Integer	No	Si
nombres	Varchar(50)	No	No
ap	Varchar(50)	No	No
am	Varchar(50)	No	No
doc_id	Varchar(50)	No	No
fecha_nac	Varchar(50)	No	No
direccion	Varchar(100)	No	No

Genero	Bool	No	No
muestra_hd1	Byte[]	No	No
muestra_hd2	Byte[]	No	No
muestra_hd3	Byte[]	No	No

Tabla No. 14: registro_personal

Atributos	Tipo de dato	Nulos	Clave primaria
id_personal	Integer	No	Si
id_persona	Integer	No	No
id_cargo	Integer	No	No
fecha_reg	Date	No	No
estado	Bool	No	No

Tabla No. 15: tipo_permiso_trabajo

Atributos	Tipo de dato	Nulos	Clave primaria
id_tipo_permiso	Integer	No	Si
descripcion	Varchar(100)	No	No
hrs_permiso	Time	No	No

Tabla No. 16: usuario

Atributos	Tipo de dato	Nulos	Clave primaria
id_usuario	Integer	No	Si
id_persona	Integer	No	No
usuario	Varchar(20)	No	No
clave	Varchar(100)	No	No
estado	Bool	No	No

Tabla No. 17: vacación

Atributos	Tipo de dato	Nulos	Clave primaria
id_personal_vacacion	Integer	No	Si
fecha_inicio	Datetime	No	No
fecha_final	Datetime	No	No
estado	Bool	No	No
id_personal	Integer	No	No

4.3 DISEÑO DE PANTALLAS

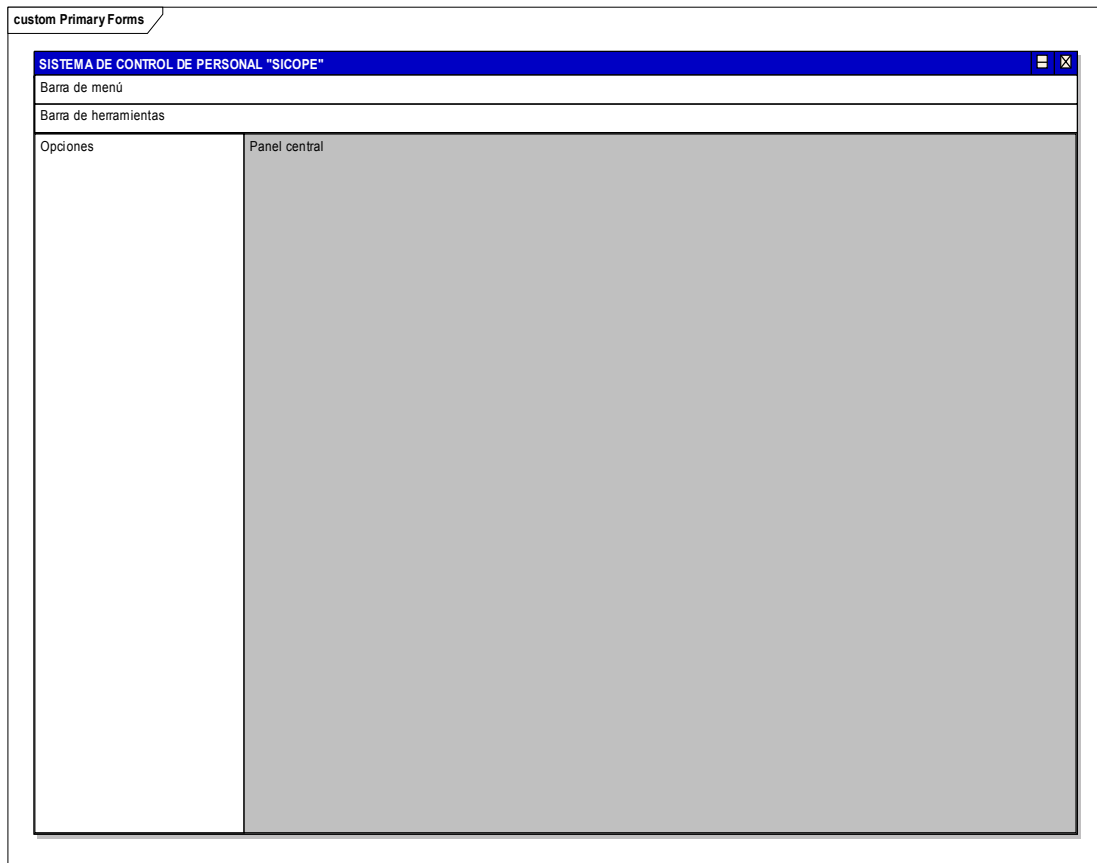


Figura 54. Pantalla Principal para el administrador del Sistema

Mediante esta pantalla el Administrador del Sistema puede navegar por el resto de las opciones disponibles, de acuerdo a la necesidad del usuario.

- En esta pantalla se visualiza la opción cerrar sesión, esta opción se agrega en la barra de menú. Se deshabilita la opción cuando existe una sesión abierta.

I. MÓDULO DEL SISTEMA



The image shows a screenshot of a web application's login screen. The window title is 'custom Iniciar sesión'. The main content area has a title 'Iniciar sesión'. Below the title, there are two input fields: 'Usuario:' and 'Clave:'. Below the input fields, there are two buttons: 'Aceptar' and 'Limpiar'.

Figura 55. Inicio de Sesión

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem de menú en la barra de menú.
- Para cargar esta pantalla se adiciona un ítem a la barra de herramienta.
- El botón aceptar se habilita, cuando los campos usuario y clave se encuentran con datos.
- El botón limpiar, se habilita cuando los campos usuario o clave tiene datos.

- Nuevo: permite crear un nuevo registro de asistencia.
- Borrar: Borra un registro de asistencia. Se habilita si hay una fila seleccionada.
- Buscar: muestra un dialogo para la búsqueda de un registro de persona. Una vez cerrado el diálogo se tiene que cargar los nombre y el código del personal.
- Cancelar: cancela las operaciones que se están realizando sobre el sistema.

Caso de uso relacionado.

Registrar asistencia sin huella digital

Tablas relacionadas:

Registro_personal, asistencia, persona

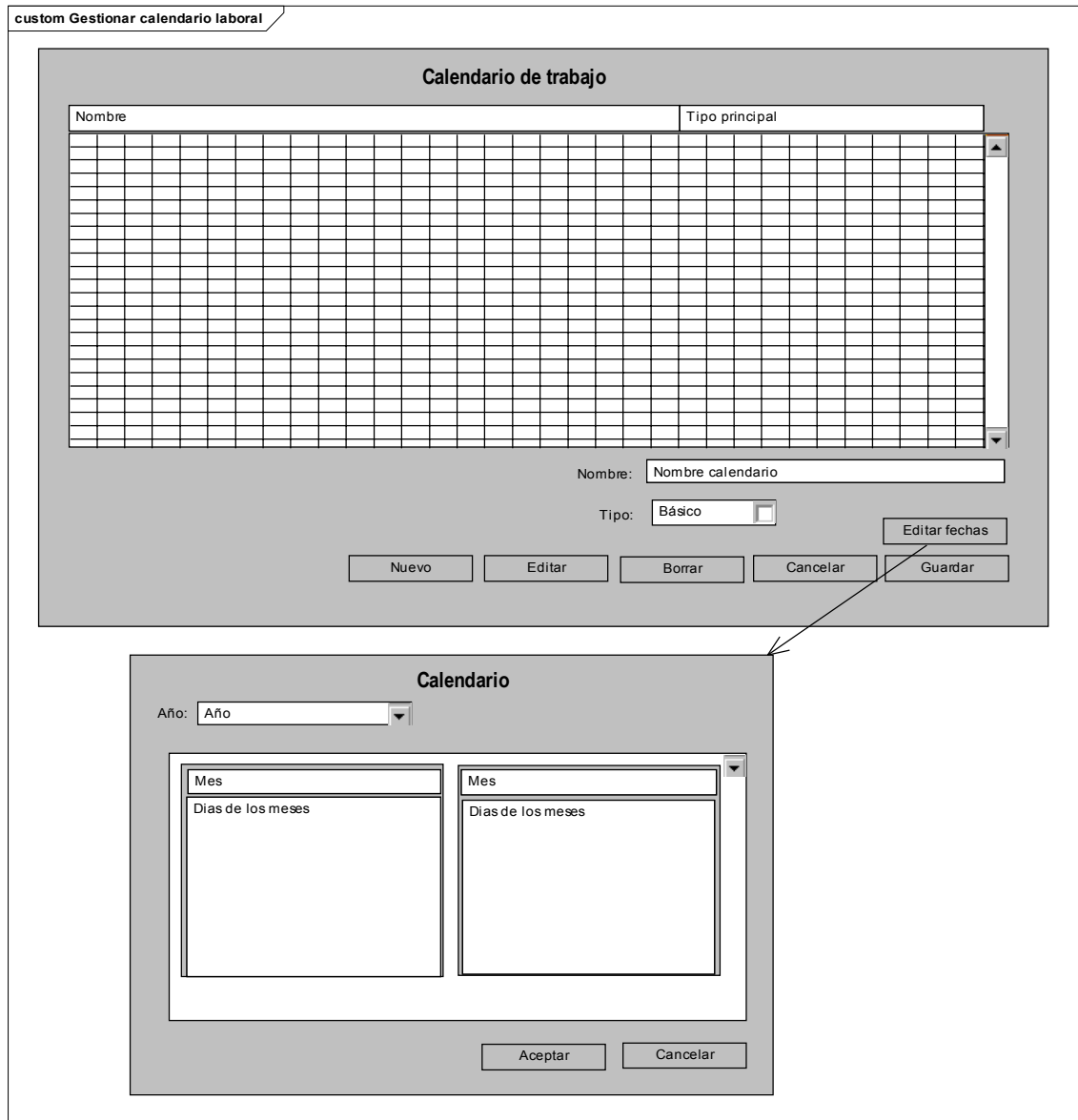


Figura 58. Gestionar Calendario Laboral

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem de menú en la barra de menú.
- Cuando se selecciona un registro, los campos se actualizan, con los datos seleccionados.
- Editar fechas: es el que permite ver un dialogo con los meses del año para mostrar las fechas habilitadas o deshabilitadas en un calendario laboral.
- Nuevo: Crea un nuevo calendario laboral.
- Editar: da la posibilidad de modificar los datos de un calendario laboral.

- Borrar: cambia el estado de un calendario laboral a inactivo.
- Cancelar: cancela la edición de cualquier dato.
- Guardar: guarda los cambios realizados sobre un objeto.

Tablas relacionadas

Cargo, calen_lab_cargo, calendario_laboral, fecha_excepcion_laboral

The image shows a software interface for managing work calendars. The main window, titled "Calendario de trabajo", features a large grid for editing dates. Below the grid, there are input fields for "Nombre" (set to "Nombre calendario") and "Tipo" (set to "Básico"). Buttons for "Nuevo", "Editar", "Borrar", "Cancelar", "Guardar", and "Editar fechas" are located at the bottom. An arrow points from the "Editar fechas" button to a secondary window titled "Calendario", which displays a calendar view for a specific year and month, with options to "Crear" or "Borrar" exceptions.

Figura 59. Gestionar Fecha de Excepción Laboral

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Año: la lista muestra el año del cual se está editando el calendario laboral.
- Mes: son los distintos meses de un año con sus fechas.
- Dentro de cada mes se encuentran cada uno de los días que le corresponden, sobre cada uno de ellos haciendo clic derecho se despliegan las opciones Crear y Borrar.
- Aceptar: acepta y guarda los cambios realizado sobre todas las fechas de excepción laboral.
- Cancelar: Cancela todos los cambios realizados.

Caso de uso relacionado.

Gestionar calendario laboral

Tablas relacionadas:

Calen_lab_cargo, calendario_laboral, fecha_excepción_laboral

- Hora final: se escribe la hora a la cual se va a terminar el permiso de trabajo.
- Observación: cualquier otro dato adicional.
- Nuevo: crea un nuevo permiso de trabajo.
- Modificar: permite la modificación del permiso de trabajo seleccionado en la lista. Sólo se habilita cuando se encuentra un permiso de trabajo seleccionado.
- Borrar: borra el estado de permiso de trabajo. Sólo se habilita cuando se encuentra un permiso de trabajo seleccionado.
- Cancelar: cancela cualquier modificación que se haya realizado sobre un registro de permiso de trabajo. Sólo se habilita cuando se ha modificado o creado un permiso de trabajo.
- Guardar: guarda los cambios realizado en los permisos de trabajo.

Tablas relacionadas:

Tipo_permiso_trabajo, registro_personal, permiso_trabajo

- Guardar: guarda los cambios realizados durante la edición o la creación de un objeto de tipo de permiso de trabajo. Sólo se habilita cuando se han modificado datos de un tipo de permiso de trabajo.

Tablas relacionadas

Tipo_permiso_trabajo

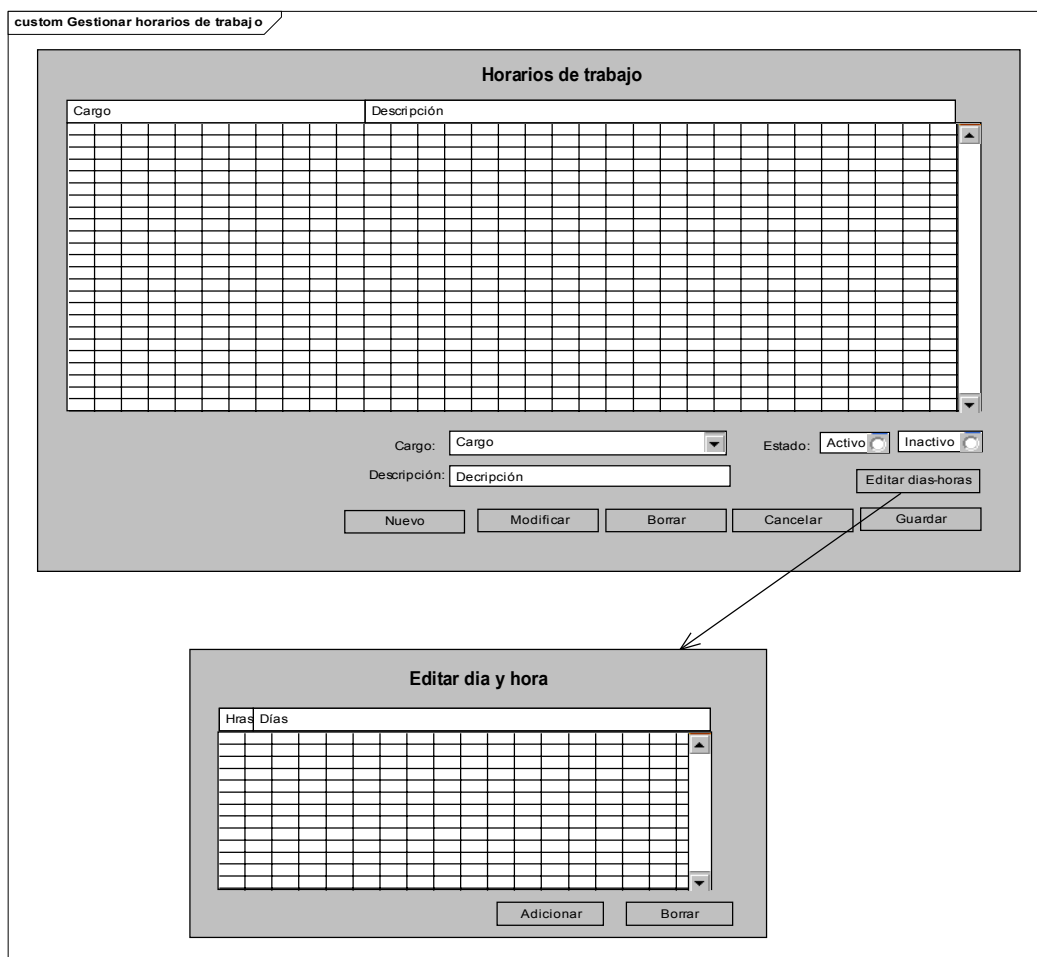


Figura 62. Gestionar Horario de Trabajo

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Cargo: con el que se relaciona el horario de trabajo.
- Nombre: que describe el horario de trabajo.

Figura 63. Gestionar Registro Personal

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Buscar: pulsando el botón muestra una pantalla de búsqueda para personas.
- Registrar huella: permite editar las muestras de huella digital.
- Cargo: Lista de cargos disponibles para designar a una persona.
- Nuevo: crea un nuevo registro de personal. Sólo se habilita cuando no hay operaciones pendientes en curso.
- Modificar: permite modificar un registro de personal. Sólo se habilita cuando no hay operaciones pendientes en curso y si existe un objeto de la lista seleccionado.
- Cancelar: revierte cualquier cambio que se realice sobre un objeto registro de personal, antes de guardar los cambios. Sólo se habilita cuando hay operaciones pendientes en curso.
- Borrar: borra un registro de personal. Sólo se habilita cuando no hay operaciones en curso y si existe un objeto de la lista seleccionado.
- Guardar: guarda los cambios realizados durante la modificación o la creación de un objeto de registro de personal. Sólo se habilita cuando se han modificado datos de un registro de personal.

Tablas relacionadas:

Cargo, registro_personal, persona

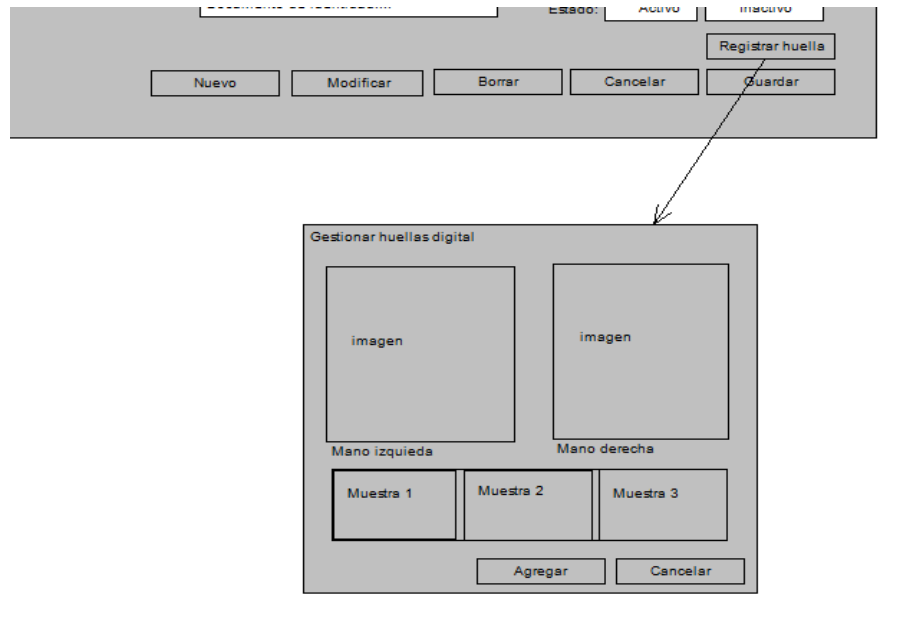


Figura 64. Gestionar Huella Digital

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Mano izquierda (Imagen): Muestra una imagen de mano con dedos seleccionables.
- Mano derecha (Imagen): Muestra una imagen de mano con dedos seleccionables.
- Muestra 1, 2, 3: ahí se ven las imágenes de las muestras tomadas.
- Agregar: permite agregar muestras de huella digital. Se habilita sólo en el caso de que haya un dedo seleccionado en cualquier imagen de mano.
- Cancelar: Cierra la pantalla de edición para las huellas digitales.

Tablas relacionadas:

Registro_personal, persona,

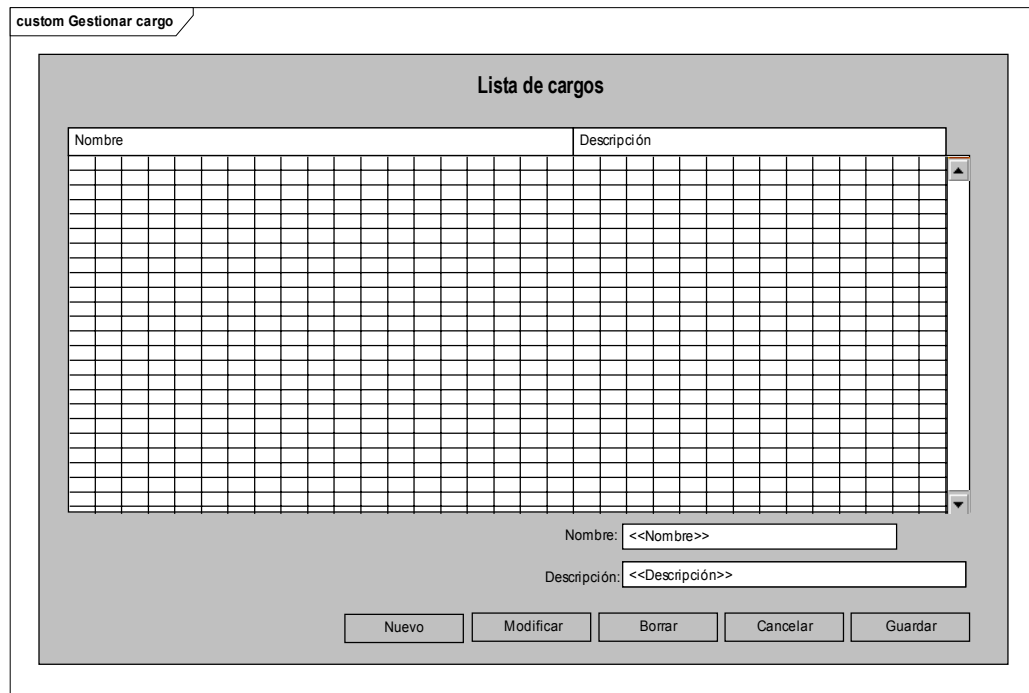


Figura 65. Gestionar Cargo

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Nombre: nombre para el cargo.
- Descripción: pequeño texto que permite conocer más sobre el cargo.
- Nuevo: crea un nuevo registro de cargo. Sólo se habilita cuando no hay operaciones pendientes en curso.
- Modificar: permite modificar un registro de cargo. Sólo se habilita cuando no hay operaciones pendientes en curso y si existe un objeto de la lista seleccionado.
- Cancelar: revierte cualquier cambio que se realice sobre un objeto registro de cargo, antes de guardar los cambios. Sólo se habilita cuando hay operaciones pendientes en curso.
- Borrar: borra un registro de cargo. Sólo se habilita cuando no hay operaciones en curso y si existe un objeto de la lista seleccionado.

- Guardar: guarda los cambios realizados durante la modificación o la creación de un objeto de registro de cargo. Sólo se habilita cuando se han modificado datos de un registro de cargo.

Tablas relacionadas:

Cargo

III. MÓDULO GENERAL

The screenshot shows a web application window titled "custom Gestionar persona". Inside, there is a section titled "Lista de personas". At the top right of this section is a search bar labeled "Buscar:" with the placeholder text "<<Ingrese parametro de busqueda>". Below the search bar is a table with the following columns: "Nombres", "Ap. paterno", "Ap. Materno", "C.I.", "Fecha de nac.", and "Dirección". The table contains several empty rows. Below the table are several input fields: "Nombres:" with a text box containing "Nombres....", "Ap. Paterno:" with a text box containing "Apellido paterno...", "Ap. Materno:" with a text box containing "Apellido materno", "Doc Id.:" with a text box containing "Documento de identidad...", "Fecha Nac.:" with a dropdown menu showing "Fecha de nacimiento", and "Dir.:" with a text box containing "Dirección...". At the bottom of the form are five buttons: "Nuevo", "Editar", "Borrar", "Cancelar", and "Guardar".

Figura 66. Gestionar Persona

Requerimientos en el diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Nombres: nombres de la persona.
- Ap. Paterno: es el apellido paterno de la persona.
- Ap. Materno: es el apellido materno de la persona.
- Doc id.: documento de identidad de la persona.

- Fecha nac.: se ingresa la fecha de nacimiento de la persona.
- Dirección: el lugar del domicilio de la persona.
- Nuevo: crea un nuevo objeto de persona. Sólo se habilita cuando no hay operaciones pendientes en curso.
- Modificar: permite modificar un objeto de persona. Sólo se habilita cuando no hay operaciones pendientes en curso y si existe un objeto de la lista seleccionado.
- Cancelar: revierte cualquier cambio que se realice sobre un objeto de persona, antes de guardar los cambios. Sólo se habilita cuando hay operaciones pendientes en curso.
- Borrar: borra un objeto de persona. Sólo se habilita cuando no hay operaciones en curso y si existe un objeto de la lista seleccionado.
- Guardar: guarda los cambios realizados durante la modificación o la creación de un objeto de persona. Sólo se habilita cuando se han modificado datos de una persona.

Tabla relacionada:

Persona

The image shows a web form interface for managing a user. The main title is "custom Gestionar persona". Inside, there is a section titled "Modulos, usuario y clave". This section contains several input fields: "Usuario:" with a text box, "Clave:" with a text box, and "Estado:" with a dropdown menu currently showing "Activo". To the right, there is a "Módulos" dropdown menu with a downward arrow, containing two selected items: "Modulo general" and "Modulo de sistema". At the bottom of the form are two buttons: "Aceptar" and "Cancelar".

Figura 67. Registrar Usuario y Clave

Requerimiento de diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- **Usuario:** el nombre de usuario ante el sistema.
- **Clave:** caracteres que se van a usar por parte de la persona para autenticarse ante el sistema.
- **Estado:** indica si el usuario y clave se encuentra activo para autenticarse ante el sistema.
- **Módulos:** la lista de módulos disponibles para asignar al usuario de sistema.
- **Aceptar:** permite validar y guardar todos los cambios.
- **Cancelar:** borra todos los cambios realizados en el sistema.

Tablas relacionadas:

Modulo, usuario, persona

custom Gestionar registro de institucion

Datos de la institución

Nombre:

Dirección:

Logo:

Vista previa...

Figura 68. Gestionar Registro de Institución

Requerimiento del diseño de pantalla.

- Para cargar esta pantalla se adiciona un ítem en la barra de menú.
- Nombre: es el nombre de la institución.
- Dirección: el domicilio legal de la institución.
- Logo: permite buscar el logo de la institución, que se va a introducir en los documentos generados por el sistema.
- Vista previa: permite ver el logo de la institución en vista previa.
- Guardar: guarda las modificaciones realizadas sobre el objeto de institución. Se habilita sólo cuando se ha modificado el objeto de institución.
- Cancelar: cancela cualquier cambio realizado en el objeto de institución.

Tablas relacionadas:

Institución

- **Imprimir:** imprime el reporte que se encuentra visible en el tab-panel.

Tablas relacionadas:

Horario_trabajo, día_horario, cargo, calen_lab_cargo,
calendario_laboral, registro_personal, vacación, permiso_trabajo,
asistencia.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- ❖ El trabajo cumple con el objetivo general propuesto al inicio del mismo, puesto que se realizó el Análisis y Diseño del “Sistema Automatizado para el Control de Personal de la Escuela de Formación de Maestros “Juan Misael Saracho”, de acuerdo a los alcances propuestos.
- ❖ Los objetivos se lograron cumplir mediante observación directa, también siguiendo las fases de desarrollo de la metodología RUP.
- ❖ Para el modelado del sistema se aplicó UML, porque tiene el propósito de visualizar, especificar y documentar sistemas orientados a objetos.
- ❖ Para la realización del trabajo fue necesario comprender a fondo el manejo de los diferentes registros y reportes de la institución.
- ❖ El sistema cubre con todos los registros de datos, controles y reportes, los mismos que dan solución a los problemas identificados, aplicando la tecnología adecuada dándole seguridad a la información emitida.
- ❖ El sistema ha sido diseñado para que se pueda programar e implementar dentro de la Escuela Superior de Formación de Maestros “Juan Misael Saracho” de Canasmoro, ya que cuenta con todas las herramientas que permiten ese trabajo. Además se ha incluido el registro de asistencia mediante huella digital, esto para que el sistema se adecue a las innovaciones tecnológicas y así se pueda tener un sistema confiable y seguro, pues no existe duplicidad en el registro de huellas dactilares.

5.2 **RECOMENDACIONES**

- ❖ Las ventajas que conlleva el sistema automatizado para el control de personal son un control confiable y seguro de la asistencia, donde se podrán consultar los reportes de asistencia y a la vez servirá como estación de registro de las entradas y salidas de los funcionarios. Elimina el uso de tarjetas de papel y relojes de marcado de tiempos. Elimina el cálculo de tiempos hechos a mano, lo cual genera errores.
- ❖ El sistema automatizado para el control de personal mediante huella digital evita el "Fraude de Asistencia" verificando la identidad de los empleados por medio de la huella digital.
- ❖ Es necesario para un buen uso del sistema y su explotación, que los usuarios tengan un conocimiento básico de computación. Los usuarios serían el Administrador del Sistema y los funcionarios como usuarios vienen a ser los que registran su asistencia a la institución.
- ❖ Se recomienda que exista más de un equipo de computación dedicado al sistema de Asistencia, uno para el Registro de Asistencia en el cual el lector de huellas se instala en el puerto USB. Otro para los datos y el Sistema y por último un equipo Terminal Simple que puede ejecutarse desde el servidor sin ningún inconveniente.
- ❖ El sistema puede ser instalado en modo Local o en Red.