



## **MANUAL TECNICO DE STAFFLOG SOFTWARE**

**Aplicación para el Manejo de Información de Empleados**

María Gabriela Rivera Valarezo  
Andrea Gabriela Vega Ortiz

Quito – Ecuador

**TABLA DE CONTENIDO**

TABLA DE CONTENIDO .....	1
TABLA DE FIGURAS.....	3
1. PRINCIPALES REQUERIMIENTOS.....	6
Requerimientos Funcionales .....	6
Requerimientos No Funcionales .....	9
2. DISEÑO.....	10
Casos de Uso .....	10
Diagrama General .....	10
F0: Administración de Usuarios .....	11
F1: Administración de Empleados .....	13
F2: Manejo de Contratos .....	17
Diagrama de Actividades de Manejo de Contratos .....	21
F3: Administración de Horas Extras .....	22
F4: Gestión de Acciones de Personal .....	24
Diagrama de Actividades de Administración de Acción de Personal .....	26
F5: Administración de Ausencias .....	27
F6: Administración de Vacaciones.....	30
Diagrama de Clases General .....	33
Diagrama de Clases con Atributos .....	34
Diagrama de Secuencia .....	38
Caso de uso: Ingresar Usuario .....	38
Caso de uso: Modificar Usuario .....	39
Caso de uso: Eliminar Usuario.....	40
Caso de uso: Ingresar Empleado .....	41

Caso de uso: Modificar Empleado .....	42
Caso de uso: Buscar Empleado .....	43
Caso de uso: Eliminar Empleado .....	44
Caso de uso: Crear Contrato.....	45
Caso de uso: Modificar Contrato .....	46
Caso de uso: Buscar Contrato .....	47
Caso de uso: Anular Contrato .....	48
Caso de uso: Ingresar Horas Extra .....	49
Caso de uso: Calcular Horas Extra .....	50
Caso de uso: Ingresar Acción de Personal.....	51
Caso de uso: Buscar Acción de Personal .....	52
Caso de uso: Ingresar Ausencias .....	53
Caso de uso: Modificar Ausencias .....	54
Caso de uso: Consultar Ausencias .....	55
Caso de uso: Ingresar Vacaciones .....	56
Caso de uso: Modificar Vacaciones.....	57
Caso de uso: Eliminar Vacaciones.....	58
Caso de uso: Consultar Vacaciones .....	59
Implementación .....	59
STAFFLOG Software .....	72
3.    REQUERIMIENTOS E INSTALACION .....	73
Requerimientos .....	73
Instalación .....	73
4.    INGRESO COMO ADMINISTRADOR .....	73
Ingreso como Administrador .....	<b>iError! Marcador no definido.</b>

## TABLA DE FIGURAS

Figura No: 1 Diagrama General.....	10
Figura No: 2 Administración de Usuarios .....	11
Figura No: 3 Administración de Empleados .....	13
Figura No: 4 Manejo de Contratos .....	17
Figura No: 5 Diagrama de Actividades de Manejo de Contratos.....	21
Figura No: 6 Administración de Horas Extras .....	22
Figura No: 7 Gestión de Acciones de Personal .....	24
Figura No: 8 Diagrama de Actividades de Administración de Acción de Personal .....	26
Figura No: 9 Administración de Ausencias .....	27
Figura No: 10 Administración de Vacaciones .....	30
Figura No: 11 Diagrama de Clases General .....	33
Figura No: 12 Diagrama de Clases Categoría Usuario .....	34
Figura No: 13 Diagrama de Clases Usuario .....	34
Figura No: 14 Diagrama de Clases Horas Extra .....	34
Figura No: 15 Diagrama de Clases Contrato.....	35
Figura No: 16 Diagrama de Clases Acción de Personal .....	35
Figura No: 17 Diagrama de Clases Ausencias .....	36
Figura No: 18 Diagrama de Clases Vacaciones .....	36
Figura No: 19 Diagrama de Clases Empleado .....	37
Figura No: 20 Diagrama de Secuencia Ingresar Usuario.....	38
Figura No: 21 Diagrama de Secuencia Modificar Usuario.....	39
Figura No: 22 Diagrama de Secuencia Eliminar Usuario .....	40
Figura No: 23 Diagrama de Secuencia Ingresar Empleado .....	41
Figura No: 24 Diagrama de Secuencia Modificar Empleado.....	42
Figura No: 25 Diagrama de Secuencia Buscar Empleado.....	43
Figura No: 26 Diagrama de Secuencia Eliminar Empleado .....	44
Figura No: 27 Diagrama de Secuencia Crear Contrato .....	45
Figura No: 28 Diagrama de Secuencia Modificar Contrato .....	46
Figura No: 29 Diagrama de Secuencia Buscar Contrato .....	47
Figura No: 30 Diagrama de Secuencia Anular Contrato.....	48
Figura No: 31 Diagrama de Secuencia Ingresar Horas Extra .....	49
Figura No: 32 Diagrama de Secuencia Calcular Horas Extra .....	50
Figura No: 33 Diagrama de Secuencia Ingresar Acción de Personal ..	51
Figura No: 34 Diagrama de Secuencia Buscar Acción de Personal ..	52

Figura No: 35 Diagrama de Secuencia Ingresar Ausencias .....	53
Figura No: 36 Diagrama de Secuencia Modificar Ausencias.....	54
Figura No: 37 Diagrama de Secuencia Consultar Ausencias.....	55
Figura No: 38 Diagrama de Secuencia Ingresar Vacaciones.....	56
Figura No: 39 Diagrama de Secuencia Modificar Vacaciones .....	57
Figura No: 40 Diagrama de Secuencia Eliminar Vacaciones .....	58
Figura No: 41 Diagrama de Secuencia Consultar Vacaciones .....	59
Figura No: 42 Página de Ingreso como Administrador .....	74



## INTRODUCCION

El objetivo de este Manual es presentar los detalles del diseño de STAFFLOG Software.

STAFFLOG Software esta creado para ayudar en el manejo de información de empleados de una empresa y automatizar los principales procesos del departamento de Recursos Humanos.

STAFFLOG Software esta desarrollado en PHP, tiene conexión a una base de datos en MySQL y trabaja con un servidor interno XAMPP, además para la implementación se hizo uso de los Patrones de Diseño Estrategia y Fachada.

En la primera parte de este manual se efectúa una descripción de los Principales Requerimientos del diseño de STAFFLOG Software.

En la segunda parte se presenta una descripción del diseño de STAFFLOG Software basada en los Diagramas UML.

En la Tercera Parte se presentan los requerimientos de Software para el funcionamiento de STAFFLOG Software y la instalación del mismo.

Finalmente, se muestra el Inicio de sesión como.

# 1. PRINCIPALES REQUERIMIENTOS

## Requerimientos Funcionales

- **F0: Administración de Usuarios**

El sistema tendrá uno o más usuarios el mismo que manejará el proceso de Acción de Personal.

- ✓ **F0.1: Ingresar Usuario**

El sistema será capaz de crear uno o más usuarios en la base de datos.

- ✓ **F0.2: Modificar Usuario**

El sistema será capaz de modificar uno o más usuarios en la base de datos.

- ✓ **F0.3: Eliminar Usuario**

El sistema será capaz de eliminar uno o más usuarios en la base de datos.

- **F1: Administración de Empleados**

El sistema tendrá una base de empleados que es personal que labora en el Hospital Gineco-Obstétrico "Isidro Ayora".

- ✓ **F1.1: Ingresar Empleado**

El sistema será capaz de registrar nuevos empleados en la base de datos.

- ✓ **F1.2: Modificar Empleado**

El sistema será capaz de modificar los datos de los empleados de la base de datos.

- ✓ **F1.3: Buscar Empleado**

El sistema será capaz de buscar y mostrar un empleado en la base de datos.

✓ **F1.4: Eliminar Empleado**

El sistema será capaz de eliminar uno o más empleados de la base de datos.

• **F2: Manejo de Contratos**

El sistema será capaz de manejar los contratos del personal que labora en el Hospital.

✓ **F2.1: Crear Contrato**

El sistema será capaz de crear un nuevo contrato para un nuevo empleado.

✓ **F2.2: Modificar Contrato**

El sistema será capaz de modificar los datos de un contrato.

✓ **F2.3: Buscar Contrato**

El sistema será capaz de buscar y mostrar los datos de un contrato.

✓ **F2.4: Anular Contrato**

El sistema será capaz de anular el contrato de un empleado eliminándolo al mismo de la base.

• **F3: Administración de Horas Extras**

El sistema será capaz de almacenar las horas extras de cada empleado dentro de la base de datos.

✓ **F3.1: Ingresar Horas Extras**

El sistema será capaz de registrar las horas extras diarias de cada empleado.

✓ **F3.2: Calcular Horas Extras**

El sistema será capaz de calcular las horas extras diarias de cada empleado.

• **F4: Gestión de Acciones Personal**

El sistema podrá generar una Acción de Personal para un empleado y enviarla a Microsoft Office Word para su correcta edición e impresión.

✓ **F4.1: Ingresar Acción Personal**

El sistema será capaz de generar una Acción de Personal para un empleado y almacenar la dirección de donde está guardada en la base de datos.

✓ **F4.2: Buscar Acción Personal**

El sistema será capaz de buscar una Acción de Personal por nombre de Empleado o por número de Acción de Personal.

• **F5: Administración de Ausencias**

El sistema será capaz de administrar y contabilizar las ausencias y/o faltas de cada empleado.

✓ **F.5.1: Ingresar ausencias**

El sistema será capaz de ingresar una nueva ausencia en el registro de cada empleado y archivarlo en la base de datos.

✓ **F.5.2 Modificar ausencias**

El sistema será capaz de modificar las fechas y las notas de las ausencias.

✓ **F.5.3: Consultar ausencias**

El sistema será capaz de mostrar las ausencias de los empleados.

• **F6: Administración de Vacaciones**

El sistema será capaz de administrar las vacaciones solicitadas por los empleados.

✓ **F6.1: Ingresar Vacaciones**

El sistema será capaz de ingresar las fechas de vacaciones de un empleado.

✓ **F6.2: Modificar Vacaciones**

El sistema será capaz de modificar las fechas de vacaciones de un empleado.

✓ **F6.3 Eliminar Vacaciones**

El sistema será capaz de eliminar una solicitud de vacaciones de un empleado.

✓ **F6.3 Eliminar Vacaciones**

El sistema será capaz de mostrar una solicitud de vacaciones de un empleado.

## Requerimientos No Funcionales

- Solo tendrá un Administrador que podrá hacer lo mismo que los otros usuarios con la única diferencia de poder crear nuevos usuarios.
- Tendrá un nombre y un logo creado por las programadoras
- El entorno del sistema será Web

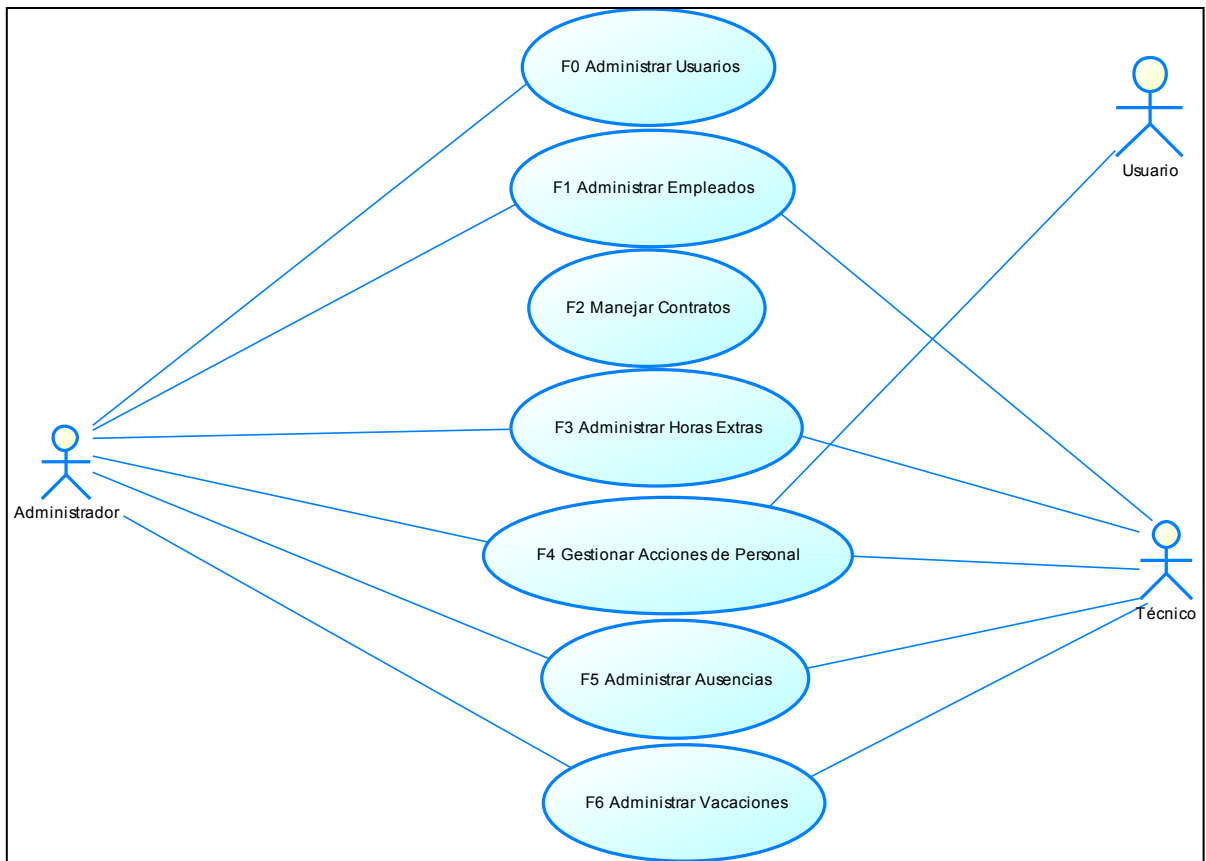
## 2. DISEÑO

### Casos de Uso

Los casos de uso son una representación gráfica del comportamiento de un sistema desde el punto de vista de un usuario, en estos se determinan los requisitos funcionales del sistema, es decir, representan las funcionalidades que el sistema puede ejecutar y así facilitan su interpretación.

### Diagrama General

En el diagrama general se puede visualizar las funcionalidades que tendrá el sistema y de qué manera interactuará el cliente con éstas.



**Figura No: 1 Diagrama General**

## F0: Administración de Usuarios

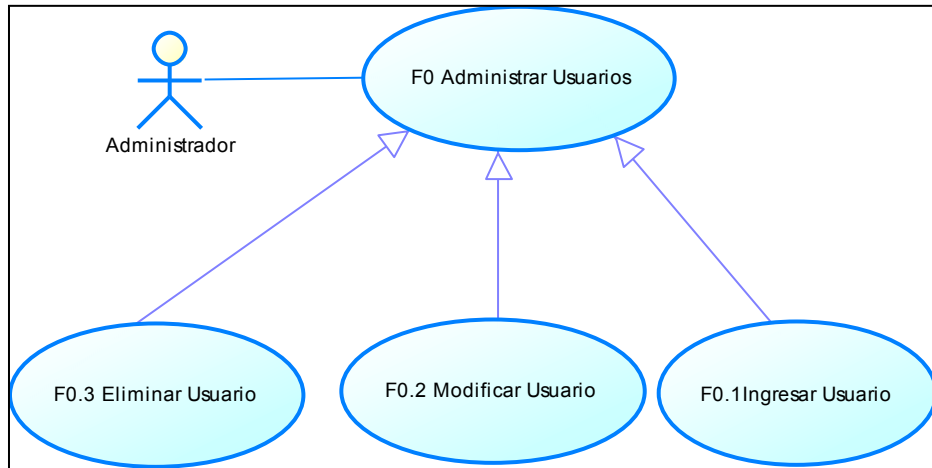


Figura No: 2 Administración de Usuarios

### • F0.1: Ingresar Usuario

**Actor:** Administrador

**Descripción:** El Actor tendrá la capacidad de ingresar un nuevo Usuario en la base de datos.

#### **Flujo Principal:**

1. *El Actor* selecciona Crear o Modificar Usuarios del Menú Usuarios.
2. *El Sistema* presenta una ventana para ingresar los datos.
3. *El Actor* ingresa los datos del Usuario.
4. *El Actor* presiona Grabar.
5. *El Sistema* verifica el código.(E1)
6. *El Sistema* almacena la información.

#### **Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F0.2 o F0.3.

**Excepciones:**

E1. Error en BDD

**• F0.2: Modificar Usuario**

**Actor:** Administrador

**Descripción:** El Actor tendrá la capacidad de modificar datos de un Usuario.

**Flujo Principal:**

1. *El Actor* selecciona Crear o Modificar Usuarios del Menú Usuarios.
2. *El Sistema* presenta una ventana para ingresar los datos.
3. *El Actor* ingresa los datos del Usuario.
4. *El Actor* presiona Grabar.
5. *El Sistema* verifica el código.(E1)
6. *El Sistema* modifica la información.

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F0.1 ó F0.3.

**Excepciones:**

E1. Error en BDD

**• F0.3: Eliminar Usuario**

**Actor:** Administrador

**Descripción:** El Actor tendrá la capacidad de eliminar un Usuario de la base de datos.

**Flujo Principal:**

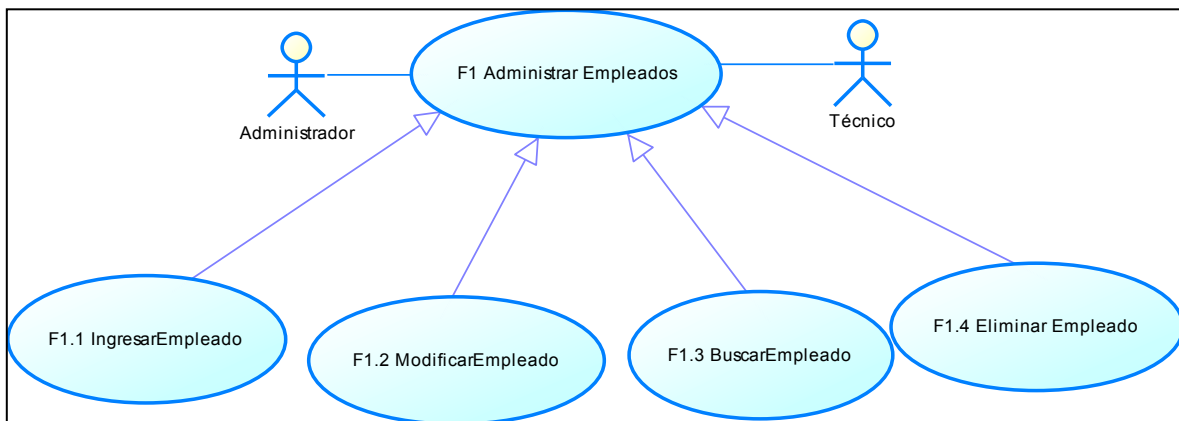
1. *El Actor* selecciona Eliminar Usuarios del Menú Usuarios.
2. *El Sistema* presenta una ventana para ingresar el código.
3. *El Actor* ingresa el código del Usuario.
4. *El Sistema* verifica el código.(E1)
5. *El Sistema* presenta el Usuario con dicho código.
6. *El Actor* presiona Eliminar.
7. *El Sistema* elimina la información.(E1)

**Flujo Alternativo:**

5. El Sistema ejecuta Caso de Uso F0.1 ó F0.2

**Excepciones:**

- E1. Error en BDD

**F1: Administración de Empleados**

**Figura No: 3 Administración de Empleados**

**• F1.1: Ingresar Empleado**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de ingresar los datos de un nuevo Empleado en la base de datos.

**Flujo Principal:**

1. *El Actor* selecciona Empleados del Menú de Creación.
2. *El Sistema* presenta una ventana para ingresar los datos.
3. *El Actor* ingresa los datos del Empleado.
4. *El Actor* presiona Insertar.
5. *El Sistema* verifica el código.(E1)
6. *El Sistema* almacena la información.

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F1.2, F1.3 ó F1.4.

**Excepciones:**

- E1. Error en BDD

• **F1.2: Modificar Empleado**

**Actores:** Administrador, Técnico

**Descripción:** El Administrador y el Técnico tendrán la capacidad de modificar los datos de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Empleados del Menú de Edición.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa un código de Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código. (E1)
6. *El Sistema* presenta los datos del Empleado.

7. *El Actor* modifica los datos del Empleado.
8. *El Actor* presiona Actualizar.
9. *El Sistema* modifica la información. (E1)

**Flujo Alternativo:**

5. El *Sistema* ejecuta Caso de Uso F1.1, F1.3 ó F1.4.

**Excepciones:**

- E1. Error en la Base de Datos

• **F1.3: Buscar Empleado**

**Actores:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de buscar un Empleado utilizando el código del mismo dentro de la base de datos.

**Flujo Principal:**

1. *El Actor* selecciona Empleados del Menú de Consultas.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa un código de Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código. (E1)
6. *El Sistema* presenta los datos del Empleado.

**Flujo Alternativo:**

6. El *Sistema* ejecuta Caso de Uso F1.1, F1.2 ó F1.4.

**Excepciones:**

- E1. Error en la Base de Datos

- **F1.4: Eliminar Empleado**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Eliminar un Empleado de la base de datos.

**Flujo Principal:**

1. *El Actor* selecciona Empleados del Menú de Eliminar.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa un código de Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código. (E1)
6. *El Sistema* presenta los datos del Empleado.
7. *El Actor* presiona Eliminar.
8. *El Sistema* elimina la información. (E1)

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F1.1, F1.2 ó F1.3

**Excepciones:**

- E1. Error en la BDD

## F2: Manejo de Contratos

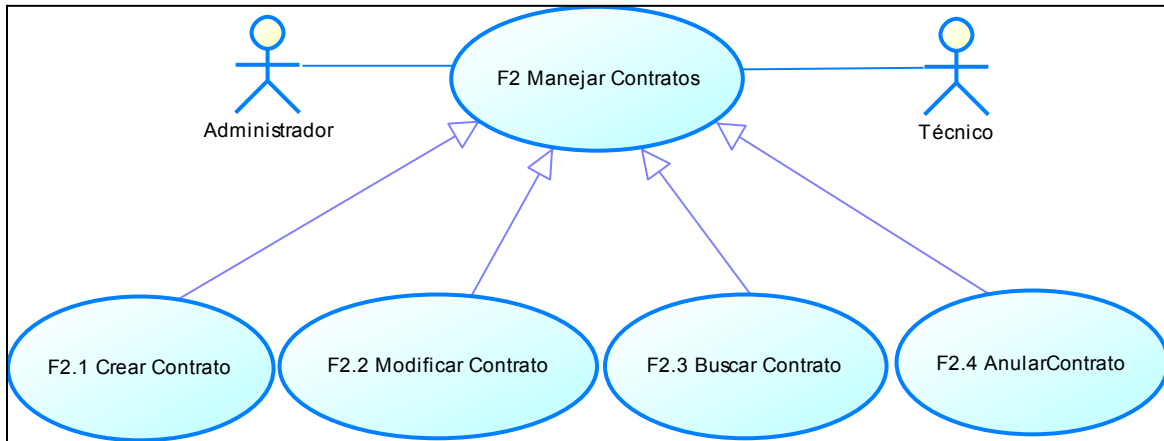


Figura No: 4 Manejo de Contratos

- **F2.1: Crear Contrato**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de crear un Contrato utilizando los datos de un empleado dentro de la base de datos.

**Flujo Principal:**

1. *El Actor* selecciona Generar Contratos del Menú de Creación.
2. *El Sistema* presenta la ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el Empleado.(E1)
6. *El Sistema* abre una ventana mostrando los campos que se requieren para ingresar un contrato nuevo.
7. *El Actor* llena los campos del nuevo contrato.
8. *El Actor* presiona Insertar.
9. *El Sistema* genera el contrato en Microsoft Word.(E2)
10. *El Sistema* almacena la información. (E1)

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F2.2, F2.3 ó F2.4.

**Excepciones:**

- E1. Error en la Base de Datos.
- E2. No tienen instalado Microsoft Word.

**• F2.2: Modificar Contrato**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de modificar un Contrato.

**Flujo Principal:**

1. *El Actor* selecciona Contratos del Menú de Edición.
2. *El Sistema* presenta la ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el Empleado.(E1)
6. *El Sistema* presenta los campos del contrato.
7. *El Actor* llena los campos del contrato.
8. *El Actor* presiona Actualizar.
9. *El Sistema* genera el contrato en Microsoft Word.(E2)
10. *El Sistema* almacena la información. (E1)

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F2.1, F2.3 ó F2.4.

**Excepciones:**

- E1. Error en la Base de Datos
- E2. No tienen instalado Microsoft Word

- **F2.3: Buscar Contrato**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de buscar un Contrato utilizando el código de un empleado dentro de la base de datos.

**Flujo Principal:**

1. *El Actor* selecciona Contratos del Menú de Consultas.
2. *El Sistema* presenta la ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el Empleado.(E1)
6. *El Sistema* presenta los campos del contrato.

**Flujo Alternativo:**

6. El Sistema ejecuta Caso de Uso F2.1, F2.2 ó F2.4.

**Excepciones:**

- E1. Error en conexión a Base de Datos

- **F2.4: Anular Contrato**

**Actor:** Administrador, Técnico

**Descripción:** El Administrador tendrá la capacidad de anular un Contrato utilizando el código de un empleado dentro de la base de datos.

### **Flujo Principal:**

1. *El Actor* selecciona Anular Contratos del Menú de Eliminar.
2. *El Sistema* presenta la ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el Empleado.(E1)
6. *El Sistema* presenta los campos del contrato.
7. *El Actor* presiona Eliminar.
8. *El Sistema* elimina la información.

### **Flujo Alternativo:**

- 6 .El Sistema ejecuta Caso de Uso F2.1, F2.2 ó F2.3

### **Excepciones:**

- E1. Error en la BDD

## Diagrama de Actividades de Manejo de Contratos

El diagrama de actividades representa gráficamente los flujos de trabajo paso a paso del Manejo de Contratos.

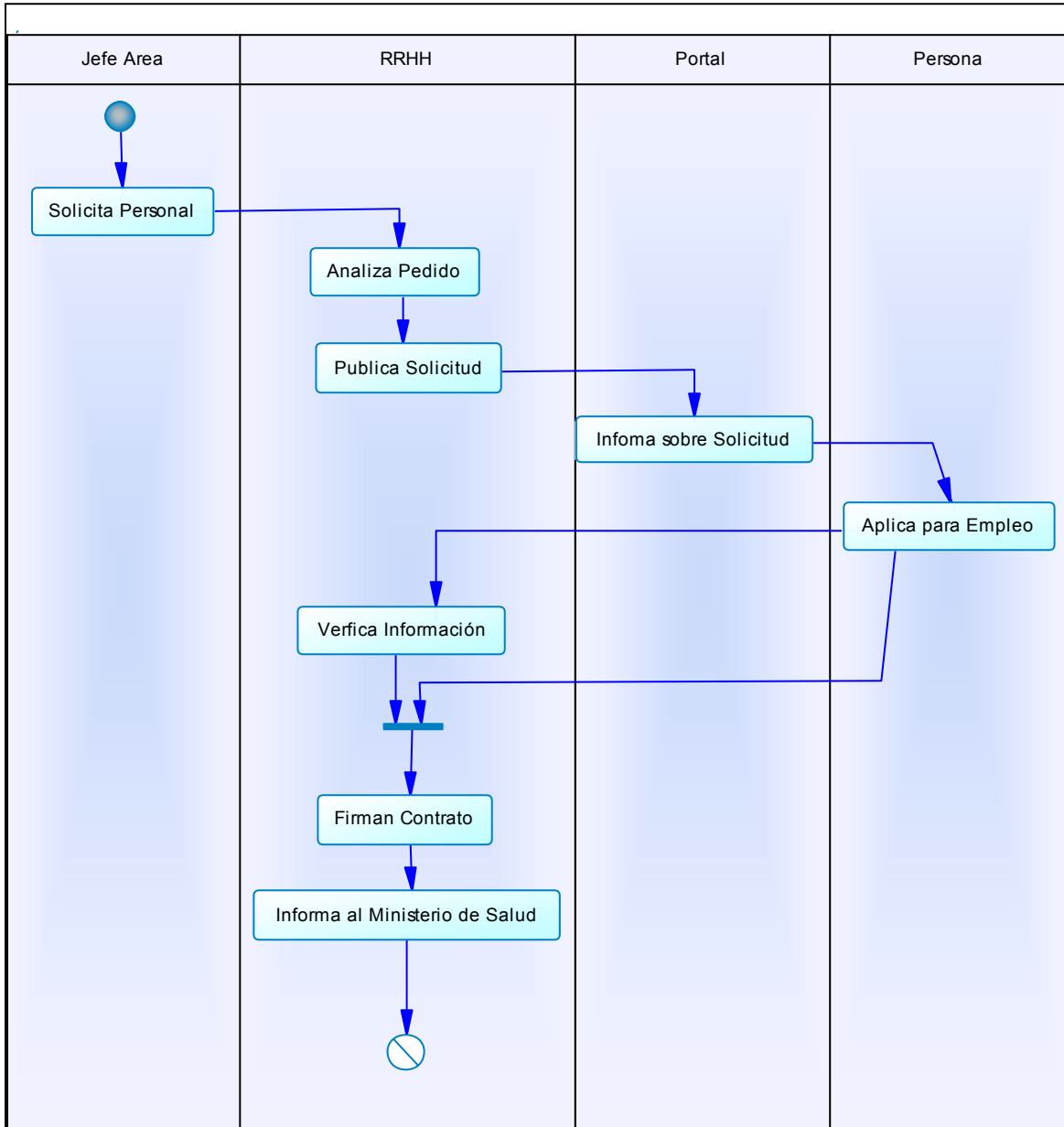


Figura No: 5 Diagrama de Actividades de Manejo de Contratos

### F3: Administración de Horas Extras

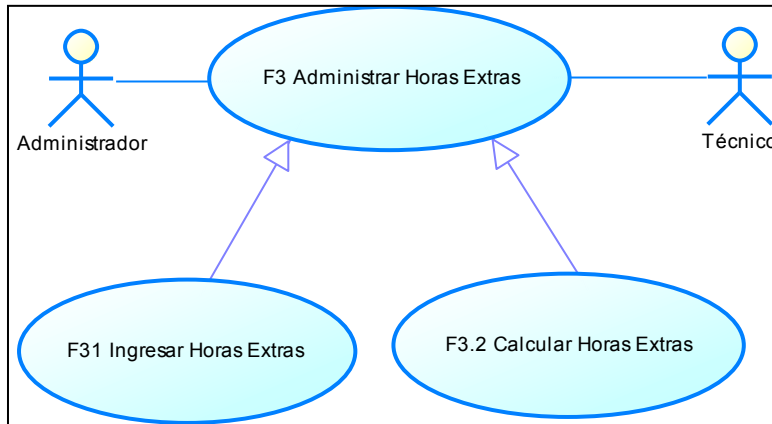


Figura No: 6 Administración de Horas Extras

- **F.3.1: Ingresar Horas Extras**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de ingresar las horas extras de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Horas Extra del Menú de Creación.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código de Empleado. (E1)
6. *El Sistema* presenta los datos del Empleado.
7. *El Actor* ingresa las horas extras del empleado.
8. *El Actor* presiona Insertar.
9. *El Sistema* guarda la información en la base de datos.(E1)

**Flujo Alterno:**

6. El *sistema* ejecuta Caso de Uso F3.2

**Excepciones:**

E1. Error en la Base de Datos

**• F.3.2: Calcular Horas Extras**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de calcular las horas extras de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Calcular Horas Extra del Menú de Consulta.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado, el mes y año.
4. *El Actor* presiona Consultar.
5. *El Sistema* verifica los datos. (E1)
6. *El Sistema* presenta los datos.

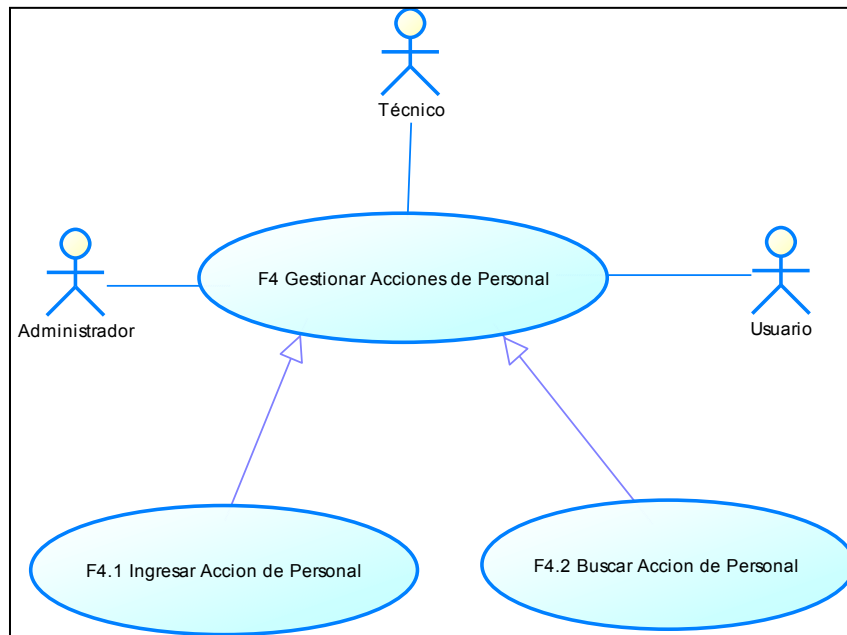
**Flujo Alterno:**

6. El *sistema* ejecuta Caso de Uso F3.1

**Excepciones:**

E1. Error en la Base de Datos

## F4: Gestión de Acciones de Personal



**Figura No: 7 Gestión de Acciones de Personal**

- **F.4.1: Ingresar Acción de Personal**

**Actor:** Administrador, Técnico, Usuario

**Descripción:** El Actor tendrá la capacidad de Ingresar una Acción de Personal de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Acción de Personal del Menú de Creación.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Insertar.(E1)
5. *El Sistema* verifica el código de Empleado.
6. *El Sistema* presenta los campos a llenar.
7. *El Actor* llena los campos de la Acción de Personal.
8. *El Actor* presiona Insertar
9. *El Sistema* genera un archivo Word.(E2)

11. *El Sistema* guarda la información. (E1)

**Flujo Alterno:**

6. *El sistema* ejecuta Caso de Uso F4.2

**Excepciones:**

E1. Error en la Base de Datos

E2. No tienen instalado Microsoft Word

- **F.4.2: Buscar Acción de Personal**

**Actores:** Administrador, Técnico, Usuario

**Descripción:** El Actor tendrá la capacidad de Ingresar una Acción de Personal de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Acción de Personal del Menú de Consulta.

2. *El Sistema* presenta la Ventana para Ingresar un código.

3. *El Actor* ingresa el código del Empleado, el mes y el año.

4. *El Sistema* verifica los datos. (E1)

5. *El Sistema* presenta los datos.

**Flujo Alterno:**

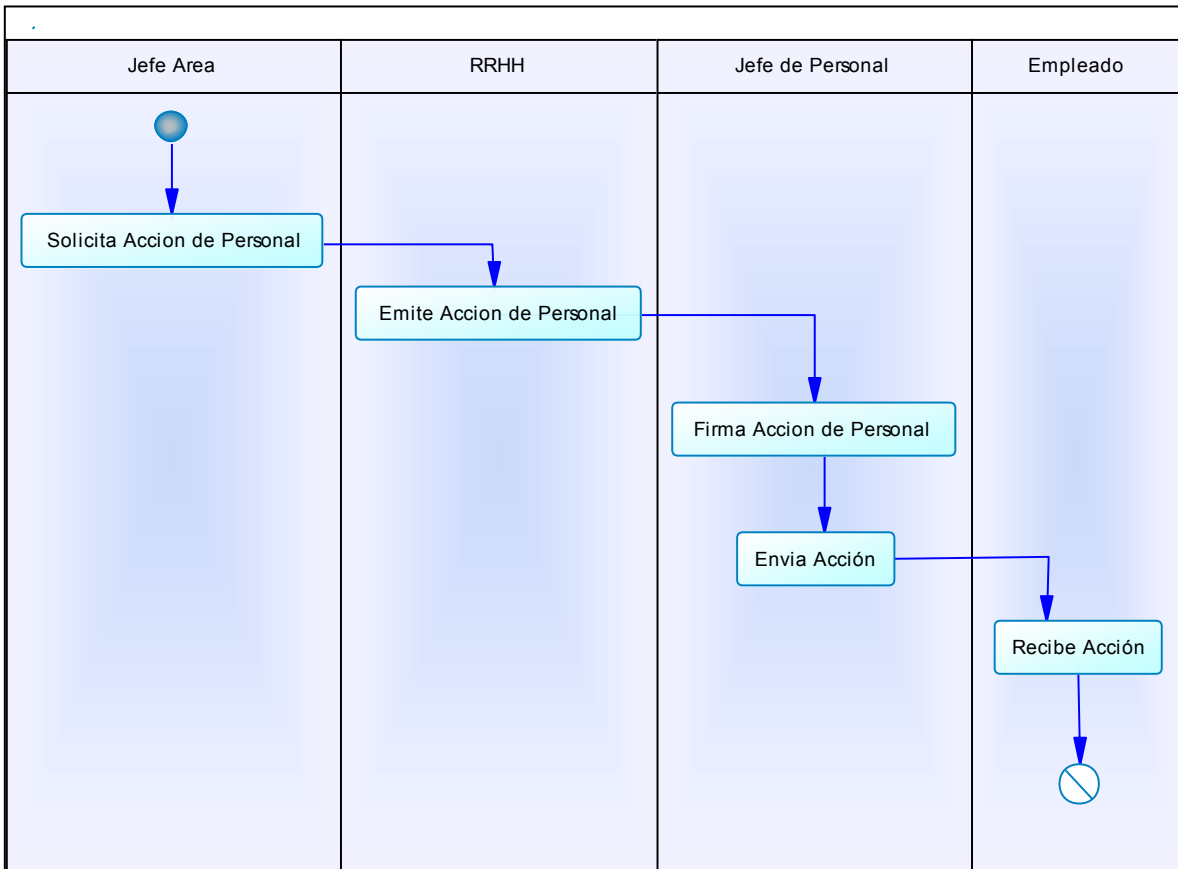
5. *El sistema* ejecuta Caso de Uso F4.1

**Excepciones:**

E1. Error en la Base de Datos

## Diagrama de Actividades de Administración de Acción de Personal

El diagrama de actividades representa gráficamente los flujos de trabajo paso a paso de la Administración de Acciones de Personal.



**Figura No: 8 Diagrama de Actividades de Administración de Acción de Personal**

## F5: Administración de Ausencias

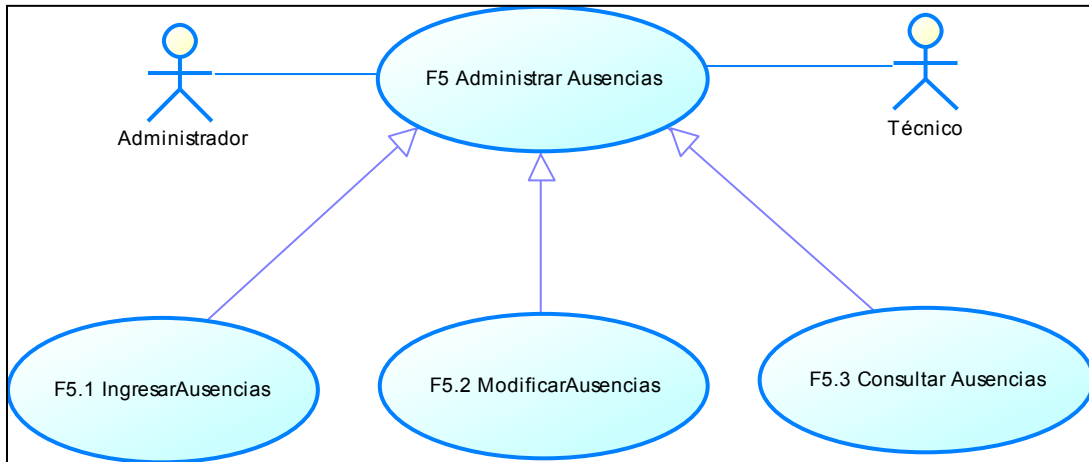


Figura No: 9 Administración de Ausencias

### • F.5.1: Ingresar Ausencias

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Ingresar una Ausencia de un Empleado.

#### Flujo Principal:

1. *El Actor* selecciona Ausencias del Menú de Creación.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código de Empleado. (E1)
6. *El Sistema* presenta los datos del Empleado.
7. *El Actor* llena los campos.
8. *El Actor* presiona Insertar.
9. *El Sistema* guarda la información. (E1)

#### Flujo Alternativo:

6. *El sistema* ejecuta Caso de Uso F5.2 ó F5.3

### **Excepciones:**

E1. Error en la Base de Datos

- **F.5.2: Modificar Ausencias**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Modificar una Ausencia de un Empleado.

### **Flujo Principal:**

1. *El Actor* selecciona Ausencias del Menú de Edición.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado.
4. *El Actor* presiona Buscar.
5. *El Sistema* verifica el código de Empleado y Código de Ausencia. (E1)
6. *El Sistema* presenta los datos del Empleado.
7. *El Sistema* presenta los datos.
8. *El Actor* llena los campos.
9. *El Actor* presiona Actualizar.
10. *El Sistema* modifica la información. (E1)

### **Flujo Alternativo:**

7. El *sistema* ejecuta Caso de Uso F5.1 ó F5.3

### **Excepciones:**

E1. Error en la Base de Datos

- **F.5.3: Consultar Ausencias**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Consultar las Ausencias de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Ausencias del Menú de Consulta.
2. *El Sistema* presenta la Ventana para Ingresar datos.
3. *El Actor* ingresa el código del Empleado, el mes y la fecha.
4. *El Actor* presiona Consultar.
5. *El Sistema* verifica los datos.(E1)
6. *El Sistema* presenta los datos.

**Flujo Alternativo:**

6. El *sistema* ejecuta Caso de Uso F5.1 ó F5.2

**Excepciones:**

E1. Error en la Base de Datos

## F6: Administración de Vacaciones

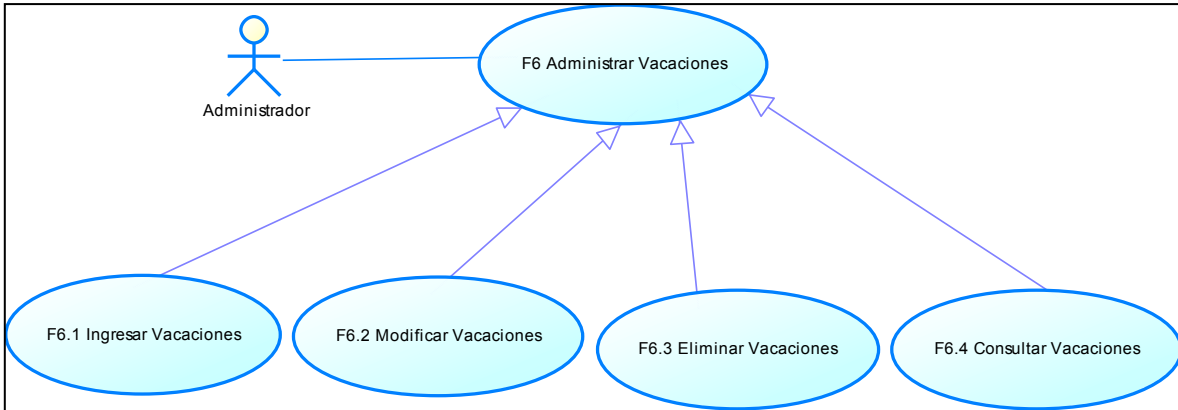


Figura No: 10 Administración de Vacaciones

### • F.6.1: Ingresar Vacaciones

**Actores:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Ingresar un periodo de Vacaciones de un Empleado.

#### Flujo Principal:

1. El Actor selecciona Vacaciones del Menú de Creación.
2. El Sistema presenta la Ventana para Ingresar un código.
3. El Actor ingresa el código del Empleado.
4. El Sistema verifica el código de Empleado. (E1)
5. El Sistema presenta los datos del Empleado.
6. El Actor llena los campos.
7. El Actor presiona Insertar.
8. El Sistema guarda la información. (E1)

#### Flujo Alternativo:

5. El sistema ejecuta Caso de Uso F6.2, F6.4 ó F6.4.

### **Excepciones:**

E1. Error en la Base de Datos

### • **F.6.2: Modificar Vacaciones**

**Actor:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Modificar un período de Vacaciones de un Empleado.

### **Flujo Principal:**

1. *El Actor* selecciona Vacaciones del Menú de Edición.
2. *El Sistema* presenta la Ventana para Ingresar un código.
3. *El Actor* ingresa el código del Empleado y el Código de Vacaciones.
4. *El Sistema* verifica el código los datos. (E1)
5. *El Sistema* presenta los datos.
6. *El Actor* modifica *los campos*.
7. *El Actor* presiona Actualizar.
8. *El Sistema* actualiza la información. (E1)

### **Flujo Alternativo:**

5. El *sistema* ejecuta Caso de Uso F6.1, F6.3 ó F6.4.

### **Excepciones:**

E1. Error en la Base de Datos

- **F.6.3: Eliminar Vacaciones**

**Actores:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Eliminar un período de Vacaciones de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Ausencias del Menú Eliminar.
2. *El Sistema* presenta la Ventana para Ingresar códigos.
3. *El Actor* ingresa el código del Empleado y el código de las Vacaciones.
4. *El Sistema* verifica los datos. (E1)
5. *El Sistema* presenta los datos.
6. *El Actor* presiona #eliminar.
7. El sistema elimina la información de la base de datos.(E1)

**Flujo Alternativo:**

5. El *sistema* ejecuta Caso de Uso F6.1, F6.2 ó F6.4

**Excepciones:**

- E1. Error en la Base de Datos

- **F.6.4: Consultar Vacaciones**

**Actores:** Administrador, Técnico

**Descripción:** El Actor tendrá la capacidad de Consultar una Ausencia de un Empleado.

**Flujo Principal:**

1. *El Actor* selecciona Vacaciones del Menú de Consulta.

2. El Sistema presenta la Ventana para Ingresarlos datos.
3. El Actor ingresa el código del Empleado, el mes y el año.
4. El Sistema verifica el los datos. (E1)
5. El Sistema presenta los datos.

**Flujo Alternativo:**

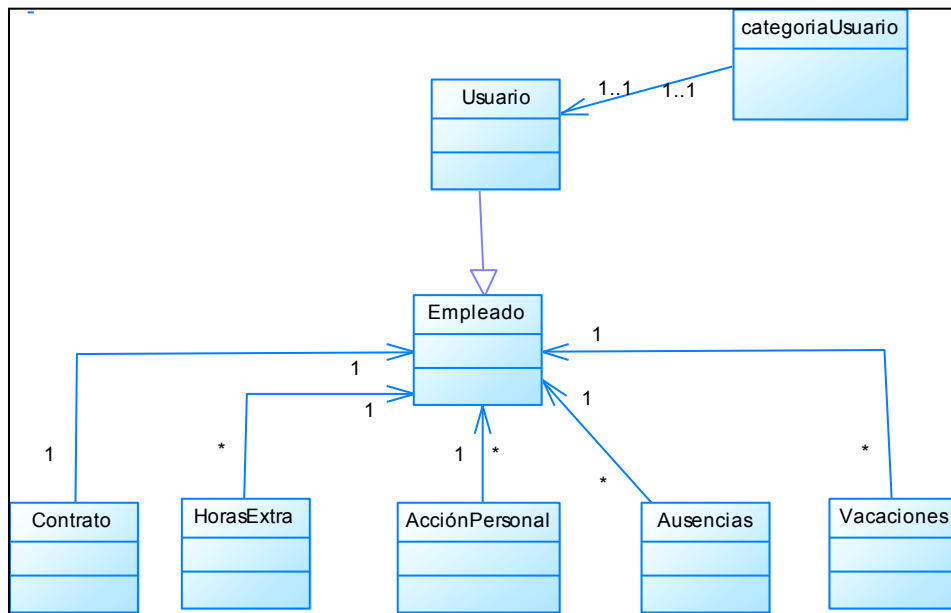
5. El sistema ejecuta Caso de Uso F6.1, F6.2 ó F6.3.

**Excepciones:**

- E1. Error en la Base de Datos

**Diagrama de Clases General**

El diagrama de clases describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos.



**Figura No: 11 Diagrama de Clases General**

## Diagrama de Clases con Atributos

En estos se mostrará cada clase de la aplicación con Atributos y principales métodos.

categoriaUsuario		
- idUsuario	: varchar	
- detalleCategoria	: varchar	
+ <<Getter>>	getIdUsuario ()	: varchar
+ <<Setter>>	setIdUsuario (varchar newIdUsuario)	: void
+ <<Getter>>	getDetalleCategoria ()	: varchar
+ <<Setter>>	setDetalleCategoria (varchar newDetalleCategoria)	: void

**Figura No: 12 Diagrama de Clases Categoría Usuario**

Usuario		
- idUsuario	: varchar	
- userUsuario	: varchar	
- passwordUsuario	: varchar	
+ <<Constructor>>	Usuario ()	
+ <<Getter>>	getUserUsuario ()	: varchar
+ <<Setter>>	setUserUsuario (varchar newUserUsuario)	: void
+ <<Getter>>	getPasswordUsuario ()	: varchar
+ <<Setter>>	setPasswordUsuario (varchar newPasswordUsuario)	: void
+ <<Getter>>	getIdUsuario ()	: varchar
+ <<Setter>>	setIdUsuario (varchar newIdUsuario)	: void

**Figura No: 13 Diagrama de Clases Usuario**

HorasExtras		
- ciHorasE	: varchar	
- ci	: char	
- numHoras	: int	
- fechaHora	: date	
+ <<Getter>>	getCiHorasE ()	: varchar
+ <<Setter>>	setCiHorasE (varchar newCiHorasE)	: void
+ <<Getter>>	getNumHoras ()	: int
+ <<Setter>>	setNumHoras (int newNumHoras)	: void
+ <<Getter>>	getFechaHora ()	: date
+ <<Setter>>	setFechaHora (date newFechaHora)	: void
+ <<Getter>>	getCi ()	: char
+ <<Setter>>	setCi (char newCi)	: void

**Figura No: 14 Diagrama de Clases Horas Extra**

Contrato		
-	ciContrato	: char
-	ci	: char
-	nombres	: char
-	apellidos	: char
-	cargo	: varchar
-	sueldo	: double
-	horario	: varchar
-	tipoContrato	: varchar
-	fechaContrato	: Date
-	duracionContrato	: varchar
-	rutacontrato	: varchar
+	<<Getter>> getCiContrato ()	: char
+	<<Setter>> setCiContrato (char newCiContrato)	: void
+	<<Getter>> getCargo ()	: varchar
+	<<Setter>> setCargo (varchar newCargo)	: void
+	<<Getter>> getSueldo ()	: double
+	<<Setter>> setSueldo (double newSueldo)	: void
+	<<Getter>> getHorario ()	: varchar
+	<<Setter>> setHorario (varchar newHorario)	: void
+	<<Getter>> getTipoContrato ()	: varchar
+	<<Setter>> setTipoContrato (varchar newTipoContrato)	: void
+	<<Getter>> getFechaContrato ()	: Date
+	<<Setter>> setFechaContrato (Date newFechaContrato)	: void
+	<<Getter>> getDuracionContrato ()	: varchar
+	<<Setter>> setDuracionContrato (varchar newDuracionContrato)	: void
+	<<Getter>> getCi ()	: char
+	<<Setter>> setCi (char newCi)	: void
+	<<Getter>> getNombres ()	: char
+	<<Setter>> setNombres (char newNombres)	: void
+	<<Getter>> getApellidos ()	: char
+	<<Setter>> setApellidos (char newApellidos)	: void
+	<<Getter>> getRutacontrato ()	: varchar
+	<<Setter>> setRutacontrato (varchar newRutacontrato)	: void

**Figura No: 15 Diagrama de Clases Contrato**

AccionPersonal		
-	ciAccionP	: varchar
-	ci	: char
-	tipoAccionP	: varchar
-	fechaAccionP	: date
-	rutaAccionP	: varchar
+	<<Getter>> getCiAccionP ()	: varchar
+	<<Setter>> setCiAccionP (varchar newCiAccionP)	: void
+	<<Getter>> getTipoAccionP ()	: varchar
+	<<Setter>> setTipoAccionP (varchar newTipoAccionP)	: void
+	<<Getter>> getFechaAccionP ()	: date
+	<<Setter>> setFechaAccionP (date newFechaAccionP)	: void
+	<<Getter>> getRutaAccionP ()	: varchar
+	<<Setter>> setRutaAccionP (varchar newRutaAccionP)	: void
+	<<Getter>> getCi ()	: char
+	<<Setter>> setCi (char newCi)	: void

**Figura No: 16 Diagrama de Clases Acción de Personal**

Ausencias		
-	ciAusencia	: char
-	ci	: char
-	diaAusencia	: int
-	mesAusencia	: varchar
-	anio	: varchar
<hr/>		
+	<<Getter>> getCiAusencia ()	: char
+	<<Setter>> setCiAusencia (char newCiAusencia)	: void
+	<<Getter>> getDia ()	: int
+	<<Setter>> setDia (int newDiaAusencia)	: void
+	<<Getter>> getMesAusencia ()	: varchar
+	<<Setter>> setMesAusencia (varchar newMesAusencia)	: void
+	<<Getter>> getAnio ()	: varchar
+	<<Setter>> setAnio (varchar newAnio)	: void
+	<<Getter>> getCi ()	: char
+	<<Setter>> setCi (char newCi)	: void

**Figura No: 17 Diagrama de Clases Ausencias**

Vacaciones		
-	ciVacaciones	: varchar
-	ci	: char
-	diaVacaciones	: int
-	mesVacaciones	: varchar
-	anio	: varchar
<hr/>		
+	<<Getter>> getCiVacaciones ()	: varchar
+	<<Setter>> setCiVacaciones (varchar newCiVacaciones)	: void
+	<<Getter>> getDia ()	: int
+	<<Setter>> setDia (int newDiaVacaciones)	: void
+	<<Getter>> getMesVacaciones ()	: varchar
+	<<Setter>> setMesVacaciones (varchar newMesVacaciones)	: void
+	<<Getter>> getCi ()	: char
+	<<Setter>> setCi (char newCi)	: void
+	<<Getter>> getAnio ()	: varchar
+	<<Setter>> setAnio (varchar newAnio)	: void

**Figura No: 18 Diagrama de Clases Vacaciones**

Empleado	
- ci	: varchar
- apellidos	: varchar = Apellidos
- nombres	: varchar = Nombres
- fechaNac	: Date
- estadoCivil	: varchar
- apellidoConyuge	: varchar
- nombreConyuge	: varchar
- trabajoConyuge	: varchar
- cargasFam	: int
- nivelInstrucc	: varchar
- titulo	: varchar
- universidad	: varchar
- fono	: varchar
- cell	: varchar
- correo	: varchar
- direccion	: varchar
- provincia	: varchar
- canton	: varchar
- ciudad	: varchar
- autoidentidad	: varchar
- contactoEmergencia	: varchar
- fonoEmerg	: varchar
- cellEmerg	: varchar
- bnco	: varchar
- noCnta	: varchar
- notaria	: varchar
- fechaDeclaracion	: Date
+ <<Constructor>>	Empleado ()
+ <<Getter>>	getCi () : varchar
+ <<Setter>>	setCi (varchar newCi) : void
+ <<Getter>>	getApellidos () : varchar
+ <<Setter>>	setApellidos (varchar newApellidos) : void
+ <<Getter>>	getApellidoConyuge () : varchar
+ <<Setter>>	setApellidoConyuge (varchar newApellidoConyuge) : void
+ <<Getter>>	getNombres () : varchar
+ <<Setter>>	setNombres (varchar newNombres) : void
+ <<Getter>>	getFechaNac () : Date
+ <<Setter>>	setFechaNac (Date newFechaNac) : void
+ <<Getter>>	getEstadoCivil () : varchar
+ <<Setter>>	setEstadoCivil (varchar newEstadoCivil) : void
+ <<Getter>>	getNombreConyuge () : varchar
+ <<Setter>>	setNombreConyuge (varchar newNombreConyuge) : void
+ <<Getter>>	getTrabajoConyuge () : varchar
+ <<Setter>>	setTrabajoConyuge (varchar newTrabajoConyuge) : void
+ <<Getter>>	getCargasFam () : int
+ <<Setter>>	setCargasFam (int newCargasFam) : void
+ <<Getter>>	getNivelInstrucc () : varchar
+ <<Setter>>	setNivelInstrucc (varchar newNivelInstrucc) : void
+ <<Getter>>	getTitulo () : varchar
+ <<Setter>>	setTitulo (varchar newTitulo) : void
+ <<Getter>>	getUniversidad () : varchar
+ <<Setter>>	setUniversidad (varchar newUniversidad) : void
+ <<Getter>>	getFono () : varchar
+ <<Setter>>	setFono (varchar newFono) : void
+ <<Getter>>	getCell () : varchar
+ <<Setter>>	setCell (varchar newCell) : void
+ <<Getter>>	getCorreo () : varchar
+ <<Setter>>	setCorreo (varchar newCorreo) : void
+ <<Getter>>	getDireccion () : varchar
+ <<Setter>>	setDireccion (varchar newDireccion) : void
+ <<Getter>>	getProvincia () : varchar
+ <<Setter>>	setProvincia (varchar newProvincia) : void
+ <<Getter>>	getCanton () : varchar
+ <<Setter>>	setCanton (varchar newCanton) : void
+ <<Getter>>	getCiudad () : varchar
+ <<Setter>>	setCiudad (varchar newCiudad) : void
+ <<Getter>>	getAutoidentidad () : varchar
+ <<Setter>>	setAutoidentidad (varchar newAutoidentidad) : void
+ <<Getter>>	getContactoEmergencia () : varchar
+ <<Setter>>	setContactoEmergencia (varchar newContactoEmergencia) : void
+ <<Getter>>	getFonoEmerg () : varchar
+ <<Setter>>	setFonoEmerg (varchar newFonoEmerg) : void
+ <<Getter>>	getCellEmerg () : varchar
+ <<Setter>>	setCellEmerg (varchar newCellEmerg) : void
+ <<Getter>>	getBnco () : varchar
+ <<Setter>>	setBnco (varchar newBnco) : void
+ <<Getter>>	getNoCnta () : varchar
+ <<Setter>>	setNoCnta (varchar newNoCnta) : void
+ <<Getter>>	getNotaria () : varchar
+ <<Setter>>	setNotaria (varchar newNotaria) : void
+ <<Getter>>	getFechaDeclaracion () : Date
+ <<Setter>>	setFechaDeclaracion (Date newFechaDeclaracion) : void

Figura No: 19 Diagrama de Clases Empleado

## Diagrama de Secuencia

Los diagramas de Secuencia muestran la interacción de los objetos durante su ciclo de vida. Especificando de mejor manera los casos de uso para así comprender lo que harán desde que el usuario ingresa al sistema.

### Caso de uso: Ingresar Usuario

Identificador: F0.1

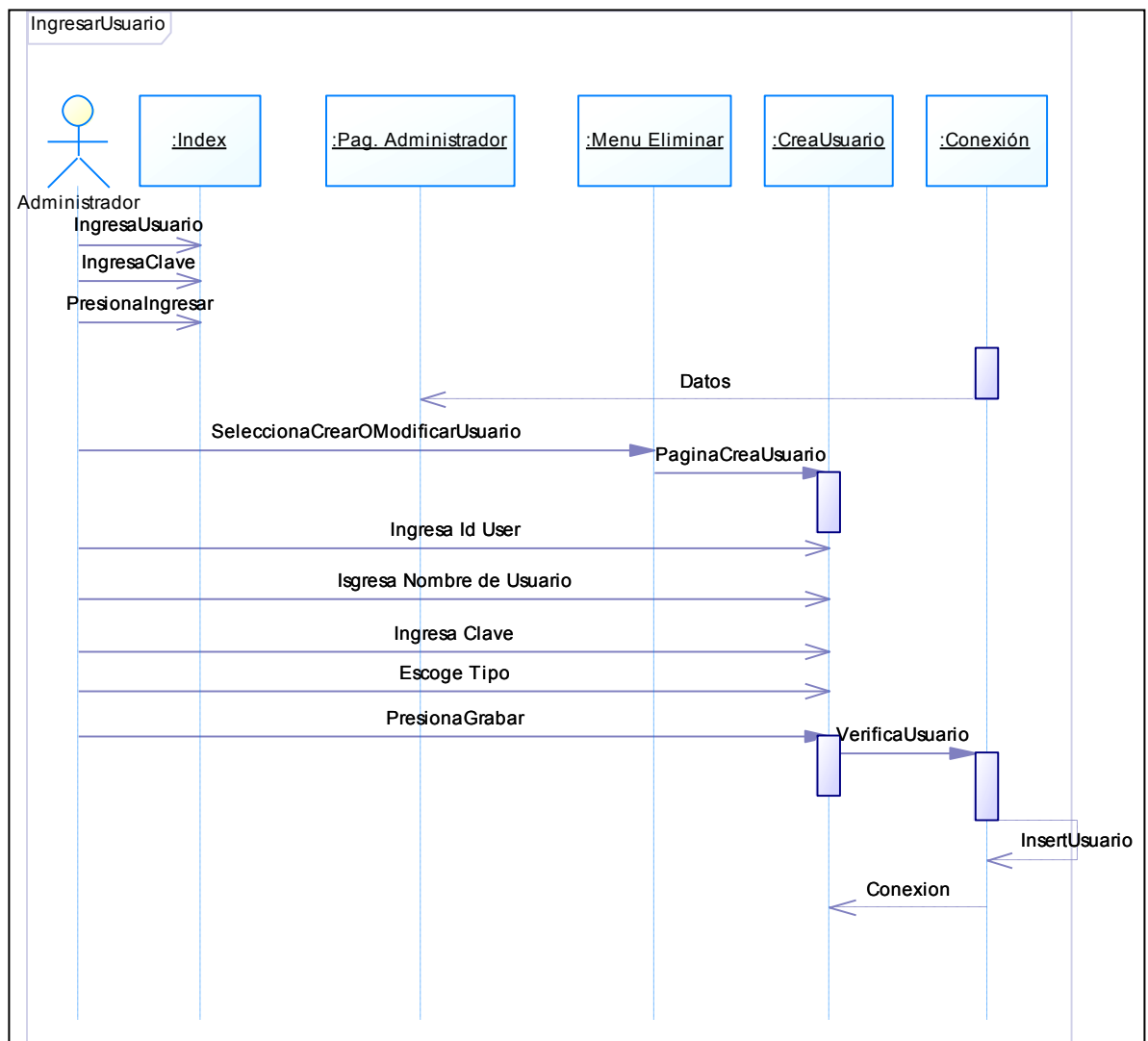
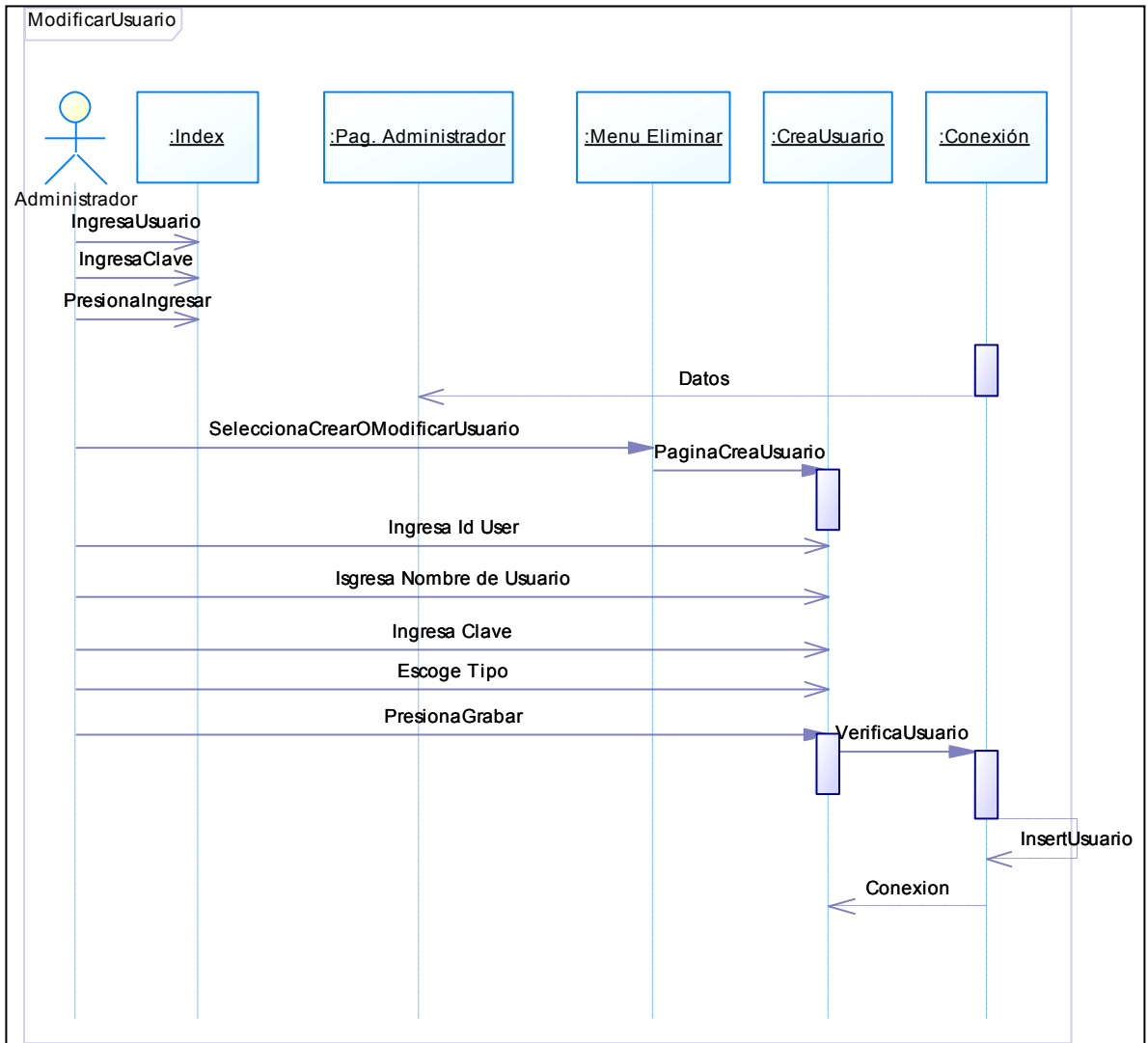


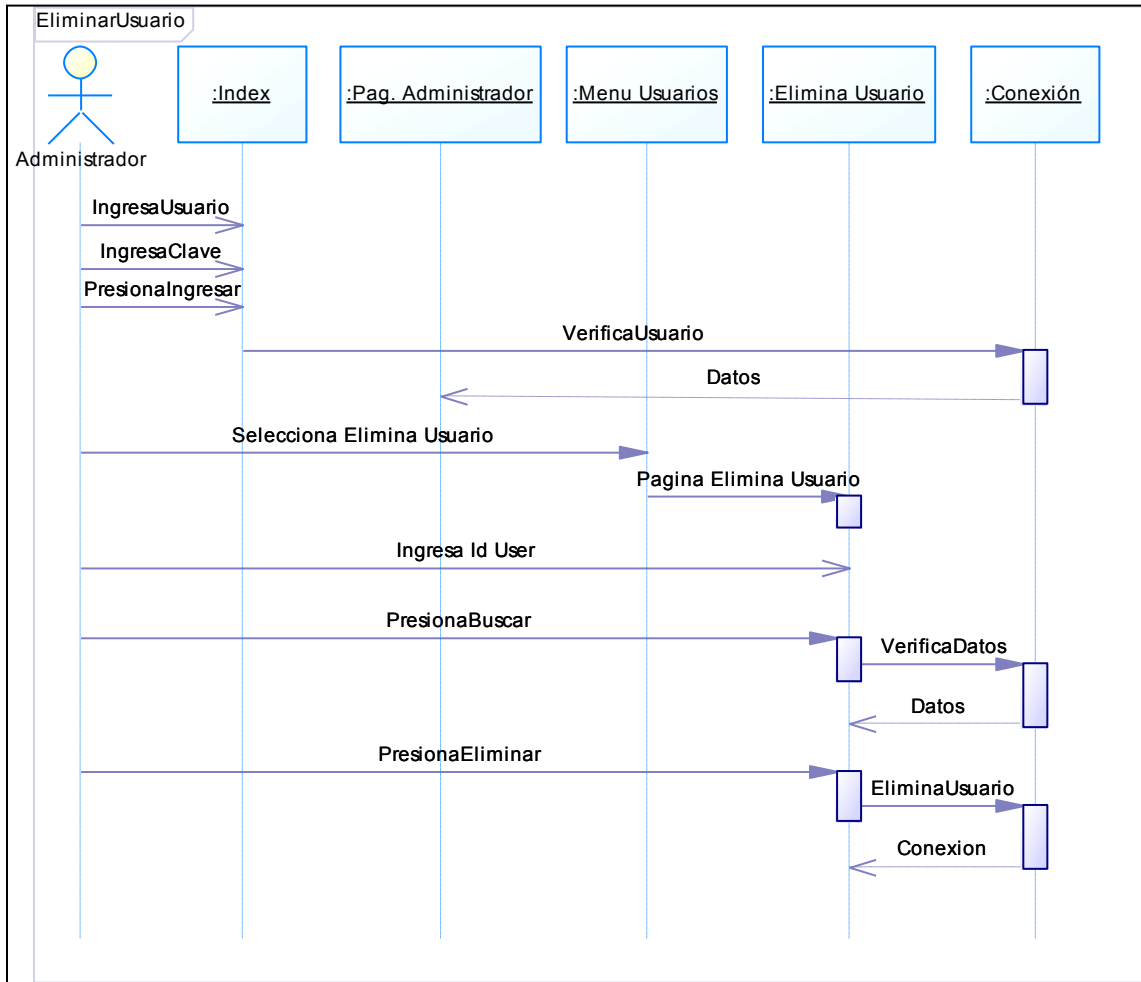
Figura No: 20 Diagrama de Secuencia Ingresar Usuario

**Caso de uso: Modificar Usuario**  
**Identificador: F0.2**



**Figura No: 21**Diagrama de Secuencia Modificar Usuario

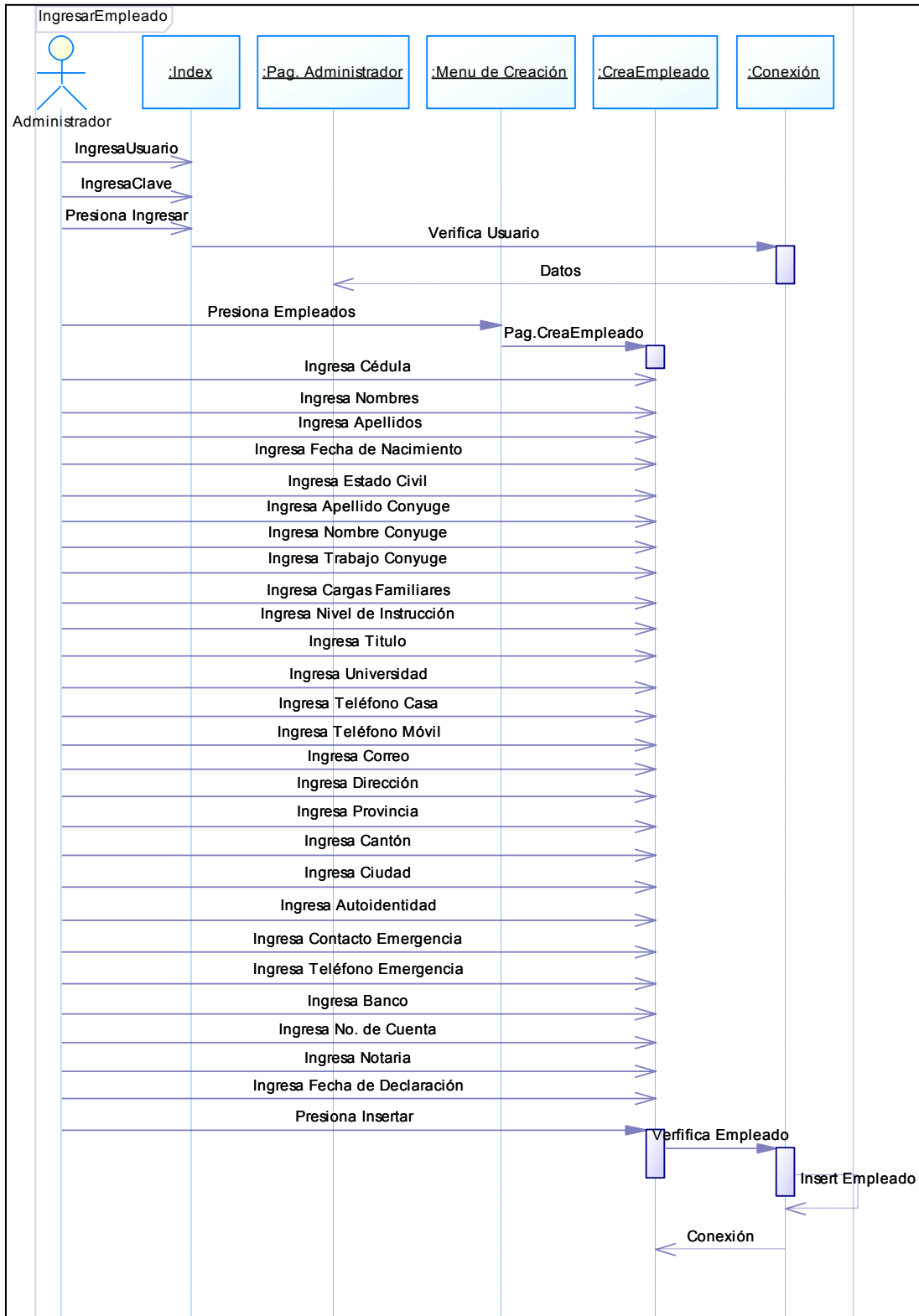
**Caso de uso: Eliminar Usuario**  
**Identificador: F0.3**



**Figura No: 22 Diagrama de Secuencia Eliminar Usuario**

## Caso de uso: Ingresar Empleado

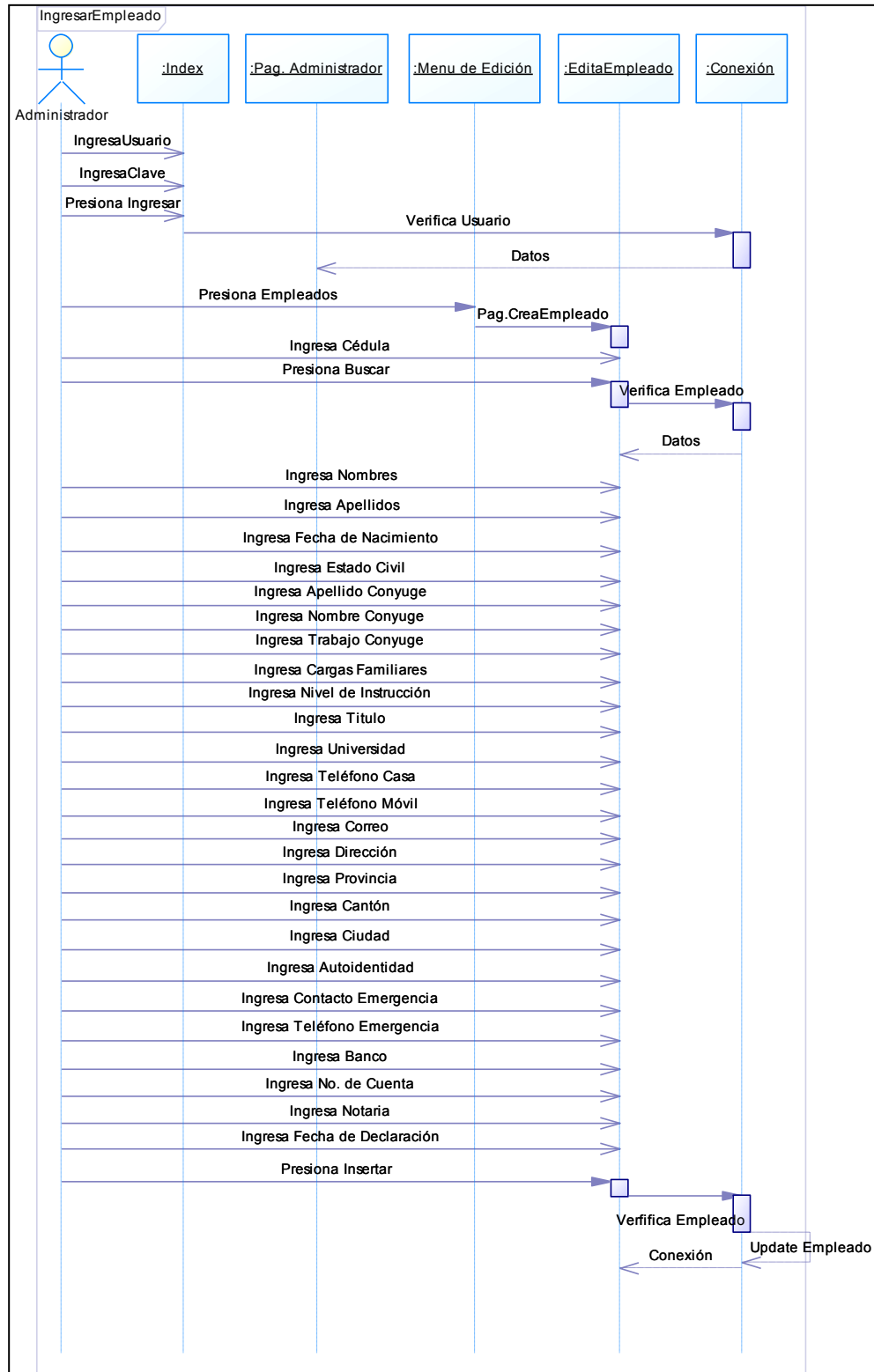
### Identificador: F1.1



**Figura No: 23 Diagrama de Secuencia Ingresar Empleado**

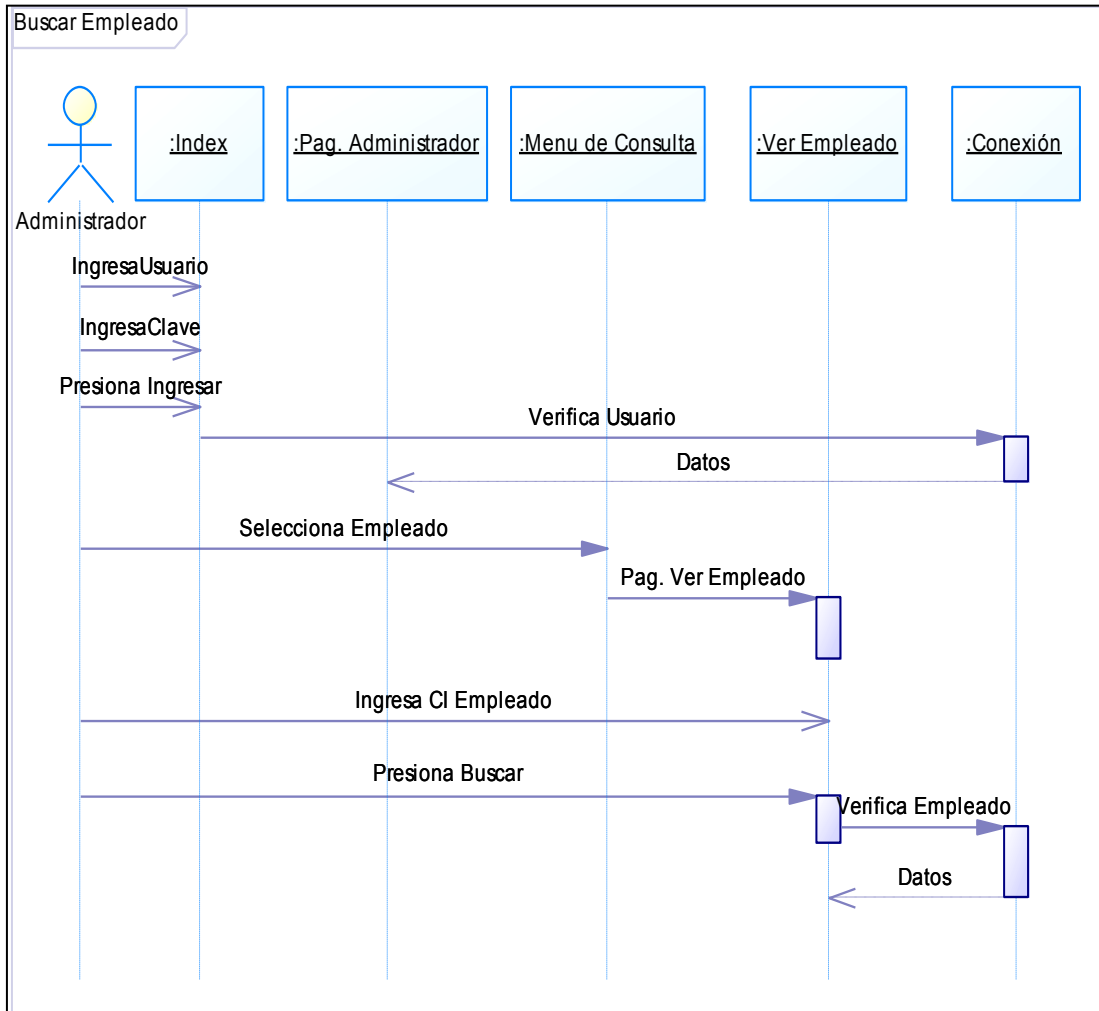
## Caso de uso: Modificar Empleado

**Identificador: F1.2**



**Figura No: 24 Diagrama de Secuencia Modificar Empleado**

**Caso de uso: Buscar Empleado**  
**Identificador: F1.3**



**Figura No: 25 Diagrama de Secuencia Buscar Empleado**

## Caso de uso: Eliminar Empleado

Identificador: F1.4

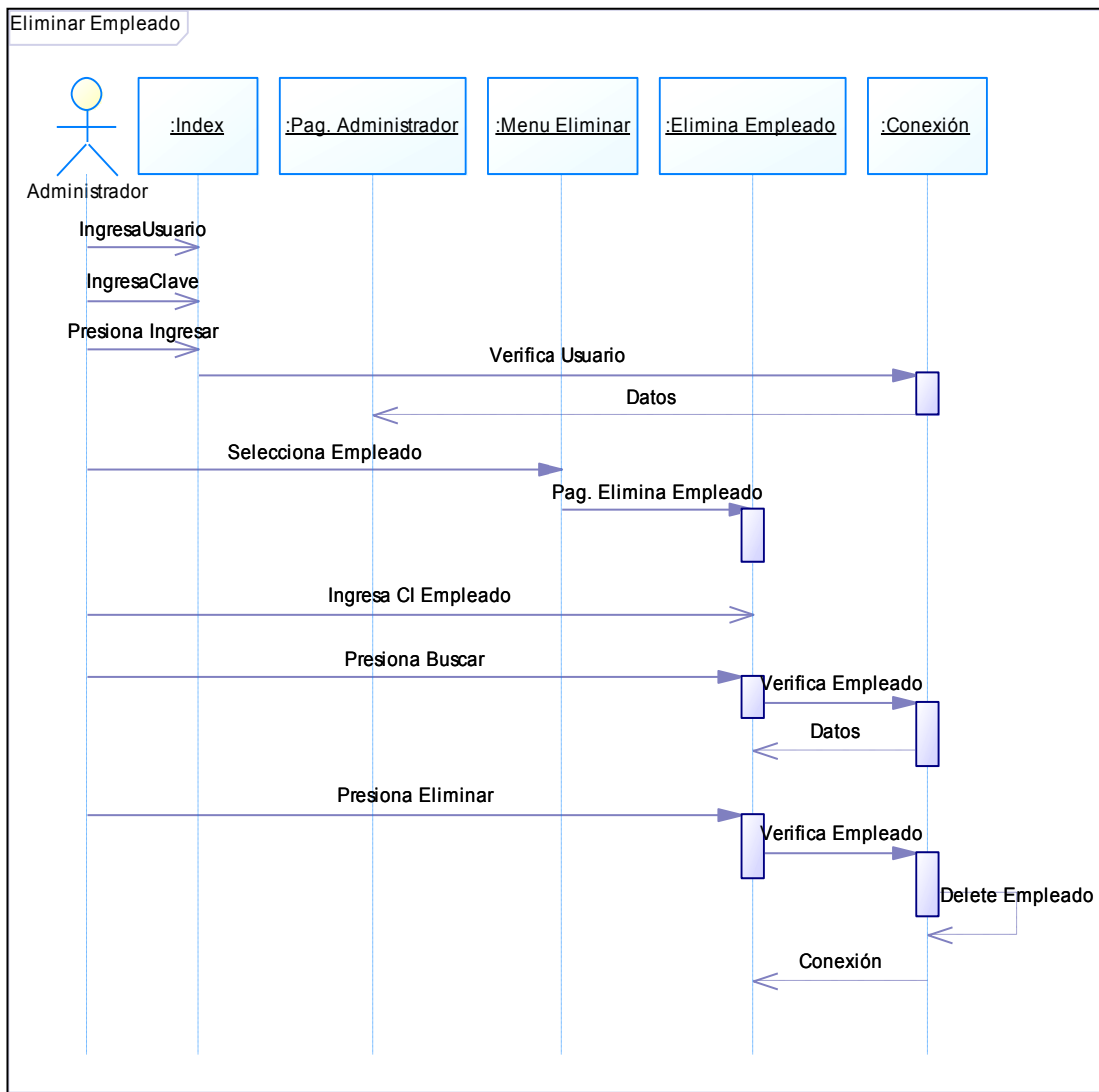
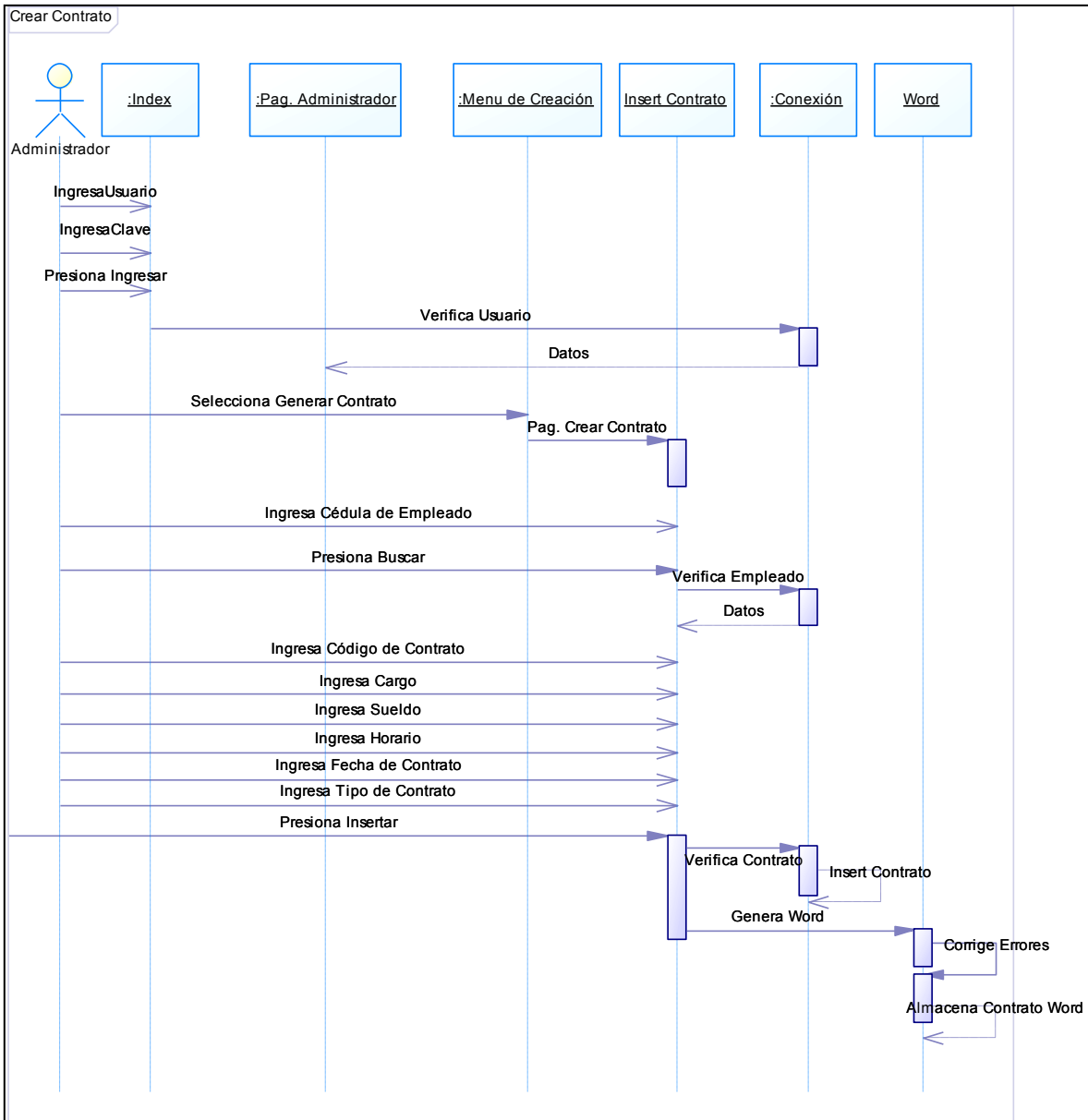


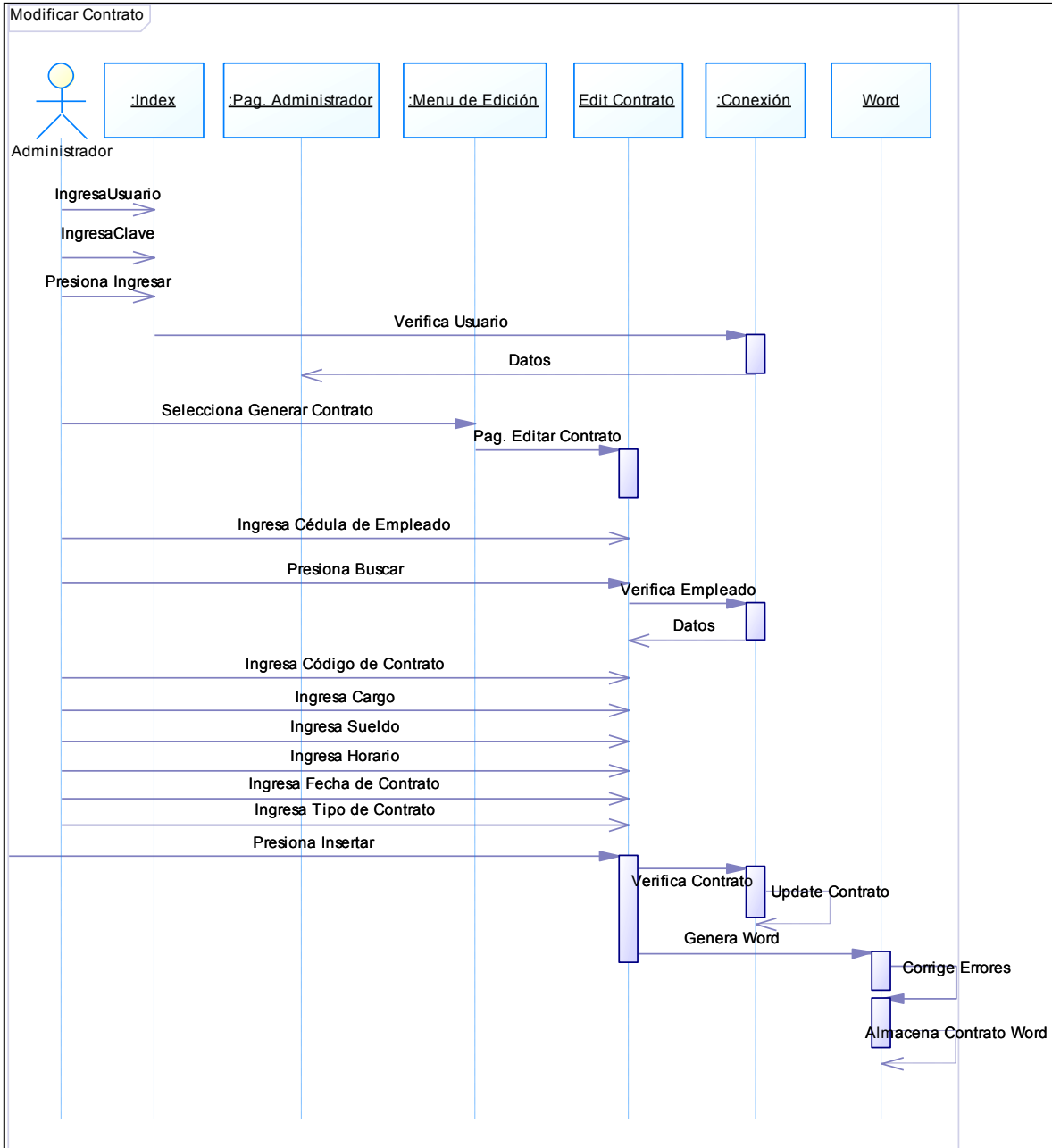
Figura No: 26 Diagrama de Secuencia Eliminar Empleado

**Caso de uso: Crear Contrato**  
**Identificador: F2.1**



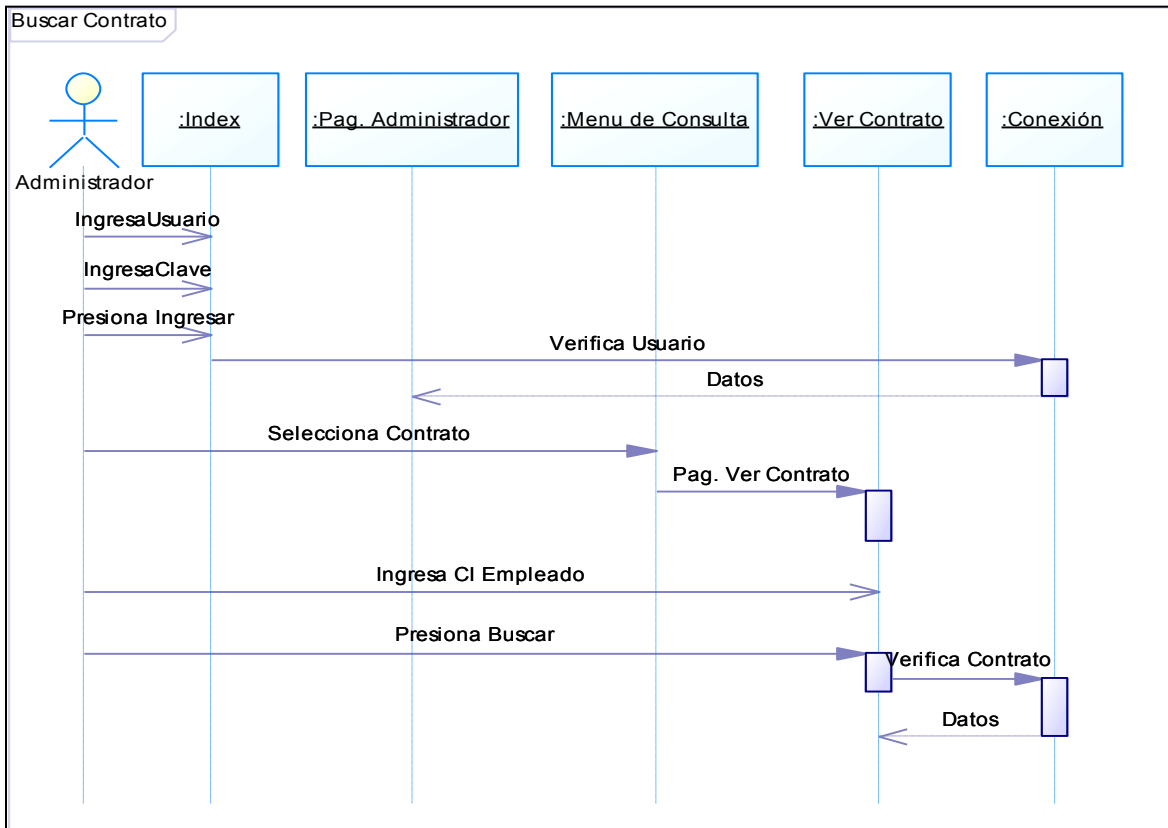
**Figura No: 27 Diagrama de Secuencia Crear Contrato**

**Caso de uso: Modificar Contrato**  
**Identificador: F2.2**



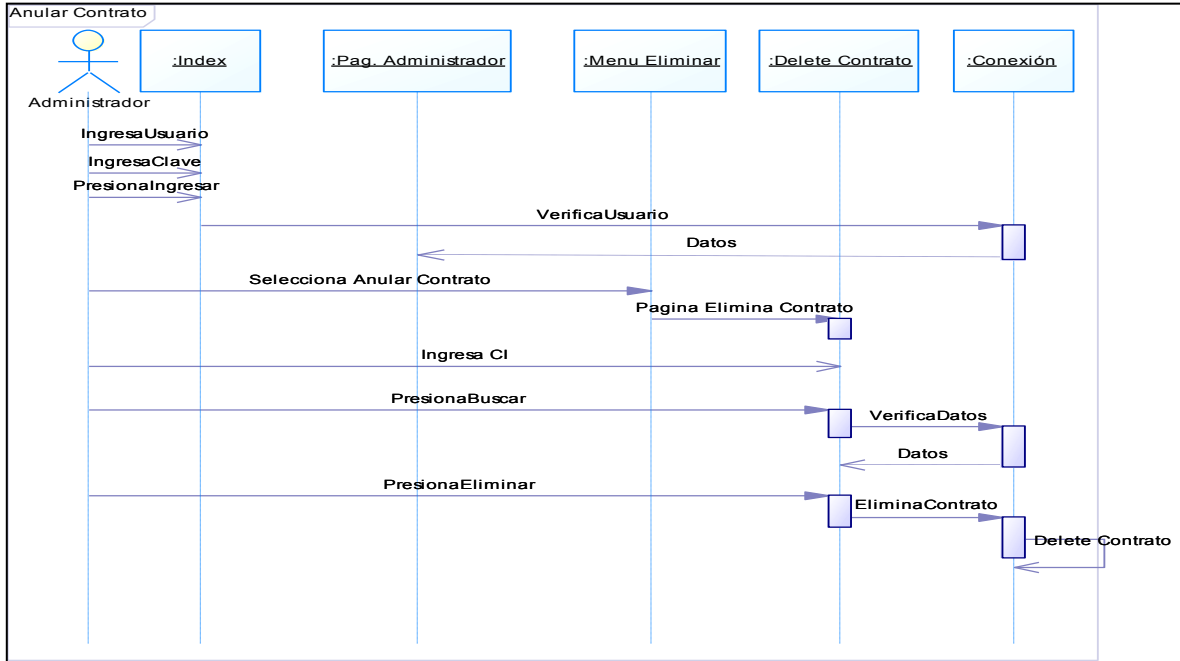
**Figura No: 28 Diagrama de Secuencia Modificar Contrato**

**Caso de uso: Buscar Contrato**  
**Identificador: F2.3**



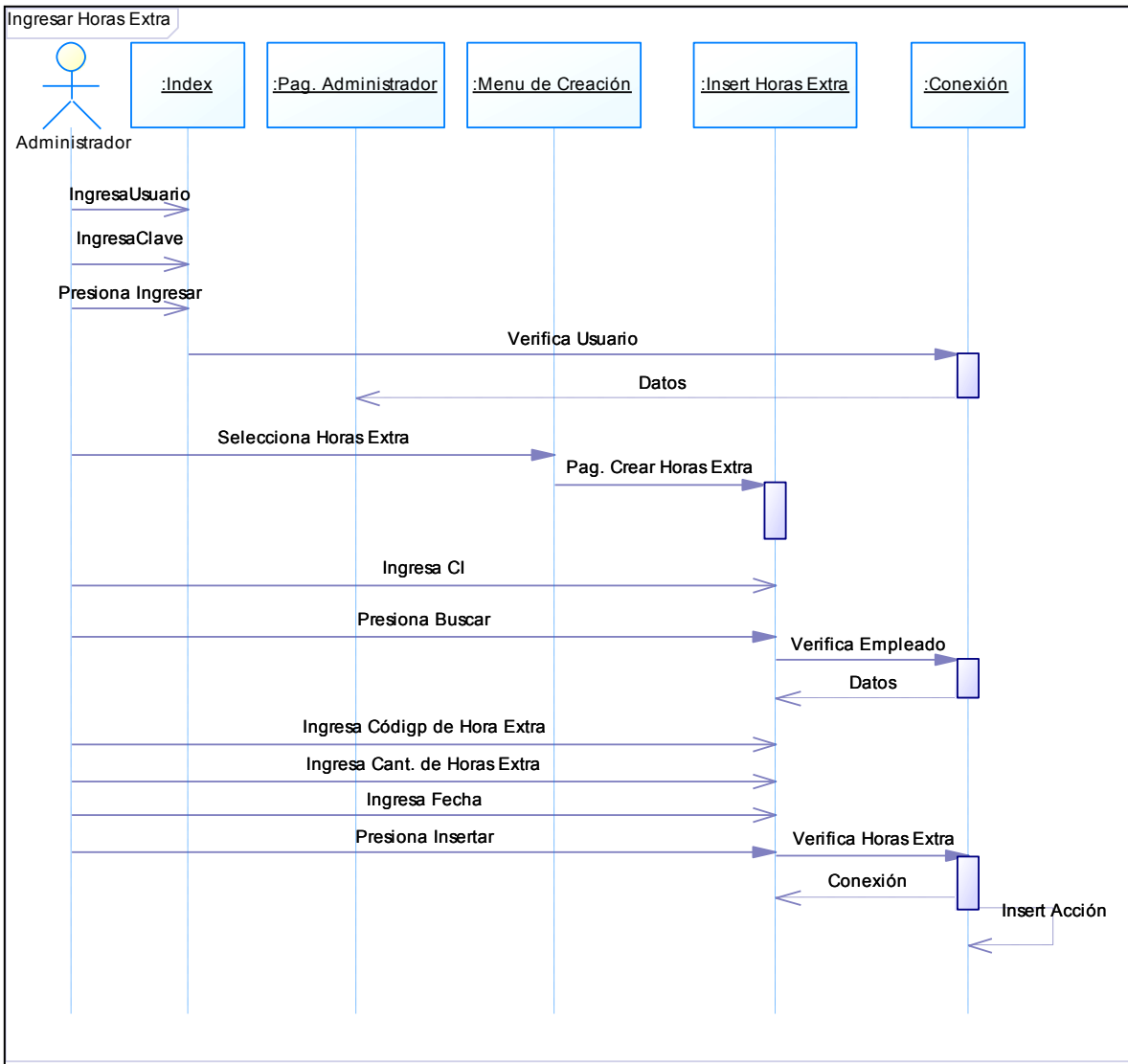
**Figura No: 29 Diagrama de Secuencia Buscar Contrato**

**Caso de uso: Anular Contrato**  
**Identificador: F2.4**



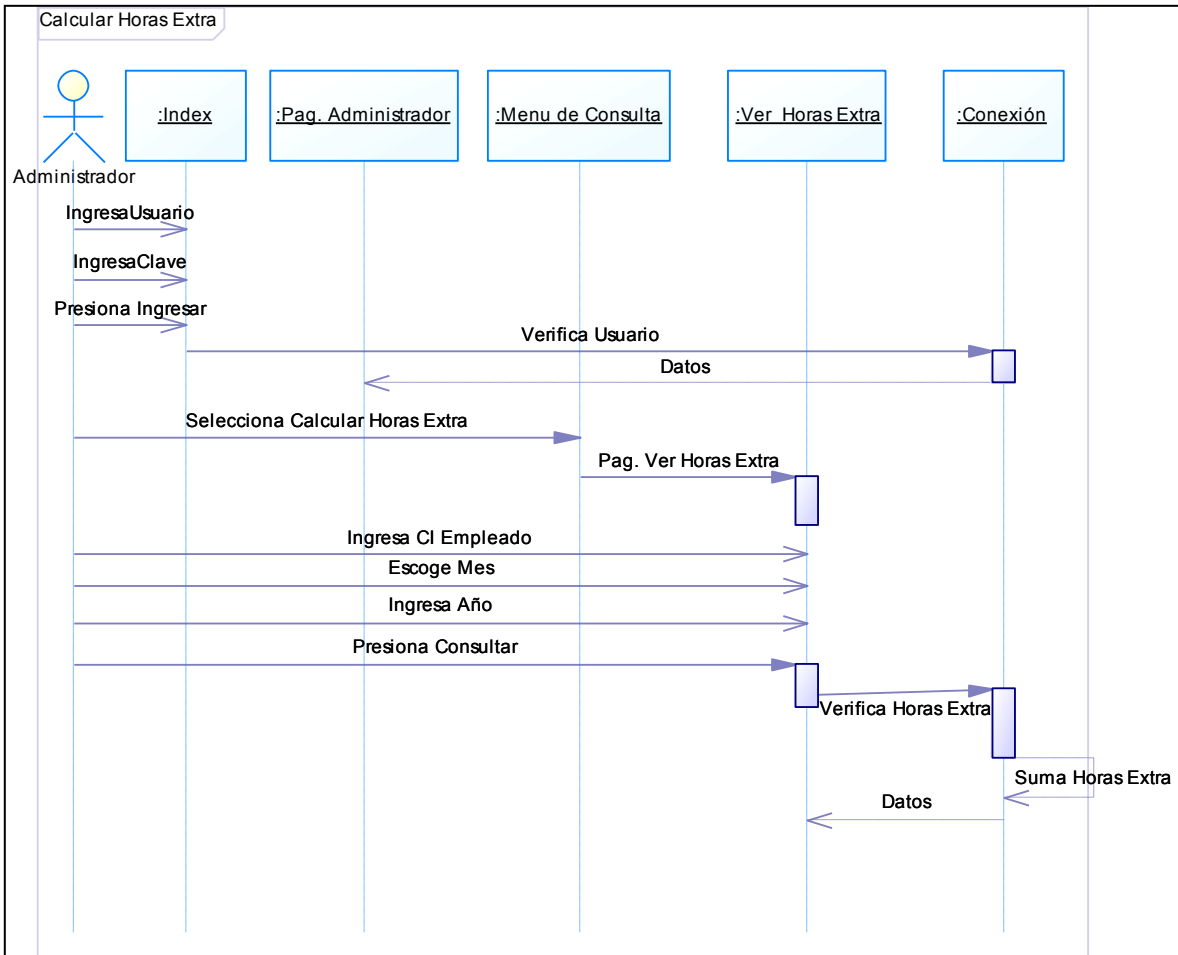
**Figura No: 30 Diagrama de Secuencia Anular Contrato**

**Caso de uso: Ingresar Horas Extra**  
**Identificador: F3.1**



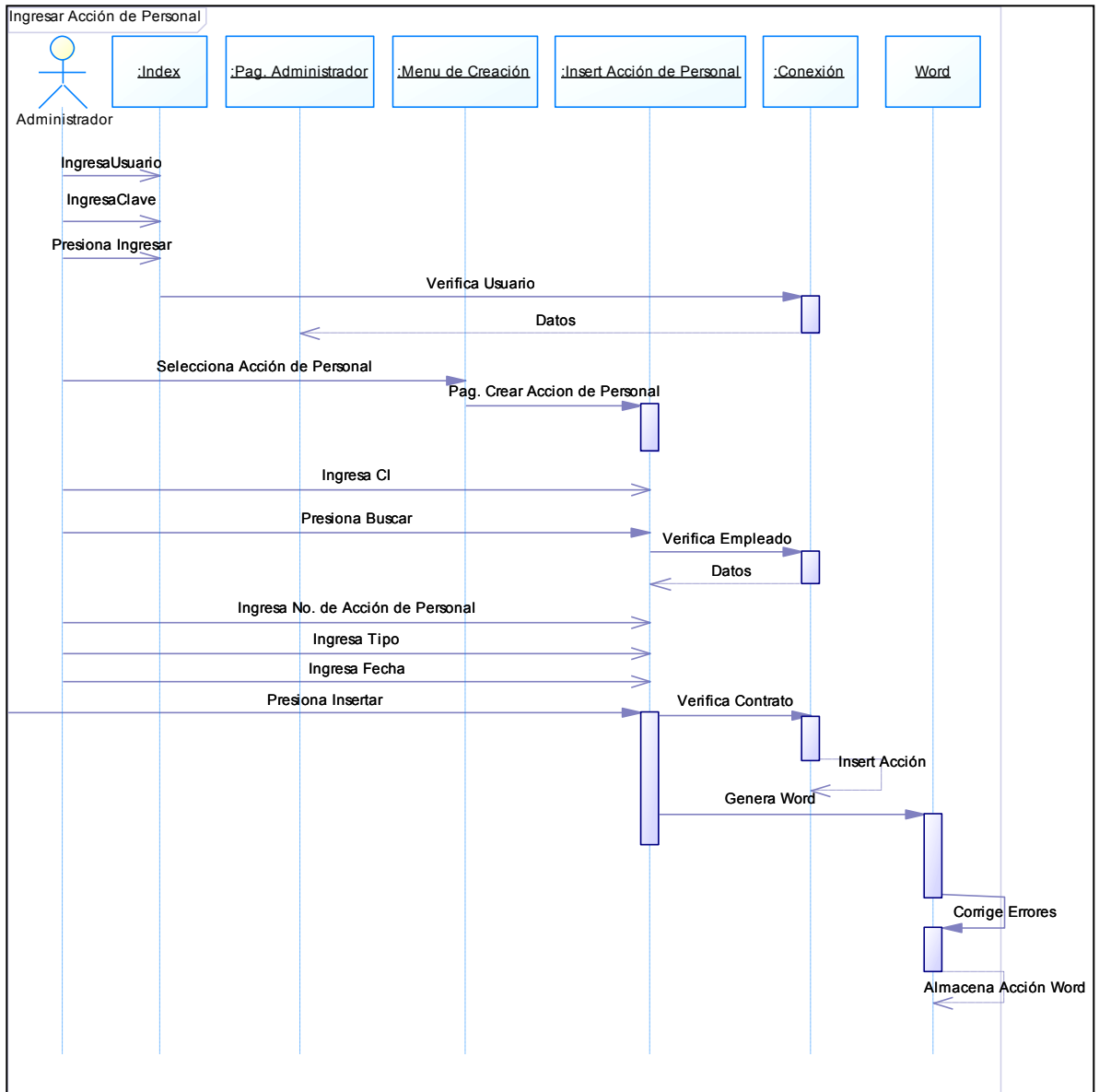
**Figura No: 31 Diagrama de Secuencia Ingresar Horas Extra**

**Caso de uso: Calcular Horas Extra**  
**Identificador: F3.2**



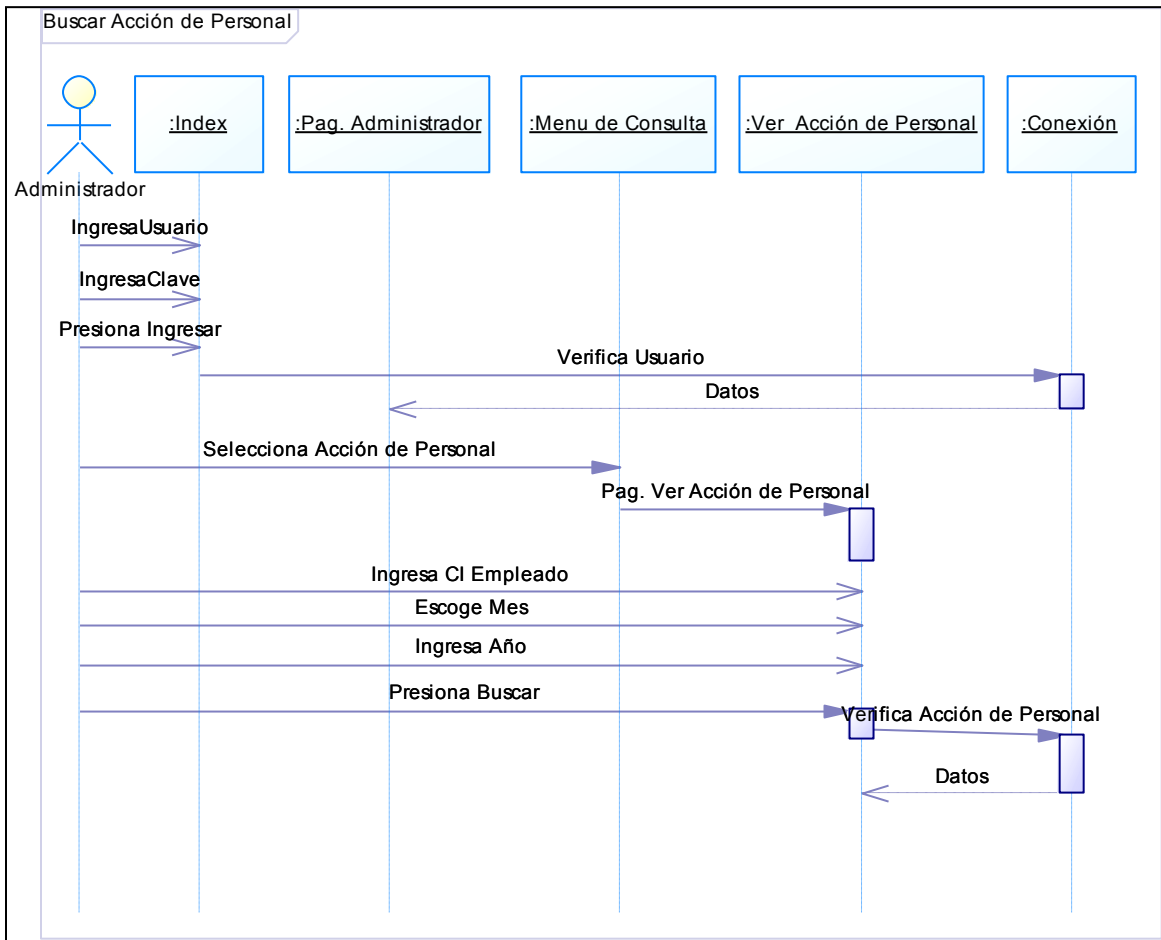
**Figura No: 32 Diagrama de Secuencia Calcular Horas Extra**

**Caso de uso: Ingresar Acción de Personal**  
**Identificador: F4.1**



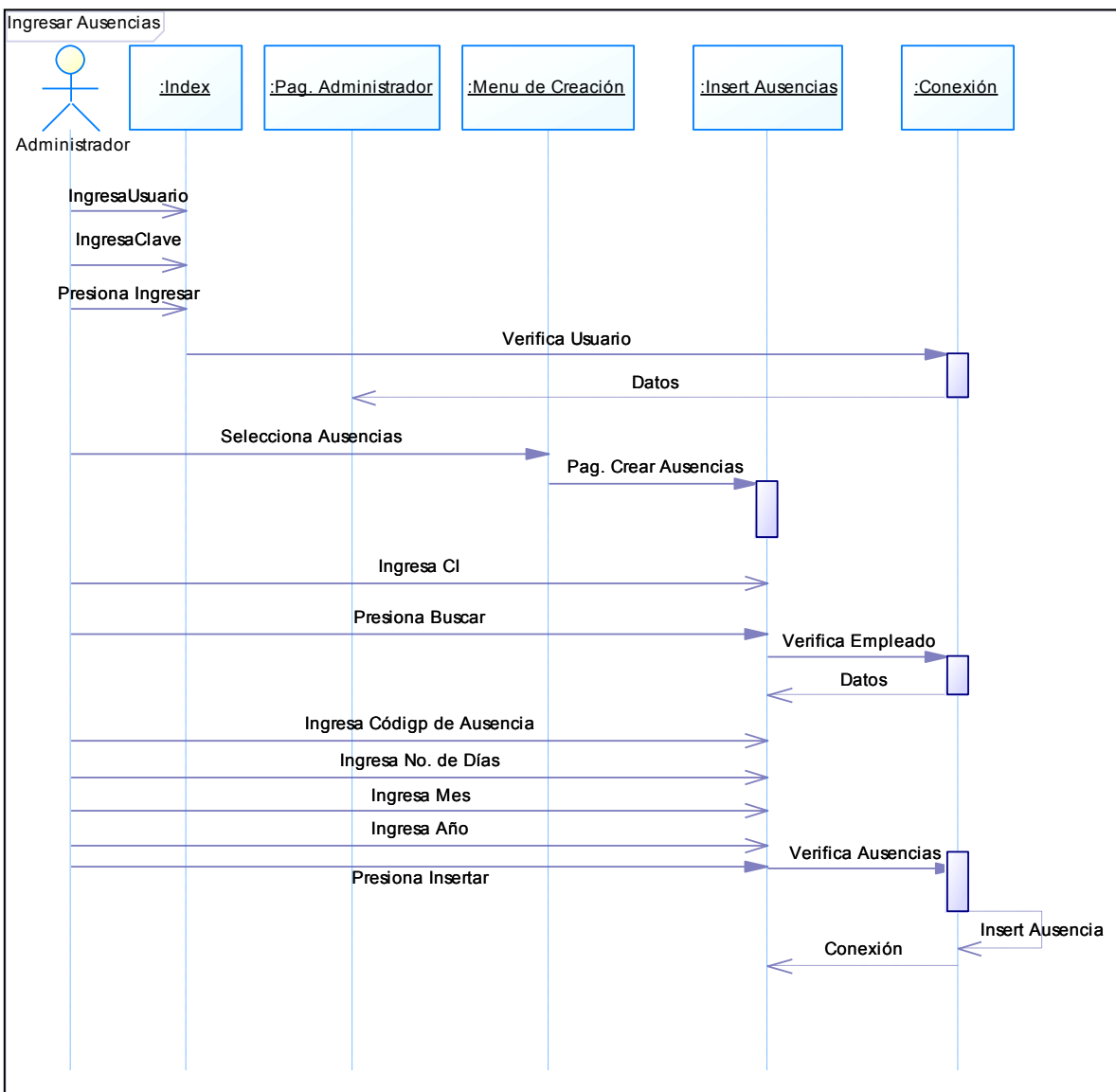
**Figura No: 33 Diagrama de Secuencia Ingresar Acción de Personal**

**Caso de uso: Buscar Acción de Personal**  
**Identificador: F4.2**



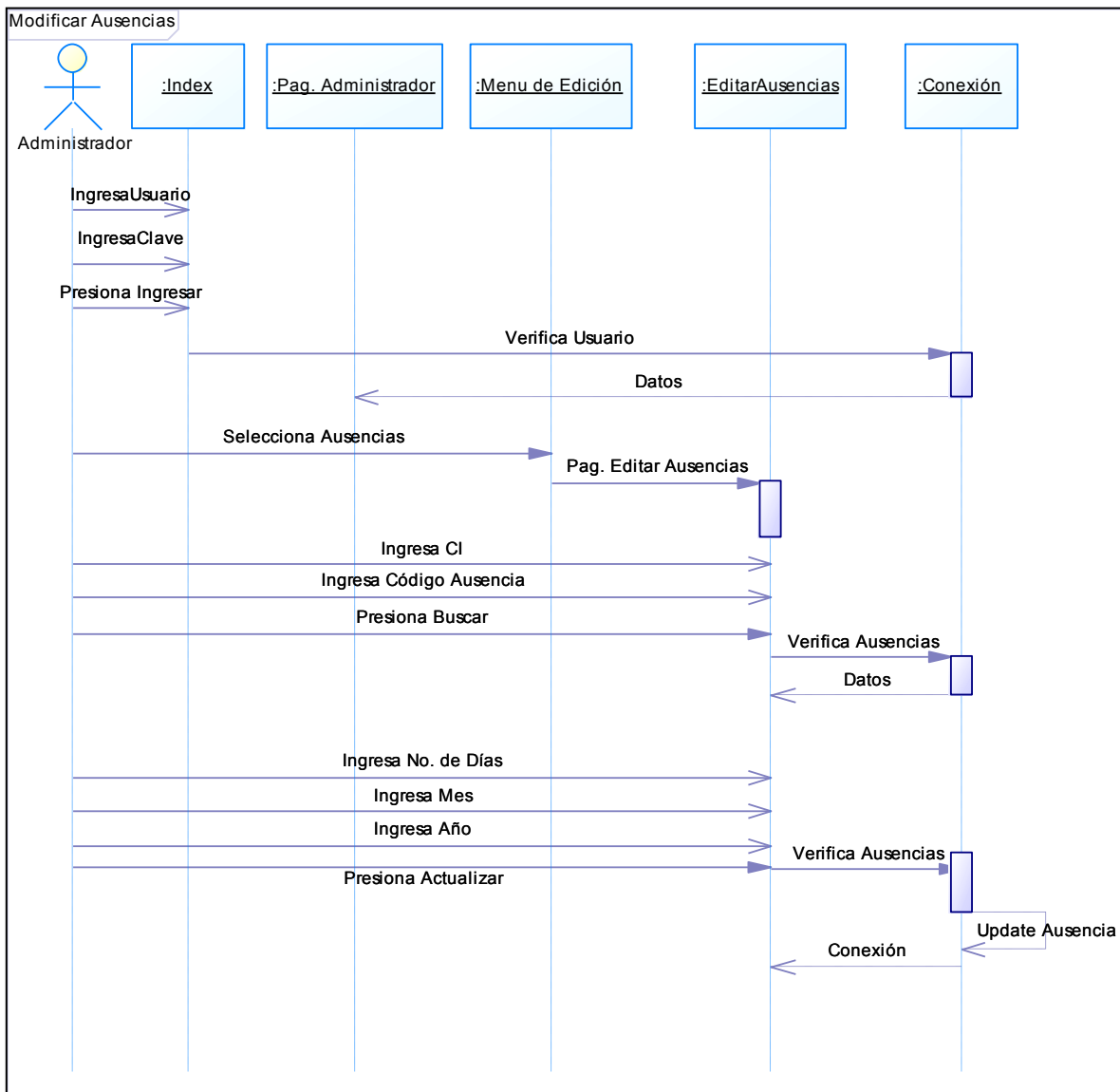
**Figura No: 34 Diagrama de Secuencia Buscar Acción de Personal**

**Caso de uso: Ingresar Ausencias**  
**Identificador: F5.1**



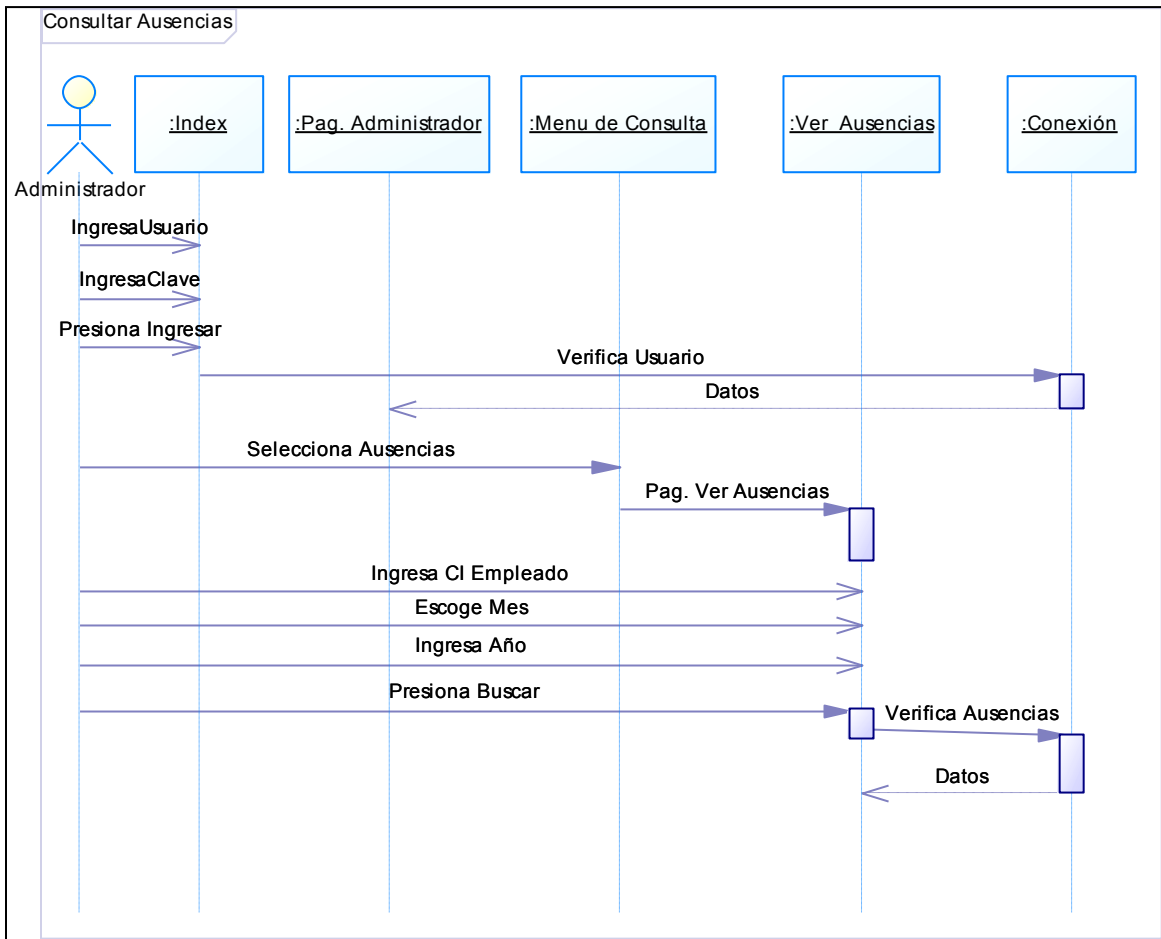
**Figura No: 35 Diagrama de Secuencia Ingresar Ausencias**

**Caso de uso: Modificar Ausencias**  
**Identificador: F5.2**



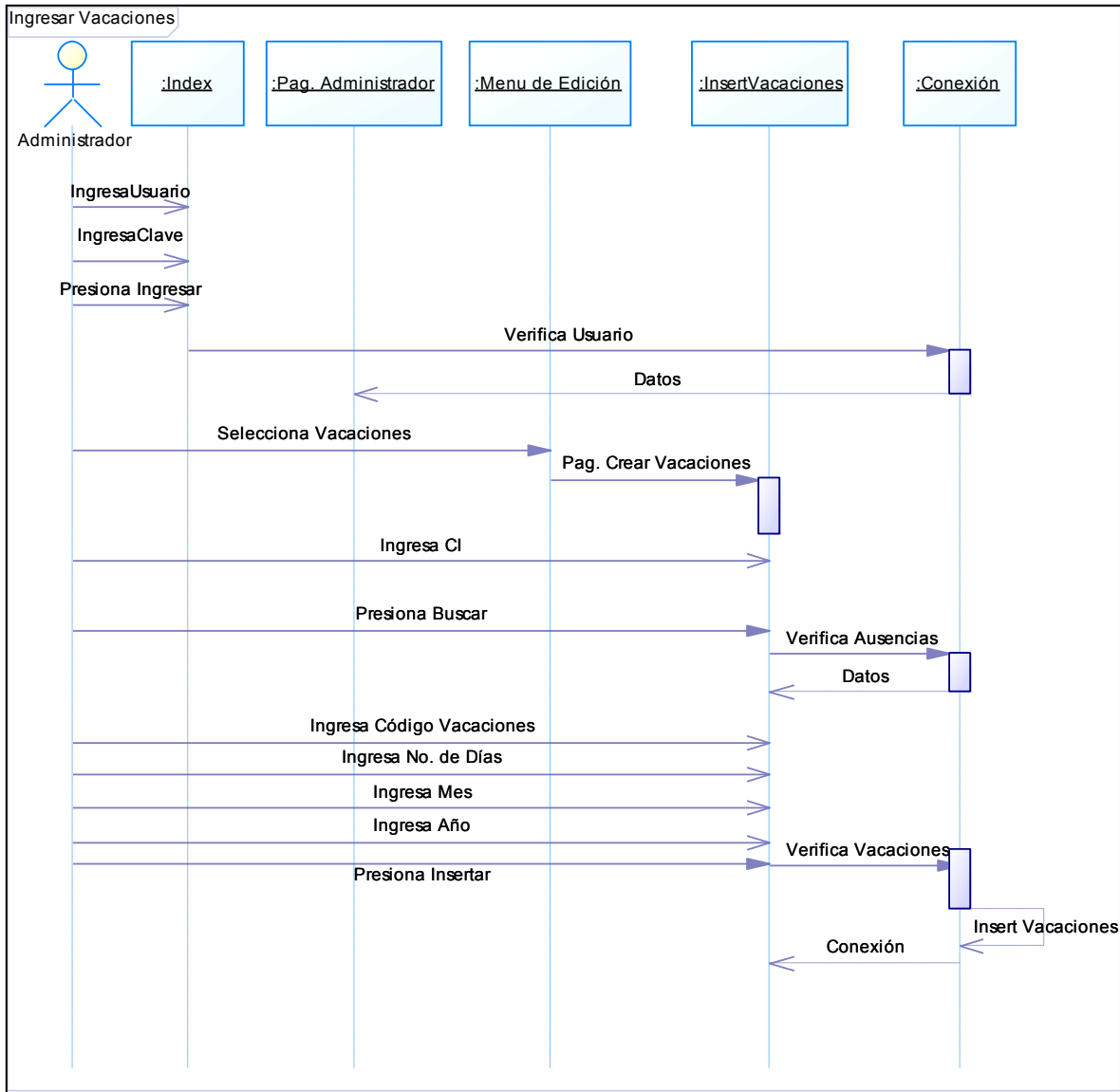
**Figura No: 36 Diagrama de Secuencia Modificar Ausencias**

**Caso de uso: Consultar Ausencias**  
**Identificador: F5.3**



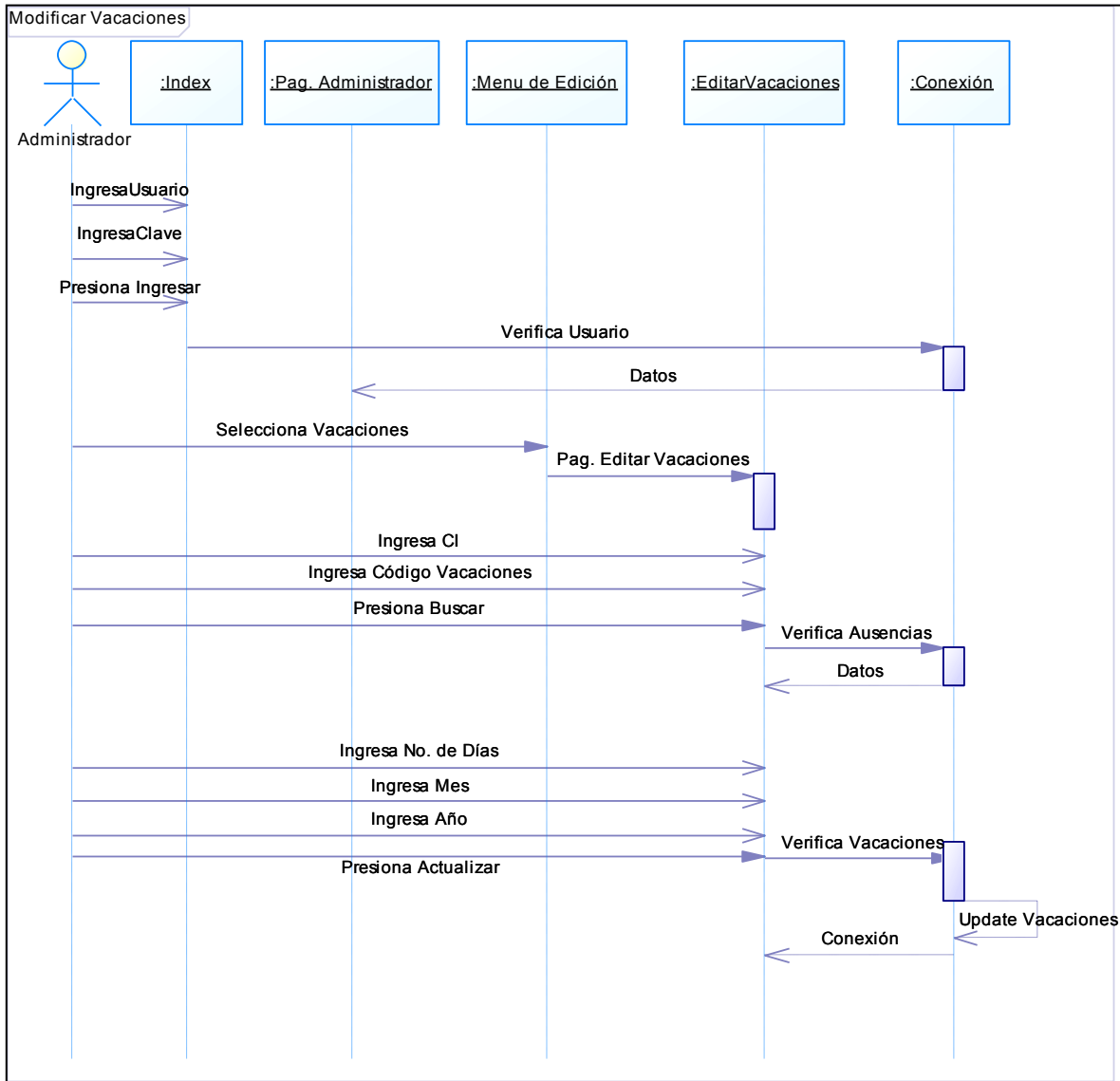
**Figura No: 37 Diagrama de Secuencia Consultar Ausencias**

**Caso de uso: Ingresar Vacaciones**  
**Identificador: F6.1**



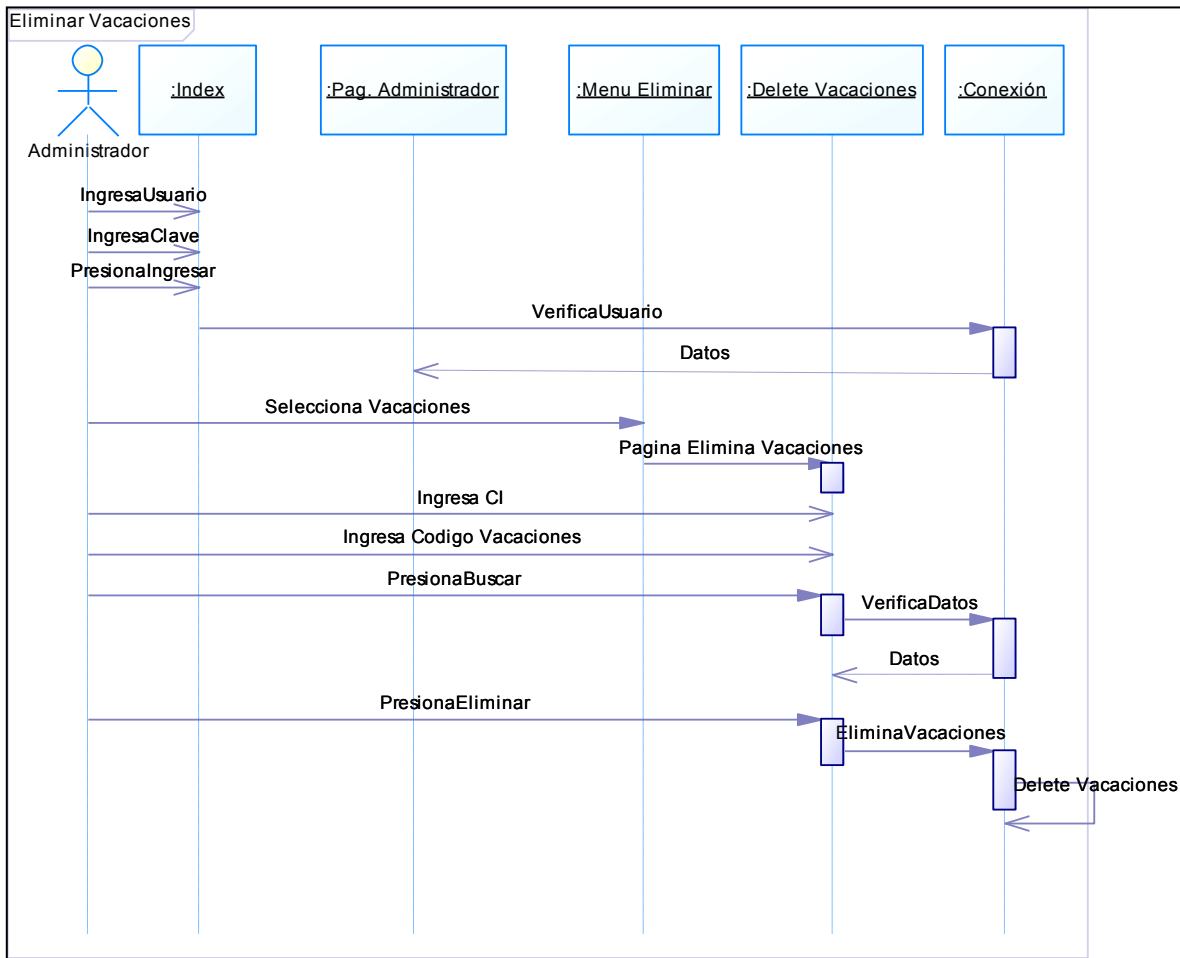
**Figura No: 38 Diagrama de Secuencia Ingresar Vacaciones**

**Caso de uso: Modificar Vacaciones**  
**Identificador: F6.2**



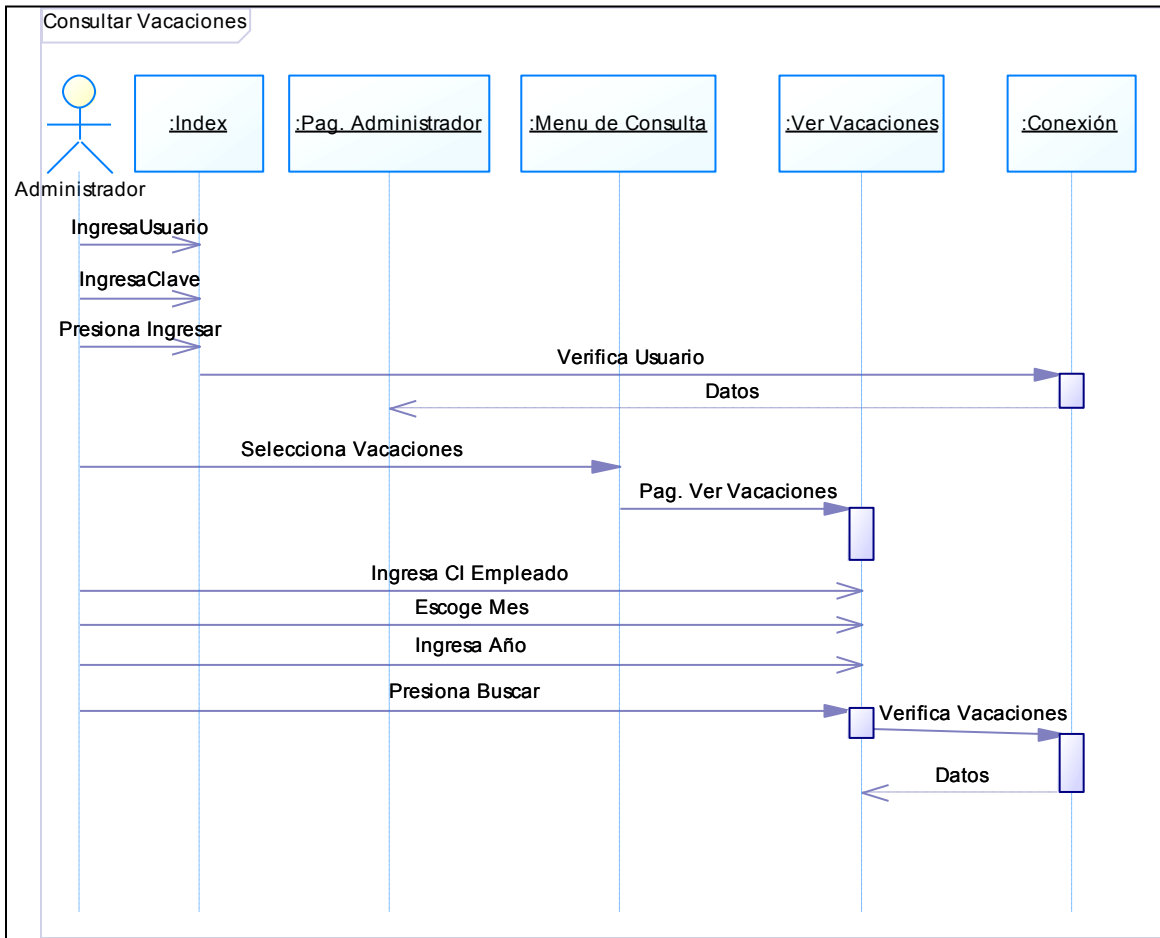
**Figura No: 39 Diagrama de Secuencia Modificar Vacaciones**

**Caso de uso: Eliminar Vacaciones**  
**Identificador: F6.3**



**Figura No: 40 Diagrama de Secuencia Eliminar Vacaciones**

**Caso de uso: Consultar Vacaciones**  
**Identificador: F6.4**



**Figura No: 41 Diagrama de Secuencia Consultar Vacaciones**

**Implementación**

En esta parte vamos a revisar la implementación del módulo Usuarios. En todos los módulos de la aplicación se verá reflejada la implementación del Patrón de Diseño Fachada el cual ayudará a que

las variables que se pasen de página a página no sean visibles al usuario llamando dentro de una interfaz visible a otra escondida que será la que realice las operaciones.

Toda la aplicación tiene código que determinará que los usuarios están registrados caso contrario no podrán hacer uso de la misma.

Por motivos de espacio solo presentaremos una parte del código, el resto se encuentra en el CD, dentro de la carpeta STAFFLOG1.

### Crear y Modificar Usuario:

En crear y modificar Usuario se visualizará el uso del Patrón de Diseño Estrategia al hacer el ingreso o modificación del usuario, para realizar cualquiera de estas dos acciones se presentara la misma interfaz y el usuario tendrá que ingresar los datos y la aplicación decidirá si hay que ingresar un nuevo usuario o simplemente modificarlo.

#### creusuario.php

```
<?php
//esta es la interfaz que verá el usuario para crear o modificar un usuario es decir la
fachada del patrón estrategia

//inicia la sesión
if (!isset($_SESSION)) {
    session_start();
}

// ** Cierra la sesión de usuario actual **
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true";
if ((isset($_SERVER['QUERY_STRING'])) && ($_SERVER['QUERY_STRING'] != "")){
    $logoutAction .="&". htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_GET['doLogout'])) &&($_GET['doLogout']=="true")){
    //se necesita limpiar todas las variables para poder cerrar la session completamente
    $_SESSION['MM_Username'] = NULL;
    $_SESSION['MM_UserGroup'] = NULL;
    $_SESSION['PrevUrl'] = NULL;
    unset($_SESSION['MM_Username']);
}
```

```

unset($_SESSION['MM_UserGroup']);
unset($_SESSION['PrevUrl']);

$logoutGoTo = "../index.php";
if ($logoutGoTo) {
    header("Location: $logoutGoTo");
    exit;
}
}
?>
<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "";
$MM_donotCheckaccess = "true";

// *** Restringe el acceso a la página: Conceder o denegar el acceso a esta página
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // Para mayor seguridad, comenzaremos asumiendo que el visitante no está
    autorizado
    $isValid = False;

    // Cuando un visitante se ha registrado en este sitio, la variable de sesión
    MM_Username igual a su nombre de usuario
    // Por lo tanto, sabemos que un usuario no está conectado si esa variable de sesión
    está en blanco.
    if (!empty($UserName)) {
        // Además de estar conectado, es posible restringir el acceso sólo a ciertos usuarios
        con base en una identificación establecido al iniciar la sesión.
        //Analiza las cadenas.
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        // o se puede restringir el acceso a uno o algunos usuarios.
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && true) {
            $isValid = true;
        }
    }
}
return $isValid;
}

```

```

$MM_restrictGoTo = "../index.php";
if (!(isset($_SESSION['MM_Username'])) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)
        $MM_referrer .= "?" . $_QUERY_STRING;
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
    header("Location: ". $MM_restrictGoTo);
    exit;
}
?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
            break;
        case "double":
            $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
            break;
        case "date":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "defined":
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
            break;
    }
    return $theValue;
}
}
}

```

```

$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
}
//en la creación del html se harán las llamadas a los formatos de cabecera, pie y
menus
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"[]>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xml:lang="en">
<head>
    <!--
    Created by Artisteer v3.1.0.45994
    Base template (without user's data) checked by http://validator.w3.org : "This page
is valid XHTML 1.0 Transitional"
    -->
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <?php include_once '../lib/titulo.php'; ?>

    <link rel="stylesheet" type="text/css" media="screen"href="../lib/css/style.css" />
    <script type="text/javascript" src="../lib/css/jquery.js"></script>
    <script type="text/javascript" src="../lib/css/script.js"></script>

</head>
<body>
<div id="art-main">
    <div class="cleared reset-box"></div>
    <?php include_once '../lib/cabecera.php'; ?>
    <div class="cleared reset-box"></div>
<div class="art-bar art-nav">
<div class="art-nav-outer">
<div class="art-nav-wrapper">
<?php include_once '../lib/menuhorizontal.php'; ?>
</div>
</div>
</div>
<div class="cleared reset-box"></div>
<div class="art-box art-sheet">
    <div class="art-box-body art-sheet-body">
        <div class="art-layout-wrapper">
            <div class="art-content-layout">

```

```

<div class="art-content-layout-row">
    <?php include_once './lib/menu.php'; ?>
    <div class="art-layout-cell art-content">
<div class="art-box art-post">
    <div class="art-box-body art-post-body">
<div class="art-post-inner art-article">
        <h2 class="art-postheader">Bienvenido Administrador -
USUARIOS</h2>
        <div class="art-postmetadataheader">
            <div class="art-postheadericons art-metadata-
icons"></div>
            </div>
        <div class="art-postcontent">
            <div id="content">
                <div class="post">
<?php //aquí se hace la llamada al php que hará el trabajo de insertar o modificar el
usuario ¿>
                <iframe frameborder="0" scrolling="auto" height="550" width="900"
src="./insertusu.php" align="center"></iframe>
                <p>&nbsp;</p>
            </div>
        </div>
        <p></p>
    </div>
    </div>
</div>
    </div>
    </div>
    </div>
    </div>
    </div>
    </div>
</body>
<?php include_once './lib/footer.inc.php'; ?>
</html>

```

## insertusu.php

```

<?php
//este php será el subsistema del patrón fachada aquí se realizarán las operaciones de
crear y modificar, también se hará la implementación del patrón estrategia
//inicia la sesión
if (!isset($_SESSION)) {
    session_start();
}

```

```

$MM_authorizedUsers = "";
$MM_donotCheckaccess = "true";

// *** Restringe el acceso a la página
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // Para mayor seguridad, comenzaremos asumiendo que el visitante no está
    autorizado
    $isValid = False;

    // Cuando un visitante se ha registrado en este sitio, la variable de sesión
    MM_Username igual a su nombre de usuario
    // Por lo tanto, sabemos que un usuario no está conectado si esa variable de sesión
    está en blanco.
    if (!empty($UserName)) {
        // Además de estar conectado, es posible restringir el acceso sólo a ciertos usuarios
        con base en una identificación establecido al iniciar la sesión.
        //Analiza las cadenas
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        //o se puede restringir el acceso a uno o algunos usuarios.
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && true) {
            $isValid = true;
        }
    }
    return $isValid;
}

$MM_restrictGoTo = "login.php";
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)
        $MM_referrer .= "?" . $_QUERY_STRING;
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
    header("Location: ". $MM_restrictGoTo);
    exit;
}

```

```

?>
<?php
    //llama al php de conexión a la base
    require_once('Connections/miconexion.php'); ?>
<?php
//realiza las consultas necesarias para crear la tabla de usuarios y poderes visualizar
if (!function_exists("GetSQLValueString")) {
function  GetSQLValueString($theValue,  $theType,  $theDefinedValue  =  "",
$theNotDefinedValue = "")
{
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

    switch ($theType) {
    case "text":
        $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
        break;
    case "long":
    case "int":
        $theValue = ($theValue != "") ? intval($theValue) : "NULL";
        break;
    case "double":
        $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
        break;
    case "date":
        $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
        break;
    case "defined":
        $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
        break;
    }
    return $theValue;
}
}
}
mysql_select_db($database_miconexion, $miconexion);
$query_Recordset3 = "SELECT IDUSUARIO FROM usuario";
$Recordset3 = mysql_query($query_Recordset3, $miconexion) or die(mysql_error());
$row_Recordset3 = mysql_fetch_assoc($Recordset3);
$totalRows_Recordset3 = mysql_num_rows($Recordset3);
$dato=0;
do{
$dato=$dato+1;
}while ($row_Recordset3 = mysql_fetch_assoc($Recordset3));

```

```

$currentPage = $_SERVER["PHP_SELF"];

$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
}

//si el usuario presiona insertar la aplicacion ejecutara los sql siguientes con los datos
//pasados por el botón insertar
if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
//identifica si el Id de usuario ya esta en la base o no (esto es para el patron
//estrategia)
mysql_select_db($database_miconexion, $miconexion);
$consulta="select * from usuario where IDUSUARIO =".$_POST['IDUSUARIO'];
$resultado=mysql_query($consulta) or die ("ERROR1!!!");
//si es usuario está en la base actualiza el registro con los nuevos valores
if (mysql_num_rows($resultado)>0)
{
    $updateSQL = "UPDATE usuario SET USERUSUARIO =
".$_POST['USERUSUARIO'].", PASSWORDUSUARIO = ".$_POST['PASSUSU']."
WHERE IDUSUARIO =".$_POST['IDUSUARIO'];
    mysql_select_db($database_miconexion, $miconexion);
    $Result1 = mysql_query($updateSQL, $miconexion) or die(mysql_error());

    $updateSQL = "UPDATE categoriausuario SET DETALLECATEGORIA =
".$_POST['cat']." WHERE IDUSUARIO =".$_POST['IDUSUARIO'];
    mysql_select_db($database_miconexion, $miconexion);
    $Result1 = mysql_query($updateSQL, $miconexion) or die(mysql_error());

//si el usuario no esta en la base agrega un usuario nuevo
}else {$insertSQL = sprintf("INSERT INTO usuario (IDUSUARIO, USERUSUARIO,
PASSWORDUSUARIO) VALUES (%s, %s, %s)",
    GetSQLValueString($_POST['IDUSUARIO'], "text"),
    GetSQLValueString($_POST['USERUSUARIO'], "text"),
    GetSQLValueString($_POST['PASSUSU'], "text"));

mysql_select_db($database_miconexion, $miconexion);
$Result1 = mysql_query($insertSQL, $miconexion) or die("ERROR2!!!");

$insertSQL = sprintf("INSERT INTO categoriausuario (IDUSUARIO,
DETALLECATEGORIA) VALUES (%s, %s)",
    GetSQLValueString($_POST['IDUSUARIO'], "text"),
    GetSQLValueString($_POST['cat'], "text"));

mysql_select_db($database_miconexion, $miconexion);

```

```

$Result1 = mysql_query($insertSQL, $miconexion) or die(mysql_error());
}

$insertGoTo = "insertusu.php";
if (isset($_SERVER['QUERY_STRING'])) {
    $insertGoTo .= (strpos($insertGoTo, '?')) ? "&" : "?";
    $insertGoTo .= $_SERVER['QUERY_STRING'];

    header(sprintf("Location: %s", $insertGoTo));
}
//determina la cantidad de usuarios que se van a mostrar en la tabla
$maxRows_Recordset1 = 5;
$pageNum_Recordset1 = 0;
if (isset($_GET['pageNum_Recordset1'])) {
    $pageNum_Recordset1 = $_GET['pageNum_Recordset1'];
}
$startRow_Recordset1 = $pageNum_Recordset1 * $maxRows_Recordset1;

mysql_select_db($database_miconexion, $miconexion);
$query_Recordset1 = "SELECT * FROM usuario";
$query_limit_Recordset1 = sprintf("%s LIMIT %d, %d", $query_Recordset1,
$startRow_Recordset1, $maxRows_Recordset1);
$Recordset1 = mysql_query($query_limit_Recordset1, $miconexion) or
die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);

if (isset($_GET['totalRows_Recordset1'])) {
    $totalRows_Recordset1 = $_GET['totalRows_Recordset1'];
} else {
    $all_Recordset1 = mysql_query($query_Recordset1);
    $totalRows_Recordset1 = mysql_num_rows($all_Recordset1);
}
$totalPages_Recordset1 = ceil($totalRows_Recordset1/$maxRows_Recordset1)-1;

$queryString_Recordset1 = "";
if (!empty($_SERVER['QUERY_STRING'])) {
    $params = explode("&", $_SERVER['QUERY_STRING']);
    $newParams = array();
    foreach ($params as $param) {
        if (strpos($param, "pageNum_Recordset1") == false &&
            strpos($param, "totalRows_Recordset1") == false) {
            array_push($newParams, $param);
        }
    }
}
if (count($newParams) != 0) {

```

```

$queryString_Recordset1 = "&" . htmlentities(implode("&", $newParams));
}
}
$queryString_Recordset1          =          sprintf("&totalRows_Recordset1=%d%s",
$totalRows_Recordset1, $queryString_Recordset1);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<link rel="SHORTCUT ICON" href="images/30m.ico">
<head>
<script language='javascript' src="Scripts/popcalendar.js"> </script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<script type="text/javascript">
<!--
function MM_validateForm() { //v4.0
  if (document.getElementById){
    var i,p,q,nm,test,num,min,max,errors="",args=MM_validateForm.arguments;
    for      (i=0;      i<(args.length-2);      i+=3)      {      test=args[i+2];
val=document.getElementById(args[i]);
    if (val) { nm=val.name; if ((val=val.value)!="") {
      if (test.indexOf('isEmail')!=-1) { p=val.indexOf('@');
        if (p<1 || p==(val.length-1)) errors+='- '+nm+' must contain an e-mail
address.\n';
      } else if (test!='R') { num = parseFloat(val);
        if (isNaN(val)) errors+='- '+nm+' must contain a number.\n';
        if (test.indexOf('inRange') != -1) { p=test.indexOf(':');
          min=test.substring(8,p); max=test.substring(p+1);
          if (num<min || max<num) errors+='- '+nm+' must contain a number
between '+min+' and '+max+'.\n';
        } } } else if (test.charAt(0) == 'R') errors += '- '+nm+' is required.\n'; }
    } if (errors) alert('The following error(s) occurred:\n'+errors);
    document.MM_returnValue = (errors == "");
  } }
//-->
</script>
</head>
<body>
<div id="id">
<form action="<?php echo $editFormAction; ?>"enctype="multipart/form-data"
method="post" id="form1">
  <!--
onmouseover="MM_validateForm('USERUSUARIO','R','PASSWORDUSUARIO','R','DIRUSU',
'R','FECHANACIUSU','R','TELUSU','RisNum','MAILUSU','RisEmail','NICKUS
U','R','PASSUSU','R');return document.MM_returnValue">-->

```

```

<input type="hidden" name="MM_insert" value="form1" />
</p>
<?php //tabla para ingresar datos de usuario ?>
<table align="left" width="400" height="200">
  <tr valign="baseline">
    <td width="193" align="right">Id de Usuario:</td>
    <td width="195"><input name="IDUSUARIO" type="text"
id="IDUSUARIO" value="" size="32" /></td>
  </tr>
  <tr valign="baseline">
    <td align="right">Nombre de Usuario:</td>
    <td><input name="USERUSUARIO" type="text" id="USERUSUARIO"
value="" size="32" /></td>
  </tr>

  <tr valign="baseline">
    <td align="right">Password:</td>
    <td><input name="PASSUSU" type="text" id="PASSUSU" value=""
size="32" /></td>
  </tr>
  <tr valign="baseline">
    <td align="right">Categoría:</td>
    <td><select name="cat">
      <option value="Administrador">Administrador</option>
      <option value="Tecnico">Tecnico</option>
      <option value="Usuario">Usuario</option>

    </select>
  </td>
  </tr>
  <tr valign="baseline">
    <td align="right">&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr valign="baseline">
    <td align="right">&nbsp;</td>
    <td><input type="submit" value="Insertar" /></td>
  </tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>

```

```

<table border="2" cellpadding="2" cellspacing="1">
  <tr>
    <td><strong>ID</strong></td>
    <td><strong>Nombre</strong></td>
    <td><strong>Contraseña</strong></td>

  </tr>
  <?php do { ?>
  <?php //tabla para presentar los usuarios que estan en la base ?>
  <tr>
    <td><?php echo $row_Recordset1['IDUSUARIO']; ?></td>
    <td><?php echo $row_Recordset1['USERUSUARIO']; ?></td>
    <td><?php echo $row_Recordset1['PASSWORDUSUARIO']; ?></td>

  </tr>
  <?php } while ($row_Recordset1 = mysql_fetch_assoc($Recordset1)); ?>
</table>
<?php //tabla para ignresar las flechas de navegacion de la tabla de
usuarios ?>
<table border="0">
  <tr>
    <td><?php if ($pageNum_Recordset1 > 0) { ?>
      <a href="<?php printf("%s?pageNum_Recordset1=%d%s",
$currentPage, 0, $queryString_Recordset1); ?>"></a>
      <?php }?>
    </td>
    <td><?php if ($pageNum_Recordset1 > 0) ?>
      <a href="<?php printf("%s?pageNum_Recordset1=%d%s",
$currentPage, max(0, $pageNum_Recordset1 - 1), $queryString_Recordset1);
?>"></a>
      <?php }?>
    </td>
    <td><?php if ($pageNum_Recordset1 < $totalPages_Recordset1) {?>
      <a href="<?php printf("%s?pageNum_Recordset1=%d%s",
$currentPage, min($totalPages_Recordset1, $pageNum_Recordset1 + 1),
$queryString_Recordset1); ?>"></a>
      <?php }?>
    </td>
    <td><?php if ($pageNum_Recordset1 < $totalPages_Recordset1) {?>
      <a href="<?php printf("%s?pageNum_Recordset1=%d%s",
$currentPage, $totalPages_Recordset1, $queryString_Recordset1); ?>"></a>
      <?php }?>
    </td>
  </tr>

```

```
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
</form>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</div>
</body>
</html>
<?php
mysql_free_result($Recordset1);
?>
```

## STAFFLOG Software

La aplicación completa está en la carpeta stafflog1 dispuesta de la siguiente manera:

- En la raíz de la carpeta se encuentran los archivos interfaces ocultas que realizan las operaciones de los Casos de Uso.
- **Carpeta Connctions:** En la que se encuentra el archivo de conexión a la base.
- **Carpeta images:** En la que se encuentran todas las imágenes que utiliza la aplicación.
- **Carpeta lib:** En la que se encuentran los archivos Menús, Encabezado y Pie de Página, Título y Calendarios.
- **Carpeta plantillas:** En la que se encuentran los archivos de la Creación de los Archivos Word para Contratos y Acción de Personal.
- **Carpeta vista:** En la que se encuentran los archivos de creación de interfaces principales para cada Caso de Uso.

### 3. REQUERIMIENTOS E INSTALACION

#### Requerimientos

- Sistema operativo: Windows Vista o versiones posteriores.
- Tener instalado **Google Chrome** versión 26.0.1410.43 m o posteriores.
- Tener Instalado el Servidor **XAMPP**.

#### Instalación

1. Para Instalar STAFFLOG Software:
2. Copiar la carpeta **stafflog1** en la siguiente dirección:  
C:\xampp\htdocs.
3. Crear la Base de datos con el archivo llamado **maternidad1.sql** desde PHPMYADMIN.
4. Editar el archivo **respald.bat** (archivo para actualizar automáticamente la base de datos) con los datos de usuario y password de su servidor.
5. Crear una tarea programada semanalmente que haga el llamado al archivo **respald.bat**.
6. Ingresar a STAFFLOG Software como ADMINISTRADOR

### 4. INGRESO COMO ADMINISTRADOR

1. Ingresar a Google Chrome.
2. En la barra de direcciones escribir: <http://localhost/stafflog1/> se presentara la siguiente ventana.



**Figura No: 42 Página de Ingreso como Administrador**

3. Ingresar los datos de Username : administrador y Password: ad123 (estos datos se pueden modificar dentro de la base de datos) la creación de usuarios puede ser vista en el Caso de Uso F0.

STAFFLOG Software v 1.0  
Gabriela Rivera [magabytarv@hotmail.com](mailto:magabytarv@hotmail.com)  
Gabriela Vega [gabyvega\\_8@hotmail.com](mailto:gabyvega_8@hotmail.com)