

18/06/2025



## PROYECTO DE DESARROLLO DE SOFTWARE

---

IVAN PALE ARENAS

MARTINEZ GARCIA ANGEL

# Documento Diseño

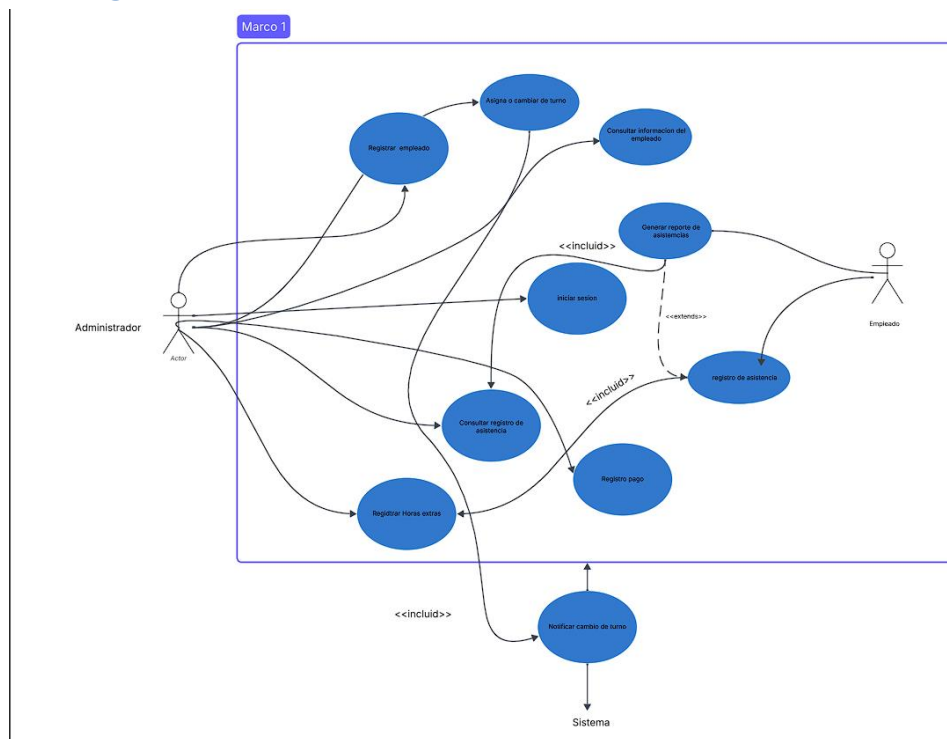
## Sistema: Gestor de Empleados

### 1. Introducción

#### 1.1 Propósito

El propósito de este documento es mostrar los diagramas que se realizaron respecto al software que creamos y la documentación ya obtenida el documento SRS para que se tenga una idea mas clara respecto a la funcionalidad y si terceros llegan a utilizar el software puedan saber cuales son nuestros objetivos y que es lo que hacemos.

#### 1.2 Diagrama de Casos de uso



### Explicación del Diagrama de Casos de Uso – Gestor de Empleados

#### Actores principales:

- **Administrador:** Responsable de gestionar empleados, turnos, reportes y pagos.

- **Empleado:** Registra su asistencia, consulta su historial, solicita turnos o trabaja horas extra.
- **Sistema:** Realiza acciones automáticas como enviar notificaciones.

#### Casos de uso principales:

##### 1. Iniciar sesión

- Acción requerida para acceder al sistema por parte de cualquier usuario (administrador o empleado).

##### 2. Registrar empleado

- Caso de uso exclusivo del administrador para dar de alta nuevos trabajadores en el sistema.

##### 3. Registro de asistencia

- El empleado marca su hora de entrada y salida.
- Tiene una relación de tipo <<extend>> con **Registrar horas extra**, ya que esta opción solo se activa si el empleado trabaja fuera del horario normal.

##### 4. Registrar horas extra

- Se ejecuta cuando un empleado trabaja tiempo adicional después de su jornada regular.
- Este caso de uso extiende a **Registro de asistencia**.

##### 5. Consultar registro de asistencia

- Puede ser realizado por el empleado para ver su historial o por el administrador para análisis.

##### 6. Generar reporte de asistencias

- Caso de uso del administrador.
- Tiene una relación <<include>> con **Consultar registro de asistencia**, ya que debe consultar los datos antes de generar el reporte.

##### 7. Asignar o cambiar turno

- El administrador modifica el horario asignado a un empleado.
- Incluye el caso de uso **Notificar cambio de turno**, ya que al modificar el turno, el sistema debe enviar una notificación.

## 8. Notificar cambio de turno

- Acción automática realizada por el sistema cuando se asigna o cambia un turno.

## 9. Consultar información del empleado

- El administrador visualiza datos como historial, turnos, rendimiento, etc.

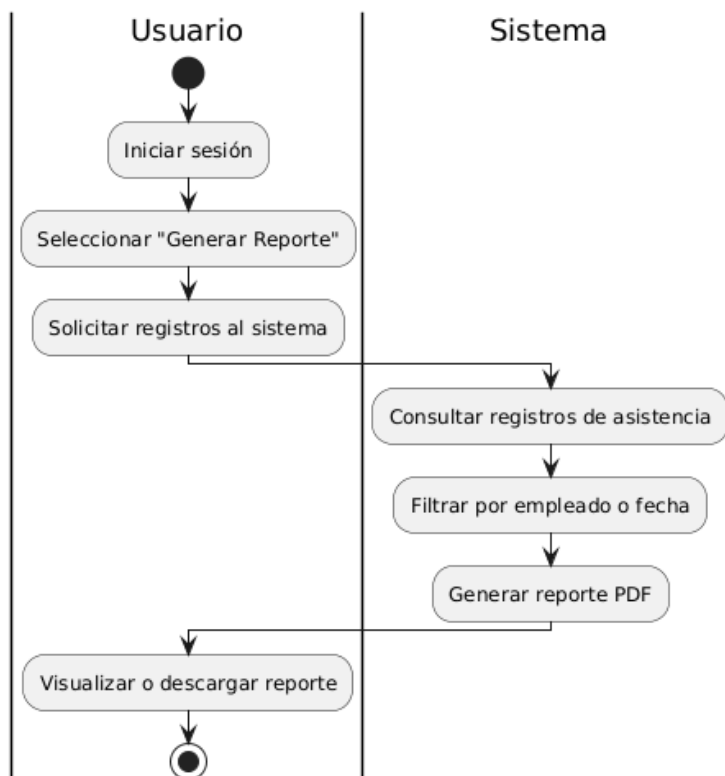
## 10. Registro de pago

- El administrador calcula y registra el pago del empleado con base en su asistencia y horas extra.

### Relación entre casos de uso:

- <<include>>: El caso de uso depende de otro para poder ejecutarse siempre.
- <<extend>>: El caso de uso adicional se activa solo si se cumplen ciertas condiciones.

## 2. Diagrama bpmn de Generar Reporte de Asistencias – BPMN



Este diagrama muestra cómo un **usuario (administrador)** inicia sesión, selecciona la opción para generar reportes y cómo el **sistema** realiza la consulta de registros para generar el archivo correspondiente.

**Flujo:**

1. El usuario inicia sesión.
2. Selecciona la opción de "Generar Reporte".
3. El sistema busca los registros de asistencia.
4. Se filtran los resultados (por fecha, empleado, etc.).
5. El sistema genera y muestra el reporte (posiblemente como PDF o en pantalla).
6. El usuario puede visualizar o descargar el archivo.

**Propósito:**

Permitir al administrador tener un control visual o descargable del comportamiento laboral de los empleados.

## 2.1 Registrar Asistencia – BPMN simulado



Simula el proceso diario que realiza un empleado al marcar su asistencia, tanto de entrada como de salida, e incluye la posibilidad de registrar **horas extra** si corresponde.

### Flujo:

1. El empleado inicia sesión.
2. Registra la hora de entrada.
3. Realiza su jornada laboral.
4. Registra la hora de salida.
5. Si trabajó tiempo adicional, registra las horas extra.

### Propósito:

Llevar el control de asistencia de forma digital, precisa y flexible.

## 2.2 Registrar Empleado – BPMN simulado



Este flujo es usado por el administrador cuando se da de alta a un nuevo empleado en el sistema.

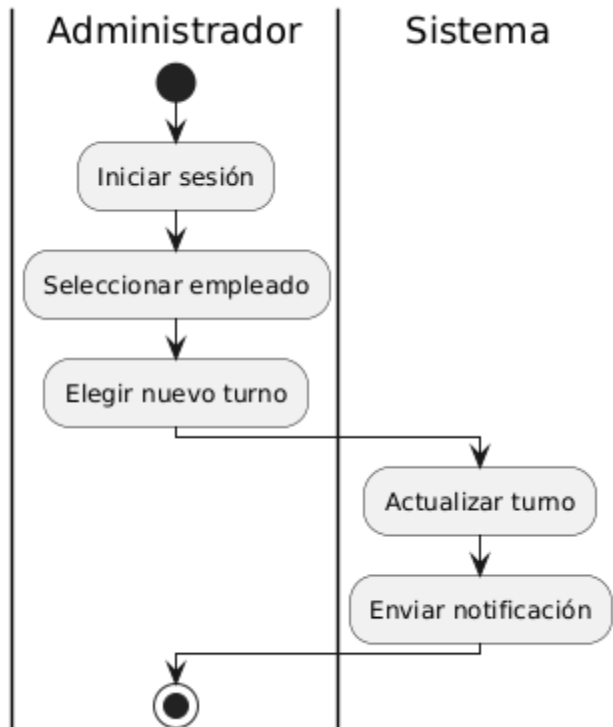
### Flujo:

1. El administrador inicia sesión.
2. Accede al módulo de gestión de empleados.
3. Captura los datos del nuevo empleado (nombre, correo, turno, etc.).
4. Valida la información.
5. El sistema guarda al nuevo empleado en la base de datos.

### Propósito:

Agregar nuevos empleados al sistema para que puedan usarlo y registrar su asistencia.

### 2.3 Asignar o Cambiar Turno – BPMN simulado



Describe cómo el administrador modifica o asigna un turno a un empleado, y cómo el sistema automáticamente le notifica del cambio.

**Flujo:**

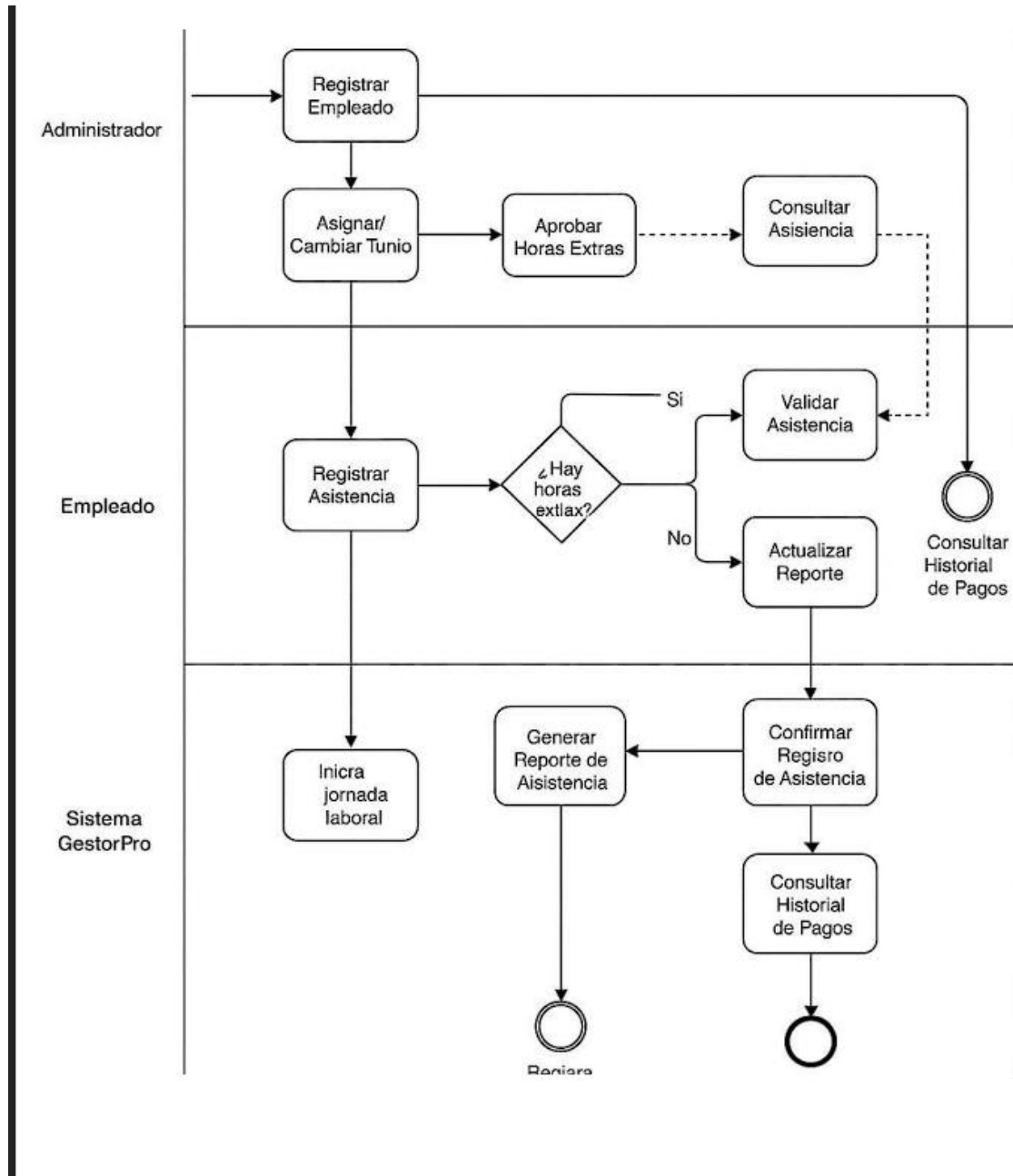
1. El administrador inicia sesión.
2. Selecciona un empleado.
3. Asigna o cambia su turno.
4. El sistema actualiza el turno en la base de datos.
5. Envía una notificación al empleado con la información del cambio.

**Propósito:**

Asegurar que los turnos sean asignados correctamente y que el empleado esté informado.



## 2.4 Diagrama BPMN general del sistema



### Administrador

- 1. Registrar empleado**  
Se da de alta un nuevo empleado.
- 2. Asignar / cambiar turno**  
Se le asigna su horario laboral.

3. **Aprobar horas extras**

Si un empleado registra horas fuera del horario normal, el administrador aprueba o rechaza.

4. **Consultar asistencia**

Consulta registros para análisis o control.

**Empleado**

5. **Registrar asistencia**

El empleado marca su entrada/salida.

6. **¿Hay horas extra?**

Si las hay, se valida la asistencia como especial.

7. **Validar asistencia**

Se procesa con condiciones adicionales.

8. **Actualizar reporte**

Se actualiza el registro del día.

9. **Consultar historial de pagos**

El empleado revisa cuánto ha cobrado y cuándo.

**Sistema GestorPro**

10. **Iniciar jornada laboral**

El sistema da por iniciada la jornada con la marca de entrada.

11. **Generar reporte de asistencia**

Se prepara el reporte para nómina o control.

12. **Confirmar registro de asistencia**

Guarda el registro final del día.

13. **Consultar historial de pagos**

Sistema proporciona al empleado su historial.

**Propósito del diagrama:** Mostrar cómo interactúan **los tres actores** (administrador, empleado, sistema) en el proceso completo desde el **registro del empleado** hasta la **generación de reportes y consulta de pagos**, incluyendo decisiones como si hay o no horas extra.



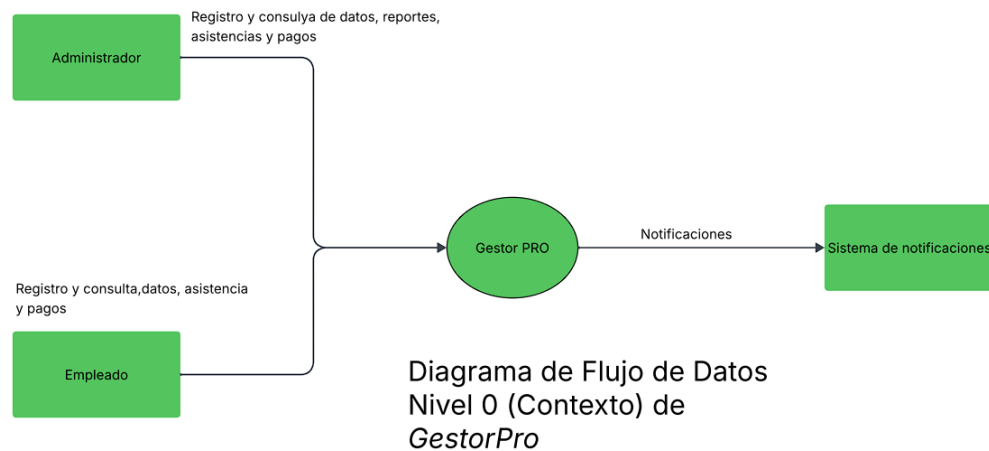
### 3. Diagrama de Flujo de Datos (DFD) Nivel 0

Este diagrama representa la vista general del sistema GestorPro y cómo se relaciona con los actores externos. Muestra el flujo de información entre el sistema principal y los usuarios que lo utilizan.

El sistema GestorPro recibe información del Administrador y del Empleado, quienes pueden registrar y consultar datos relacionados con asistencias, reportes y pagos.

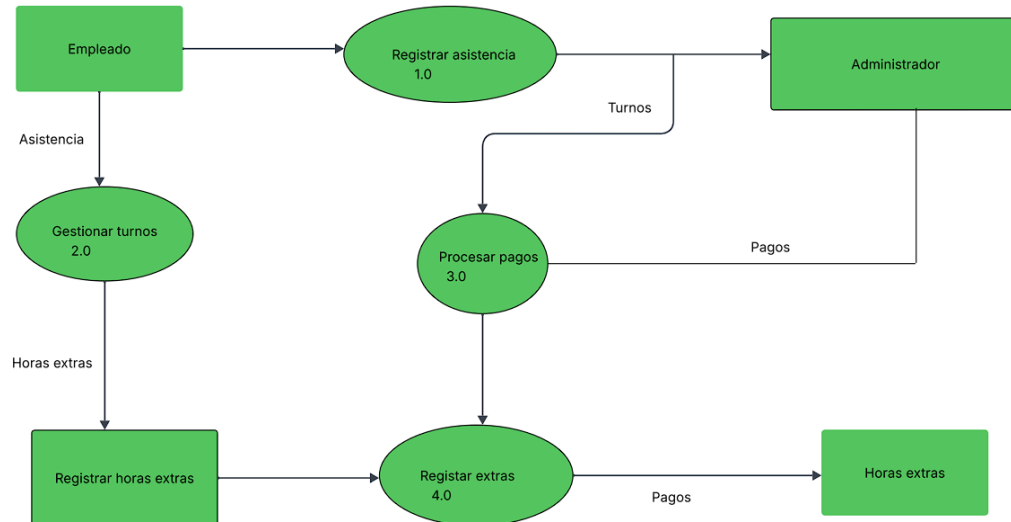
Por otro lado, el sistema también se comunica con un Sistema de notificaciones, al cual le envía información para generar alertas o avisos automáticos (por ejemplo, avisar un cambio de turno o confirmar un registro).

Este diagrama se considera de Nivel 0 porque no entra en detalles internos del sistema, sino que muestra solo las conexiones externas más importantes. Sirve para entender qué actores interactúan con el sistema y qué tipo de datos se intercambian.



### Diagrama de Flujo de Datos nivel 1

Diagrama de Flujo de Datos  
Nivel 1 (Contexto) de  
*GestorPro*



Este diagrama muestra una vista más detallada del funcionamiento interno del sistema GestorPro. A diferencia del Nivel 0, aquí se representan los **procesos principales dentro del sistema** y cómo fluye la información entre ellos.

Los actores externos siguen siendo el **Empleado**, el **Administrador** y el módulo de **Horas extras**, pero ahora se detallan **cuatro procesos internos numerados**:

1. **Registrar asistencia (1.0):**

El empleado registra su entrada o salida. Esta información se envía tanto al administrador como al proceso de pagos.

2. **Gestionar turnos (2.0):**

Se administra el control de turnos. Usa datos de asistencia y permite asignar o cambiar horarios según la disponibilidad.

3. **Procesar pagos (3.0):**

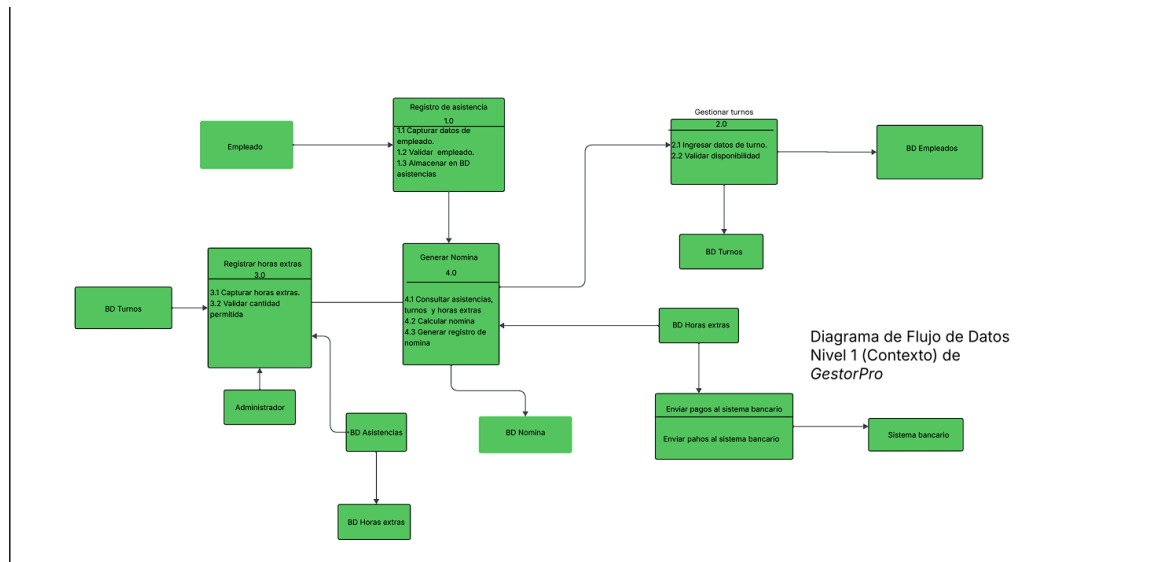
Calcula el pago correspondiente en función de la asistencia y los turnos registrados. También se comunica con el administrador.

4. **Registrar extras (4.0):**

Registra las horas extra trabajadas, las cuales también son tomadas en cuenta para completar el pago del empleado. Este proceso recibe datos desde el módulo de **Registrar horas extras**.

Este diagrama permite entender con mayor precisión **cómo están organizados internamente los procesos del sistema y cómo se relacionan entre sí**, ayudando a identificar puntos clave para el diseño y desarrollo de cada módulo.

## Diagram de Flujo de Datos nivel 2



Este diagrama muestra los procesos detallados dentro del sistema **GestorPro**, con las interacciones entre los actores (Empleado y Administrador) y los procesos internos del sistema. A diferencia del Nivel 0, aquí se visualiza cómo cada proceso se descompone en subprocesos específicos.

Los actores siguen siendo:

- Empleado
- Administrador

Además, se muestra cómo el sistema interactúa con las **bases de datos** (BD de turnos, asistencia, empleados y horas extras) y con el **sistema bancario** para generar pagos.

### Flujos de datos y procesos internos:

#### 1. Empleado:

- **Registrar asistencia (1.0):**

El empleado registra su entrada y salida. Este proceso se descompone en subprocesos:

- Capturar datos de empleado
- Validar empleado
- Almacenar en la base de datos de asistencias

- 2. **Gestionar turnos (2.0):**

El administrador gestiona los turnos, asegurando que los datos sean correctos y que haya disponibilidad. La información se almacena en la BD de turnos.

- 3. **Registrar horas extras (3.0):**

El empleado registra las horas trabajadas fuera de su turno regular. Este proceso se valida en dos subprocesos:

- Capturar horas extras
- Validar la cantidad permitida de horas extras  
Los datos se almacenan en la BD de horas extras.

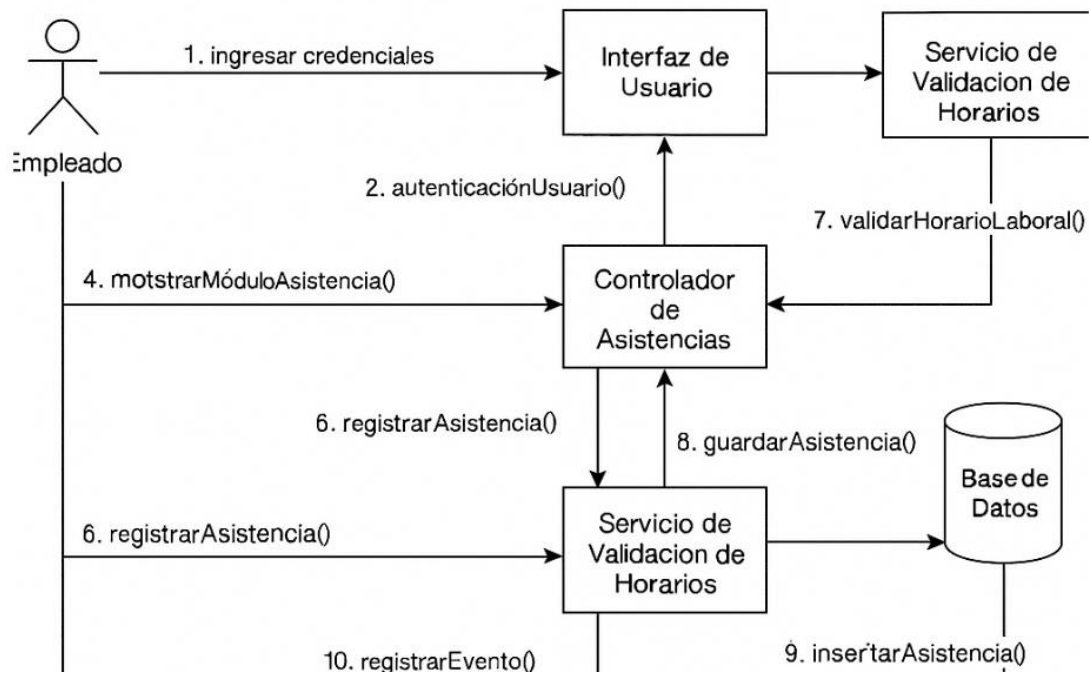
- 4. **Generar nómina (4.0):**

El administrador procesa la nómina, utilizando los datos de asistencia, turnos y horas extras:

- Consultar asistencias, turnos y horas extras
- Calcular la nómina
- Generar el registro de la nómina

## Diagrama de Comunicaciones UML 2 – Módulo de Asistencia en GestorPro

GestorPro - Diagrama de Comunicaciones UML 2



Este es un diagrama de comunicaciones UML, también conocido como diagrama de colaboración, que muestra cómo se comunican los diferentes componentes del sistema cuando un empleado registra su asistencia.

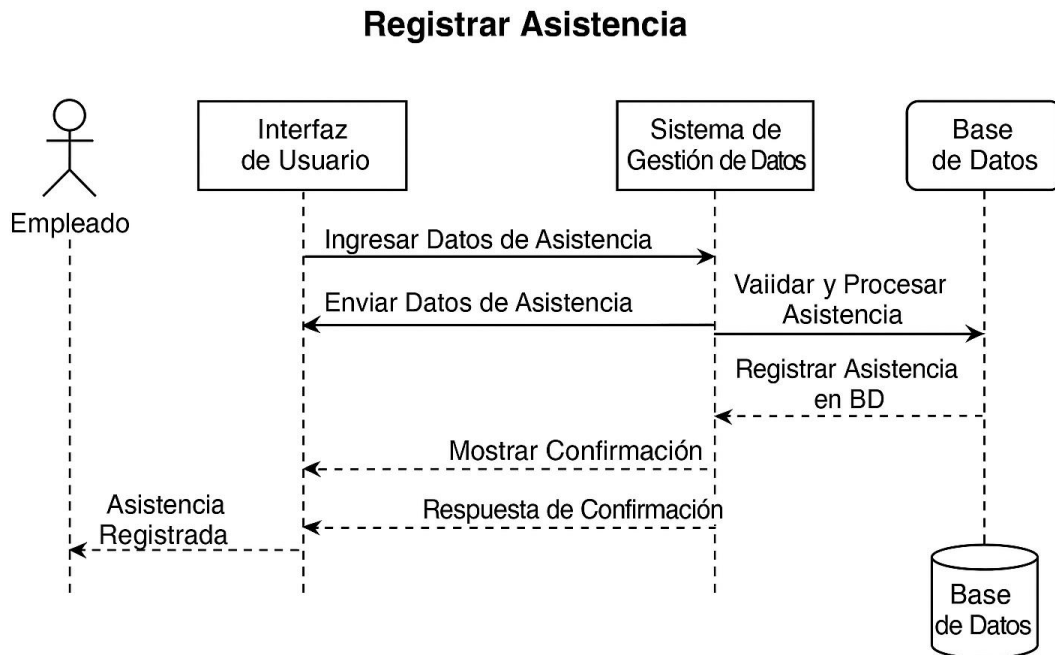
Descripción del flujo paso a paso:

1. El empleado ingresa sus credenciales en la interfaz de usuario.
2. La interfaz de usuario solicita la autenticación del usuario al sistema.
3. Una vez autenticado, el sistema permite que se muestre el módulo de asistencia al empleado.
4. El empleado accede al módulo de asistencia.
5. El empleado solicita el registro de asistencia.
6. El controlador de asistencias procesa esta solicitud y la pasa al Servicio de Validación de Horarios.
7. Este servicio se encarga de validar el horario laboral del empleado (por ejemplo, si está dentro del turno correspondiente).



8. Si es válido, el servicio indica al controlador que guarde la asistencia.
9. El sistema realiza la inserción del registro de asistencia en la base de datos.
10. Finalmente, se genera un evento de registro (por ejemplo, para una bitácora o log).

## Diagrama de secuencia



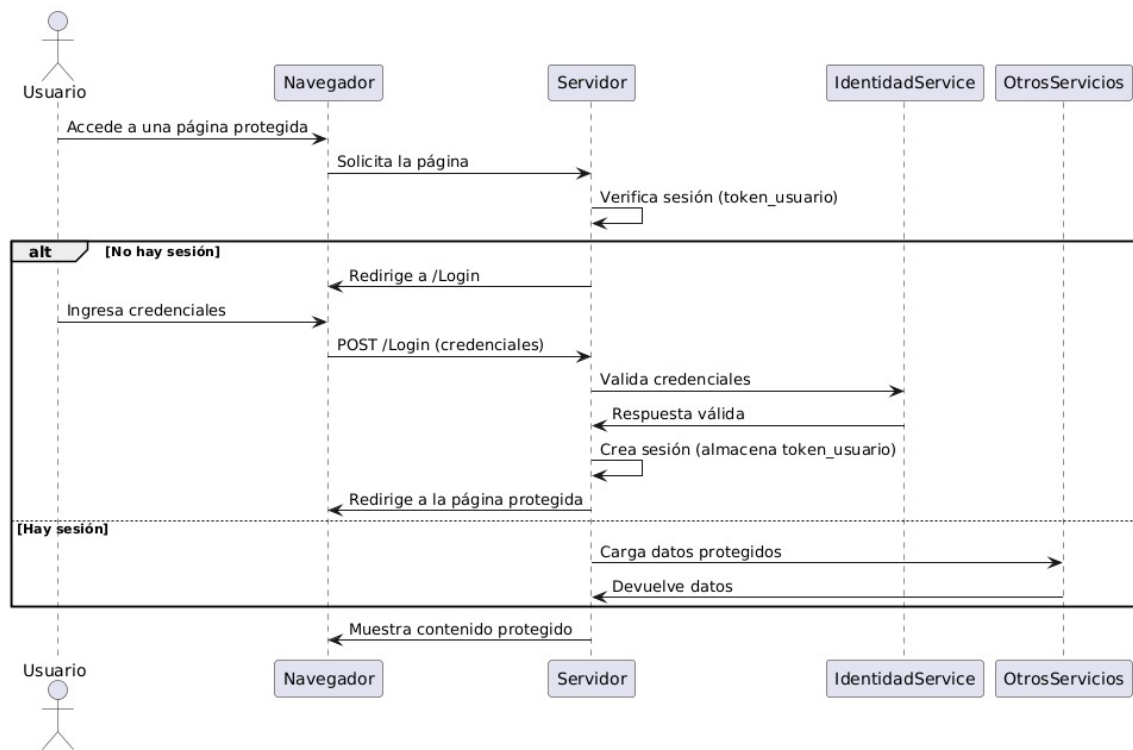
Este diagrama de secuencia muestra cómo se lleva a cabo el proceso de registrar la asistencia de un empleado dentro del sistema GestorPro, detallando el orden en que los componentes se comunican entre sí.

### Flujo del proceso:

1. El empleado inicia el proceso ingresando sus datos de asistencia a través de la interfaz de usuario.
2. La interfaz de usuario envía los datos al sistema de gestión de datos.
3. El sistema valida y procesa la información recibida.
4. Si todo es correcto, el sistema procede a registrar la asistencia en la base de datos.
5. Una vez guardada la información, la base de datos responde con una confirmación.

6. El sistema genera una respuesta de confirmación y la envía a la interfaz.
7. La interfaz muestra al empleado que la asistencia ha sido registrada correctamente.

### Diagrama de Secuencia – Proceso de Autenticación y Acceso a Página Protegida



Este diagrama de secuencia representa el flujo completo de cómo un usuario accede a una página protegida dentro de una aplicación web, y cómo el sistema gestiona la verificación de sesión, el inicio de sesión y la carga de datos protegidos.

#### Flujo del proceso:

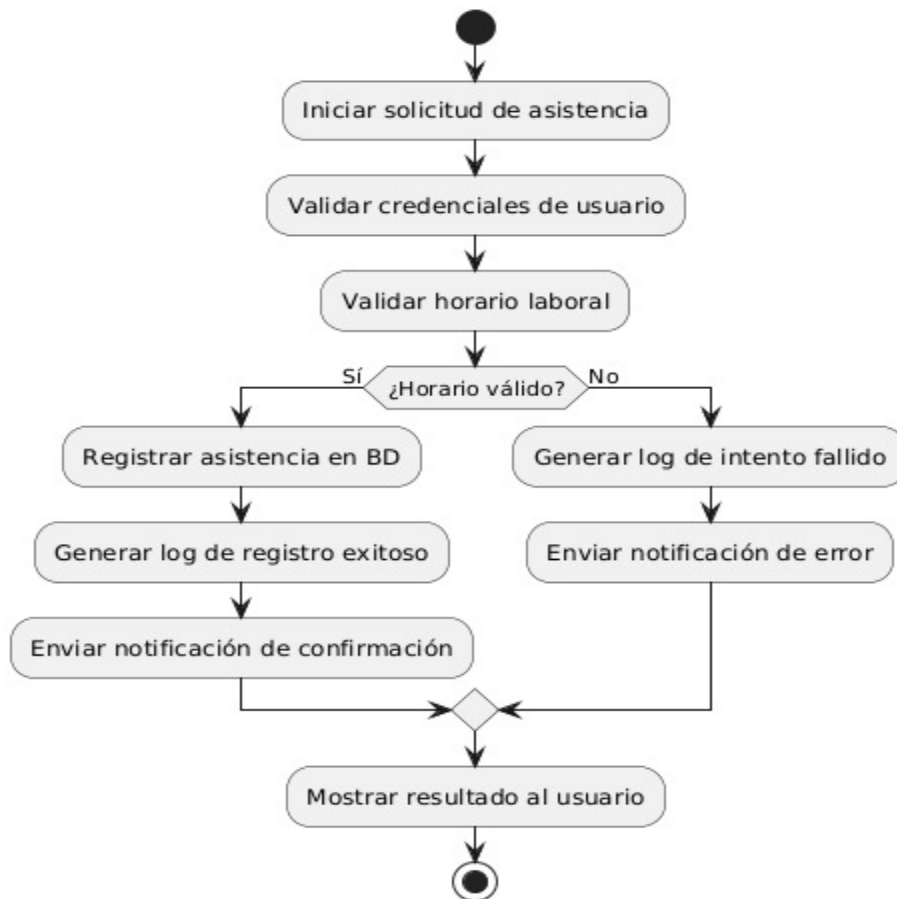
1. El usuario intenta acceder a una página protegida desde su navegador.
2. El navegador solicita la página al servidor.
3. El servidor verifica si hay una sesión activa revisando el token\_usuario.

#### Rama condicional:

- [No hay sesión]:
  - El servidor redirige al navegador hacia /Login.

- El usuario ingresa sus credenciales.
  - El navegador envía un POST con las credenciales al servidor.
  - El servidor valida las credenciales consultando el servicio de identidad.
  - Si la respuesta es válida, el servidor crea una nueva sesión (almacena el token del usuario).
  - Redirige al usuario a la página protegida.
- [Hay sesión]:
    - El servidor carga los datos protegidos desde otros servicios o bases de datos.
    - Devuelve los datos al navegador.
    - El navegador muestra el contenido protegido al usuario.

### Diagrama de Actividad – Registro de Asistencia



Este diagrama de actividad describe el flujo lógico del proceso para registrar una asistencia dentro del sistema, desde que el usuario inicia la solicitud hasta que se le muestra el resultado.

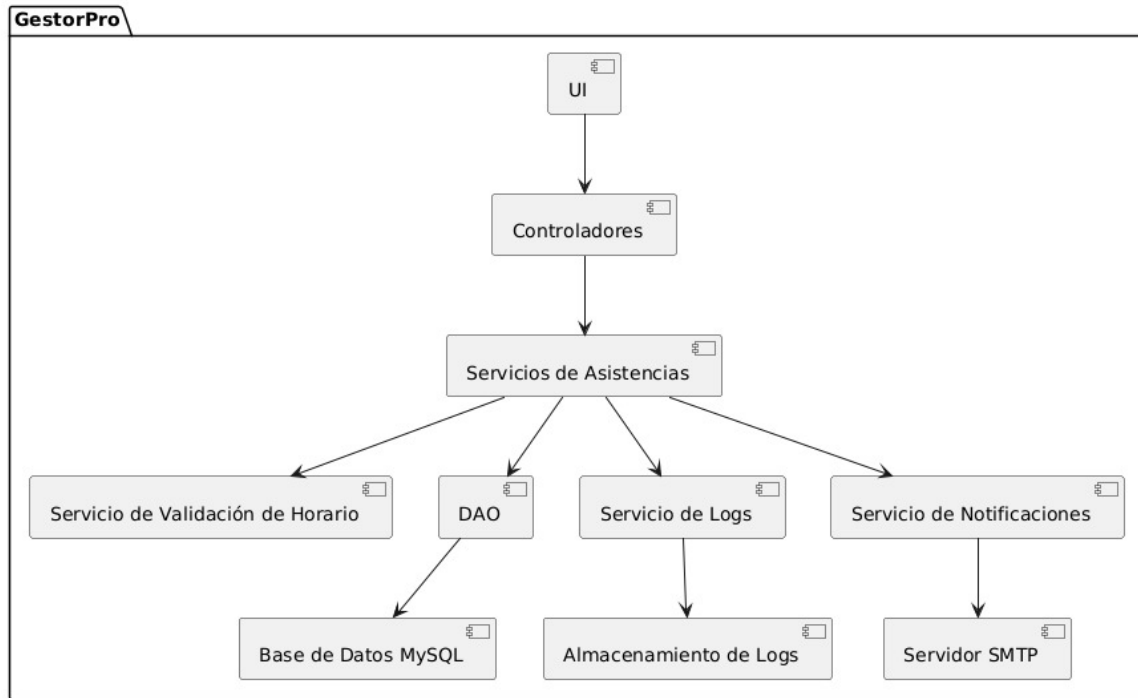
**Flujo paso a paso:**

1. Inicio del proceso:  
El usuario inicia la solicitud de asistencia (por ejemplo, al hacer clic en “Registrar entrada”).
2. Validar credenciales de usuario:  
El sistema comprueba que el usuario esté autenticado correctamente.
3. Validar horario laboral:  
Se verifica si la hora en la que se intenta registrar la asistencia está dentro del horario permitido.
4. Decisión: ¿Horario válido?
  - Si es válido:
    - Se registra la asistencia en la base de datos.
    - Se genera un log de éxito.
    - Se envía una notificación de confirmación al usuario.
  - Si no es válido:
    - Se genera un log de intento fallido.
    - Se envía una notificación de error al usuario (por ejemplo, "fuera de horario").
5. Mostrar resultado al usuario:  
Al final, el sistema muestra al usuario el resultado del proceso, sea exitoso o no.
6. Fin del proceso.

**Propósito del diagrama:**

Este diagrama ayuda a visualizar cómo se gestiona internamente el registro de asistencia en función de la validez del horario y las credenciales del usuario. Permite entender la lógica de negocio, identificar decisiones clave y los puntos de notificación o registro de eventos.

## Diagrama de Componentes – Módulo de Asistencias en GestorPro



Este es un diagrama de componentes UML, el cual representa cómo está estructurado internamente el sistema GestorPro para manejar el módulo de asistencias. Este tipo de diagrama muestra los componentes de software del sistema y cómo se relacionan entre sí.

### Estructura del sistema:

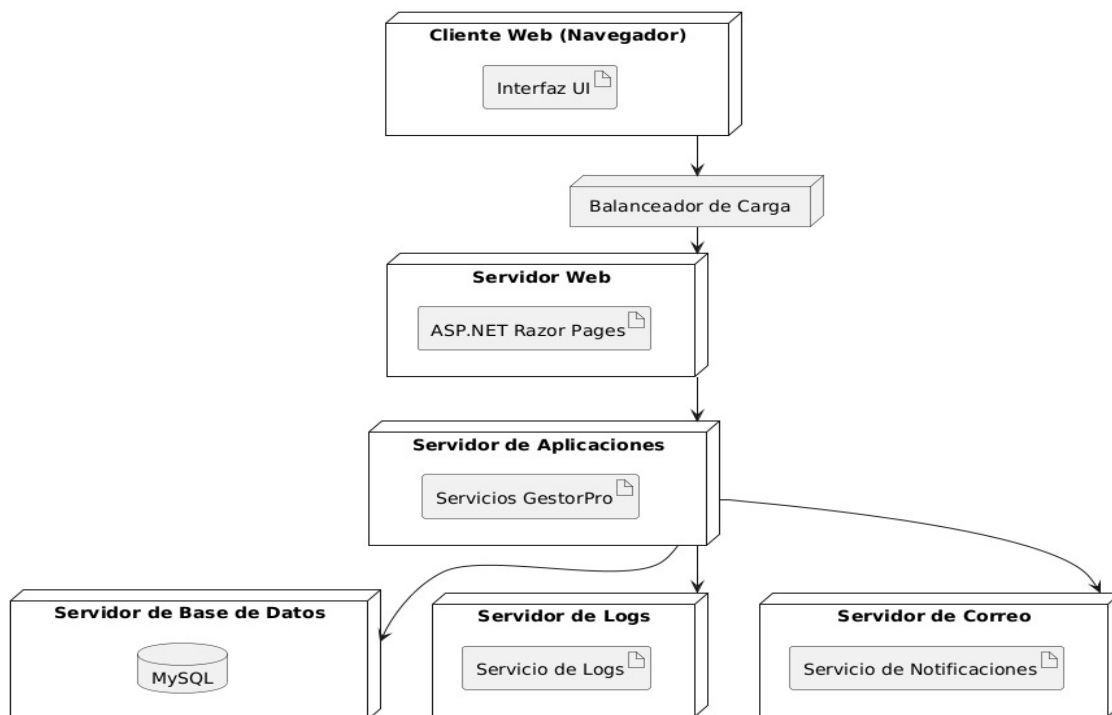
1. UI (Interfaz de Usuario):  
Es el punto de entrada del usuario. Permite interactuar con el sistema para registrar asistencias o consultar información.
2. Controladores:  
Reciben las acciones desde la UI y las procesan, canalizando las solicitudes hacia los servicios necesarios.
3. Servicios de Asistencias:  
Es el módulo central que coordina todo el proceso de registro de asistencias. Se comunica con los siguientes componentes:
  - Servicio de Validación de Horario:  
Verifica si la hora del registro está dentro del turno válido.

- DAO (Data Access Object):  
Encargado de acceder a la Base de Datos MySQL, donde se almacenan los registros de asistencia.
- Servicio de Logs:  
Registra eventos del sistema (por ejemplo, intentos exitosos o fallidos de asistencia).  
Se conecta con el componente de Almacenamiento de Logs.
- Servicio de Notificaciones:  
Envía confirmaciones o alertas por correo u otros medios.  
Usa un Servidor SMTP para el envío de correos.

### Propósito del diagrama:

Este diagrama permite entender la arquitectura modular del sistema GestorPro. Muestra cómo los diferentes componentes interactúan para ofrecer las funcionalidades del módulo de asistencias, asegurando validación, registro, trazabilidad (logs) y notificación automática.

### Diagrama de Despliegue – Arquitectura de GestorPro



Este es un diagrama de despliegue UML, que representa cómo se distribuyen los componentes del sistema GestorPro en los distintos servidores que forman parte de su infraestructura.

**Componentes y servidores involucrados:**

**1. Cliente Web (Navegador):**

- Representa al usuario accediendo desde un navegador.
- Utiliza la Interfaz UI (frontend del sistema).

**2. Balanceador de Carga:**

- Recibe las peticiones del cliente y las distribuye entre los servidores web para mejorar el rendimiento y la disponibilidad del sistema.

**3. Servidor Web:**

- Ejecuta la parte visual del sistema usando ASP.NET Razor Pages.
- Se encarga de mostrar las vistas y recibir solicitudes del usuario.

**4. Servidor de Aplicaciones:**

- Aloja los Servicios GestorPro, que contienen la lógica del sistema (como registro de asistencia, procesamiento de nómina, etc.).
- Este servidor se comunica con otros servicios externos para cumplir sus funciones.

**5. Servidor de Base de Datos:**

- Utiliza MySQL para almacenar datos como usuarios, asistencias, turnos y pagos.

**6. Servidor de Logs:**

- Contiene el Servicio de Logs, que registra eventos del sistema como accesos, errores o acciones críticas.

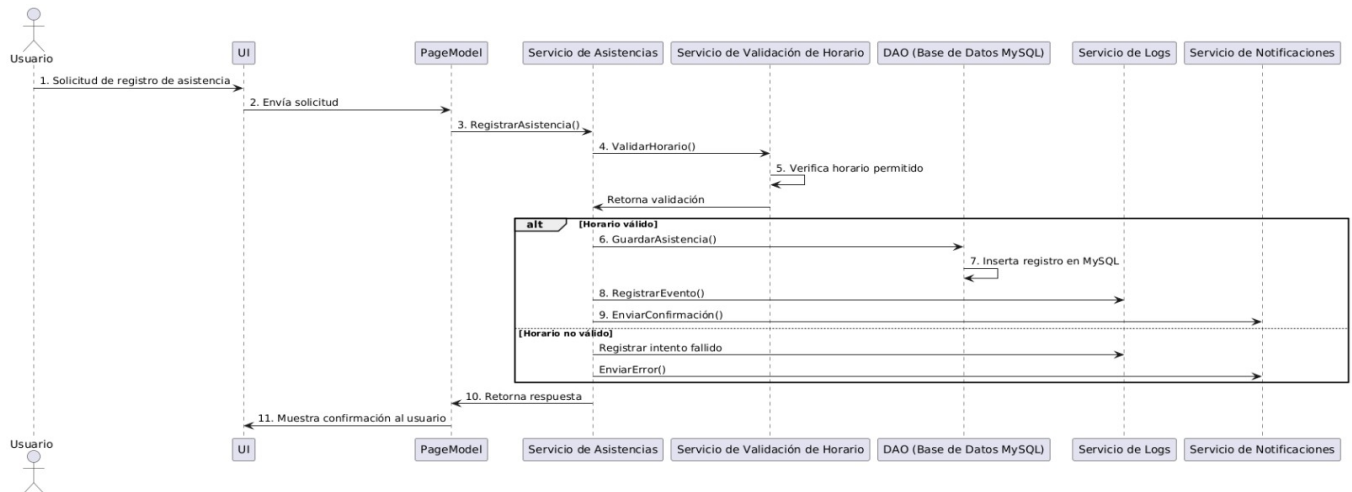
**7. Servidor de Correo:**

- Contiene el Servicio de Notificaciones, encargado de enviar correos electrónicos al usuario (por ejemplo, confirmaciones o alertas).

### Propósito del diagrama:

Este diagrama muestra la arquitectura física y lógica del sistema en producción, permitiendo entender cómo se conectan y distribuyen los componentes en distintos servidores. Es clave para tareas de despliegue, mantenimiento y escalabilidad del sistema.

## Diagrama de Secuencia – Registro de Asistencia con Validación de Horario



Este diagrama de secuencia muestra el flujo completo que sigue el sistema GestorPro para registrar una asistencia, incluyendo la validación del horario, el registro en base de datos, y el envío de notificaciones.

### Flujo paso a paso:

1. El usuario solicita registrar una asistencia desde la interfaz (UI).
  2. La UI envía la solicitud al PageModel, quien llama al método RegistrarAsistencia() del Servicio de Asistencias.
  3. Este servicio solicita al Servicio de Validación de Horario que valide si el horario actual es permitido.
  4. El Servicio de Validación consulta los datos necesarios y retorna la validación.
- **[Horario válido]:**
    - Se llama al método GuardarAsistencia() para registrar la asistencia.
    - El sistema utiliza el DAO (Base de Datos MySQL) para insertar el registro.
    - Luego se registra el evento en el Servicio de Logs.
    - También se envía una notificación al usuario mediante el Servicio de Notificaciones.
  - **[Horario no válido]:**
    - El sistema registra un intento fallido en los logs.
    - Se envía un mensaje de error al usuario a través del Servicio de Notificaciones.



5. Finalmente, se retorna una respuesta a la interfaz y se muestra al usuario la confirmación o el mensaje de error.

**Propósito del diagrama:**

Este diagrama permite comprender claramente cómo interactúan los distintos componentes del sistema al registrar una asistencia, mostrando tanto los escenarios exitosos como los fallidos. Es útil para desarrollo, pruebas y documentación técnica.

*i*