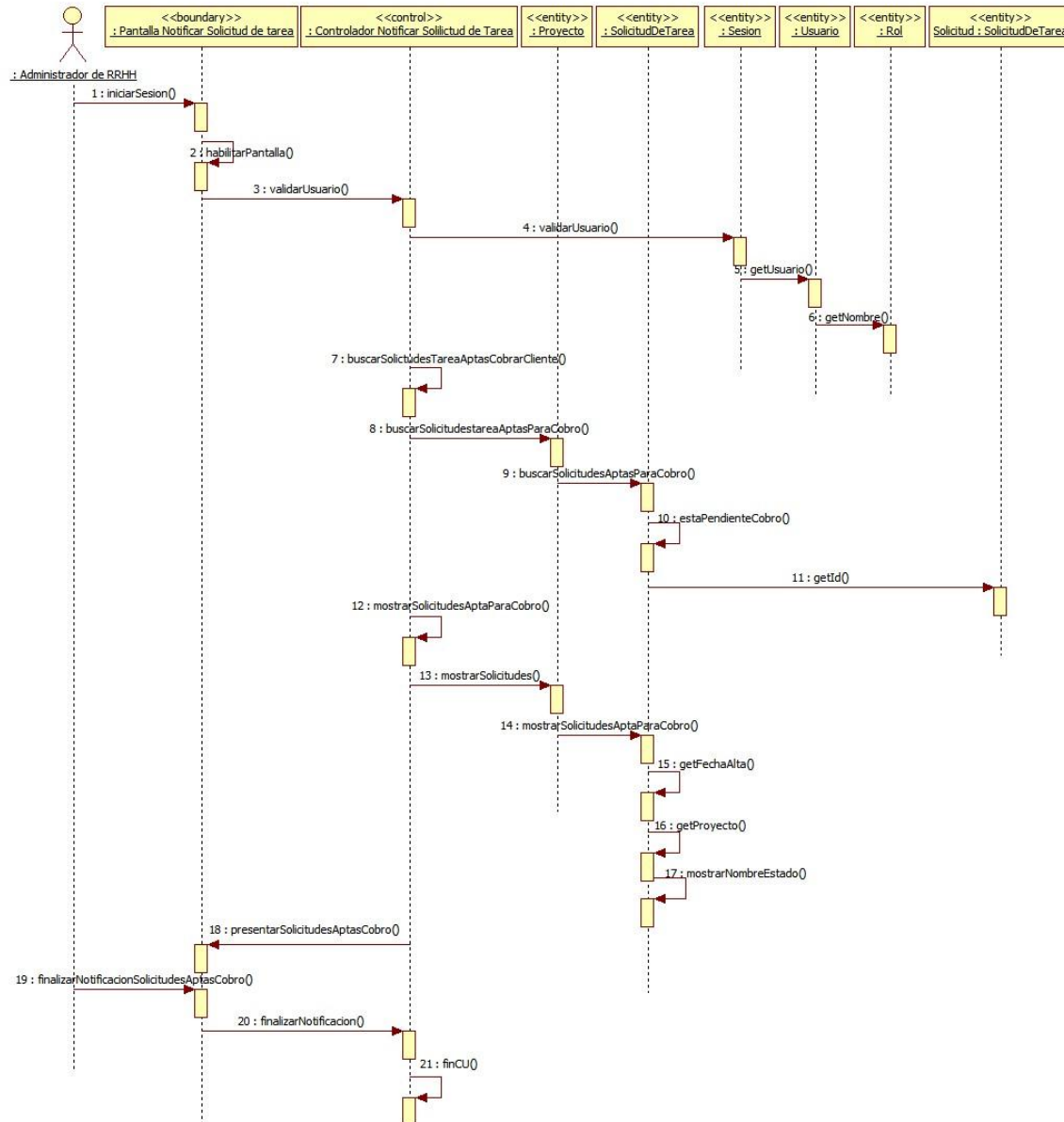




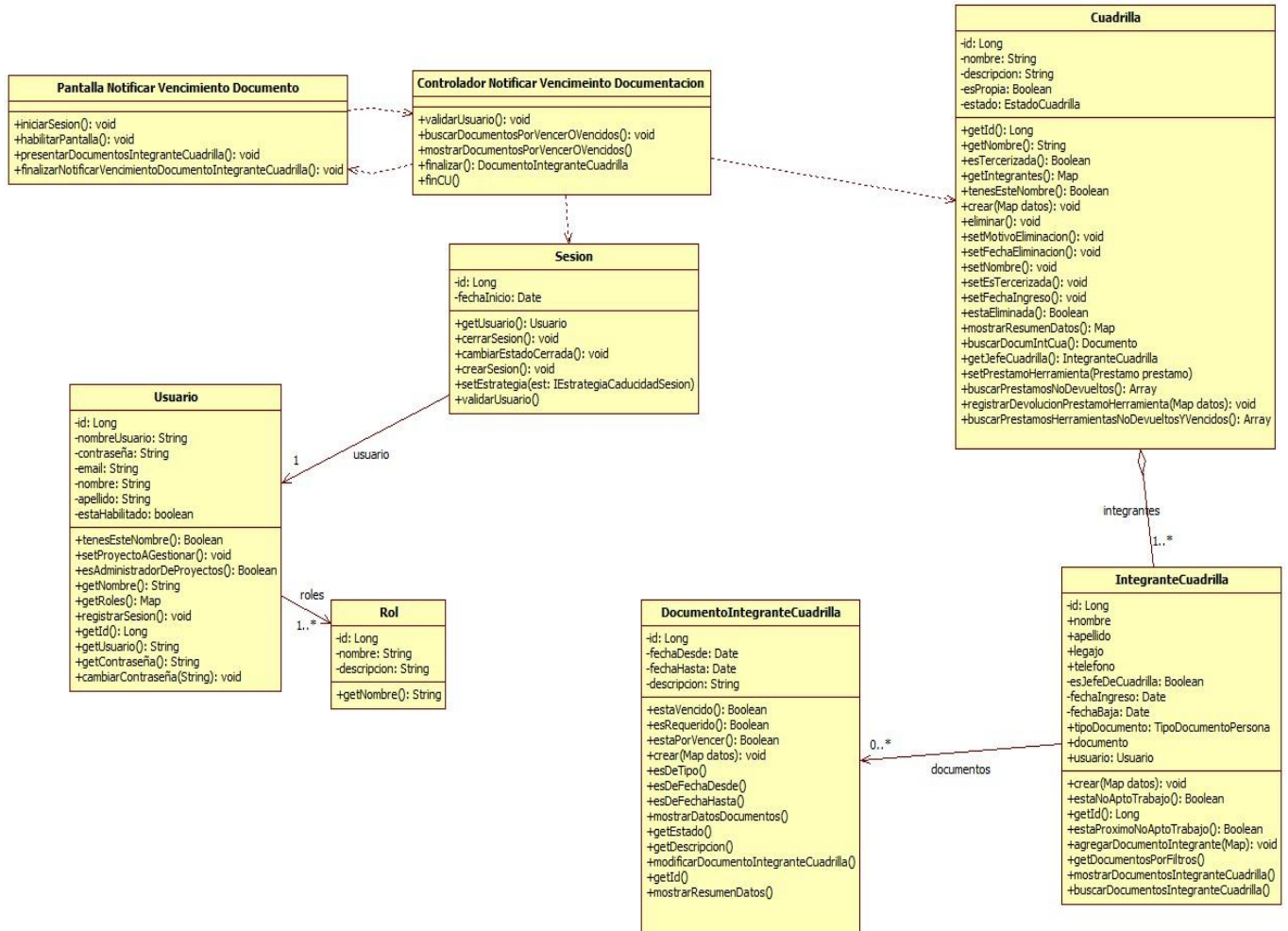
## Diagrama de secuencia





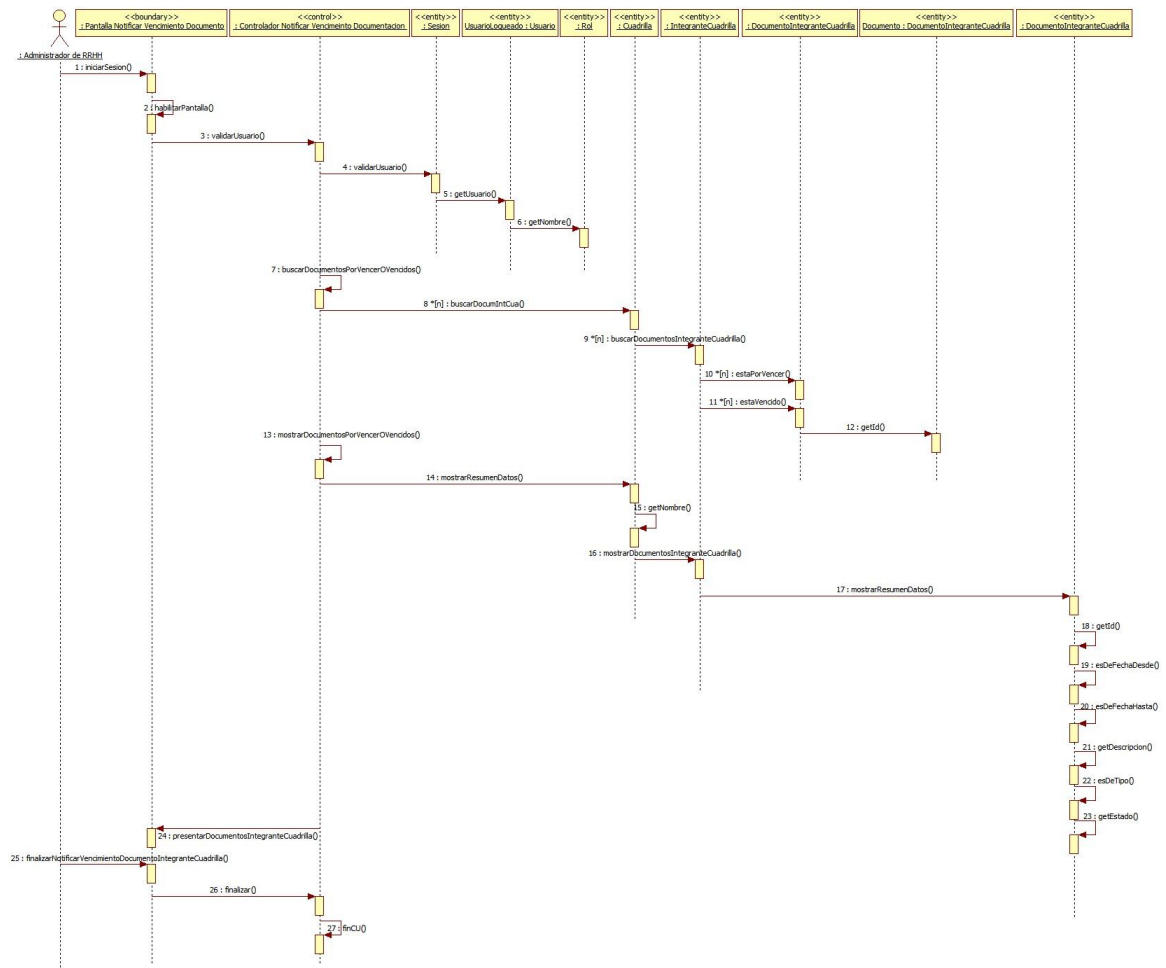
## Realización CU39: Notificar vencimiento documento de integrante de cuadrilla

### Clases intervinientes





## Diagrama de secuencia

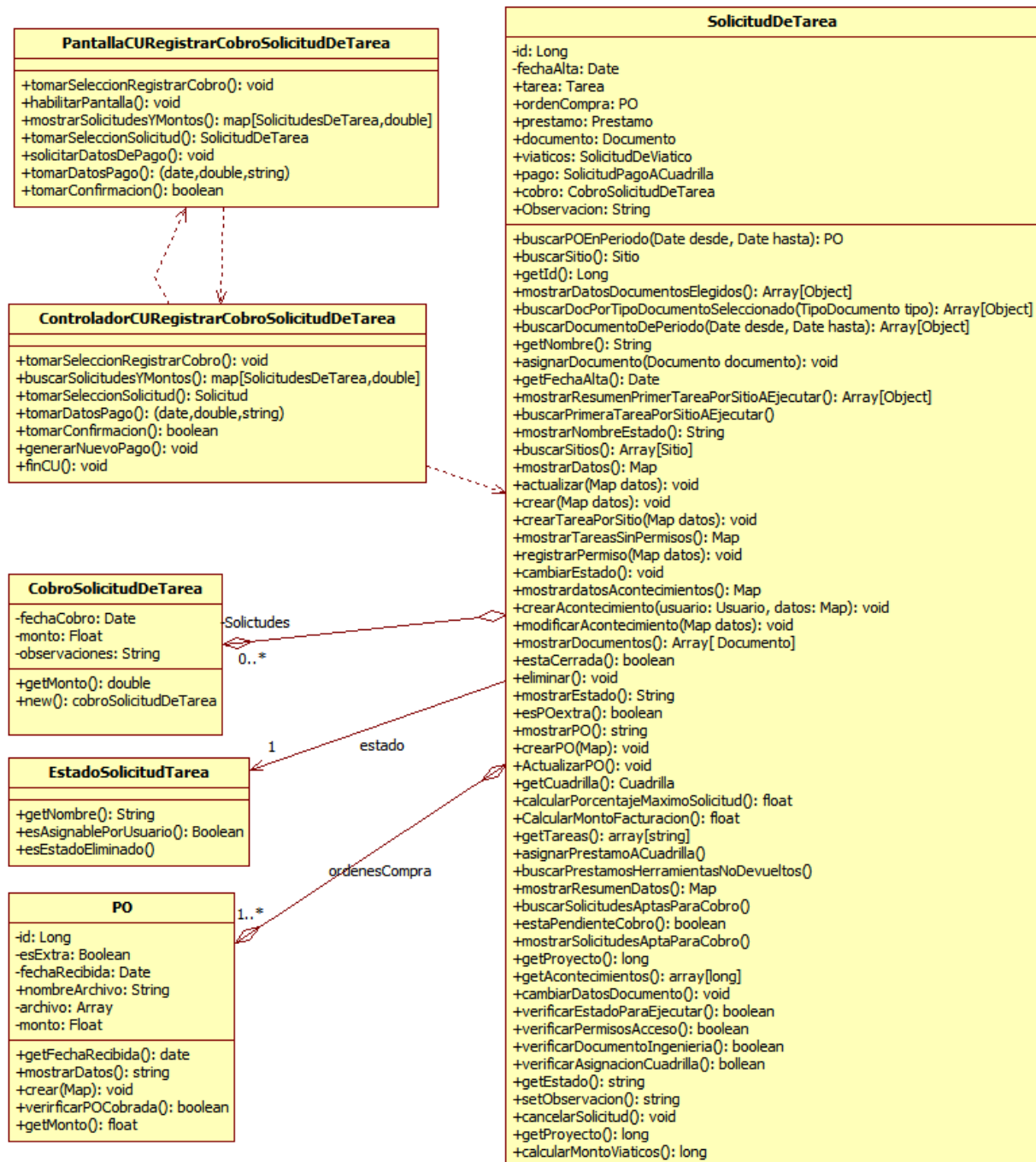




## Paquete Administración

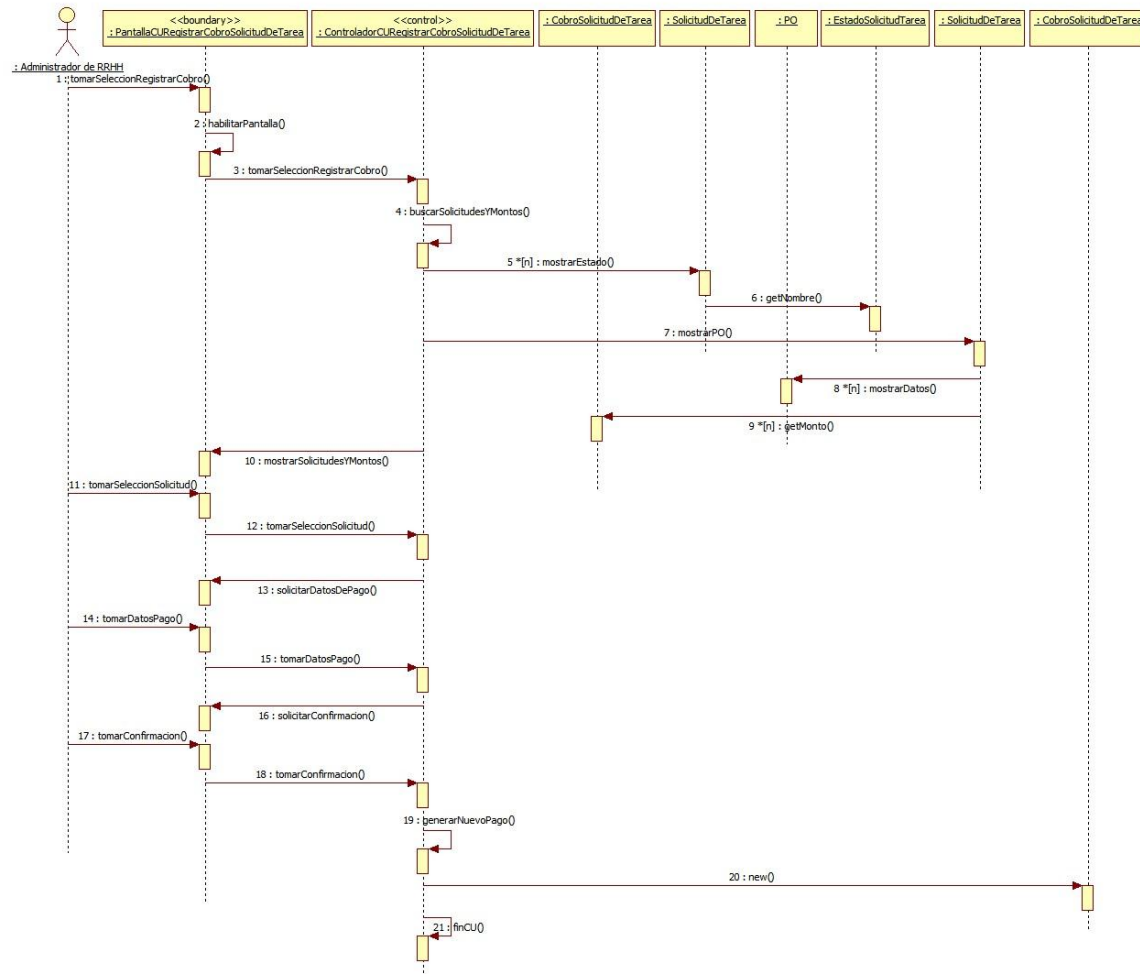
### Realización CU33: Registrar cobro de solicitud de tarea

#### Clases intervinientes





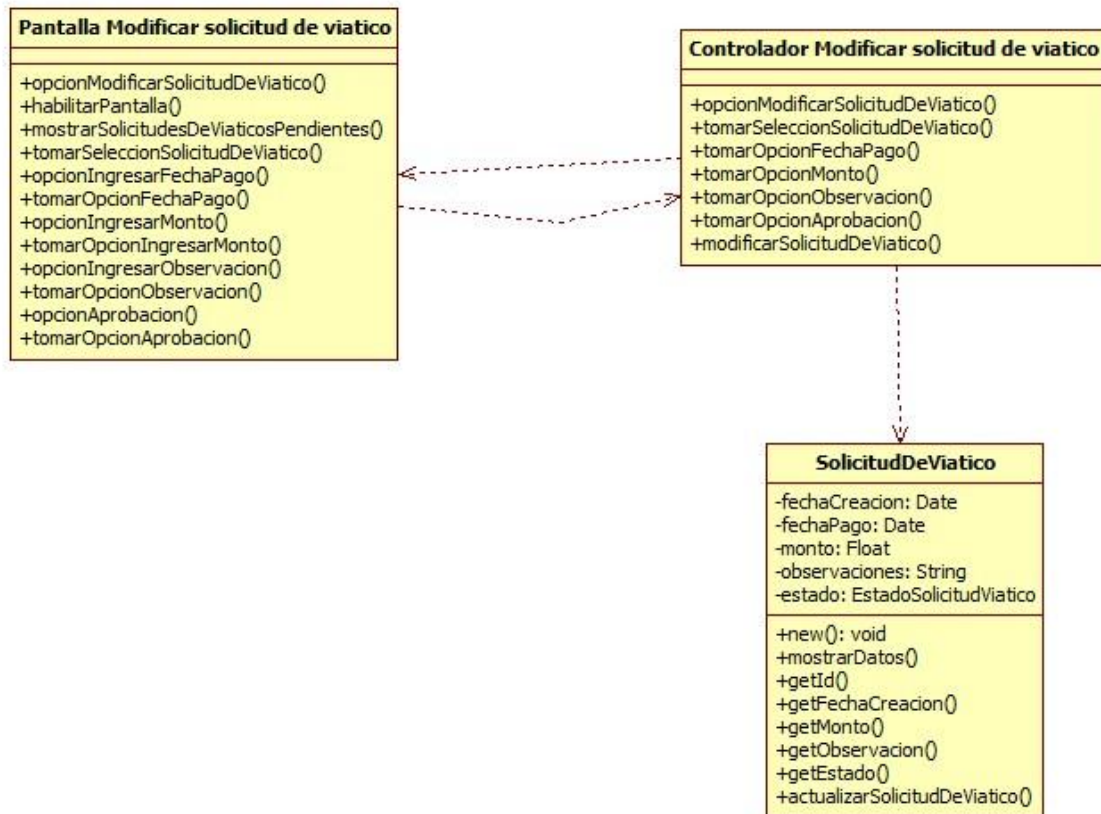
## Diagrama de secuencia





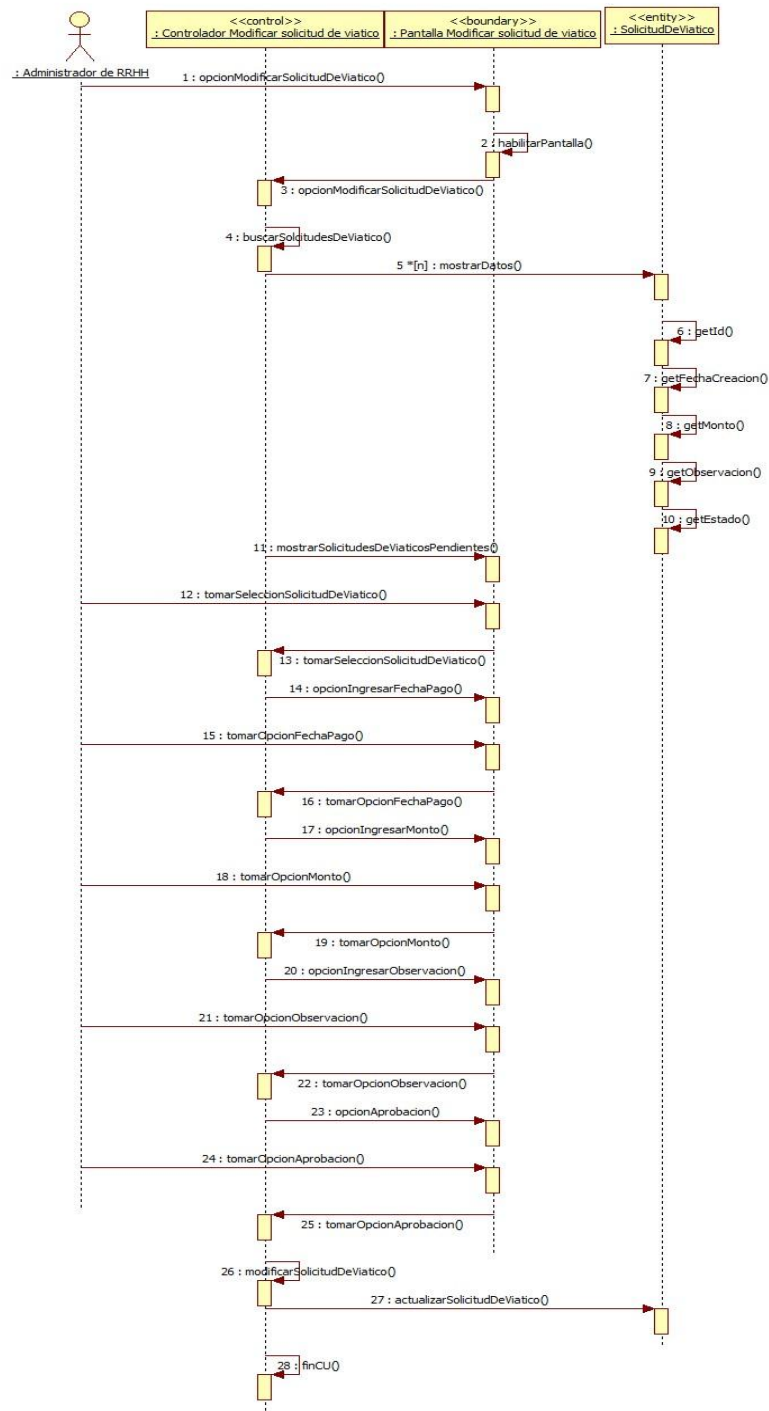
## Realización CU9: Modificar solicitud de viático

### Clases intervinientes





## Diagrama de secuencia

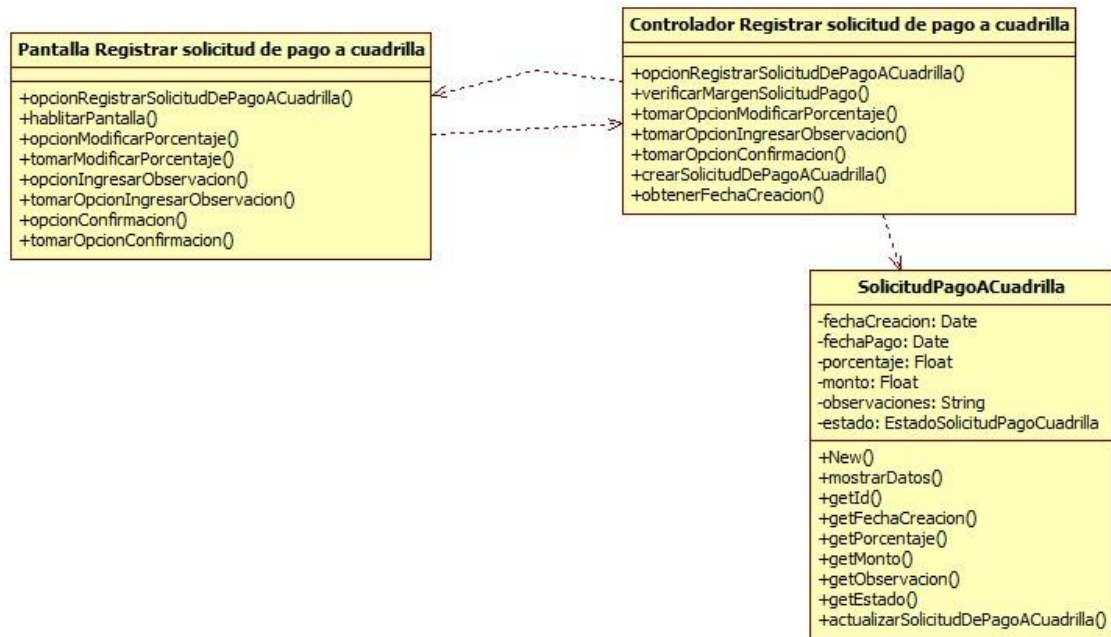






## Realización CU34: Modificar solicitud de pago a cuadrilla

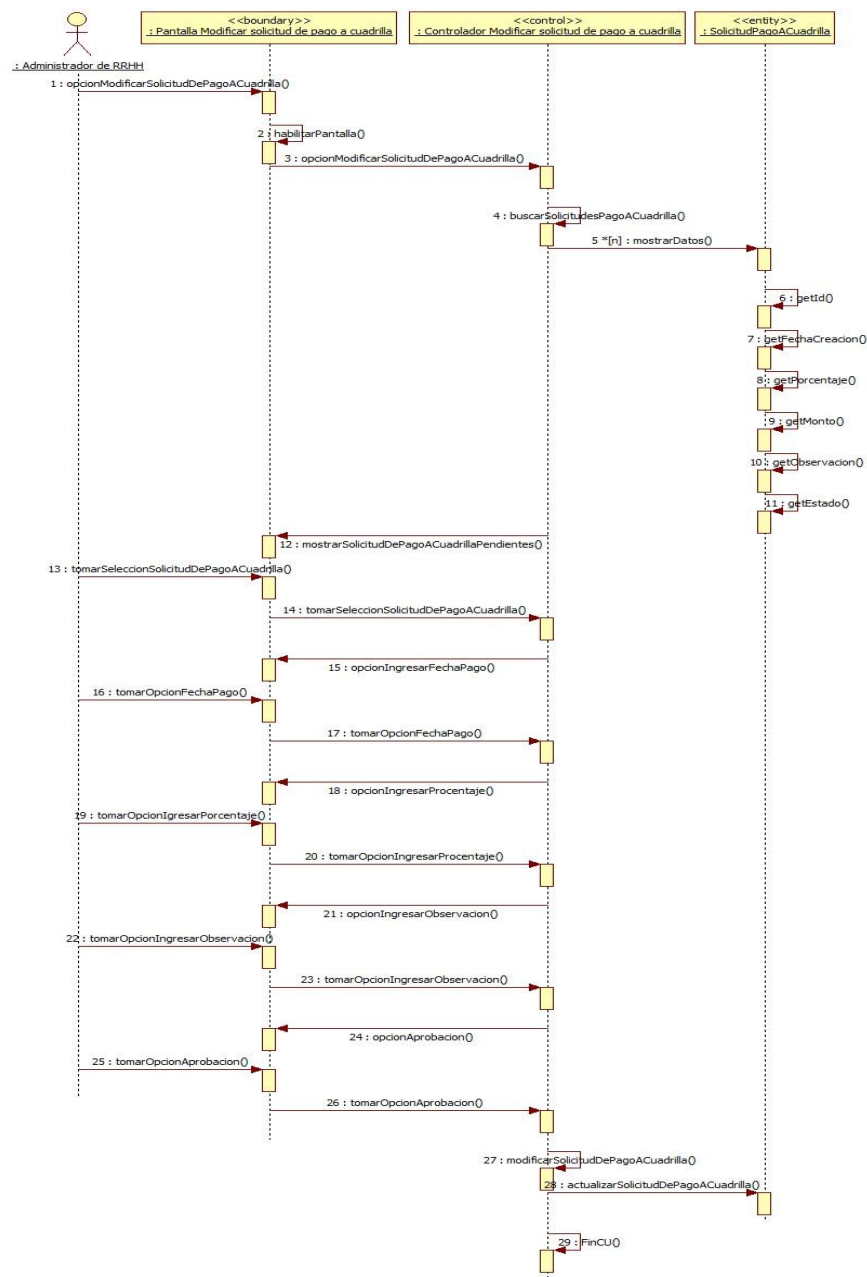
### Clases intervinientes







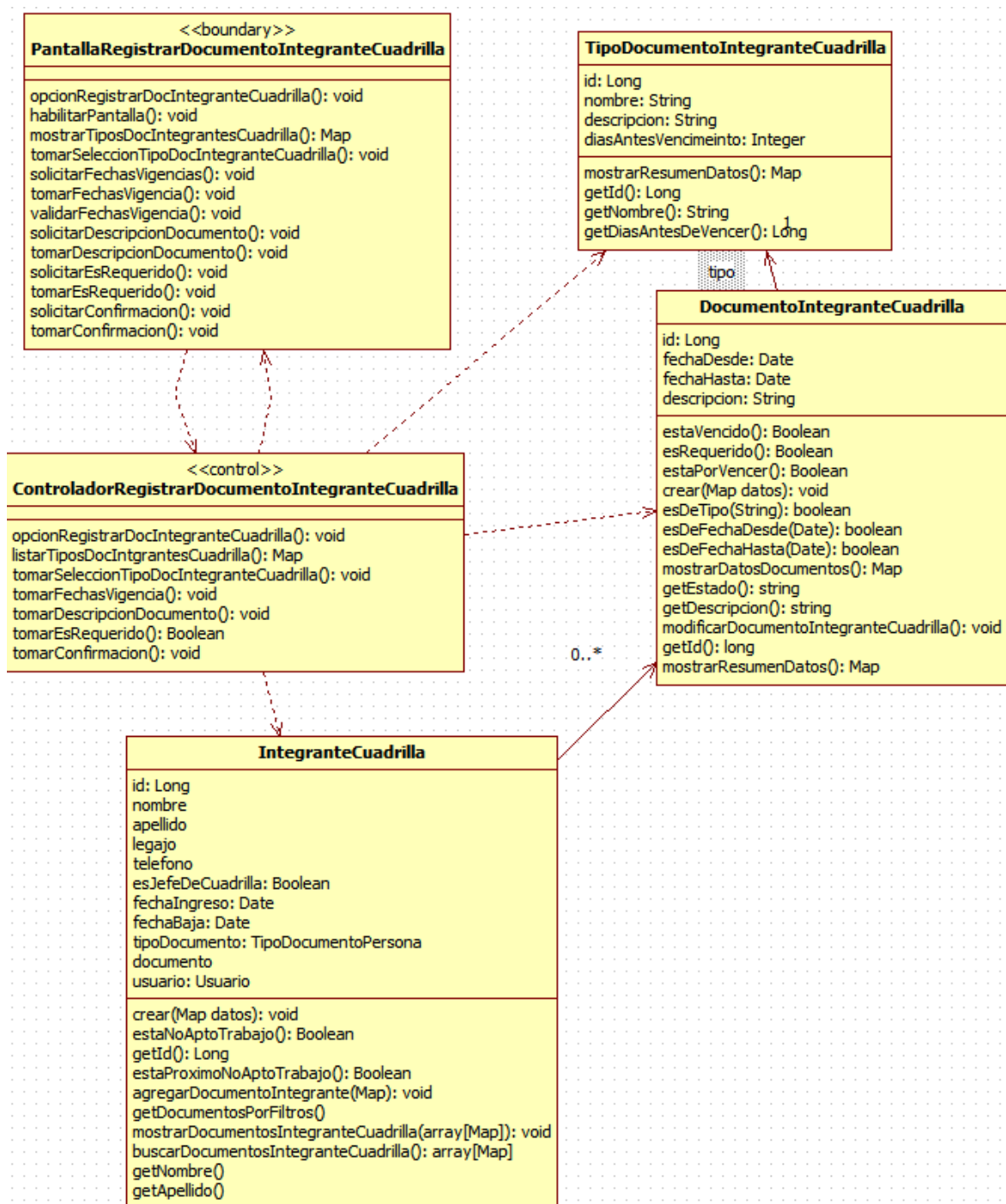
## Diagrama de secuencia





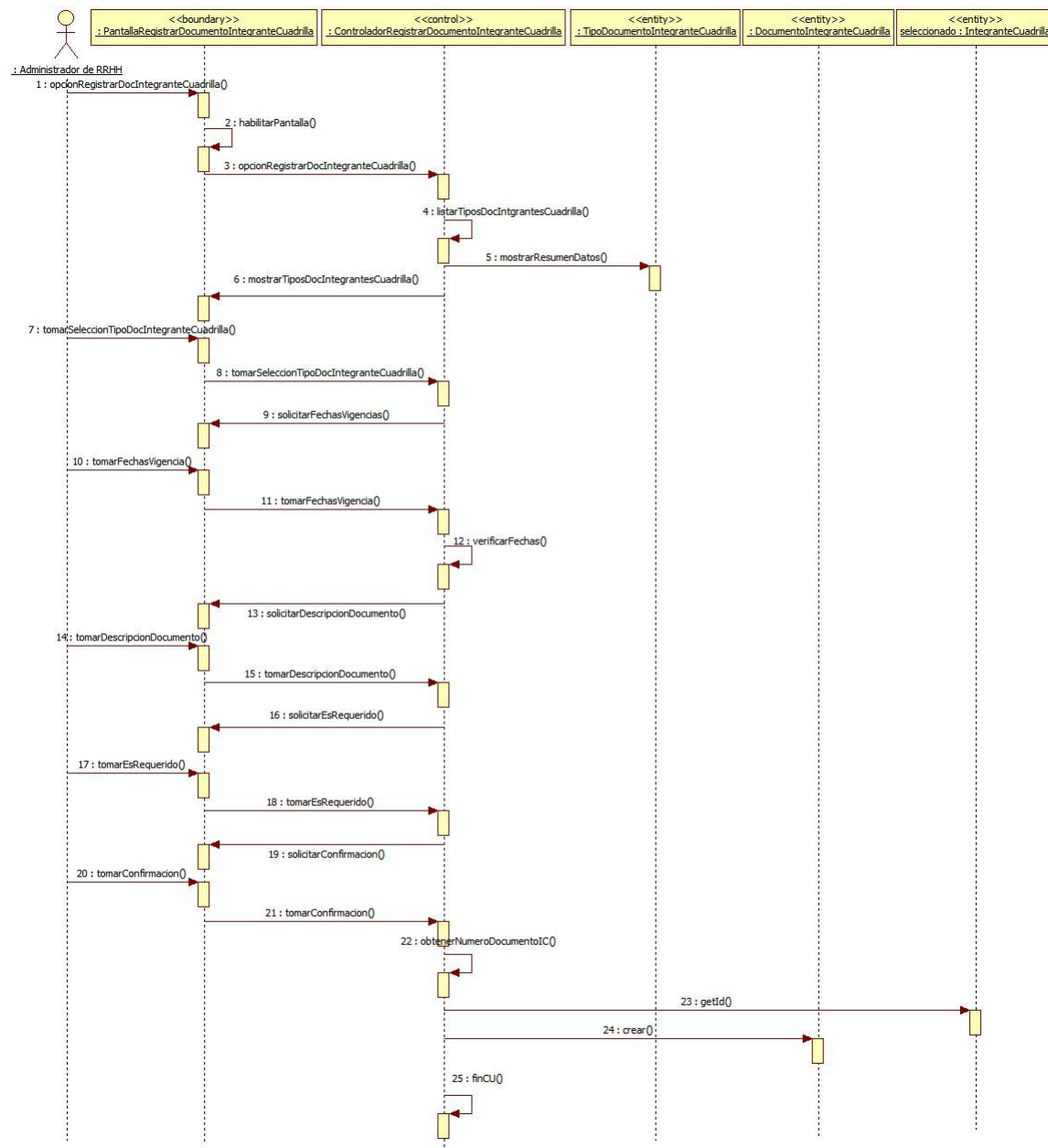
## Realización CU35: Registrar documento de integrante de cuadrilla

### Clases intervinientes





## Diagrama de secuencia

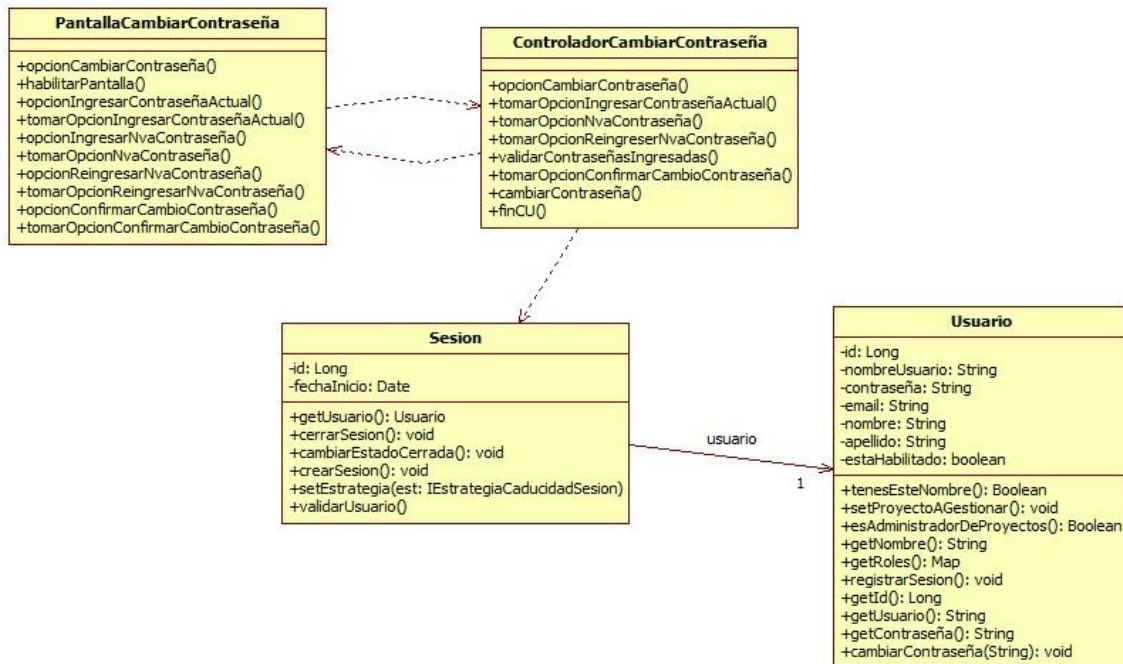




## Paquete Soporte

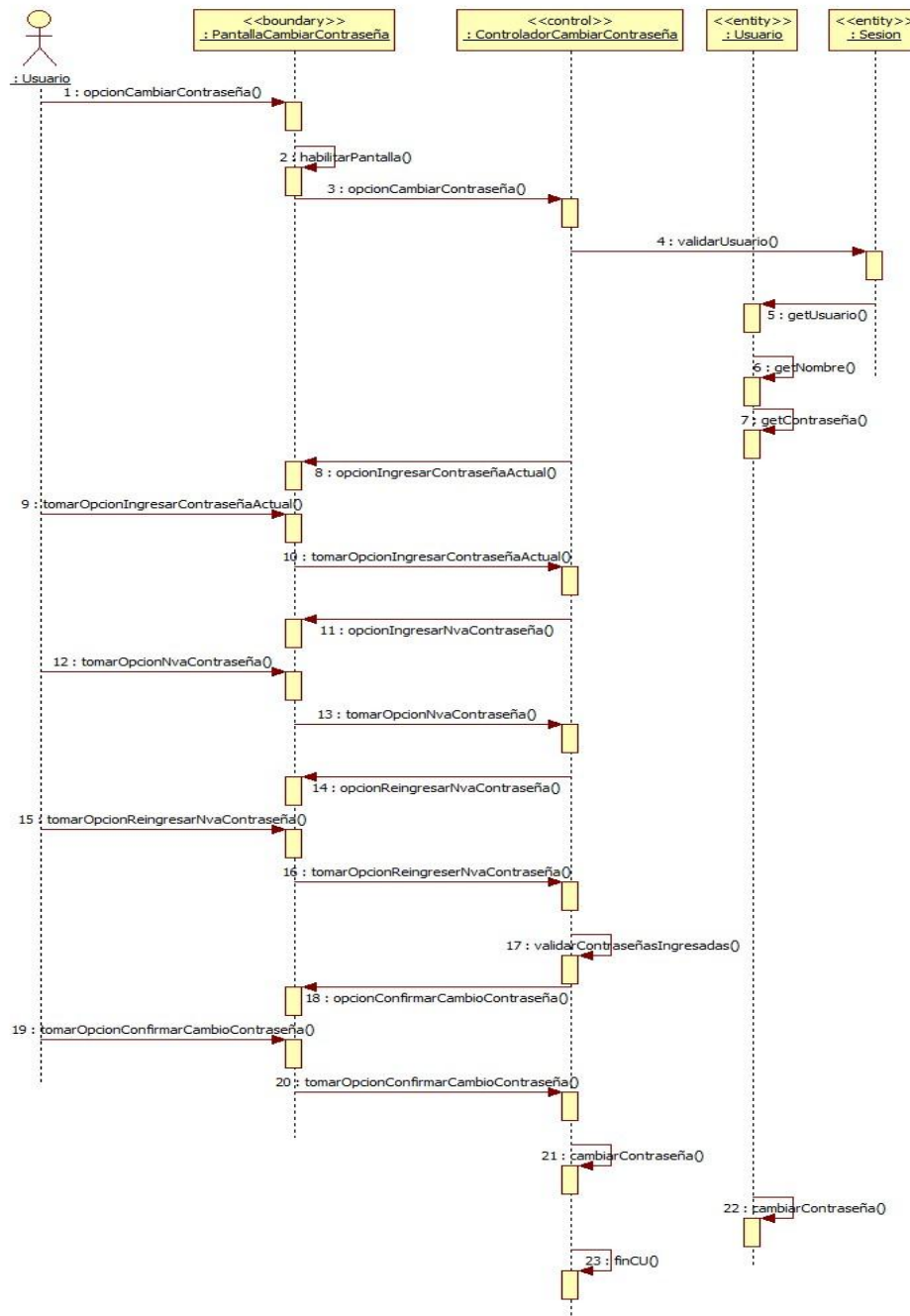
### Realización CU47: Cambiar contraseña

#### Clases intervinientes





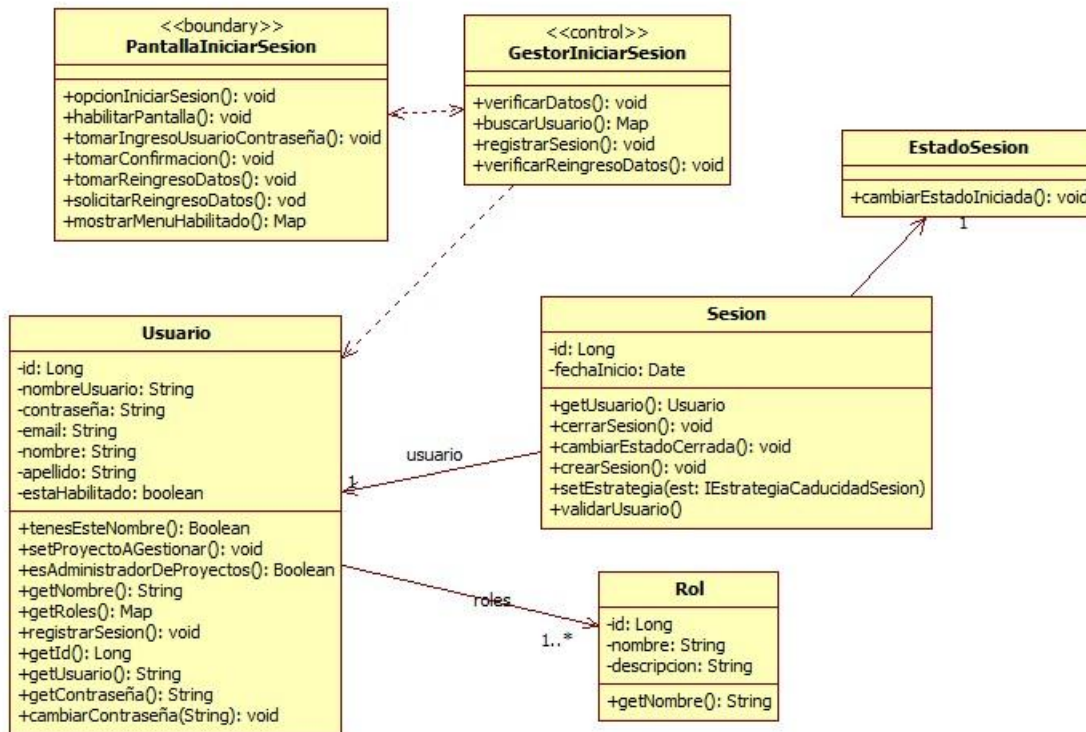
## Diagrama de secuencia





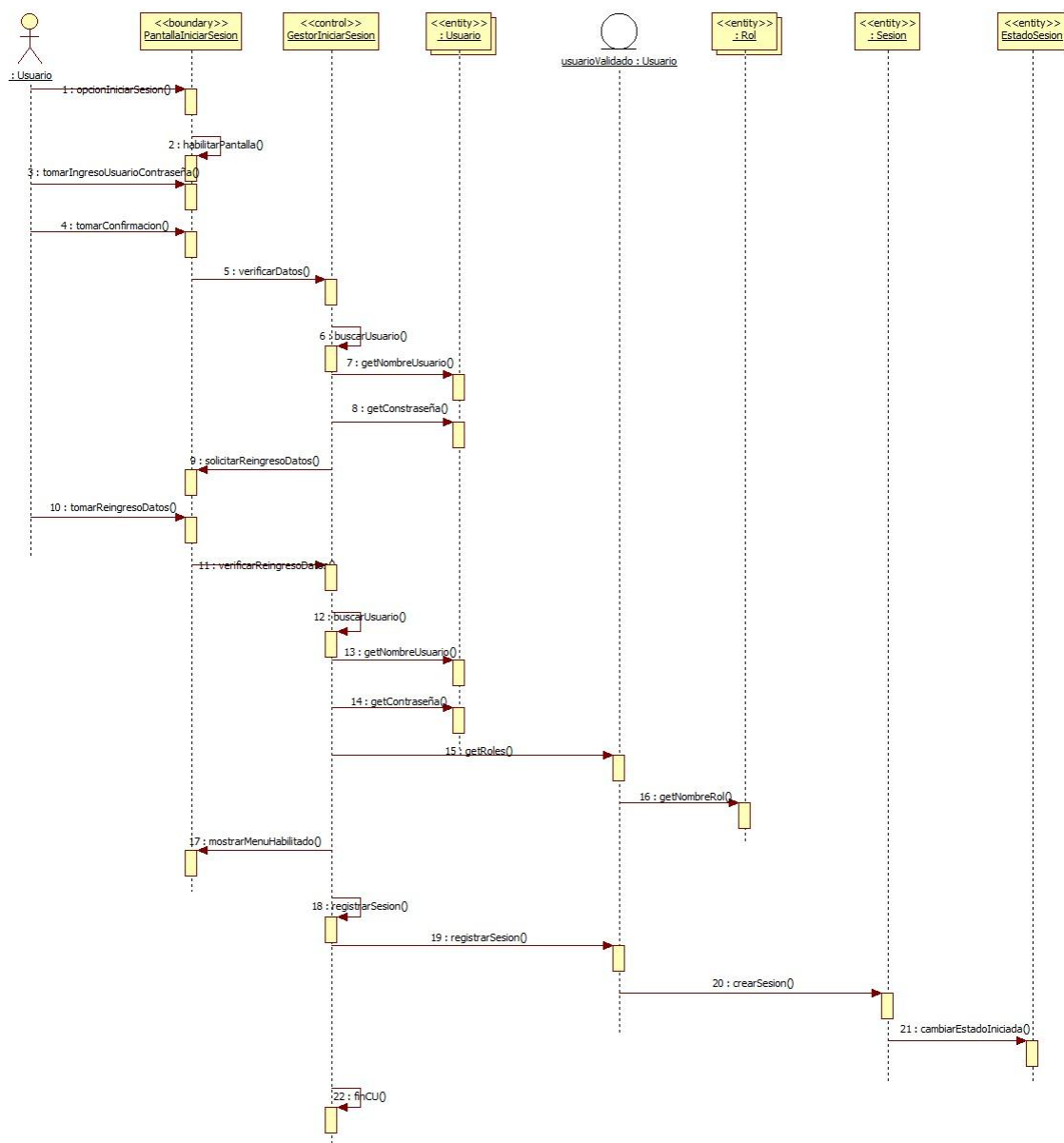
## Realización CU48: Iniciar sesión

### Clases intervinientes





## Diagrama de secuencia

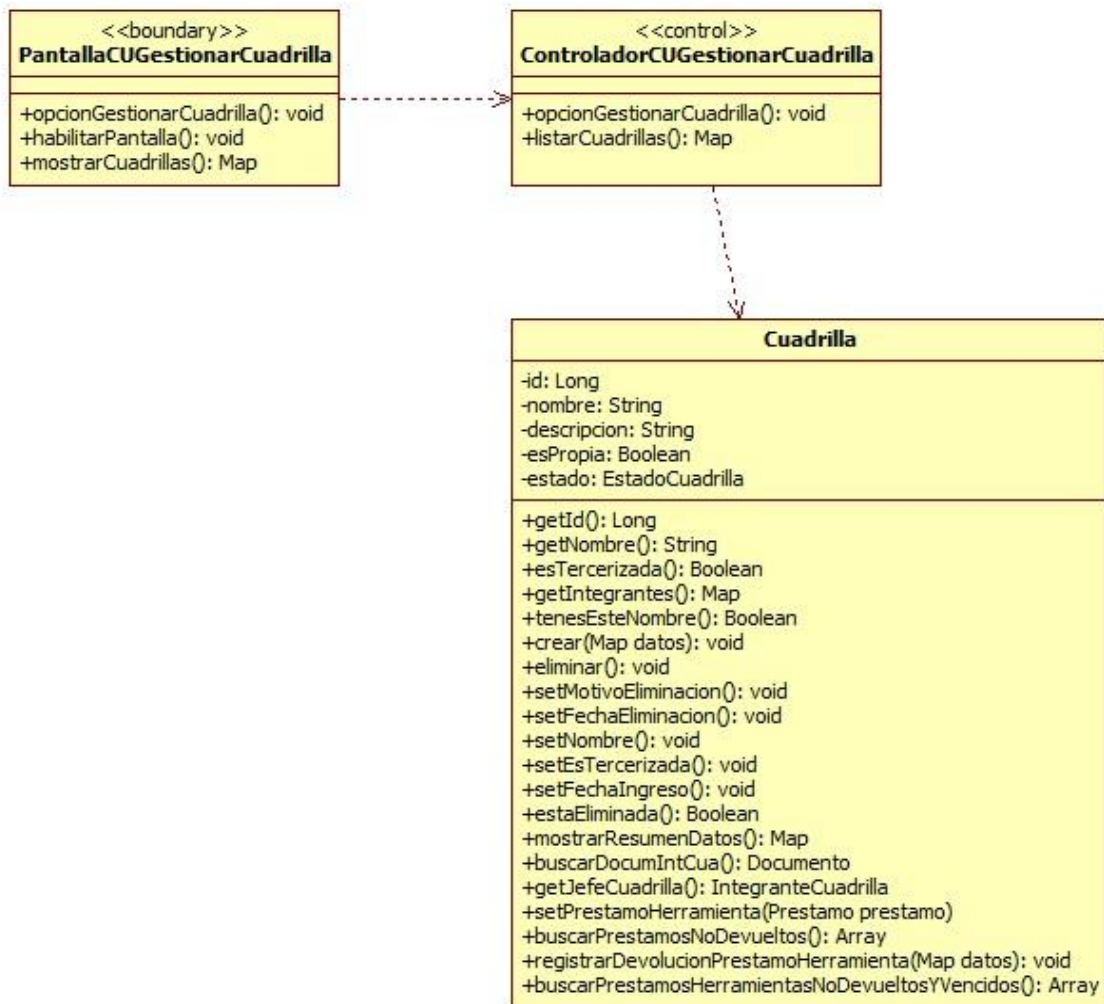






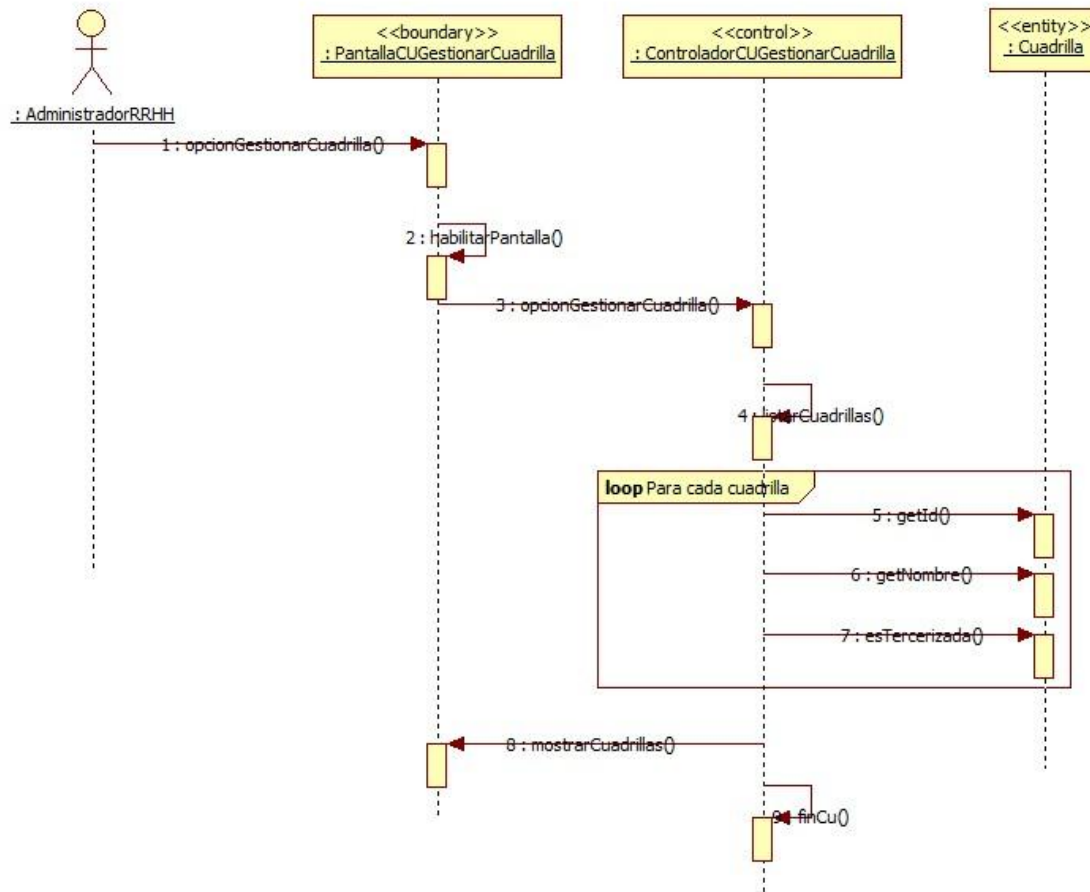
## Realización CU60: Gestionar cuadrilla

### Clases intervinientes





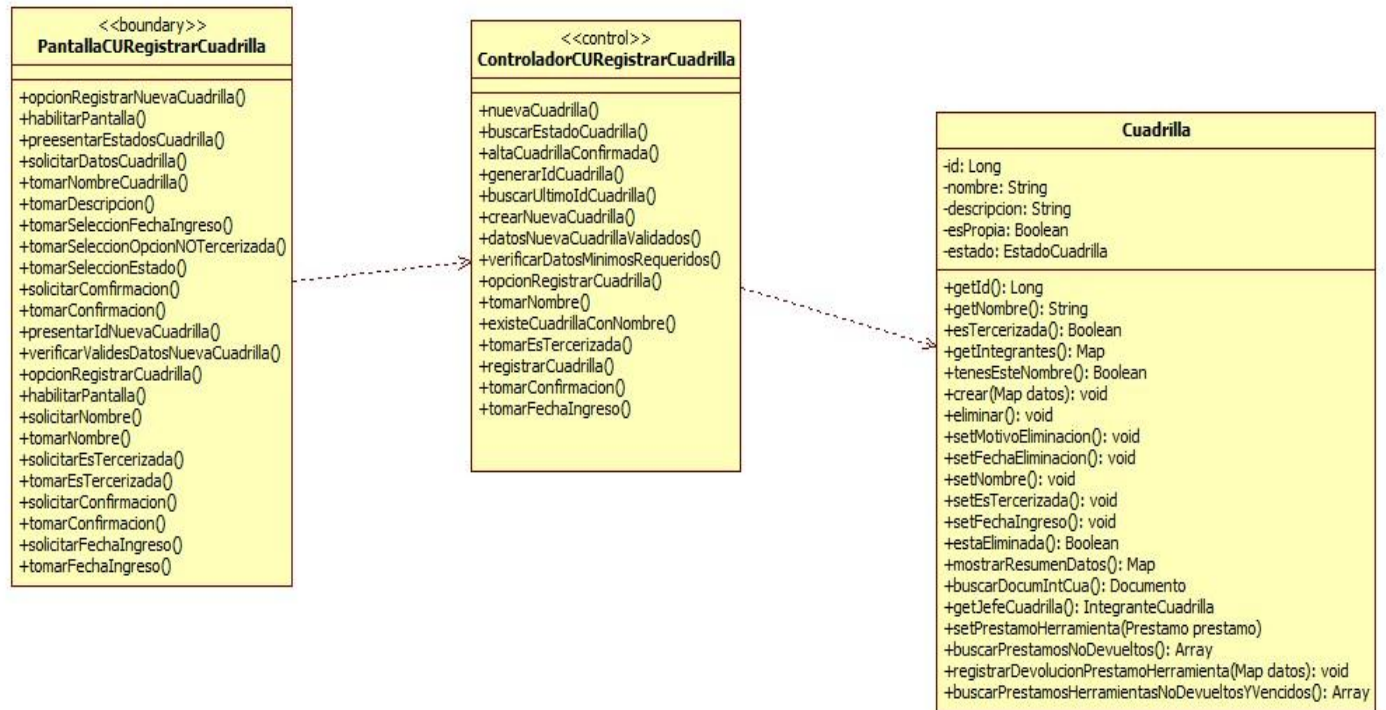
## Diagrama de secuencia





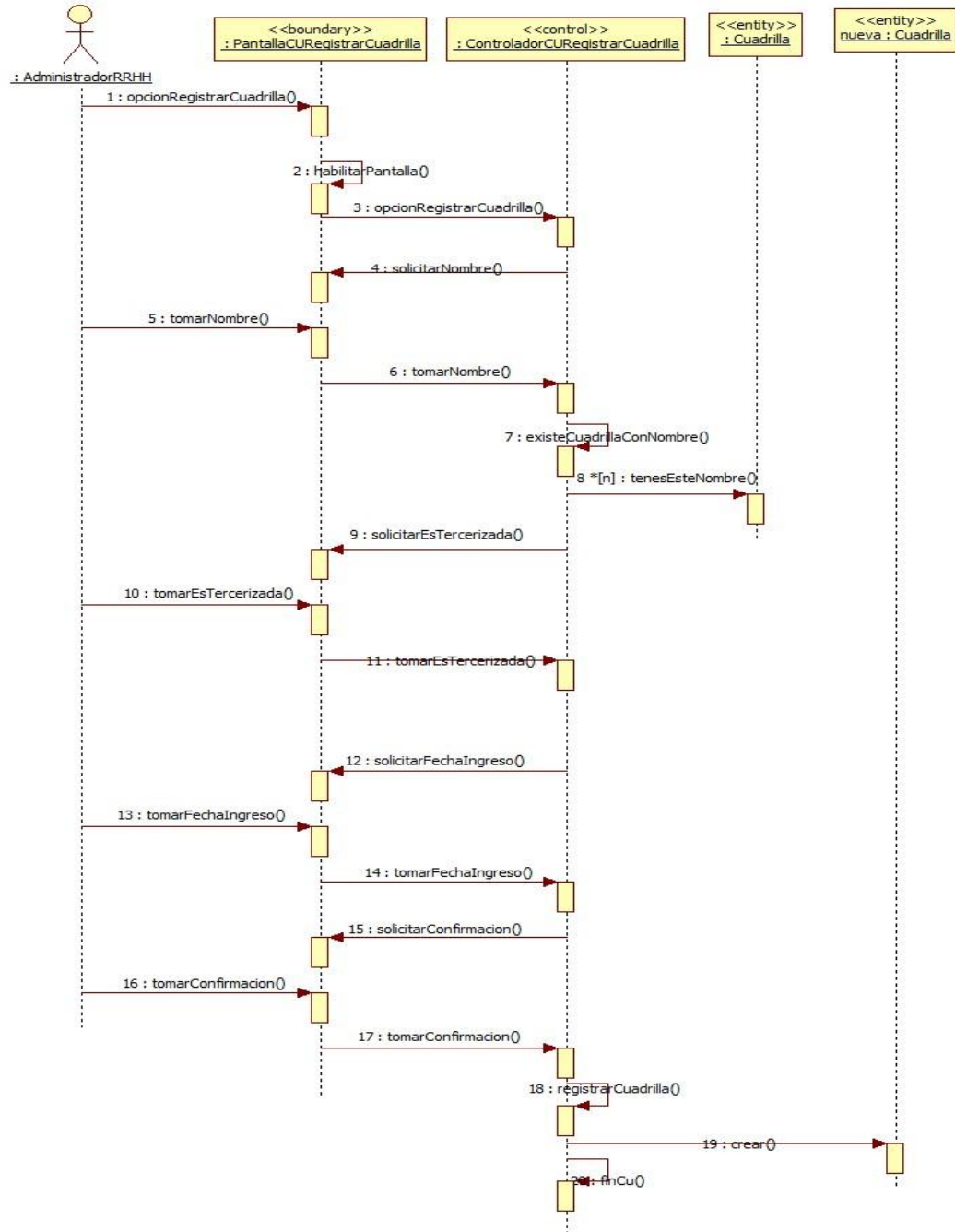
## Realización CU62: Registrar cuadrilla

### Clases intervinientes





## Diagrama de secuencia





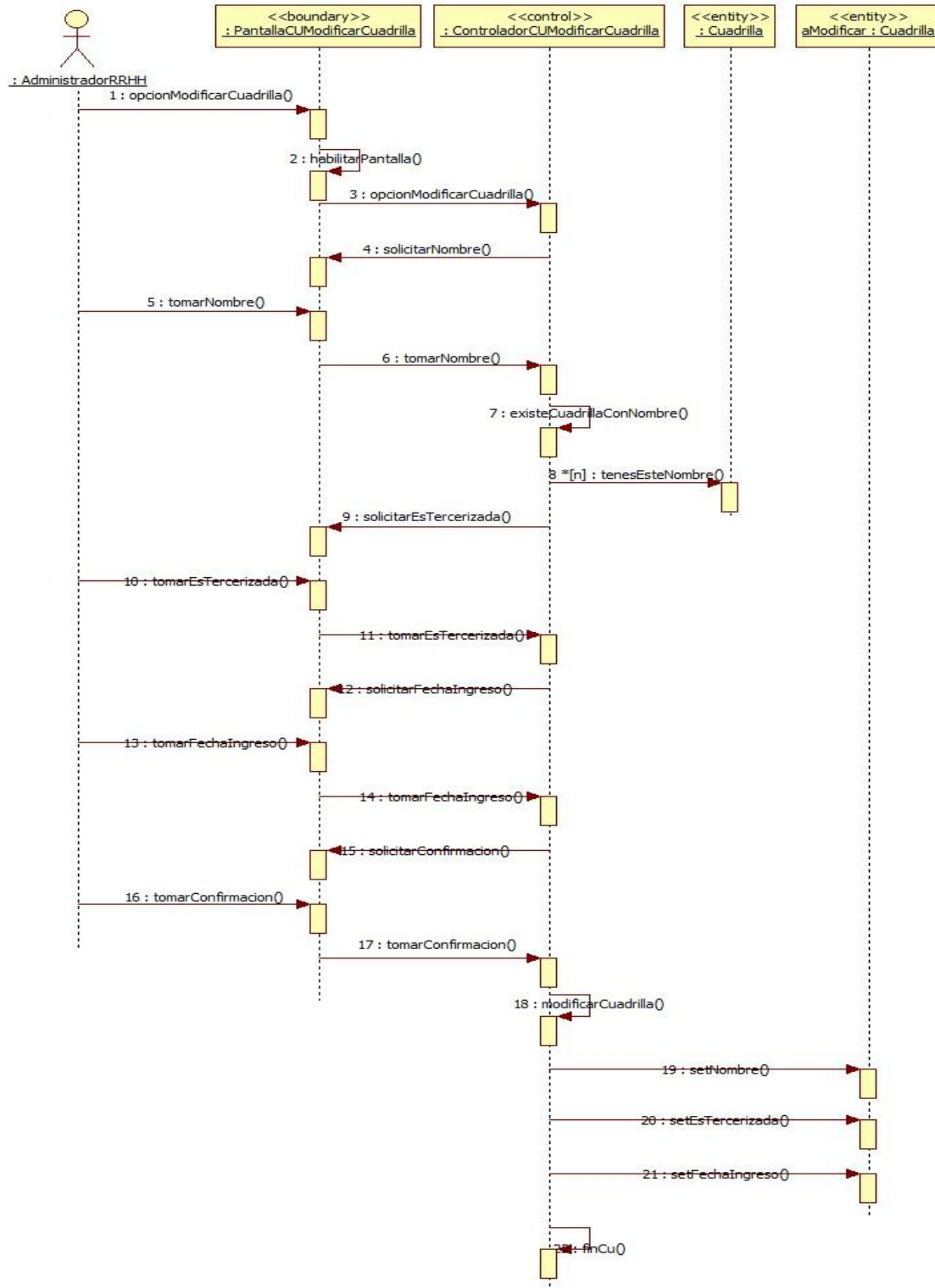
## Realización CU64: Modificar cuadrilla

### Clases intervinientes





## Diagrama de secuencia

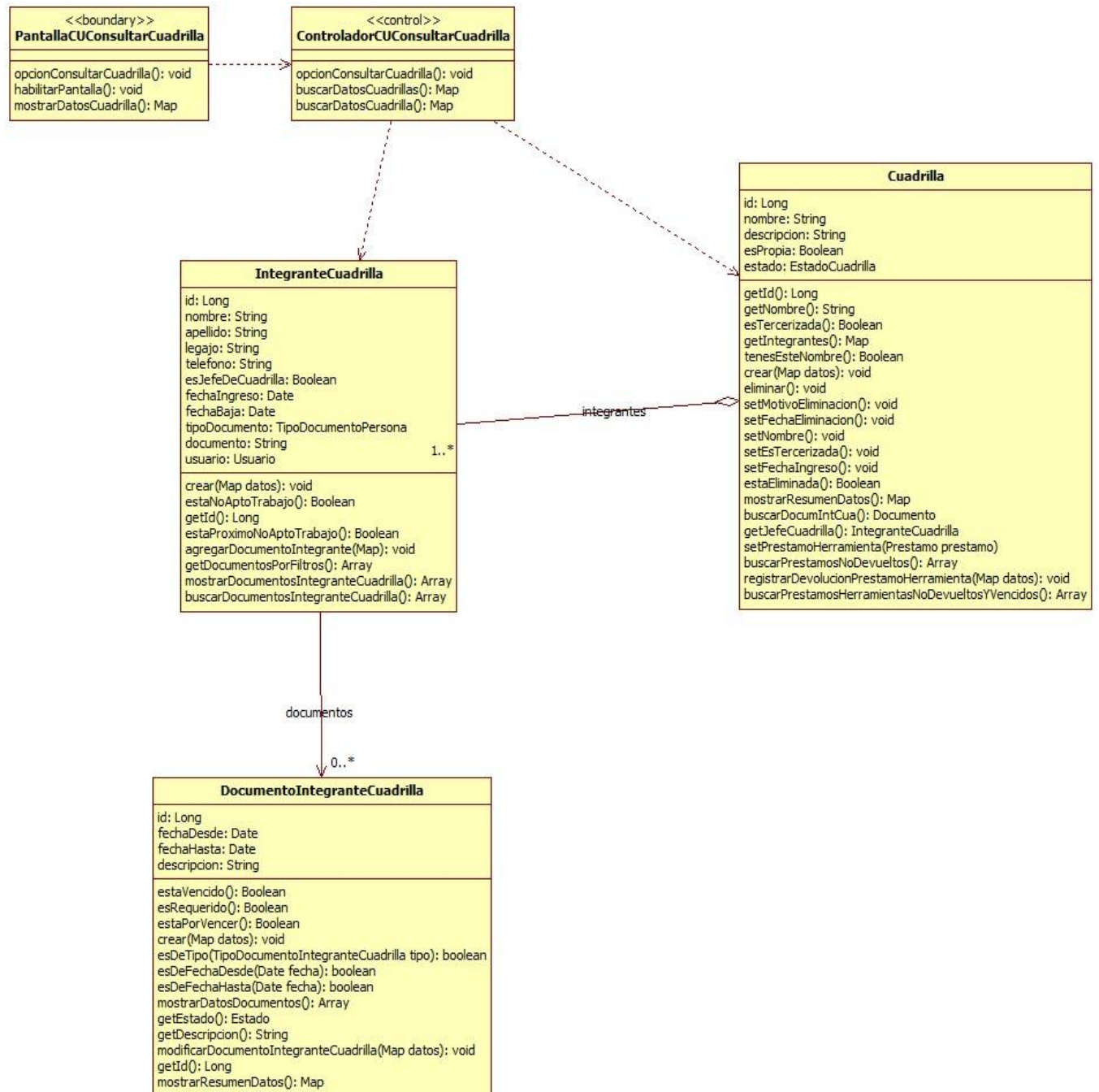






## Realización CU61: Consultar cuadrilla

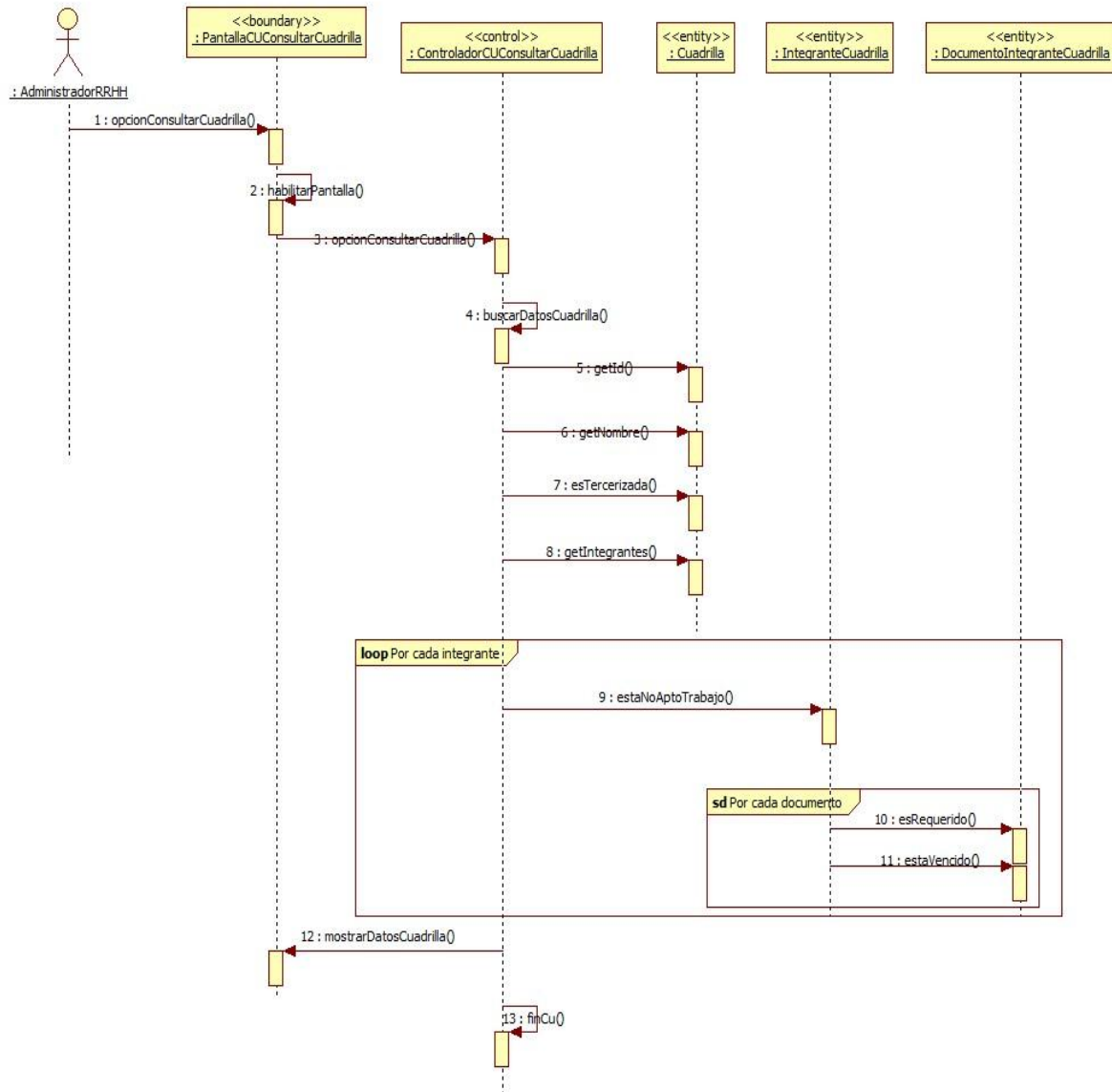
### Clases intervinientes







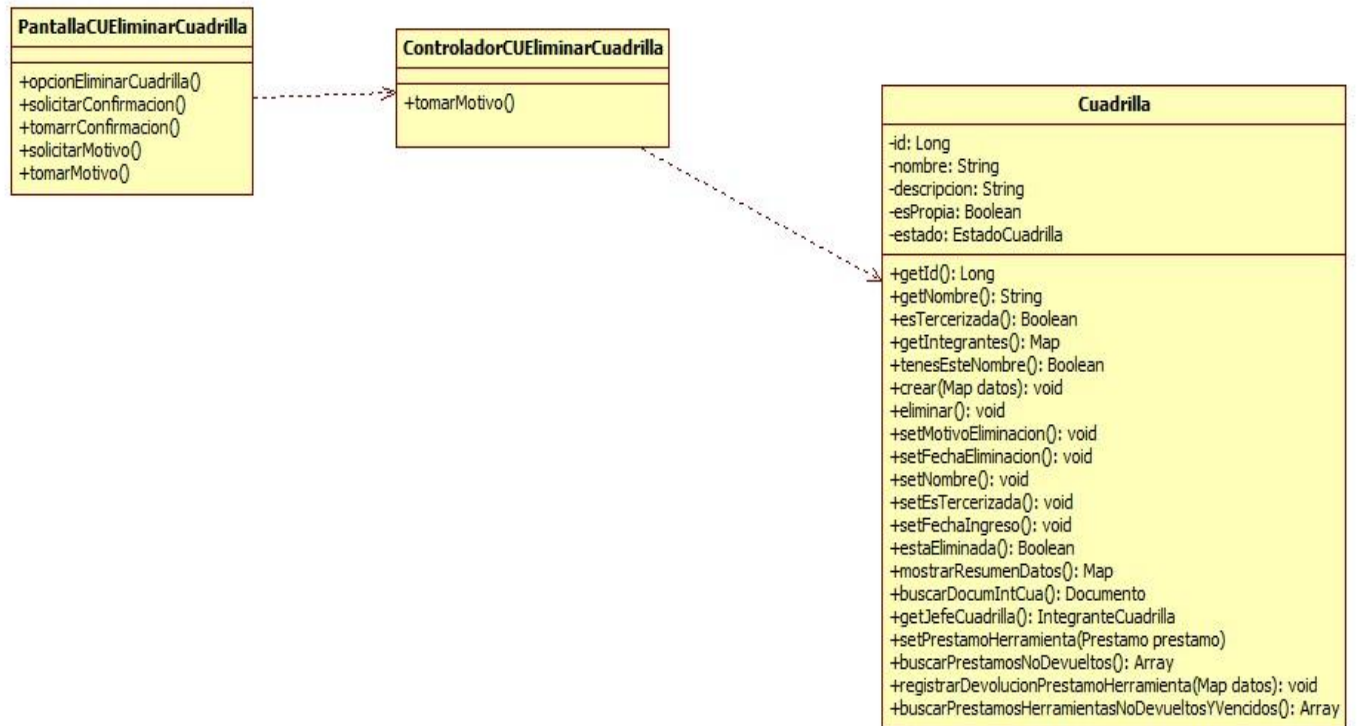
## Diagrama de secuencia





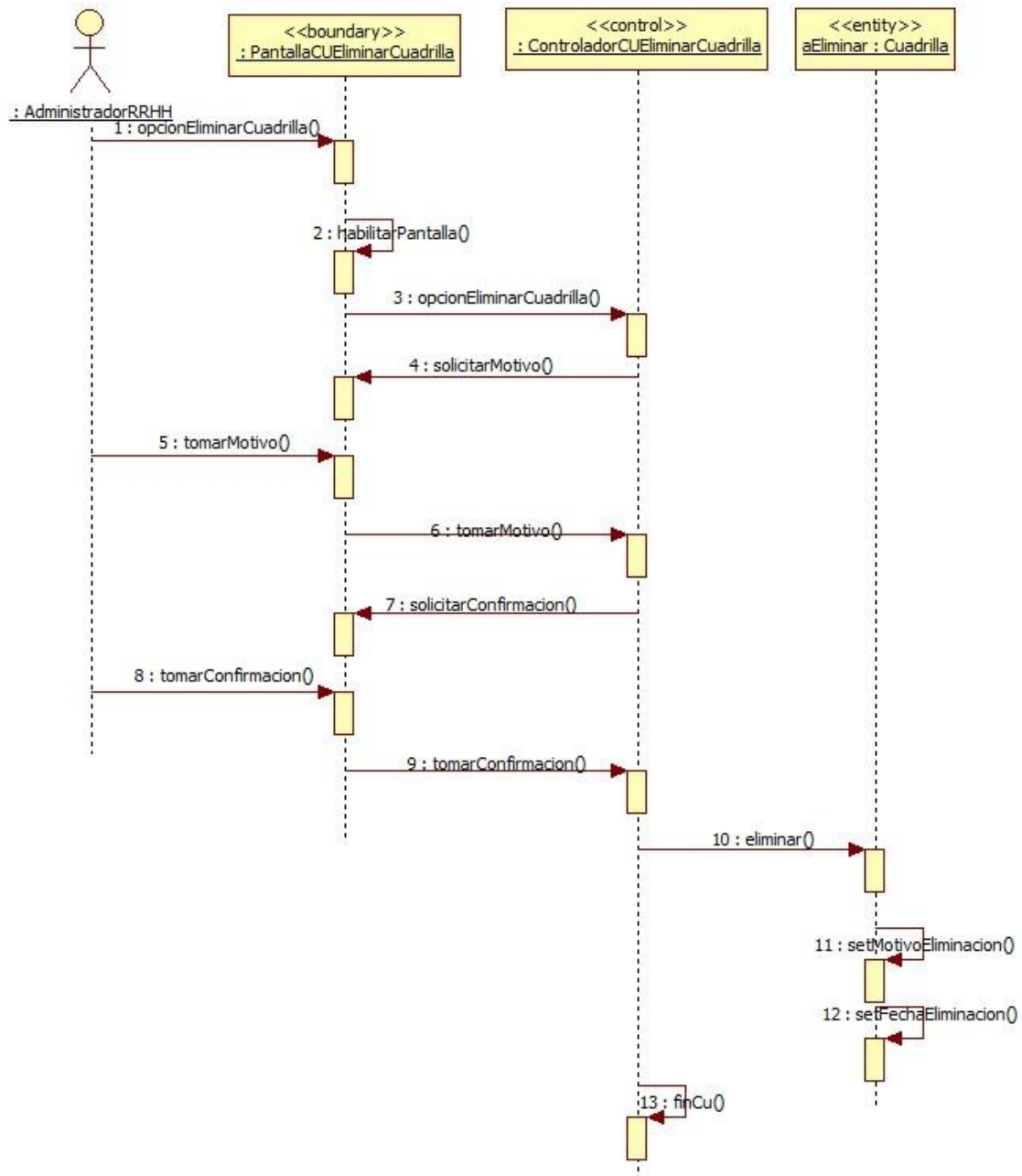
## Realización CU63: Eliminar cuadrilla

### Clases intervinientes





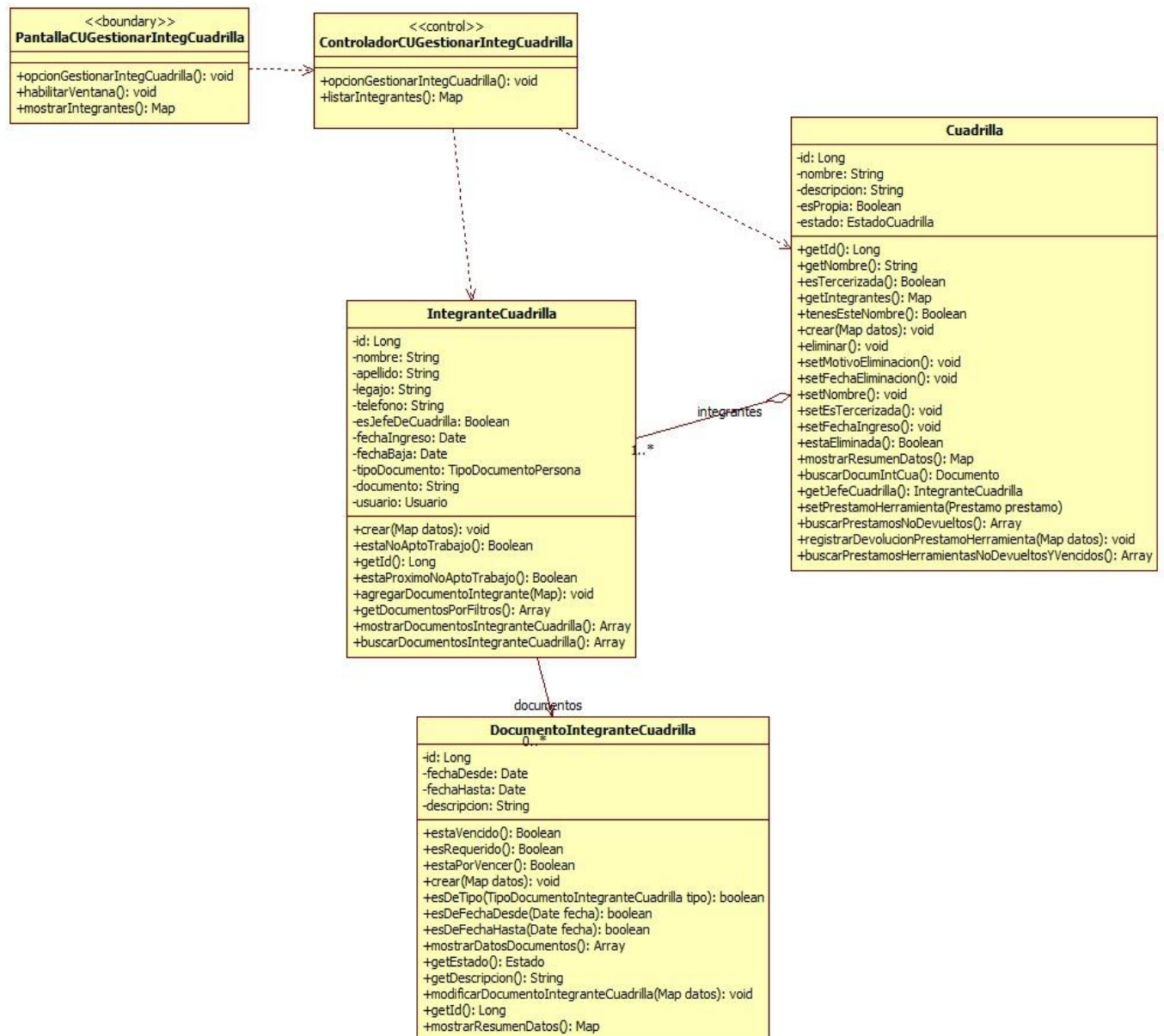
## Diagrama de secuencia





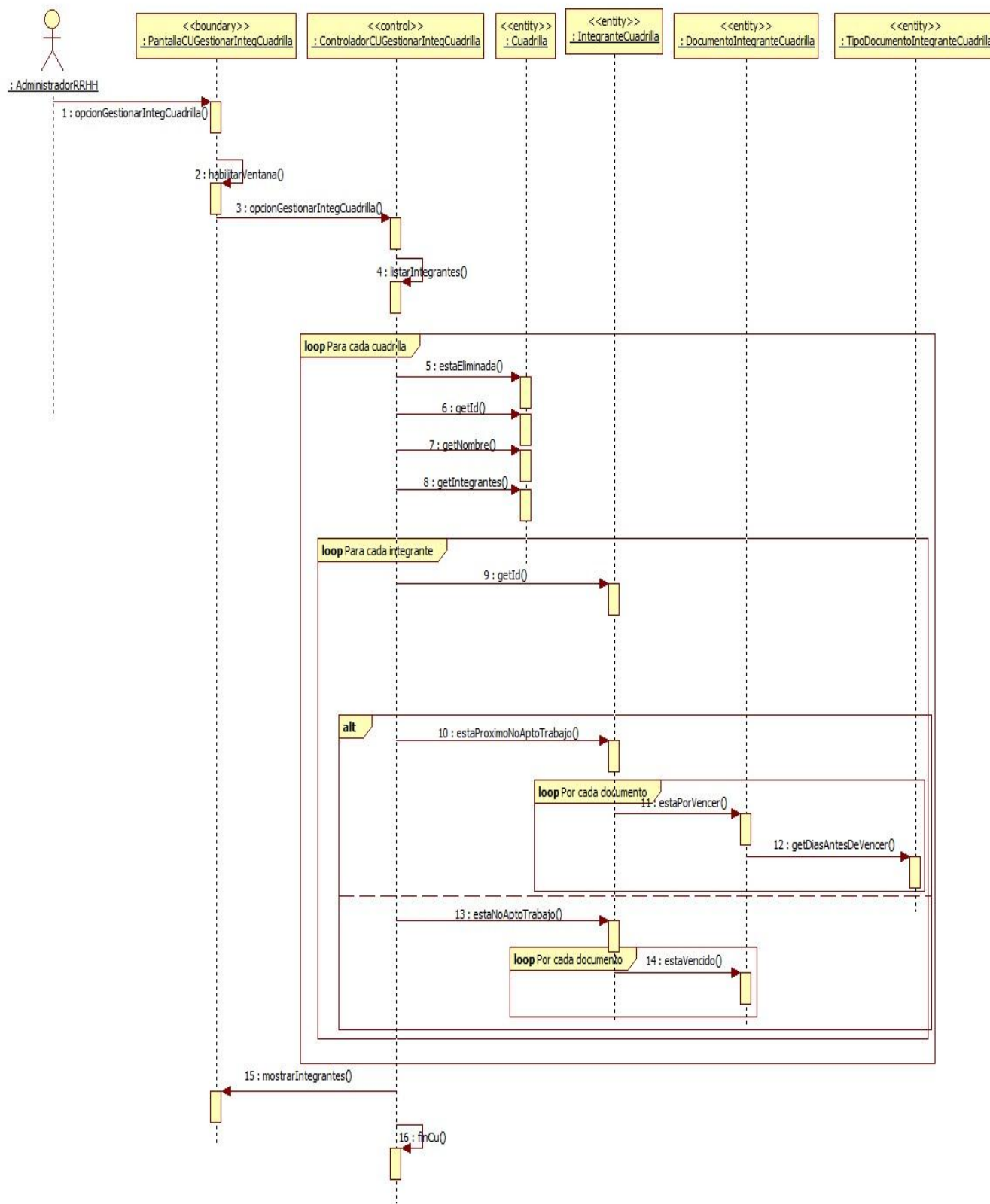
## Realización CU55: Gestionar integrante de cuadrilla

### Clases intervinientes



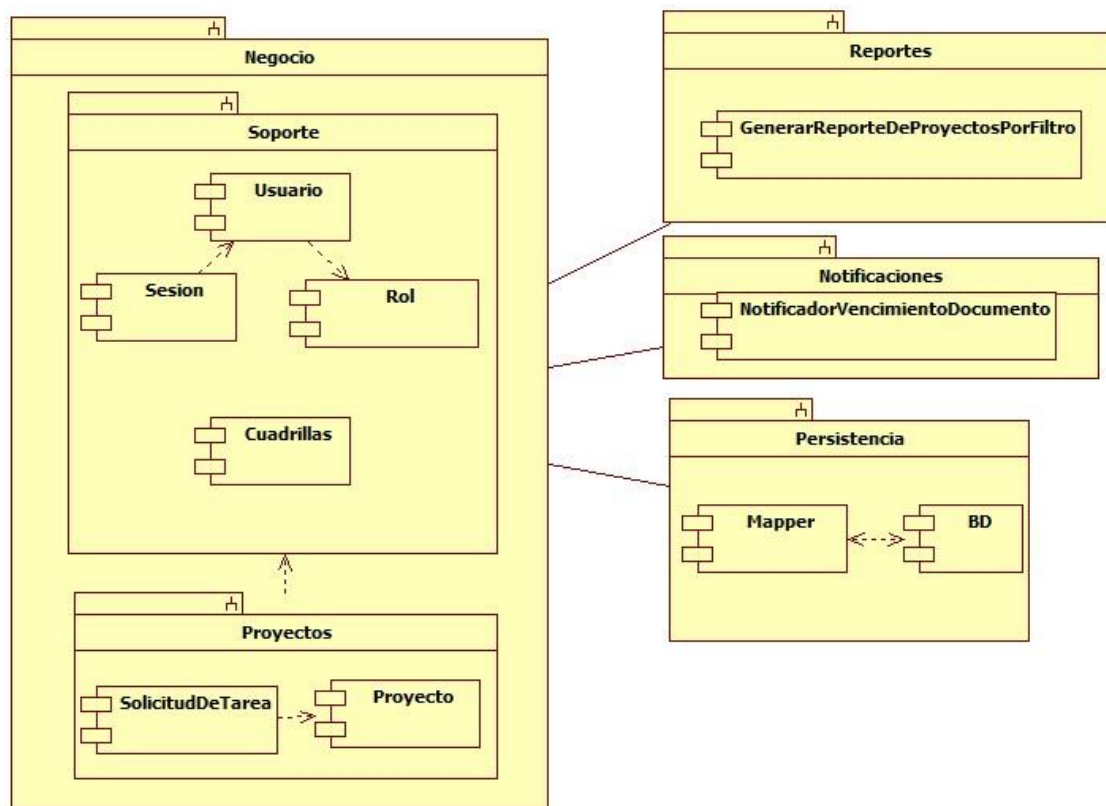


## Diagrama de secuencia





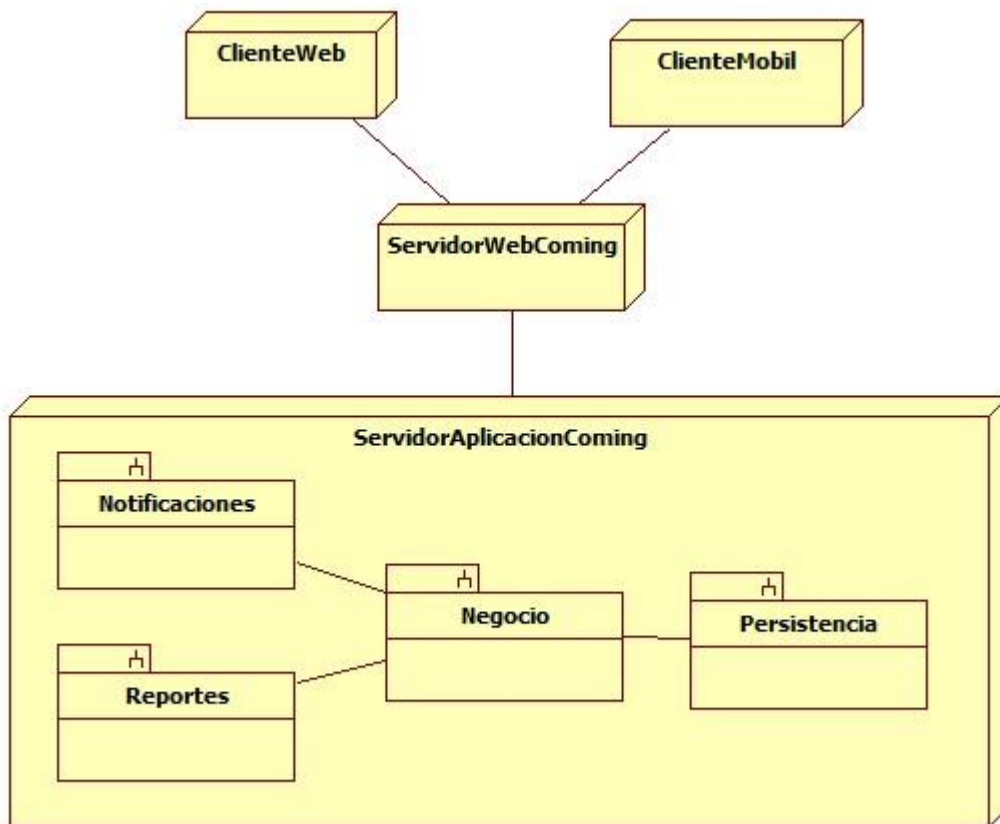
## Diagrama de componentes





## Diagramas de despliegue

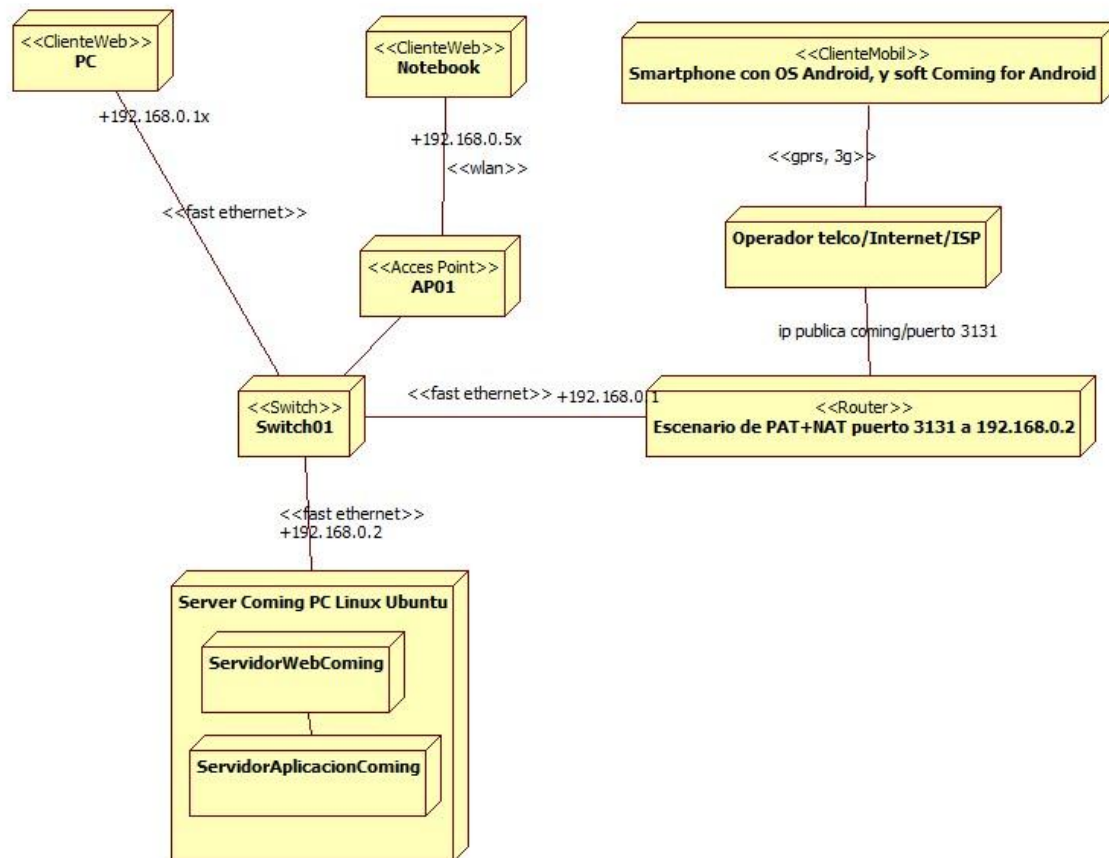
### Despliegue lógico







## Despliegue físico



## Diseño de Web Services

### Propósito

El propósito del diseño de web services es describir detalladamente los servicios web necesarios para ofrecer una interfaz a la aplicación móvil con el sistema web.



## Formatos

- Los datos que se enviarán o recibirán desde los web services deben respetar el formato JSON (<http://www.json.org/>).
- Fechas  
Las fechas hacia y desde los web services deben respetar el siguiente formato: DDMMYYYY (D: día, M: mes, Y: año). Ej 01122012 para el 01/12/2012.
- Strings  
Los Strings hacia y desde los web services no pueden ser string nulos (ej. <null> o null), los valores posibles son el string vacío ("" ) o algún string no vacío ("esto es no vacío").

## Errores

Código	Descripción
0	Éxito.
1	Fallo.



## Login

Propósito: loguea a un usuario al sistema.

Controlador: webservices

Acción: login

Método http: GET

Parámetros Url:

- nombreUsuario. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".
- password. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Respuesta Éxito:

- `{"error":{"codigo":<Long>,"descripcion":<String>},"usuario":{"id":<Long>,"nombre":<String>}}`

Donde código de error es 0.

Repuesta de fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 1.

## Tareas

Método http: GET

Propósito: busca las tareas para una cuadrilla dado el id de usuario del jefe de cuadrilla.

Controlador: mobile

Acción: tareas

Parámetros Url:

- usuarioid. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Respuestas Éxito:

- `{"error":{"codigo":<Long>,"descripcion":<String>},"tareas":[{"id":<Long>,"nombreTipoTarea":<String>,"fechaInicioEstimada":<String>,"fechaFinEstimada":<String>,"fechaInicioReal":<String>,"fechaFinReal":<String>,"estado":<String>,"descripcion":<String>}, ..., { ... }]}`

Donde código de error es 0.

Respuestas Fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 1.

Método http: POST

Propósito: actualiza una tarea dada.

Controlador: mobile

Acción: tareas

Parámetros Url:

- tareaid. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Cuerpo del POST:

- Tarea en formato JSON. La aplicación móvil enviara este JSON:  
`{"observaciones":<String>,"estado":<String>,"fechaFinReal":<String>,"fechaInicioReal":<String>,"id":<Long>,"nombreTipoTarea":<String>}`

Donde "estado" es el nombre del estado.

Respuestas Éxito o fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 0 o 1 según corresponda.



## Acontecimientos

Propósito: busca los acontecimientos de una tarea.

Método http: GET

Controlador: Mobile

Acción: acontecimientos

Parámetros Url:

- tareald. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Respuestas Éxito:

- `{"error":{"codigo":<Long>,"descripcion":<String>}acontecimientos":[{"id":<Long>,"nombreTipo":<String>,"usuariold":<String>,"fechaCreacion":<String>,"descripcion":<String>}]}`

Donde código de error es 0.

Respuestas Fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 1.

Método http: POST

Propósito: registra un acontecimiento de una tarea.

Controlador: mobile

Acción: tarea

Parámetros Url:

- tareald. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Cuerpo del POST:

- Acontecimiento en formato JSON. La aplicación móvil enviara este JSON:  
`{"usuariold":<Long>,"descripcion":<String>,"fechaCreacion":<String>,"nombreTipo":<String>}`

Respuestas Éxito o fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 0 o 1 según corresponda.

Método http: PUT

Propósito: modifica un acontecimiento de una tarea.

Controlador: mobile

Acción: acontecimientos

Parámetros Url:

- acontecimientold. Tipo String. Valores posibles: cualquier cadena no nula, incluye a la cadena vacía "".

Cuerpo del PUT:

- Acontecimiento en formato JSON. La aplicación móvil enviara este JSON:  
`{"usuariold":<Long>,"descripcion":<String>,"fechaCreacion":<String>,"nombreTipo":<String>}`

Respuestas Éxito o fracaso:

- `{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 0 o 1 según corresponda.



## Tipos de Acontecimientos

Método http: GET

Propósito: Busca los tipos de acontecimientos.

Controlador: mobile

Acción: tipoAcontecimientos

Respuestas Éxito:

- `{"error":{"codigo":<Long>,"descripcion":<String>},"tipoAcontecimientos":[{"nombre":<String>}]}`

Donde código de error es 0.

Respuestas Fracaso:

`{"error":{"codigo":<Long>,"descripcion":<String>}}`

Donde código de error es 1.



## Diseño de datos

### Diagrama de entidad relación

A continuación se muestra el diagrama de entidad relación de la base de datos que fue generado por el framework de desarrollo Groovy and Grails a partir de las clases implementadas.

