

Contenido

7. Modelo de diseño.....	1
7.1. Diagrama de clases de diseño	1
7.2. Modelo de objetos de software y sus colaboraciones	2
7.2.1. CU1-Acceso al sistema.....	3
Diagrama de secuencia del sistema	3
Diagrama de colaboración de diseño	3
7.2.2. CU2-Introducir Expediente de Preso.....	4
Diagrama de secuencia del sistema	4
Diagrama de colaboración de diseño	4
7.2.3 CU3-Dar de alta actividad	5
Diagrama de colaboración de diseño	5
7.2.4 CU4-Dejar de ofrecer	5
Diagrama de colaboración de diseño	6
7.2.5 CU5-Asignar Actividades a Presos	6
Diagrama de colaboración de diseño	7
7.2.6 CU6-Validar Actividad.....	8
Diagrama de colaboración de diseño	8
7.2.7 CU7-Cálculo de los días redimidos	9
Diagrama de colaboración de diseño	10
7.2.8 CU8-Introduce horas trabajadas presos.....	10
Diagrama de colaboración de diseño	11

7. Modelo de diseño

7.1. Diagrama de clases de diseño

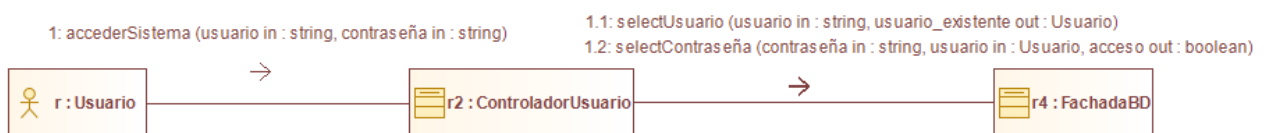
<Imagen del diagrama de clases realizado en Modelio. Se genera un diagrama de clases en Modelio basado en el diagrama de tablas de Toad, incluyendo atributos. Se añaden la clase Fachada_BD y los controladores e interfaces necesarios para el diseño de los casos de uso seleccionados. Según se van necesitando métodos en los diagramas de de colaboración de diseño se van añadiendo en el diagrama de clases.>

7.2.1. CU1-Acceso al sistema

Diagrama de secuencia del sistema



Diagrama de colaboración de diseño



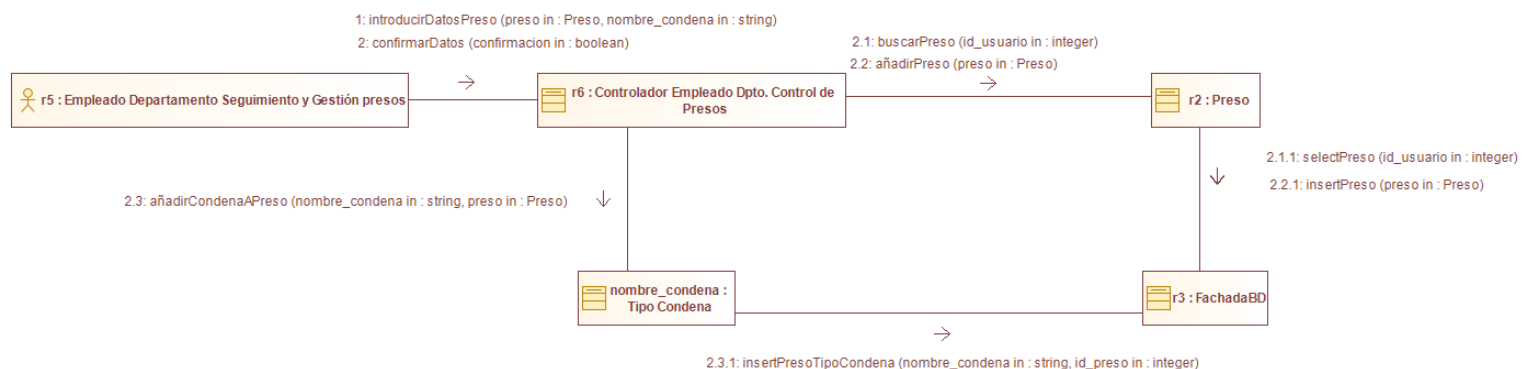
- **accederSistema():** método utilizado para iniciar sesión en el sistema. Al método se le pasará por parámetro el nombre del usuario (String usuario) y la contraseña para verificar si existe en la base de dato.
- **selectUsuario():** query de la base de datos utilizada para devolver el objeto usuario, si existiese un usuario con ese nombre.
- **selectContraseña():** dado el objeto usuario (obtenido en selectUsuario()) comprueba si para ese usuario la contraseña introducida es correcta y devuelve un boolean indicando si es correcta o no.

7.2.2. CU2-Introducir Expediente de Preso

Diagrama de secuencia del sistema



Diagrama de colaboración de diseño



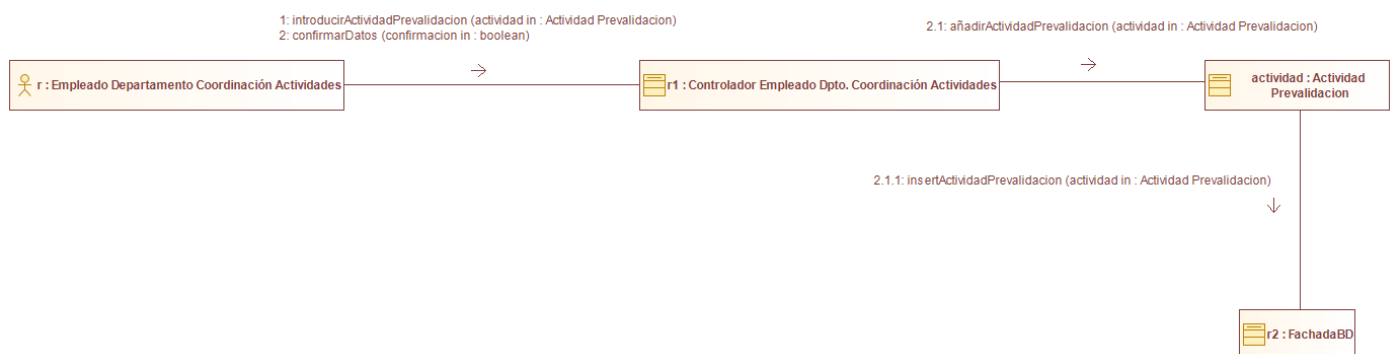
- **confirmarDatos():** método que sirve para confirmar que los datos enviados son correctos y pueden ser introducidos en la base de datos y empleados para realizar operaciones.
- **introducirDatosPreso():** método al que se le pasa por parámetro un objeto preso y el nombre de la condena para introducirlo en la base de datos.
- **buscarPreso():** método utilizado para buscar en la base de datos si ya existe un preso con ese id, que previamente ha sido pasado por parámetro.
- **añadirCondenaAPreso():** método utilizado para relacionar al preso con su condena correspondiente.

7.2.3 CU3-Dar de alta actividad

Diagrama de secuencia del sistema



Diagrama de colaboración de diseño



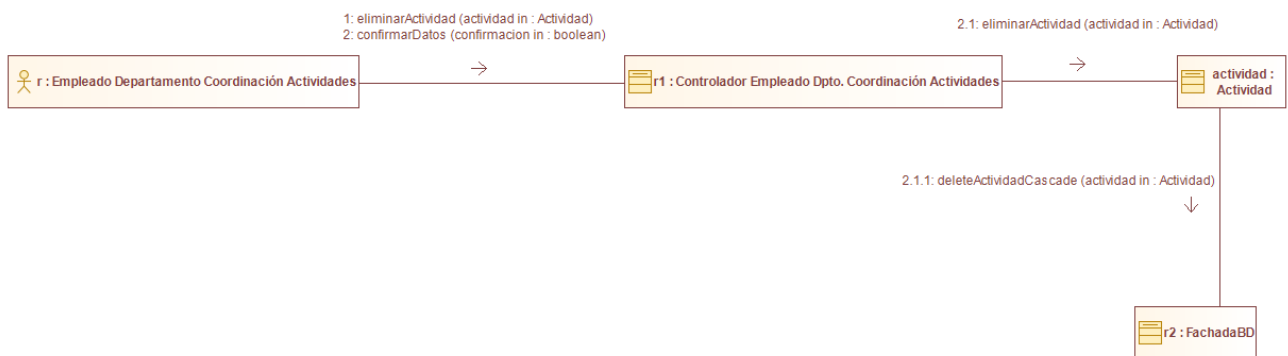
- **introducirActividadPrevalidacion()**: método al que se le pasa por parámetro un objeto actividad prevalidación con todos sus datos para introducirlo en la base de datos.
- **añadirActividadPrevalidacion()**: método que crea una nueva instancia de objeto Actividad Prevalidación y lo añade a la base de datos.

7.2.4 CU4-Dejar de ofrecer

Diagrama de secuencia del sistema



Diagrama de colaboración de diseño



- **eliminarActividad()**: método utilizado para eliminar una actividad de la base de datos.
- **deleteActividadCascade()**: método realizado para eliminar de la base de datos una actividad. Se utiliza Cascade por si la actividad a eliminar estuviese asignada aún a algún preso.

7.2.5 CU5-Asignar Actividades a Presos

Diagrama de secuencia del sistema

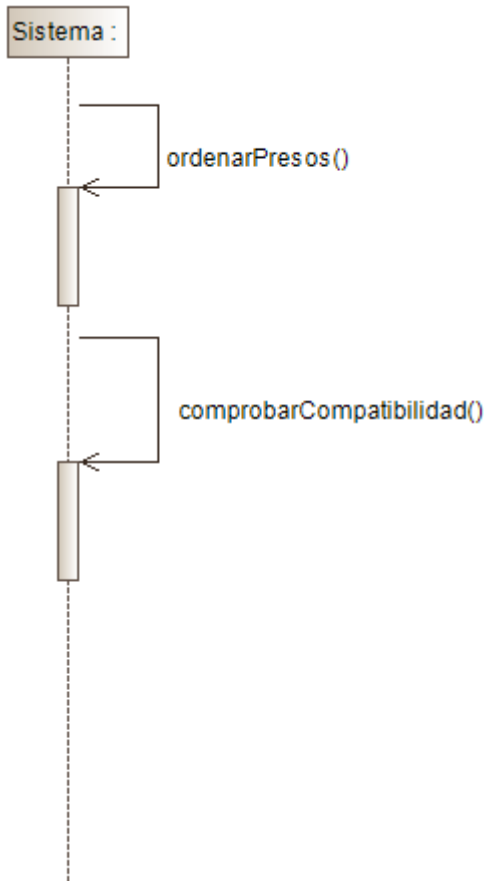
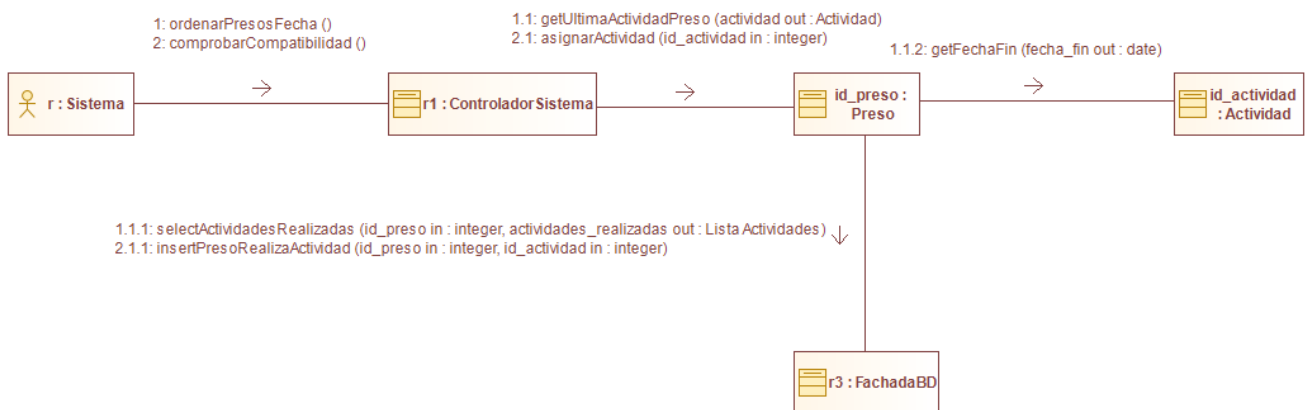


Diagrama de colaboración de diseño



- **ordenarPresosFecha()**: método de Controlador Sistema que itera por todos los presos del sistema, obteniendo sus actividades más recientes por fecha de fin, y los ordena en una lista de presos de tal forma que es aplicable como ranking de prioridad (siendo el preso que más reciente tiene la última actividad el que menos prioridad tiene).
- **getUltimaActividadPreso()**: método ejecutado para cada una de las instancias objeto Preso creadas en la iteración de ordenarPresosFecha() y que se encarga de iterar por una lista de actividades realizadas por el preso, retornando la actividad que es más reciente.
- **selectActividadesRealizadas()**: query a la base de datos que devuelve una lista de actividades que el preso con el id pasado como parámetro ha realizado.
- **getFechaFin()**: método que se encarga de devolver la fecha de fin de una actividad.

- **comprobarCompatibilidad():** método que se encarga de comprobar la compatibilidad entre el tipo de condena del recluso y la actividad seleccionada por este como de mayor preferencia.
- **asignarActividad():** método que asigna una actividad a un preso, habiendo ya comprobado su compatibilidad y teniendo en cuenta su posición en la lista de “ordenarPresosFecha()” y añade esta información a la tabla correspondiente de la base de datos.

7.2.6 CU6-Validar Actividad

Diagrama de secuencia del sistema

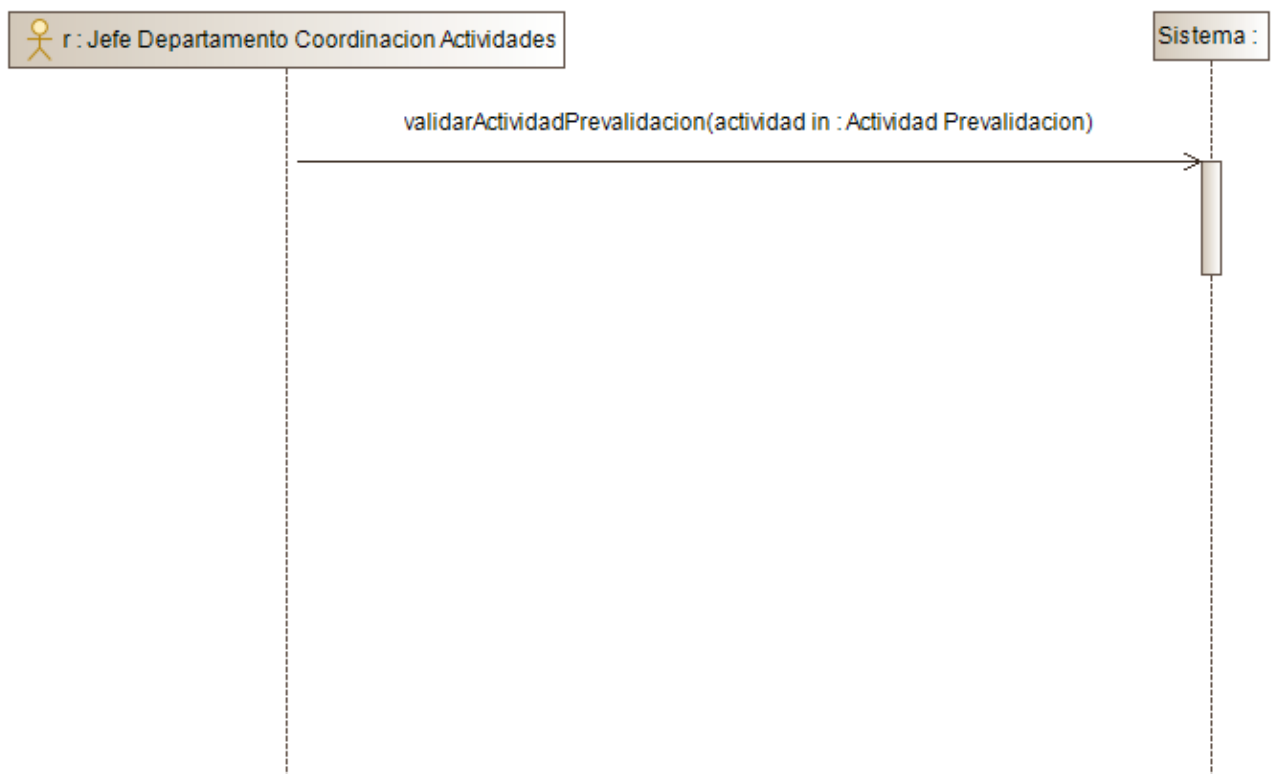
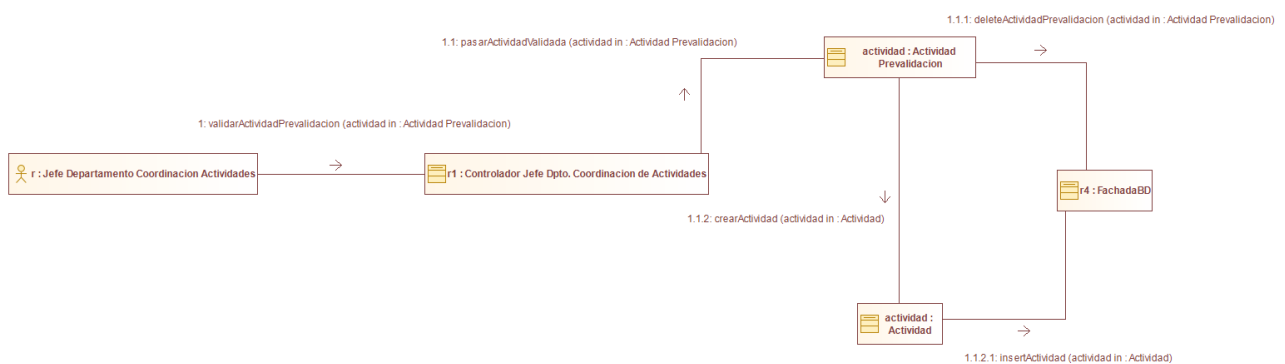


Diagrama de colaboración de diseño



- **validarActividadPrevalidacion():** método que ejecuta el jefe del departamento de coordinación de actividades y que se encarga de pasar una actividad sin validar (Actividad Prevalidación) a una actividad disponible en el sistema y visible a los presos (Actividad).
- **pasarActividadValidada():** método ejecutado por el objeto de la actividad seleccionada por el jefe de departamento que se encarga de borrar y añadir los datos en la base de datos y las tablas correspondientes.
- **crearActividad():** crea una nueva instancia de la tabla Actividad con los mismos datos que poseía la instancia de Actividad Prevalidación desde la que se ha llamado.
- **deleteActividadPrevalidacion():** se encarga de eliminar de la base de datos la Actividad Prevalidación.
- **insertActividad():** añade a la base de datos una nueva Actividad con los datos correspondientes a la Actividad Prevalidación.

7.2.7 CU7-Cálculo de los días redimidos

Diagrama de secuencia del sistema

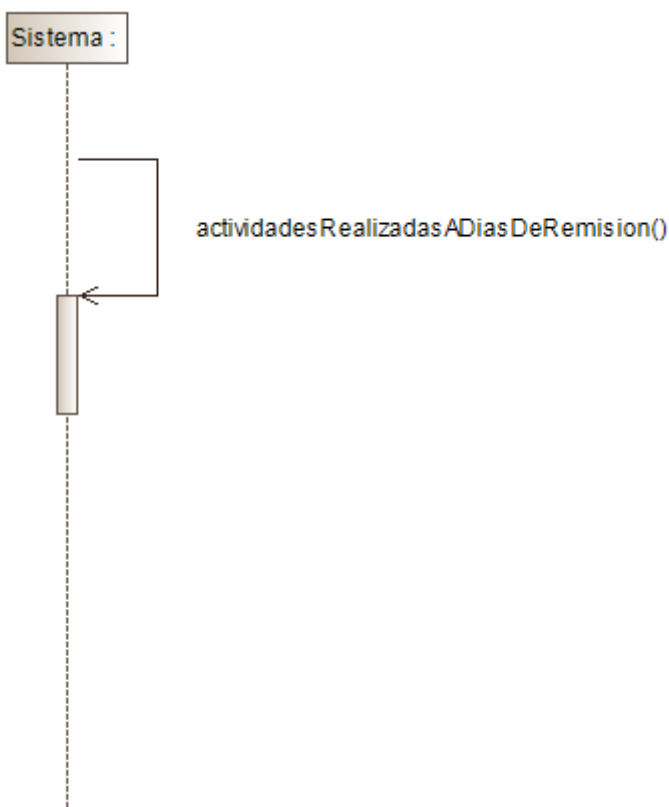
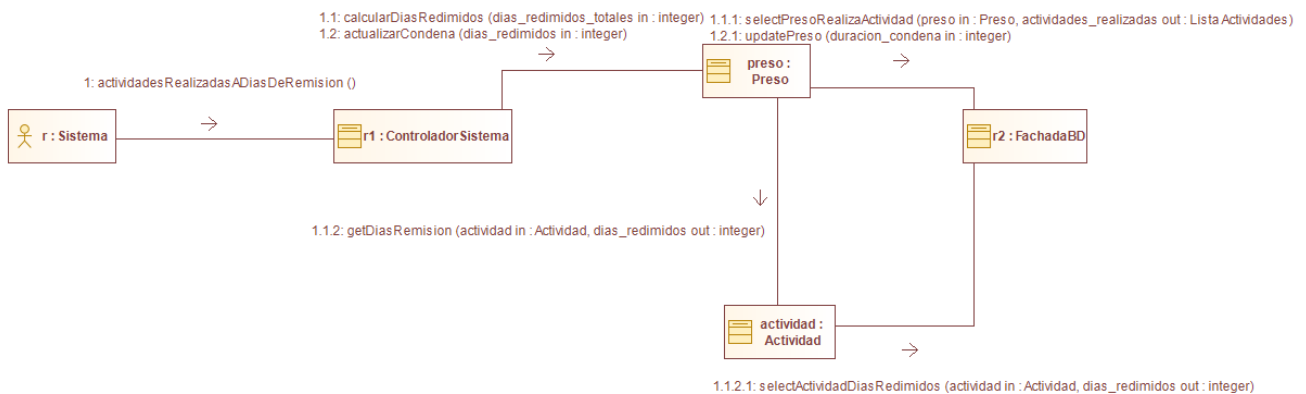


Diagrama de colaboración de diseño



- **actividadesRealizadasADiasDeRemision()**: método ejecutado por el propio sistema de forma automática el último día de cada mes que se encarga de transformar las actividades que han realizado los presos en días de remisión de condena.
- **calcularDiasRedimidos()**: calcula el total de días redimidos por un preso en ese mes de actividades.
- **actualizarCondena()**: método que se encarga de disminuir la condena en tantos días como días redimidos se hayan calculado para el preso y actualizar la base de datos.
- **getDiasRemision()**: método que devuelve los días de remisión que valen cada una de las actividades que el preso ha realizado durante el mes.

7.2.8 CU8-Introduce horas trabajadas presos

Diagrama de secuencia del sistema

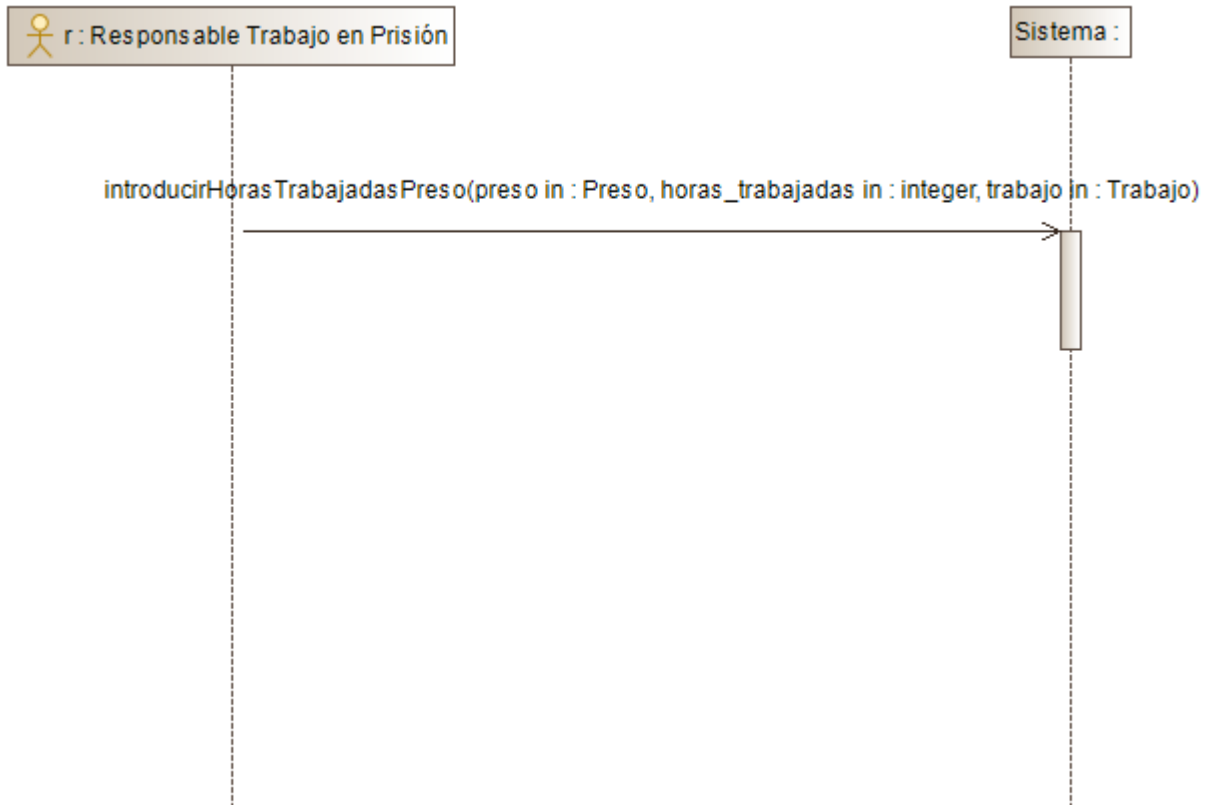
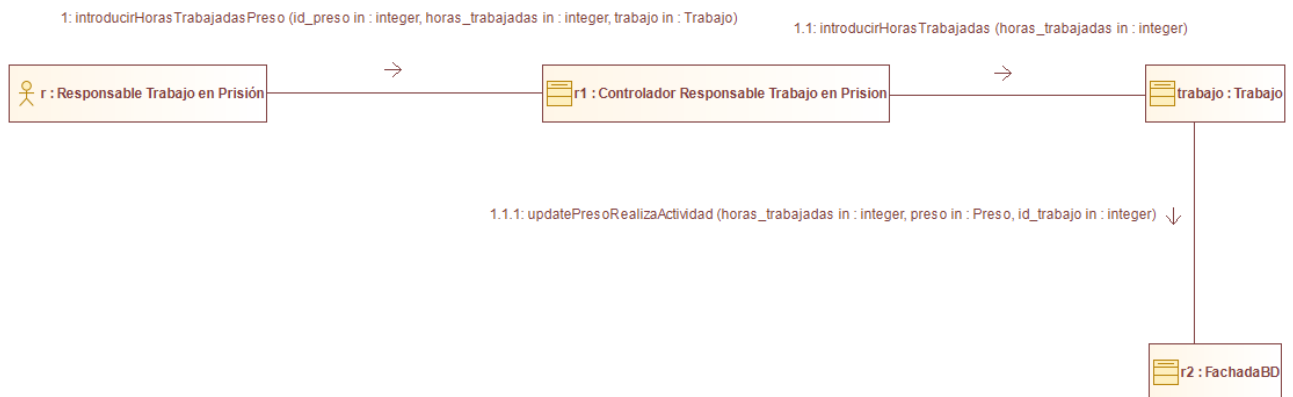


Diagrama de colaboración de diseño



- **introducirHorasTrabajadas():** método que se encarga de actualizar la base de datos con las horas trabajadas por el preso en su puesto de trabajo.
- **updatePresoRealizaActividad():** mediante el id_preso se realiza el update en la tabla preso_actividad para introducir las horas que ha trabajado.