

Proyecto Desarrollo de Software

Pale Arenas Iván

Martinez Garcia Angel



Universidad Veracruzana

Documento de Diseño - GestorPro

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 más 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.

Actores Principales

Actores:

Administrador: Crea empleados, asigna turnos, visualiza reportes, aprueba/rechaza solicitudes de cambio de turno, asigna pagos, y visualiza empleados por turno.

Empleado: Inicia sesión, consulta su turno actual, solicita cambio de turno, registra asistencia, y consulta su historial de pagos.

Sistema: Envía notificaciones automáticas (como avisos de cambio de turno aprobado/rechazado).

Casos de uso:

1. Iniciar sesión (Empleado y Administrador)
2. Cerrar sesión (Empleado y Administrador)
3. Iniciar jornada (Empleado y Administrador)
4. Finalizar jornada (Empleado y Administrador)
5. Consultar jornada realizada (Empleado y Administrador)
6. Registrar nuevo empleado (Administrador)
7. Eliminar empleado (Administrador)
8. Consultar turnos (Empleado y Administrador)
9. Ver empleados por turno (Administrador)
10. Solicitar cambio de turno (Empleado)
11. Aprobar cambio de turno (Administrador)
12. Rechazar cambio de turno (Administrador)
13. Registrar horas extra (Administrador)
14. Consultar horas extra (Empleado y Administrador)
15. Editar registro de pago (Administrador)
16. Consultar historial de pagos (Empleado y Administrador)
17. Ver perfil / turno actual (Empleado y Administrador)
18. Ver términos y condiciones (Público)
19. Ver preguntas frecuentes (Público)

Diagrama de Casos de Uso

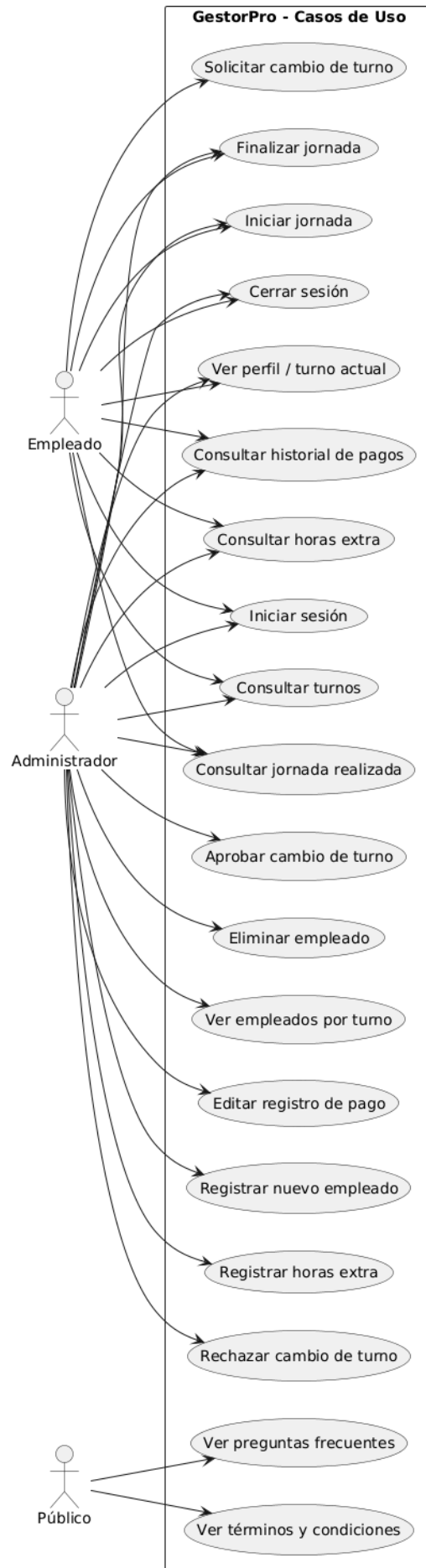
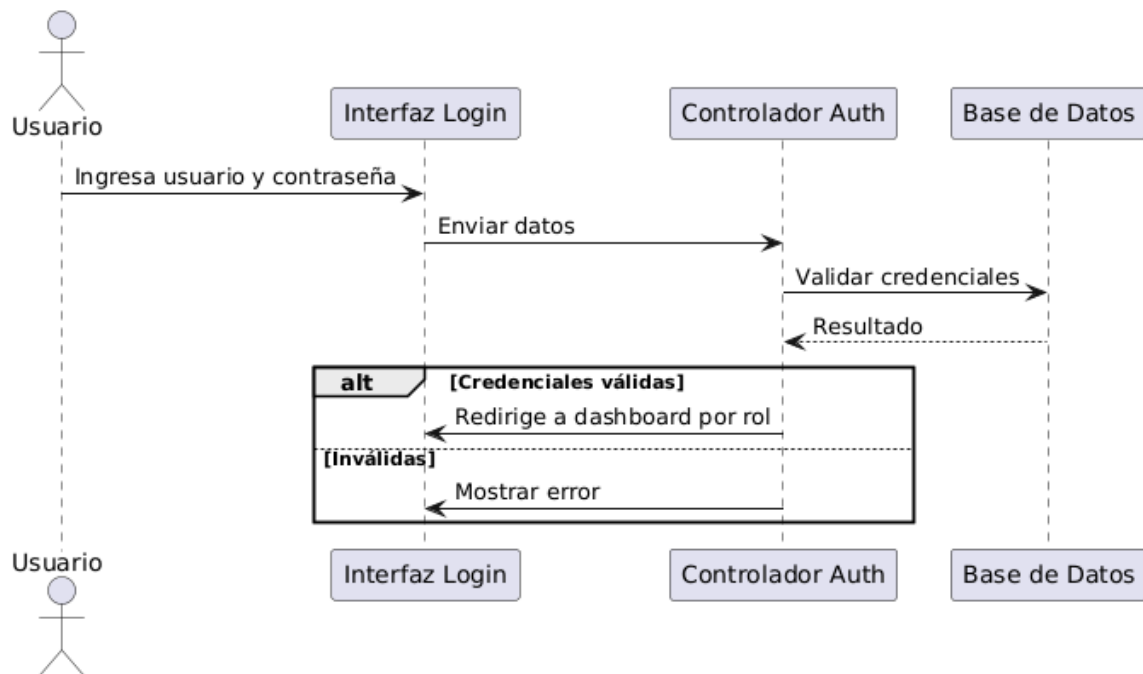


Diagrama de Secuencia - Iniciar Sesión



Propósito del Diagrama:

Este diagrama permite visualizar paso a paso cómo el sistema procesa una solicitud de inicio de sesión desde que el usuario envía sus credenciales hasta que es autenticado y redirigido al módulo correspondiente según su rol. Es útil para identificar puntos de validación crítica, manejo de errores y flujo de navegación según perfil.

Estructura del sistema involucrado:

1. **Usuario:** Actor que provee credenciales para ingresar al sistema.
2. **Interfaz de Usuario (UI):** Muestra el formulario y captura los datos.
3. **Controlador Auth:** Lógica que maneja la autenticación del usuario.
4. **Base de Datos (BD):** Repositorio que contiene las credenciales y roles.

Flujo Principal:

1. El usuario ingresa su ID y contraseña en el formulario de login.
2. El controlador recibe los datos, consulta la base de datos y valida.
3. Si las credenciales son válidas, redirige al usuario a su dashboard.
4. Si no son válidas, muestra un mensaje de error.

Uso del Diagrama:

Ideal para analizar seguridad, validación de datos y control de acceso en sistemas multi-rol como GestorPro.

Diagrama de Actividad UML - Iniciar Jornada



Propósito del Diagrama:

Este diagrama permite observar el flujo lógico interno que sigue el sistema cuando un empleado o administrador inicia su jornada laboral. Ayuda a verificar cómo se asegura que el usuario tenga sesión activa, cómo se calcula la hora y cómo se registra dicha información.

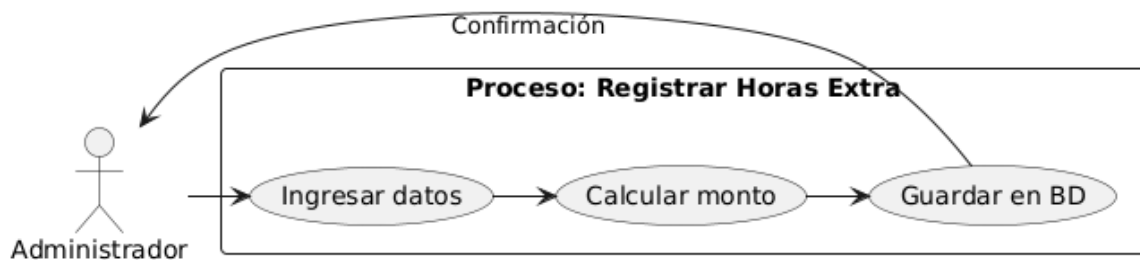
Flujo y Decisiones:

1. El sistema valida que haya sesión iniciada.
2. Se obtiene la hora actual del sistema.
3. Se registra como inicio de jornada en la base de datos.
4. Se devuelve al usuario una confirmación visual o notificación.

Utilidad del Diagrama:

Este tipo de representación es útil para diseñar validaciones automáticas, detectar posibles puntos de error como la doble entrada o asegurar que se registran correctamente los tiempos laborales para generar reportes o pagos.

Diagrama de flujo de datos (DFD) - Registrar horas extra



Propósito del Diagrama:

Este DFD muestra cómo el administrador registra horas extra para un empleado. Se detallan los pasos de captura de datos, cálculo de montos, almacenamiento y retroalimentación al usuario. Este diagrama permite entender claramente la interacción entre las capas del sistema y el procesamiento interno sin entrar en detalles técnicos de implementación.

Componentes involucrados:

1. **Administrador:** Actor que introduce los datos.
2. **Formulario de Registro:** Punto de entrada de los datos (número de horas, fecha, ID).
3. **Calculador de Montos:** Lógica que determina el monto extra (usualmente a razón de \$50/hora).

4. **DAO / Acceso a BD:** Módulo que guarda los datos en la base de datos.
5. **Sistema de Confirmación:** Retorna un mensaje al usuario con éxito o error.

Flujo de Proceso:

1. El administrador accede al formulario e ingresa datos.
2. El sistema calcula el pago correspondiente.
3. Se guarda el registro en la base de datos.
4. Se informa visualmente al usuario que fue registrado correctamente.

Importancia:

Ayuda a entender cómo las acciones administrativas afectan la compensación del empleado, lo cual puede estar vinculado al módulo de pagos.

Diagrama de Actividad UML - Solicitar Cambio de Turno



Propósito del Diagrama:

Este diagrama permite visualizar el proceso de negocio que sigue un empleado cuando desea solicitar un cambio de turno. Muestra claramente las acciones humanas (interacción del empleado) y las respuestas automatizadas del sistema, como guardar la solicitud y marcarla como pendiente.

Participantes:

- **Empleado:** Usuario que solicita un nuevo turno.
- **Sistema GestorPro:** Automatiza el registro de la solicitud y su estado.
- **Base de Datos:** Almacena la solicitud junto con el estado “pendiente”.

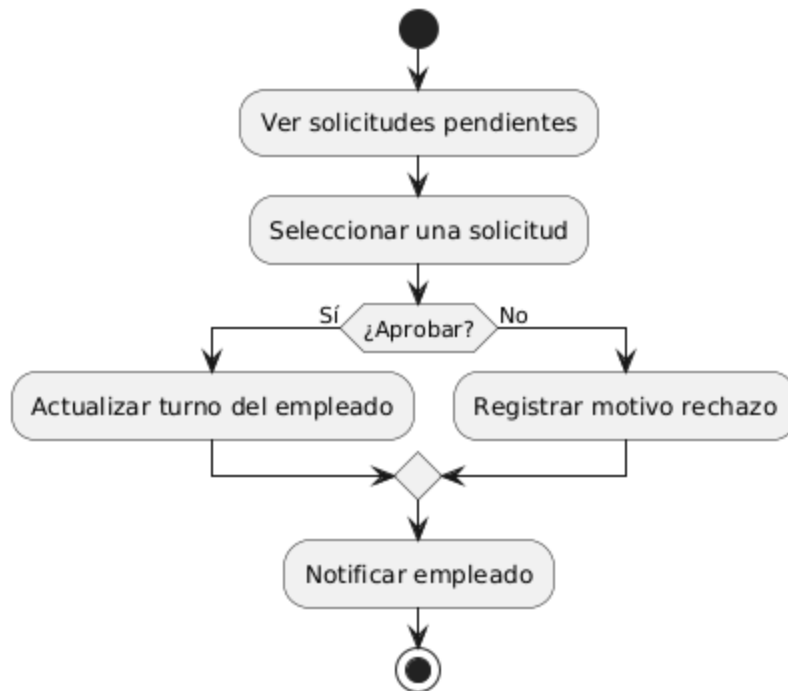
Flujo de Proceso:

1. El empleado ingresa a la opción de cambio de turno.
2. Selecciona un turno nuevo desde una lista.
3. El sistema registra la solicitud en la base de datos.
4. Marca la solicitud como “pendiente de aprobación”.
5. Se muestra una confirmación visual al usuario.

Importancia:

Este flujo ayuda a mantener una trazabilidad clara de quién pidió el cambio, cuándo, y en qué estado se encuentra, facilitando la gestión para el administrador.

Diagrama de Actividad UML - Aprobar o rechazar solicitud de turno



Propósito del Diagrama:

Mostrar el flujo de decisiones que sigue el administrador cuando revisa una solicitud de cambio de turno, permitiendo entender cómo impacta en la asignación de horarios y cómo se notifica al empleado.

Flujo de Actividades:

1. El administrador revisa la lista de solicitudes.
2. Selecciona una solicitud específica.
3. Decide entre aprobar o rechazar.
4. Si aprueba, se actualiza el turno del empleado.

5. Si rechaza, debe registrar un motivo.
6. En ambos casos, se notifica al empleado del resultado.

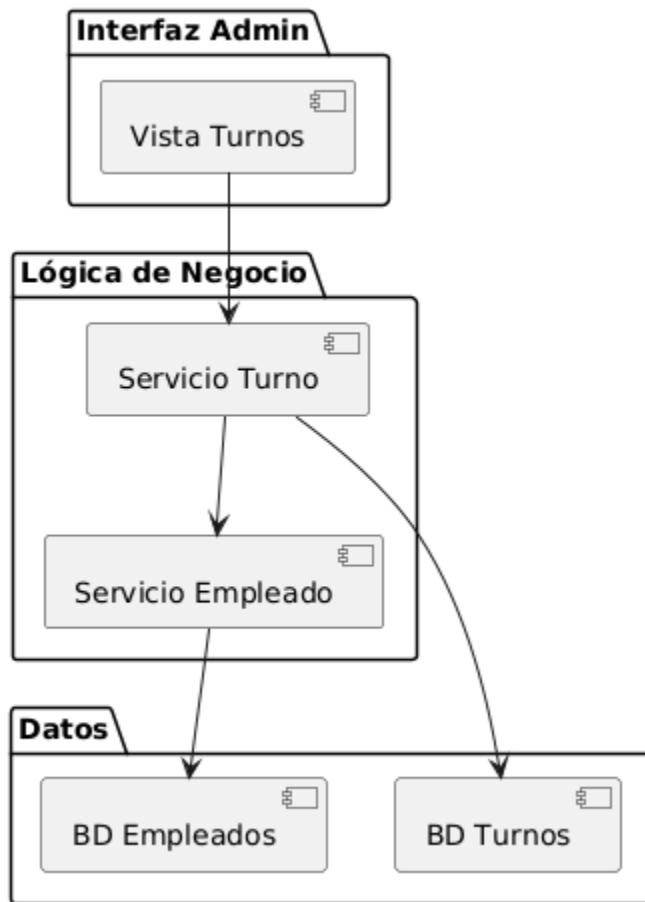
Estructura del Sistema:

- **Vista de Solicitudes:** Interfaz donde el admin ve solicitudes.
- **Controlador de Turnos:** Aplica cambios y validaciones.
- **BD:** Actualiza estado de solicitud y turno del empleado.

Utilidad:

Permite rastrear los criterios de decisión y estructurar los pasos necesarios para que una solicitud tenga un desenlace claro y comunicable.

Diagrama de componentes UML - Ver empleados por turno



Propósito del Diagrama:

Este diagrama muestra cómo se organiza el sistema internamente para mostrar al administrador qué empleados están asignados a cada turno. Se enfoca en la separación lógica entre interfaz, controlador y modelo.

Estructura del Sistema:

1. **Componente Vista Turnos:** Renderiza una lista de turnos.
2. **Componente Controlador Turno:** Solicita datos relacionados con los empleados asignados.

3. **Servicio Turno:** Lógica que filtra empleados por ID de turno.

4. **Base de Datos (Empleado, Turno):** Almacena relaciones.

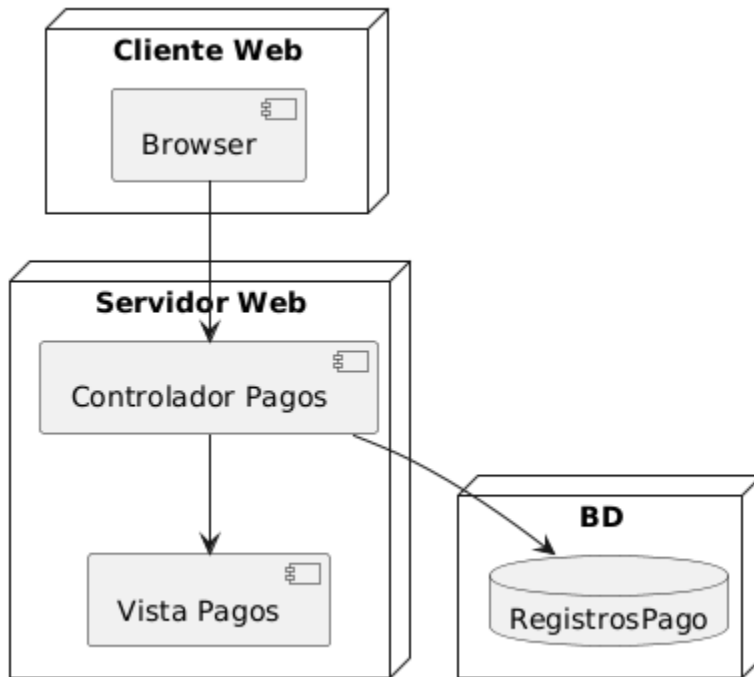
Flujo:

1. El admin accede a la vista de turnos.
2. El controlador llama al servicio para cada turno.
3. El servicio obtiene los empleados asociados desde la BD.
4. Se muestra una tabla/lista organizada.

Importancia:

Ayuda a comprender la organización interna del personal por horarios y facilita decisiones como redistribución de carga laboral o validación de jornadas.

Diagrama de despliegue UML - Consultar historial de pagos



Propósito del Diagrama:

Ilustrar la arquitectura de software y hardware que participa cuando un empleado o administrador consulta el historial de pagos. Este tipo de diagrama muestra cómo se distribuyen los componentes en el cliente, servidor y base de datos.

Componentes:

- **Cliente Web (Navegador):** Donde se ejecuta la acción.
- **Servidor Web:** Procesa la lógica y comunica con la base de datos.
- **Base de Datos MySQL:** Contiene los registros de pagos.

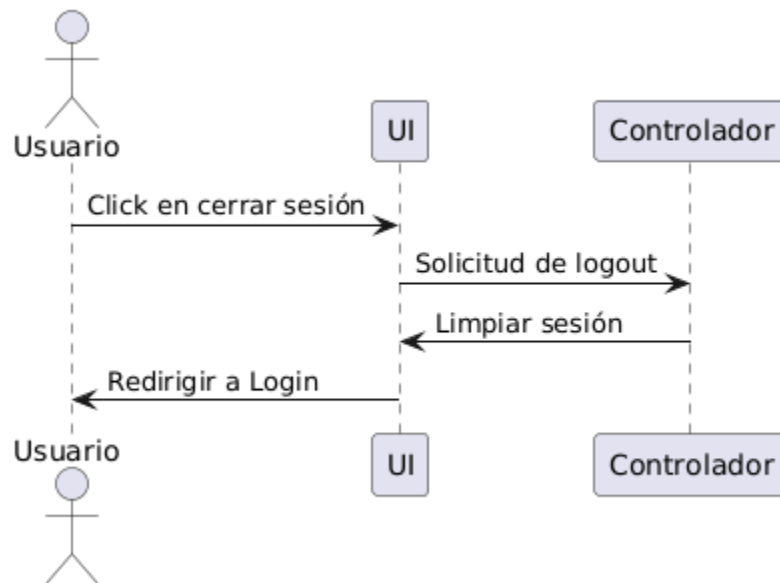
Flujo:

1. El usuario accede al historial desde su panel.
2. El controlador procesa la solicitud.
3. Consulta los pagos relacionados desde la BD.
4. Muestra la información en pantalla.

Utilidad:

Este tipo de diagrama es ideal para documentar arquitecturas distribuidas, y muestra cómo la interacción de diferentes componentes garantiza la correcta presentación de información al usuario final.

Diagrama de secuencia UML - Cerrar Sesión



Propósito del Diagrama:

Describir los pasos que se ejecutan internamente cuando un usuario decide cerrar sesión, permitiendo observar cómo se destruyen variables de sesión y se redirige al login.

Flujo del Diagrama:

1. El usuario hace clic en “Cerrar sesión”.
2. La vista llama al controlador para cerrar sesión.
3. El controlador limpia la sesión del usuario.
4. Redirige al usuario al formulario de login.

Componentes:

- **Usuario**
- **Vista (UI)**
- **Controlador Auth**

Utilidad:

Sencillo pero esencial. Este proceso asegura que los datos de sesión se manejen correctamente para proteger la integridad y privacidad del sistema.

Diagrama de actividad UML - Eliminar Empleado



Propósito del Diagrama:

Este diagrama ayuda a representar el flujo de pasos que debe realizar un administrador para eliminar a un empleado del sistema. Incluye la validación visual, confirmación y efecto sobre la base de datos.

Flujo:

1. El administrador accede a la lista de empleados.
2. Selecciona el botón de eliminar.
3. Se muestra una ventana/modal de confirmación.
4. Si se confirma, se elimina al empleado en la BD.

5. El sistema muestra un mensaje de éxito.

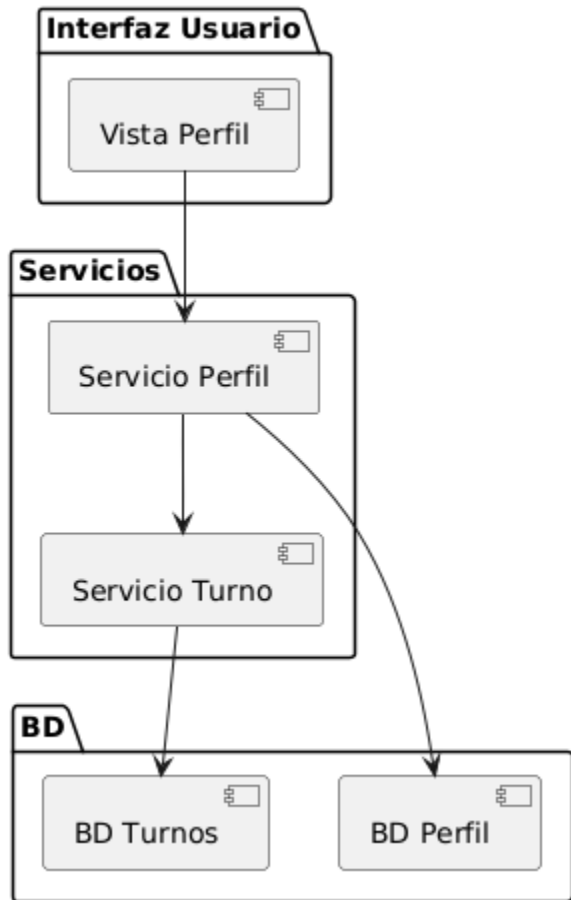
Componentes involucrados:

- Vista de empleados
- Controlador de PanelAdmin
- Servicio Empleado / Repositorio
- Base de Datos

Importancia:

Evita eliminaciones accidentales mediante confirmación explícita y mantiene integridad del sistema al eliminar correctamente referencias relacionadas.

Diagrama de componentes UML - Ver Perfil / Turno Actual



Propósito del Diagrama:

Describir cómo se organiza la consulta del perfil del empleado junto con su turno asignado actual. Ideal para mostrar la integración entre distintos servicios y modelos.

Estructura del sistema:

1. **Vista Perfil:** Muestra datos del usuario y su turno actual.
2. **Controlador Perfil:** Solicita datos al servicio.
3. **Servicio de Usuario:** Recupera datos personales del empleado.

4. **Servicio de Turnos:** Consulta el turno asignado.
5. **BD:** Empleado y Turno relacionados por clave foránea.

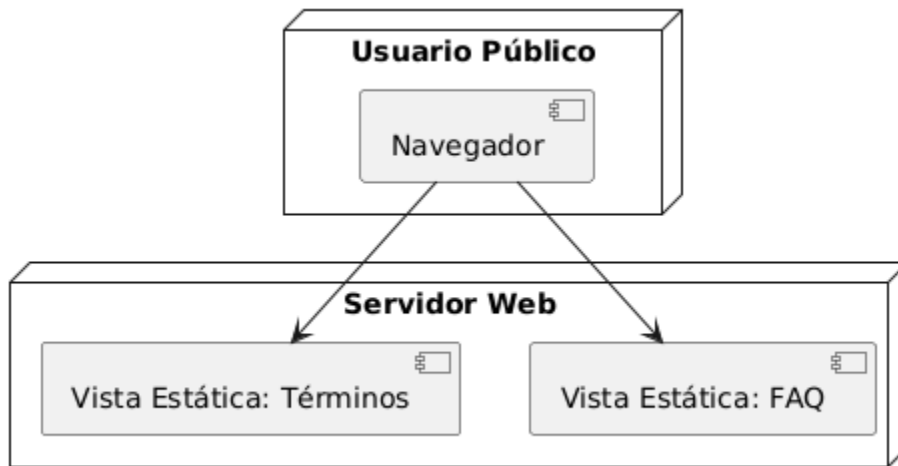
Flujo:

1. El empleado abre su perfil.
2. El sistema carga su información personal y turno.
3. Se visualiza en pantalla.

Utilidad:

Proporciona al usuario claridad sobre su identidad en el sistema y su configuración actual (turno), clave para la experiencia del usuario y validación de solicitudes.

Diagrama de despliegue UML - Ver Términos y Condiciones / Preguntas Frecuentes



Propósito del Diagrama:

Mostrar cómo se distribuyen los componentes para cargar vistas estáticas accesibles al público, sin necesidad de autenticación.

Estructura:

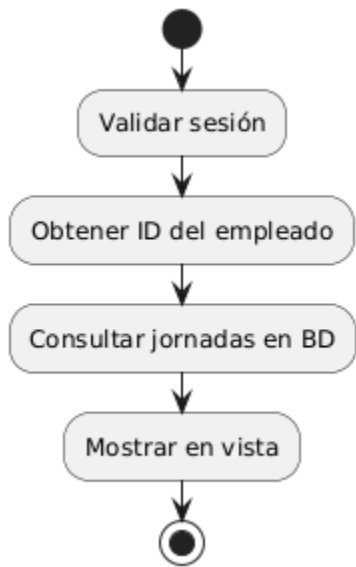
- Navegador (Cliente web)
- Servidor Web (host de vistas estáticas)
- Archivos HTML/Razor de términos y preguntas frecuentes

Flujo:

1. Un visitante accede a /Terminos o /Preguntas.
2. El servidor devuelve el archivo estático correspondiente.
3. El visitante lo visualiza en el navegador.

Utilidad:

Clarifica que estas vistas no requieren acceso a controladores complejos ni lógica dinámica. Es contenido público, útil para transparencia legal y apoyo al usuario.

Diagrama de actividad UML - Consultar Jornada Realizada**Propósito del Diagrama:**

Mostrar cómo se recupera y presenta el historial de jornadas del usuario, ya sea por fecha, quincena o empleado.

Estructura:

1. El usuario accede a la sección de registro.
2. El sistema consulta los registros de asistencia asociados al usuario.
3. Se presentan ordenados por fecha o periodo.

4. El usuario puede filtrar por fechas o exportar.

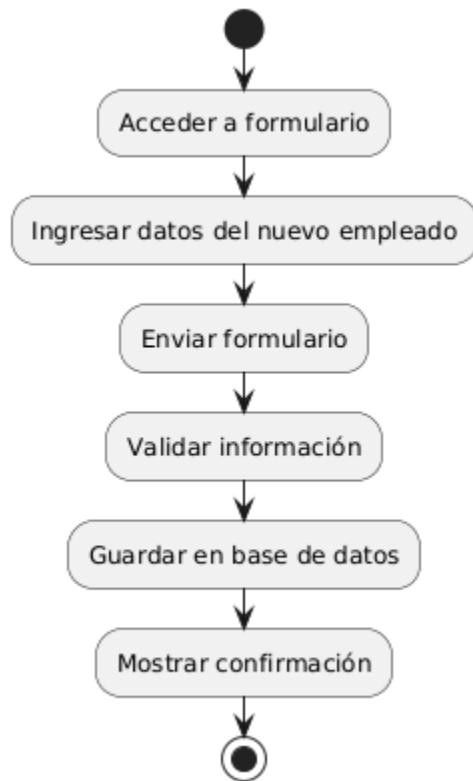
Componentes:

- Vista Registro.cshtml
- Controlador RegistroController
- Base de Datos: RegistrosAsistencia
- Servicios de Consulta o DAO

Utilidad:

Permite al usuario tener evidencia de sus jornadas y al administrador validar el comportamiento de los empleados.

Diagrama de actividad UML - Registrar nuevo empleado



Propósito del Diagrama:

Representar el flujo completo para el alta de un nuevo empleado en el sistema por parte del administrador. Este proceso asegura que se capturen los datos necesarios, se valide la información y se almacene correctamente en la base de datos.

Flujo del Proceso:

1. El administrador accede al Panel de Administración.
2. Selecciona la opción "Registrar nuevo empleado".
3. El sistema muestra un formulario con campos como: ID, nombre, correo, contraseña, turno, rol.

4. El administrador llena los datos y da clic en “Guardar”.
5. El sistema valida que todos los campos requeridos estén completos y sean válidos.
6. Si los datos son válidos:
 - Se crea un nuevo registro en la base de datos.
 - Se muestra mensaje de éxito.
7. Si hay errores:
 - Se muestran los errores en pantalla.
 - Se mantiene el formulario con los datos ingresados.

Diagrama de flujo de datos (DFD) - Editar registro de pago



Propósito del Diagrama:

Este DFD describe cómo se procesa la modificación de un registro de pago de un empleado. Permite identificar los flujos de datos desde la interfaz hasta la base de datos, asegurando precisión en el cálculo y trazabilidad en la actualización.

Flujo del Proceso:

1. El administrador accede al historial de pagos de un empleado.
2. Selecciona un registro a editar.
3. El sistema muestra el formulario con los valores actuales: período, pago base, horas extra, total.
4. El administrador modifica los campos necesarios.
5. Al hacer clic en “Guardar”:
 - Se recalcula automáticamente el nuevo total.

- Se actualiza el registro en la base de datos.

6. El sistema confirma la operación y redirige nuevamente al historial.