

diseño Iteración 2

Alexander Moreno Borrego

Carlos Jesús Fernández Basso

Francisco Santolalla Quiñonero



17 de ABRIL de 2013

Universidad de Granada

Contenido

[Control de Versiones 3](#_Toc354009983)

[Modelado de requisitos 4](#_Toc354009984)

[Modelado funcional – Diagramas de casos de uso 4](#_Toc354009985)

[Casos de uso: Gestión de actividades 6](#_Toc354009986)

[Casos de uso: Gestión categorías 10](#_Toc354009987)

[Casos de uso: Gestión de equipos 14](#_Toc354009988)

[Casos de uso: Gestión de grupos 18](#_Toc354009989)

[Casos de uso: Gestión de pagos 22](#_Toc354009990)

[Casos de uso: Gestión de temporadas 26](#_Toc354009991)

[Casos de uso: Gestión de instalaciones 30](#_Toc354009992)

[Identificar subsistemas – Diagramas de casos de uso 34](#_Toc354009993)

[Diagrama de CU: Gestión de Actividades 34](#_Toc354009994)

[Diagrama de CU: Gestión de Grupos 35](#_Toc354009995)

[Diagrama de CU: Gestión de Temporadas 36](#_Toc354009996)

[Diagrama de CU: Gestión de Equipos 37](#_Toc354009997)

[Diagrama de CU: Gestión de Instalaciones 38](#_Toc354009998)

[Diagrama de CU: Gestión de Pagos 39](#_Toc354009999)

[Diagrama de CU: Gestión de Categorias 40](#_Toc354010000)

[Requisitos no funcionales 41](#_Toc354010001)

[Operaciones del sistema – Diagramas de secuencia de sistema 42](#_Toc354010002)

[Gestión de Actividades 42](#_Toc354010003)

[*Diagrama de secuencia:Insertar Actividad* 42](#_Toc354010004)

[*Diagrama de secuencia:Consultar Actividad* 43](#_Toc354010005)

[*Diagrama de secuencia:Eliminar Actividad* 44](#_Toc354010006)

[*Diagrama de secuencia: ModificarActividad* 45](#_Toc354010007)

[Gestión de Categorías 46](#_Toc354010008)

[*Diagrama de secuencia: Insertar Categoría* 46](#_Toc354010009)

[*Diagrama de secuencia: Consultar Categoría* 47](#_Toc354010010)

[*Diagrama de secuencia: Eliminar Categoría* 48](#_Toc354010011)

[*Diagrama de secuencia: Modificar Categoría* 48](#_Toc354010012)

[Gestión de Equipos 49](#_Toc354010013)

[*Diagrama de secuencia: Insertar Equipo* 49](#_Toc354010014)

[*Diagrama de secuencia: Consultar Equipo* 50](#_Toc354010015)

[*Diagrama de secuencia: Eliminar Equipo* 51](#_Toc354010016)

[*Diagrama de secuencia: Modificar Equipo* 52](#_Toc354010017)

[Gestión de Grupos 53](#_Toc354010018)

[*Diagrama de secuencia: Insertar Grupo* 53](#_Toc354010019)

[*Diagrama de secuencia: Consultar Grupo* 54](#_Toc354010020)

[*Diagrama de secuencia: Eliminar Grupo* 55](#_Toc354010021)

[*Diagrama de secuencia: Modificar Grupo* 56](#_Toc354010022)

[Gestión de Instalaciones 57](#_Toc354010023)

[*Diagrama de secuencia: Insertar Instalaciones* 57](#_Toc354010024)

[*Diagrama de secuencia: Consultar Instalaciones* 58](#_Toc354010025)

[*Diagrama de secuencia: Eliminar Instalaciones* 59](#_Toc354010026)

[*Diagrama de secuencia: Modificar Instalaciones* 60](#_Toc354010027)

[Gestión de Pagos 61](#_Toc354010028)

[*Diagrama de secuencia: Insertar Pago* 61](#_Toc354010029)

[*Diagrama de secuencia: Consultar Pago* 62](#_Toc354010030)

[*Diagrama de secuencia: Eliminar Pago* 62](#_Toc354010031)

[*Diagrama de secuencia: Modificar Pago* 63](#_Toc354010032)

[Gestión de Temporadas 64](#_Toc354010033)

[*Diagrama de secuencia: Insertar Temporada* 64](#_Toc354010034)

[*Diagrama de secuencia: Consultar Temporada* 65](#_Toc354010035)

[*Diagrama de secuencia: Eliminar Temporada* 66](#_Toc354010036)

[*Diagrama de secuencia: Modificar Temporada* 67](#_Toc354010037)

[Análisis 68](#_Toc354010038)

[Identificar clases, atributos y relaciones 68](#_Toc354010039)

[Clase: Actividad 68](#_Toc354010040)

[Clase: Categoría 68](#_Toc354010041)

[Clase: Equipo 69](#_Toc354010042)

[Clase: Grupo Entrenamiento 69](#_Toc354010043)

[Clase: Pago Temporada (clase de asociación entre las clases Temporada y Alumno) 69](#_Toc354010044)

[Clase: Pago Actividad (clase de asociación entre Alumno y Actividad) 70](#_Toc354010045)

[Clase: Cuota Precio 70](#_Toc354010046)

[Clase: Temporada 70](#_Toc354010047)

[Clase: Instalación 71](#_Toc354010048)

[Clase: Horario (clase de asociación entre Instalación y GrupoEntrenamiento) 71](#_Toc354010049)

[Modelado estático – Diagrama de clases 72](#_Toc354010050)

[Modelado del comportamiento externo – Contratos 74](#_Toc354010051)

[Contratos de los casos de uso de la Gestión de Actividades. 74](#_Toc354010052)

[Contratos de Caso de uso: Insertar Actividad. 74](#_Toc354010053)

[Contratos de Caso de uso: Consultar Actividad. 76](#_Toc354010054)

[Contratos de Caso de uso: Eliminar Actividad. 76](#_Toc354010055)

[Contratos de Caso de uso: Insertar/Eliminar alumnos Actividad. 78](#_Toc354010056)

[Contratos de Caso de uso: Insertar/Eliminar instalaciones Actividad. 78](#_Toc354010057)

# **Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 10/3/2013 | 2.1 | Añadidos casos de uso (instalaciones, equipos, grupos pagos, temporadas, actividades), diagramas de casos de uso y diagramas de secuencia |
| 17/4/2013 | 2.2 | Realizadas las modificaciones y correcciones indicadas por el equipo de planificación y el profesor |
|  | 2.3 | Realizadas la tarea de análisis de la fase de diseño |

# **Modelado de requisitos**

## Modelado funcional – Diagramas de casos de uso

En esta etapa del modelado de requisitos se captura el propósito general del sistema:

* Se analiza qué debe hacer el sistema.
* Se obtiene una visión contextualizada del sistema.
* Identifica y delimita el sistema.
* Se determinan características, cualidades y restricciones que debe satisfacer el sistema.

Los requisitos funcionales que se han obtenido en el sistema son los siguientes:

* Un entrenador puede modificar los datos de sus alumnos relativos al grupo (comentarios, observaciones).
* El administrador es el encargado de gestionar las fichas de los alumnos, así como algunas actividades esporádicas.
* La notificación de los pagos mensuales de los alumnos se recibirá mediante un tipo de archivo procesable por nuestro sistema (sql, xml, etc).
* Hay que llevar un inventario de las equipaciones entregabas o no por cada temporada.
* Cada grupo de entrenamiento se compone de entre 15-20 alumnos y se distribuyen entre dos días de la semana a especificar.
* No puede haber dos grupos de entrenamiento a la misma hora, al mismo día y en la misma pista de la instalación.
* Un alumno solo puede jugar en el equipo al que está inscrito y puede competir en dos equipos, uno de su categoría y otro de una superior.
* La edad mínima de los alumnos tiene que ser 5 años, siendo la máxima 18 años.
* Los alumnos menores de 9 años, no tienen por qué pertenecer a un equipo.
* Los alumnos mayores de 9 años tienen que competir en al menos un equipo.
* La categoría benjamín y alevín solo pueden competir y entrenar en las pistas de mini-basket de las instalaciones.
* Se hará distinción entre 2 tipos de usuario para el acceso: administrador y entrenador. Cada uno de ellos podrá visualizar una parte distinta de la aplicación dependiendo de los privilegios que tenga.
* El administrador será el encargado de introducir los datos de los entrenadores en el sistema y proporcionar los usuarios y claves asociadas a cada entrenador.
* El administrador podrá modificar las contraseñas del sistema una vez creadas las cuentas en el mismo.
* Hay 5 categorías posibles: benjamín, alevín, infantil, cadete y junior.

En nuestro sistema hemos identificado los siguientes casos de uso:

|  |  |
| --- | --- |
| **Administrador** | **Entrenador** |
| Gestión de actividades:   * Insertar actividad * Eliminar actividad * Modificar actividad * Consultar actividad   Gestión de categorías:   * Insertar categoría * Eliminar categoría * Modificar categoría * Consultar categoría   Gestión de equipos:   * Insertar equipo * Dar de baja equipo * Modificar equipo * Consultar equipo   Gestión de grupos:   * Insertar grupos * Dar de baja grupos * Modificar grupos * Consultar grupos   Gestión de pagos:   * Insertar pagos * Consultar pagos * Modificar pagos * Eliminar pago   Gestión de temporadas:   * Insertar temporada * Modificar temporada * Consultar temporada * Eliminar temporada   Gestión de instalaciones:   * Insertar instalación * Modificar instalación * Consultar instalaciones * Eliminar instalación | Gestión de actividades   * Consultar actividad   Gestión de categorías   * Consultar categoría   Gestión de equipo   * Modificar equipo * Consultar equipo   Gestión de grupos   * Modificar grupos * Consultar grupos   Gestión de temporadas   * Consultar temporada   Gestión de instalaciones   * Consultar instalaciones |

El **actor administrador** será el encargado de realizar la mayor parte de operaciones del sistema para las altas, bajas y modificaciones de los datos del mismo.

El **actor entrenador** sólo podrá acceder a ciertos contenidos del sistema para consultar información o para en ciertos casos, modificar información relativa a los alumnos en el campo observaciones del mismo.

A continuación se detallan la descripción de los casos de uso del sistema.

### Casos de uso: Gestión de actividades

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar actividad |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con una nueva actividad. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información de la actividad no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La nueva actividad se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de dar de alta una actividad. 2. El sistema muestra un formulario con los campos a rellenar de la nueva actividad al administrador.    1. Opcionalmente se insertan los alumnos que van a disfrutar de la actividad.    2. Opcionalmente se asignan las instalaciones que se crean necesarias para llevar a cabo la actividad. 3. El administrador rellena el formulario con todos los campos necesarios y guarda la actividad. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la actividad. 2. La nueva actividad ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar actividad |
| **Resumen** | Se permite al administrador modificar los datos de una actividad del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador. |
| **Precondiciones** | La actividad a modificar debe existir previamente en el sistema y se sabrá qué actividad se va a modificar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de modificar actividad. 2. El sistema muestra un formulario con los datos actuales de la actividad en cuestión.    1. Opcionalmente se insertan o eliminan los alumnos que van a disfrutar de la actividad.    2. Opcionalmente se reasignan las instalaciones que se crean necesarias para llevar a cabo la actividad. 3. El administrador modifica los campos que desee y guarda la actividad. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la actividad. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar actividad |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con una actividad existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir la actividad que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario las actividades consultadas. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar actividad. 2. El sistema muestra los campos por los que se pueden buscar actividades. 3. El usuario rellena uno o varios de los campos para buscarlas y ejecuta la búsqueda. 4. El sistema muestra una lista con las actividades que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ninguna actividad, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar actividad |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con una actividad existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir la actividad que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué actividad se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La actividad eliminada ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar actividad. 2. El sistema muestra un mensaje de confirmación de eliminación dela correspondiente actividad. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema la actividad. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación de la actividad. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión categorías

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar categoría |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con una nueva categoría. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información de la categoría no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La nueva categoría se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de dar de alta una categoría. 2. El sistema muestra un formulario con los campos a rellenar de la nueva categoría al administrador. 3. El administrador rellena el formulario con todos los campos necesarios y guarda la categoría. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la categoría. 2. La nueva categoría ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar categoría |
| **Resumen** | Se permite al administrador modificar los datos de una categoría del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador. |
| **Precondiciones** | La categoría a modificar debe existir previamente en el sistema y se sabrá qué categoría se va a modificar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de modificar categoría. 2. El sistema muestra un formulario con los datos actuales de la categoría en cuestión. 3. El administrador modifica los campos que desee y guarda la categoría. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la categoría. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar categoría |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con una categoría existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir la categoría que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario las categorías consultadas. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar categoría. 2. El sistema muestra los campos por los que se pueden buscar categorías. 3. El usuario rellena uno o varios de los campos para buscarlas y ejecuta la búsqueda. 4. El sistema muestra una lista con las categorías que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ninguna categoría, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar categoría |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con una categoría existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir la categoría que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué categoría se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La categoría eliminada ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar categoría. 2. El sistema muestra un mensaje de confirmación de eliminación dela correspondiente categoría. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema la categoría. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación de la categoría. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión de equipos

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar equipo |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con un nuevo equipo. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información del equipo no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El nuevo equipo se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de dar de alta un equipo. 2. El sistema muestra un formulario con los campos a rellenar del nuevo equipo al administrador.    1. Opcionalmente se insertan los alumnos que van a formar parte del equipo. 3. El administrador rellena el formulario con todos los campos necesarios y guarda el equipo. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar el equipo. 2. El nuevo equipo ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | El equipo se asignará a una temporada en el paso 2. |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar equipo |
| **Resumen** | Se permite al usuario modificar los datos de un equipo del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | El equipo a modificar debe existir previamente en el sistema y se sabrá qué equipo se va a modificar.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El usuario selecciona la opción de modificar equipo. 2. El sistema muestra un formulario con los datos actuales del equipo en cuestión. 3. Opcionalmente se insertan o eliminan los alumnos que van a formar parte del equipo. 4. El usuario modifica los campos que desee y guarda el equipo. |
| **Cursos alternativos** | 1. El usuario cierra la ventana sin guardar el equipo. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar equipo |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con un equipo existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir el equipo que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario los equipos consultados. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar equipo. 2. El sistema muestra los campos por los que se pueden buscar equipos. 3. El usuario rellena uno o varios de los campos para buscarlos y ejecuta la búsqueda. 4. El sistema muestra una lista con los equipos que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ningún equipo, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar equipo |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con un equipo existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir el equipo que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué equipo se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El equipo eliminado ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar equipo. 2. El sistema muestra un mensaje de confirmación de eliminación del correspondiente equipo. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema el equipo. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación del equipo. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión de grupos

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar grupo |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con un nuevo grupo. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información del grupo no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El nuevo grupo se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de dar de alta un grupo. 2. El sistema muestra un formulario con los campos a rellenar del nuevo grupo al administrador.    1. Opcionalmente se insertan los alumnos que van a formar parte del grupo. 3. El administrador rellena el formulario con todos los campos necesarios y guarda el grupo. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar el grupo. 2. El nuevo grupo ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | El grupo se asignará a una temporada en el paso 2. |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar grupo |
| **Resumen** | Se permite al usuario modificar los datos de un grupo del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador, entrenador |
| **Precondiciones** | El grupo a modificar debe existir previamente en el sistema y se sabrá qué grupo se va a modificar.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El usuario selecciona la opción de modificar grupo. 2. El sistema muestra un formulario con los datos actuales del grupo en cuestión. 3. Opcionalmente se insertan o eliminan los alumnos que van a formar parte del grupo. 4. El usuario modifica los campos que desee y guarda el grupo. |
| **Cursos alternativos** | 1. El usuario cierra la ventana sin guardar el grupo. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar grupo |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con un grupo existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir el grupo que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario los grupos consultados. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar grupo. 2. El sistema muestra los campos por los que se pueden buscar grupos. 3. El usuario rellena uno o varios de los campos para buscarlos y ejecuta la búsqueda. 4. El sistema muestra una lista con los grupos que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ningún grupo, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar grupo |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con un grupo existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir el grupo que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué grupo se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El grupo eliminado ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar grupo. 2. El sistema muestra un mensaje de confirmación de eliminación del correspondiente grupo. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema el grupo. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación del grupo. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión de pagos

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar pago |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con un nuevo pago. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información del pago no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El nuevo pago se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de insertar un pago. 2. El sistema muestra un formulario con los campos a rellenar del nuevo pago al administrador. 3. El administrador rellena el formulario con todos los campos necesarios y guarda el pago. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar el pago. 2. El nuevo pago ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar pago |
| **Resumen** | Se permite al administrador modificar los datos de un pago del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador. |
| **Precondiciones** | El pago a modificar debe existir previamente en el sistema y se sabrá qué pago se va a modificar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de modificar pago. 2. El sistema muestra un formulario con los datos actuales del pago en cuestión. 3. El administrador modifica los campos que desee y guarda el pago. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar el pago. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar pago |
| **Resumen** | Se permite al administrador consultar del sistema información relacionada con un pago existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir el pago que se quiere consultar para poder llevar a cabo la operación.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al administrador los pagos consultados. |
| **Curso normal** | 1. El administrador selecciona la opción de consultar pago. 2. El sistema muestra los campos por los que se pueden buscar pagos. 3. El administrador rellena uno o varios de los campos para buscarlos y ejecuta la búsqueda. 4. El sistema muestra una lista con los pagos que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ningún pago, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar pago |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con un pago existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir el pago que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué pago se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | El grupo eliminado ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar pago. 2. El sistema muestra un mensaje de confirmación de eliminación del correspondiente pago. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema el pago. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación del pago. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión de temporadas

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar temporada |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con una nueva temporada. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información de la temporada no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La nueva temporada se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de insertar una temporada. 2. El sistema muestra un formulario con los campos a rellenar de la nueva temporada al administrador. 3. El administrador rellena el formulario con todos los campos necesarios y guarda la temporada. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la temporada. 2. La nueva temporada ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar temporada |
| **Resumen** | Se permite al administrador modificar los datos de una temporada del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador. |
| **Precondiciones** | La temporada a modificar debe existir previamente en el sistema y se sabrá qué temporada se va a modificar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de modificar temporada. 2. El sistema muestra un formulario con los datos actuales de la temporada en cuestión. 3. El administrador modifica los campos que desee y guarda la temporada. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la temporada. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Consultar temporada |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con una temporada existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir la temporada que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario las temporadas consultadas. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar temporada. 2. El sistema muestra los campos por los que se pueden buscar temporadas. 3. El usuario rellena uno o varios de los campos para buscarlos y ejecuta la búsqueda. 4. El sistema muestra una lista con las temporadas que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ninguna temporada, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar temporada |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con una temporada existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir la temporada que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué temporada se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La temporada eliminada ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar temporada. 2. El sistema muestra un mensaje de confirmación de eliminación de la correspondiente temporada. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema la temporada. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación de la temporada. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

### Casos de uso: Gestión de instalaciones

|  |  |
| --- | --- |
| **Nombre del CU** | Insertar instalación |
| **Resumen** | Se permite al administrador introducir en el sistema información relacionada con una nueva instalación. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | La información de la instalación no debe existir previamente en el sistema.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La nueva instalación se ha creado con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de dar de alta una instalación. 2. El sistema muestra un formulario con los campos a rellenar de la nueva instalación al administrador. 3. El administrador rellena el formulario con todos los campos necesarios y guarda la instalación. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la instalación. 2. La nueva instalación ya existe en el sistema. El sistema informa del error y vuelve al paso 2. 3. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Modificar instalación |
| **Resumen** | Se permite al administrador modificar los datos de una instalación del sistema. |
| **Dependencias** | - |
| **Actores** | Administrador. |
| **Precondiciones** | La instalación a modificar debe existir previamente en el sistema y se sabrá qué instalación se va a modificar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | Los cambios realizados se almacenan con éxito. |
| **Curso normal** | 1. El administrador selecciona la opción de modificar instalación. 2. El sistema muestra un formulario con los datos actuales de la instalación en cuestión. 3. El administrador modifica los campos que desee y guarda la instalación. |
| **Cursos alternativos** | 1. El administrador cierra la ventana sin guardar la instalación. 2. Falta algún dato por rellenar. El sistema informa del error y vuelve el paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

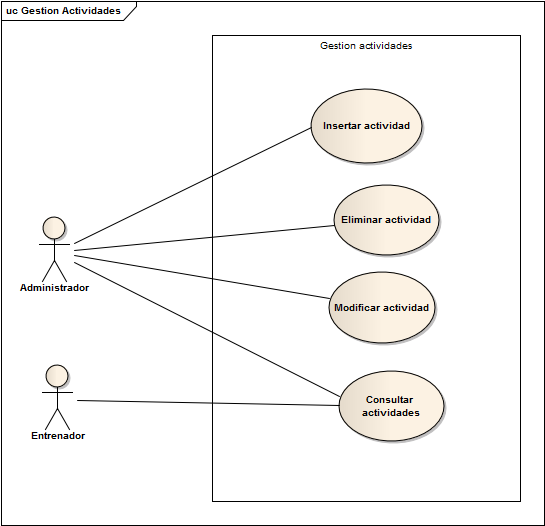
|  |  |
| --- | --- |
| **Nombre del CU** | Consultar instalación |
| **Resumen** | Se permite al usuario consultar del sistema información relacionada con una instalación existente. |
| **Dependencias** | - |
| **Actores** | Administrador, Entrenador |
| **Precondiciones** | Debe de existir la instalación que se quiere consultar para poder llevar a cabo la operación.  El usuario debe estar identificado en el sistema. |
| **Postcondición** | El sistema muestra al usuario las instalaciones consultadas. |
| **Curso normal** | 1. El usuario selecciona la opción de consultar instalación. 2. El sistema muestra los campos por los que se pueden buscar instalaciones. 3. El usuario rellena uno o varios de los campos para buscarlas y ejecuta la búsqueda. 4. El sistema muestra una lista con las instalaciones que coinciden con los criterios de búsqueda. |
| **Cursos alternativos** | 1. El sistema no encuentra ninguna instalación, lanza un error y vuelve al paso 2. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

|  |  |
| --- | --- |
| **Nombre del CU** | Eliminar instalación |
| **Resumen** | Se permite al administrador eliminar del sistema información relacionada con una instalación existente. |
| **Dependencias** | - |
| **Actores** | Administrador |
| **Precondiciones** | Debe de existir la instalación que se quiere eliminar para poder llevar a cabo la operación y se sabrá qué instalación se va a eliminar.  El administrador debe estar identificado en el sistema. |
| **Postcondición** | La instalación eliminada ya no formará parte del sistema. |
| **Curso normal** | 1. El administrador selecciona la opción de eliminar instalación. 2. El sistema muestra un mensaje de confirmación de eliminación dela correspondiente instalación. 3. El administrador confirma la decisión. 4. El sistema elimina del sistema la instalación. |
| **Cursos alternativos** | 1. El administrador rechaza la eliminación de la instalación. |
| **Observaciones** | - |
| **Requisitos no funcionales** | Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortara la ejecución del proceso. |

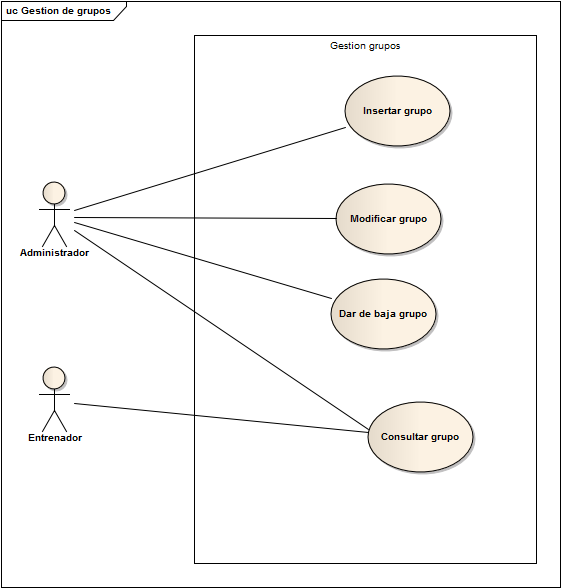
## Identificar subsistemas – Diagramas de casos de uso

A continuación se detallan los distintos diagramas de casos de uso de los que se compone el sistema analizado.

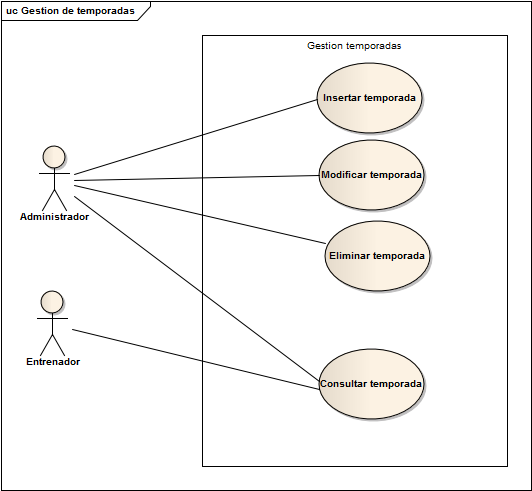
### Diagrama de CU: Gestión de Actividades

****

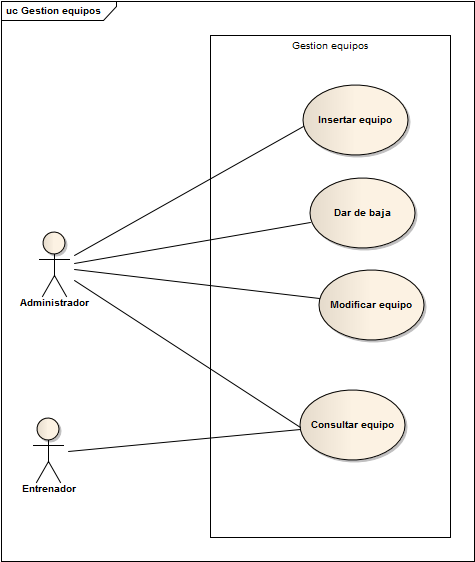
### Diagrama de CU: Gestión de Grupos

****

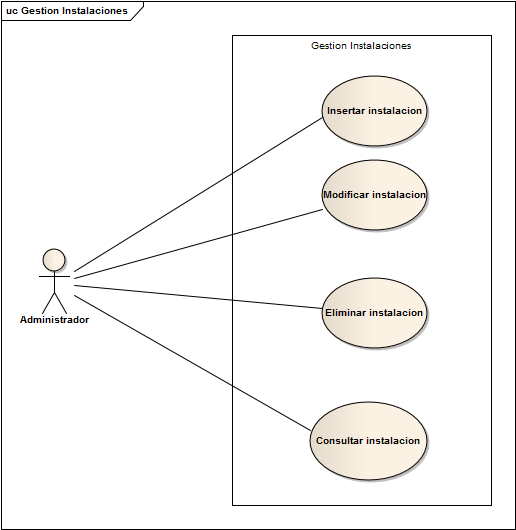
### Diagrama de CU: Gestión de Temporadas

****

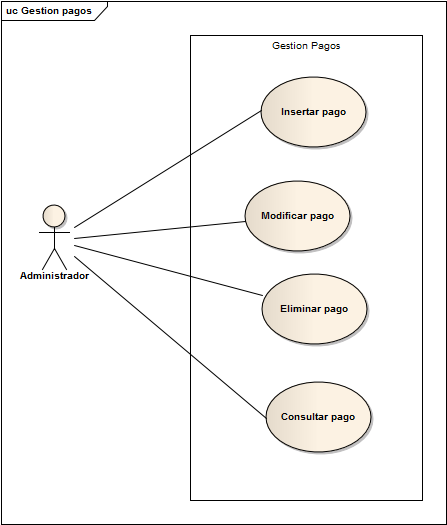
### Diagrama de CU: Gestión de Equipos



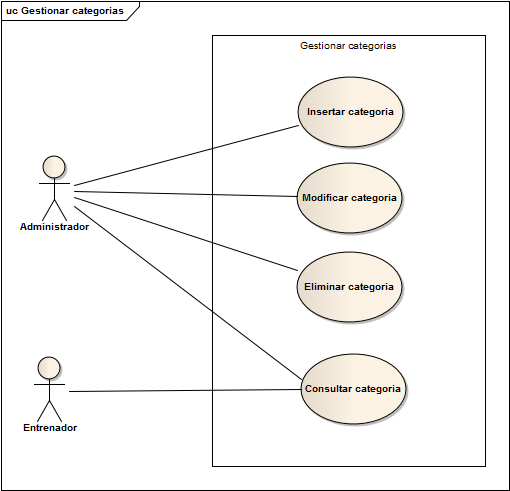
### Diagrama de CU: Gestión de Instalaciones



### Diagrama de CU: Gestión de Pagos



### Diagrama de CU: Gestión de Categorias



## Requisitos no funcionales

Estos requisitos son aquellos que describen cualidades o restricciones del sistema que no se relacionan de forma directa con el comportamiento funcional del mismo. A continuación se especifican los más importantes del sistema:

- No requiere un conocimiento específico para la utilización del software.

- La aplicación tendrá un manual de uso.

- La base de datos estará implementada en un lenguaje objeto relacional como mysql.

- La aplicación estará realizada en el lenguaje de programación Java.

- Se creara un script de tarea programada (cron) para la copia de seguridad de la base de datos debido a alguna inconsistencia del sistema.

- La interfaz debe reflejar claramente la distinción entre las distintas partes del sistema.

- La documentación del código fuente será llevada a cabo mediante la aplicación javadoc.

- El sistema se desplegara sobre una distribución Microsoft Windows.

- El administrador se encargara de tareas de mantenimiento.

- Cuando el sistema detecte una anomalía mostrara al usuario el mensaje de error pertinente y abortara la ejecución del proceso.

- El código fuente de la aplicación seguirá un estilo uniforme y normalizado para todos los módulos del mismo.

- El formato de las fechas será dd/mm/yy.

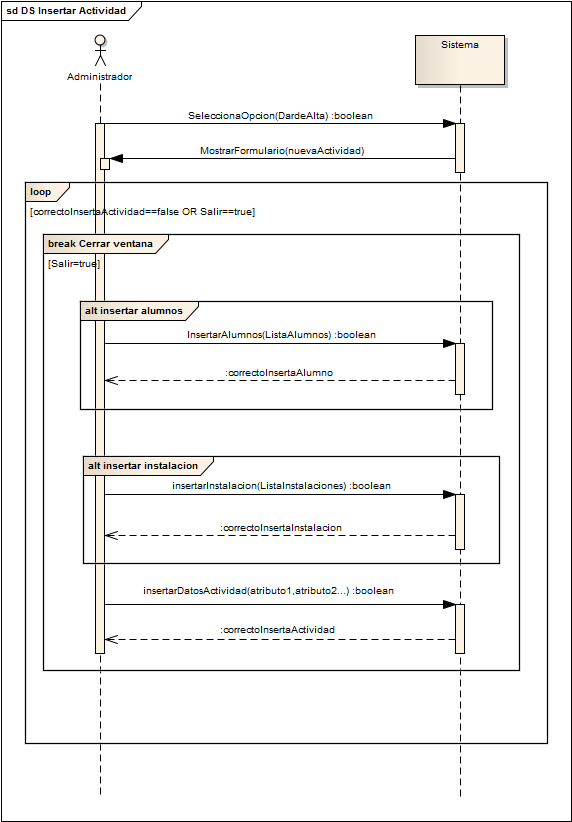
- El control de asistencia no es controlado, pero se puede incluir en la sección de observaciones de los alumnos.

## Operaciones del sistema – Diagramas de secuencia de sistema

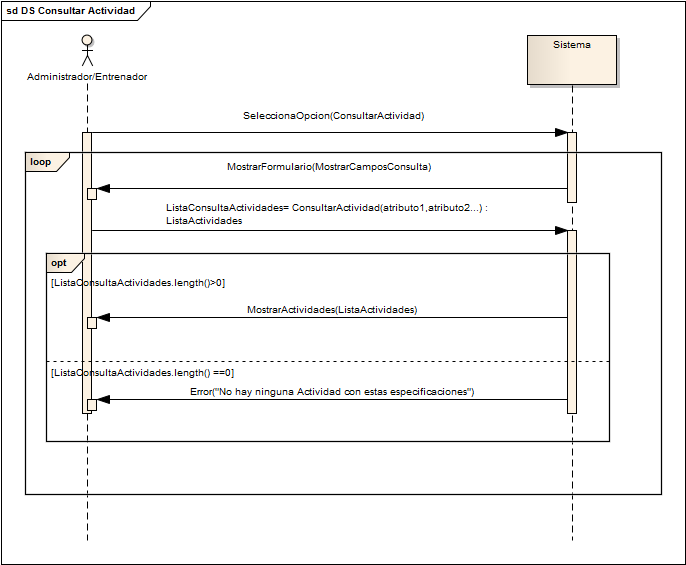
Se van a describir diagramas de secuencia de las operaciones del sistema.

### Gestión de Actividades

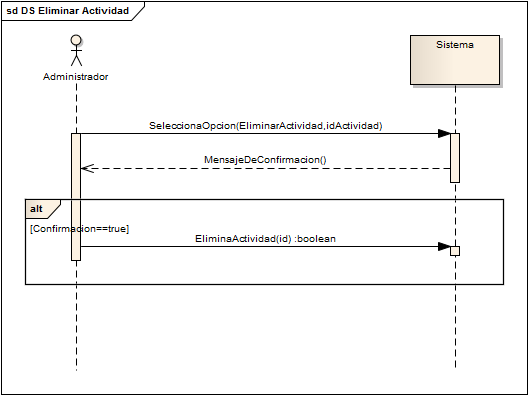
#### *Diagrama de secuencia:Insertar Actividad*



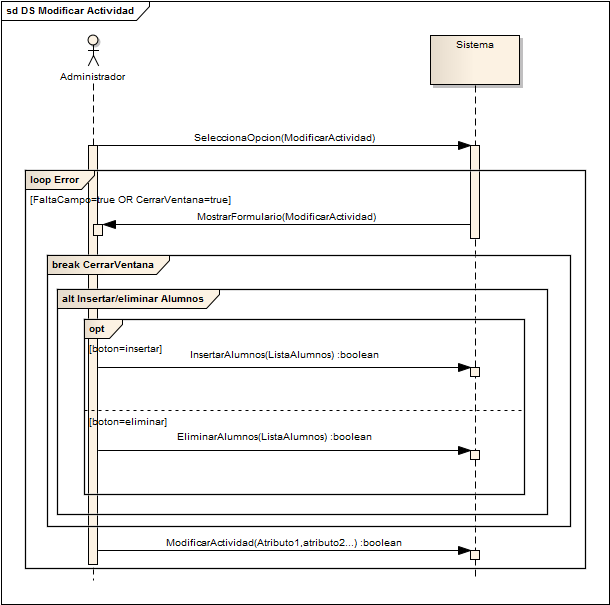
#### *Diagrama de secuencia:Consultar Actividad*

****

#### *Diagrama de secuencia:Eliminar Actividad*

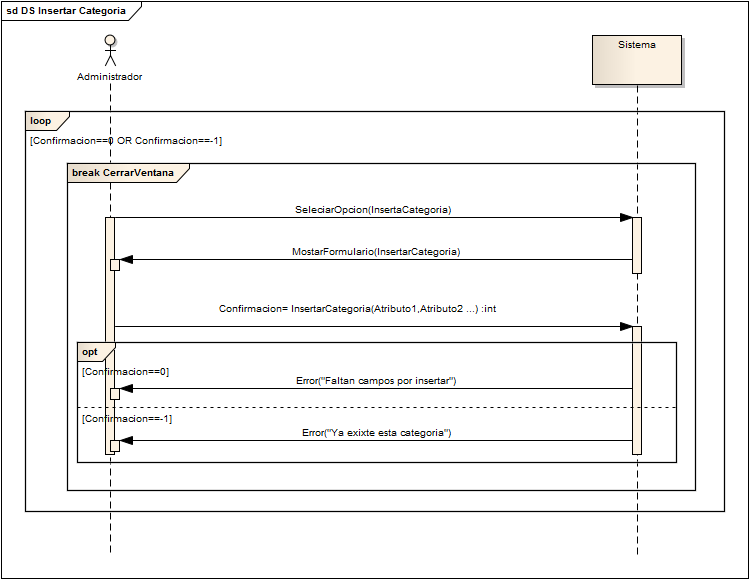
****

#### *Diagrama de secuencia: ModificarActividad*

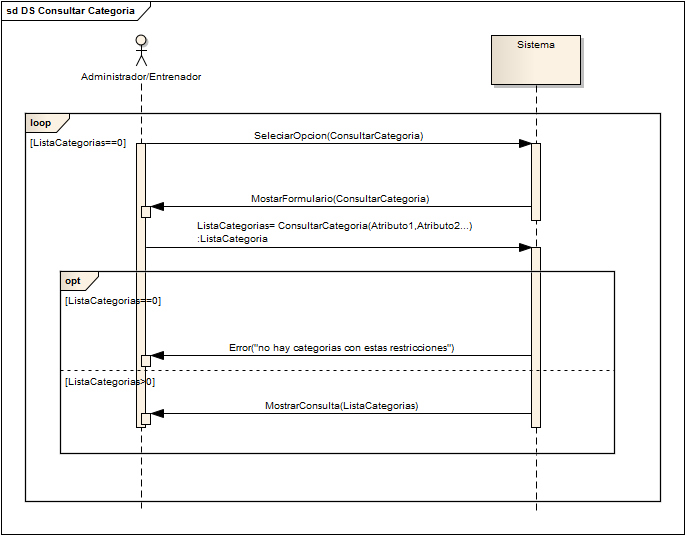


### Gestión de Categorías

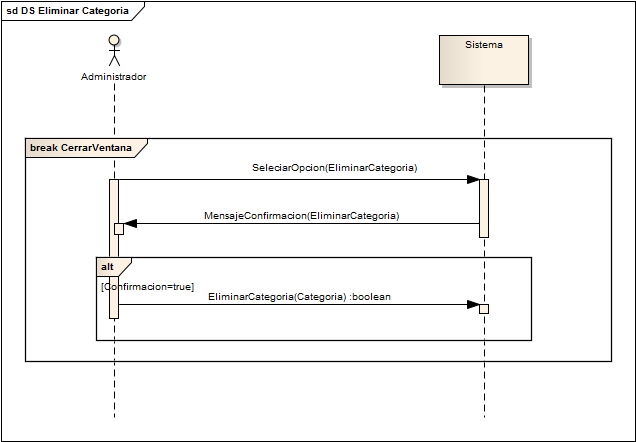
#### *Diagrama de secuencia: Insertar Categoría*



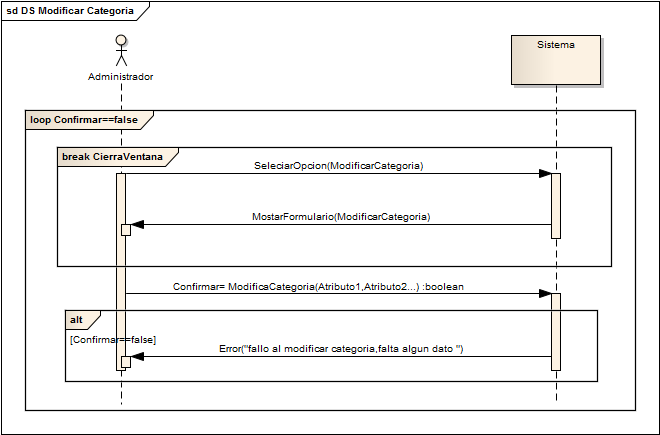
#### *Diagrama de secuencia: Consultar Categoría*



#### *Diagrama de secuencia: Eliminar Categoría*

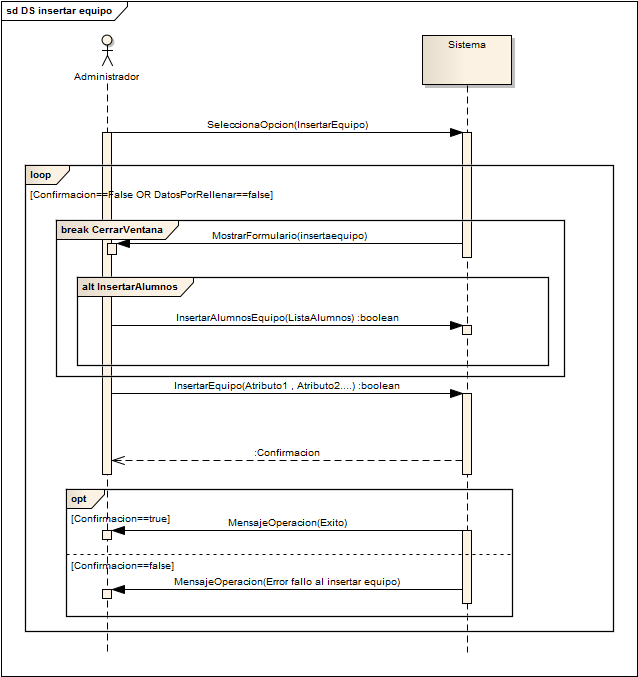


#### *Diagrama de secuencia: Modificar Categoría*

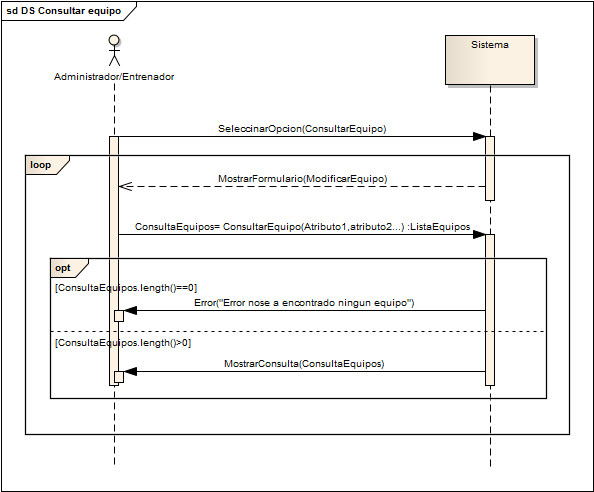


### Gestión de Equipos

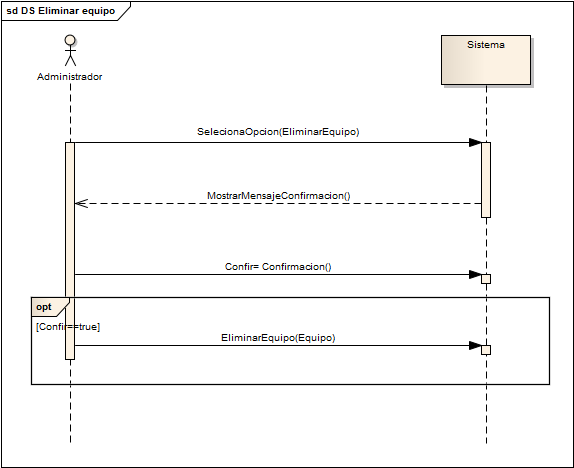
#### *Diagrama de secuencia: Insertar Equipo*



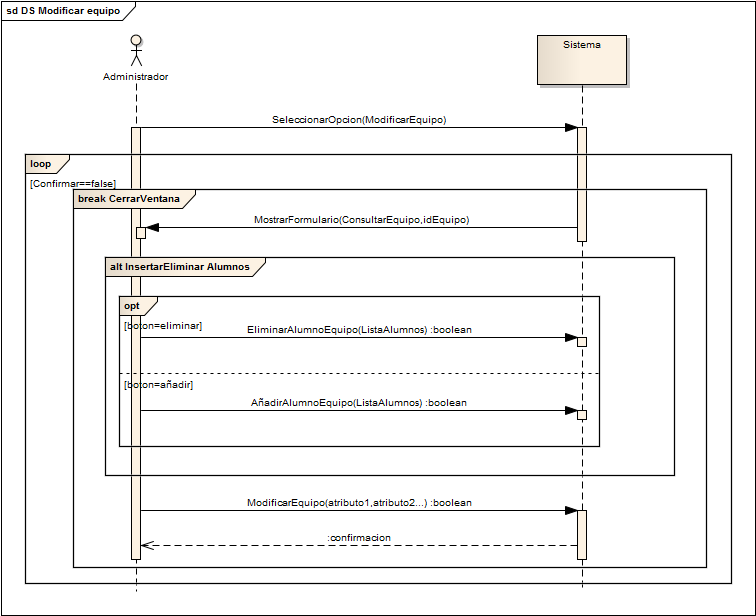
#### *Diagrama de secuencia: Consultar Equipo*



#### *Diagrama de secuencia: Eliminar Equipo*

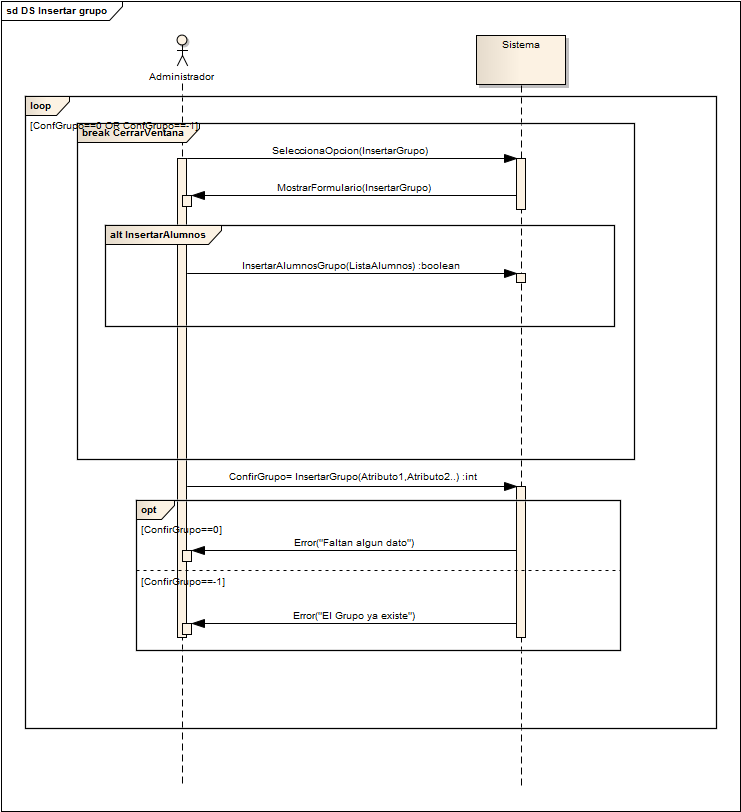


#### *Diagrama de secuencia: Modificar Equipo*

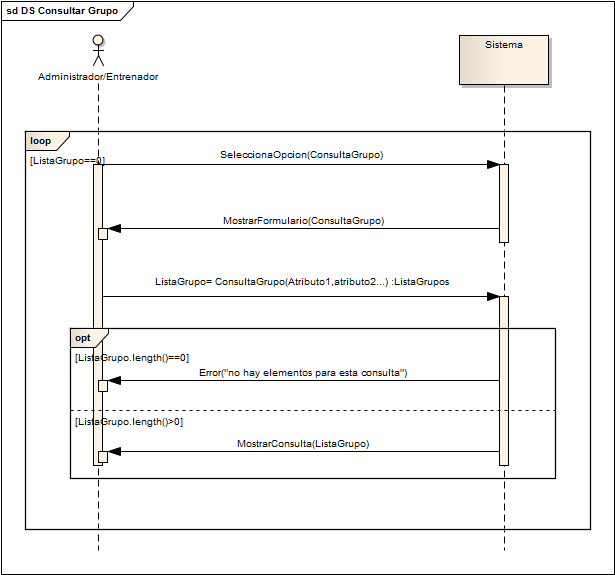


### Gestión de Grupos

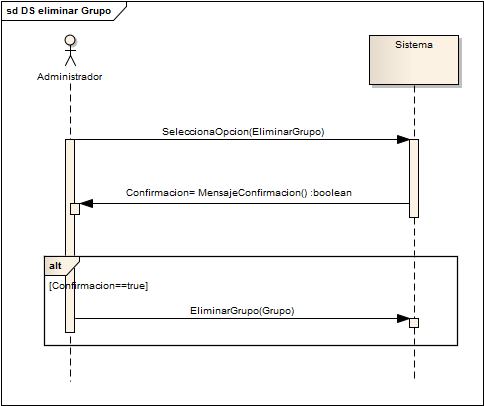
#### *Diagrama de secuencia: Insertar Grupo*



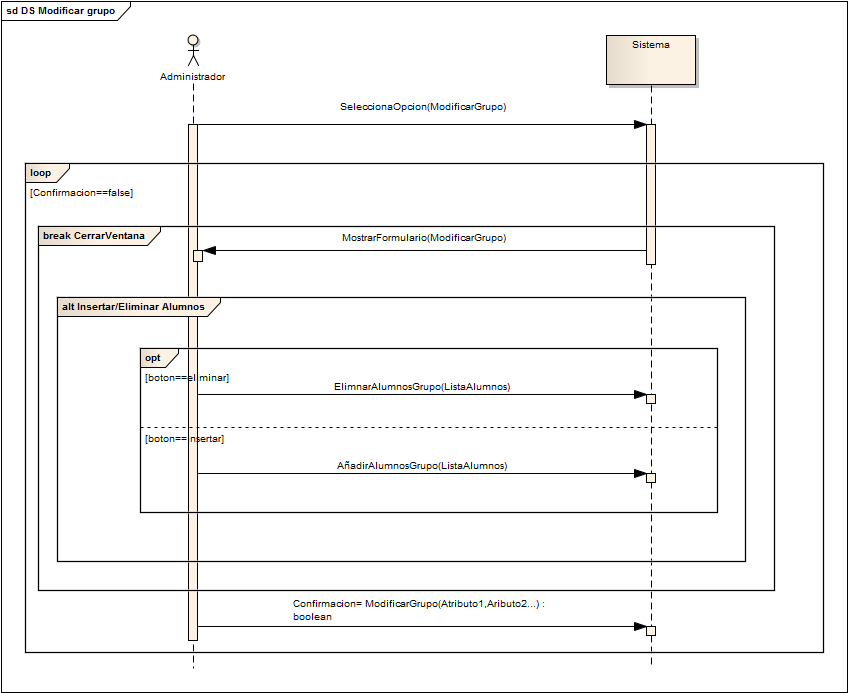
#### *Diagrama de secuencia: Consultar Grupo*



#### *Diagrama de secuencia: Eliminar Grupo*

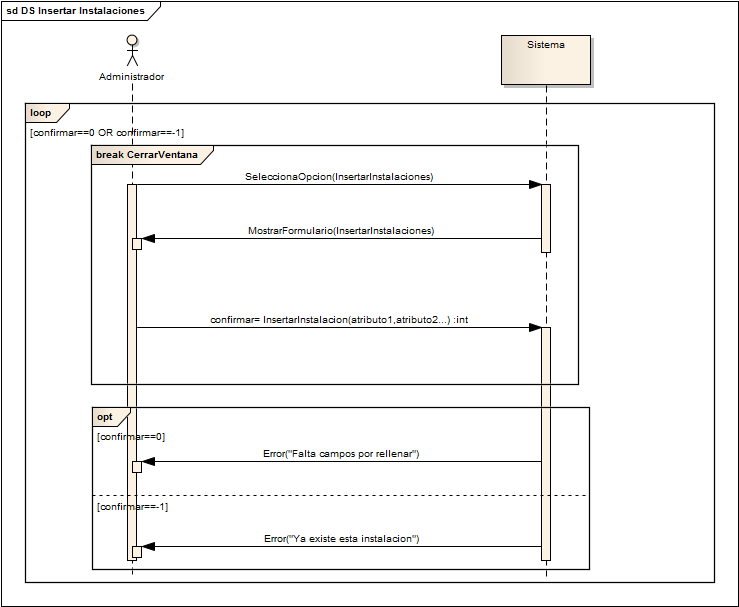


#### *Diagrama de secuencia: Modificar Grupo*

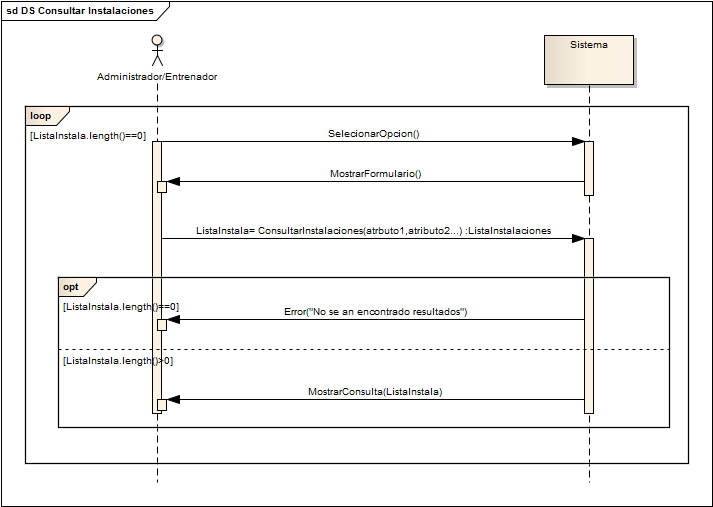


### Gestión de Instalaciones

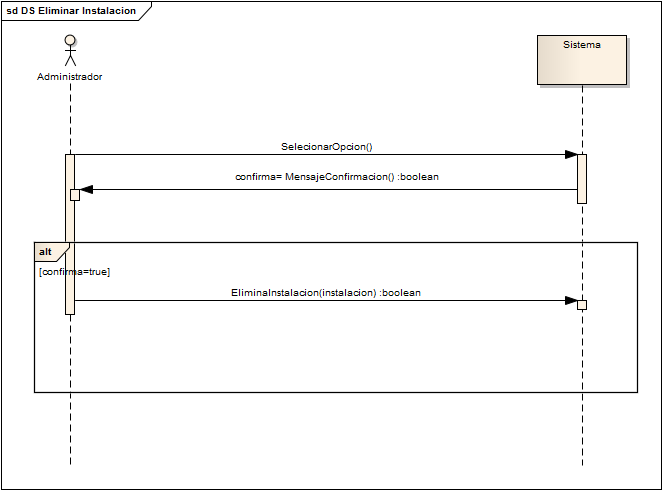
#### *Diagrama de secuencia: Insertar Instalaciones*



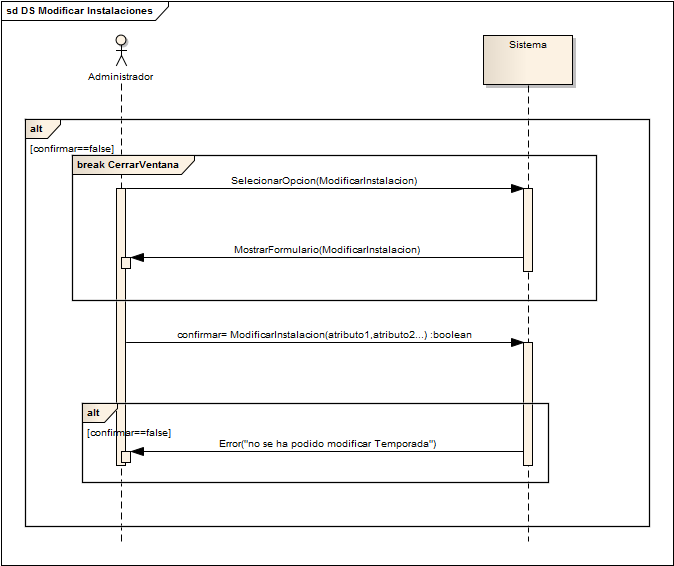
#### *Diagrama de secuencia: Consultar Instalaciones*



#### *Diagrama de secuencia: Eliminar Instalaciones*

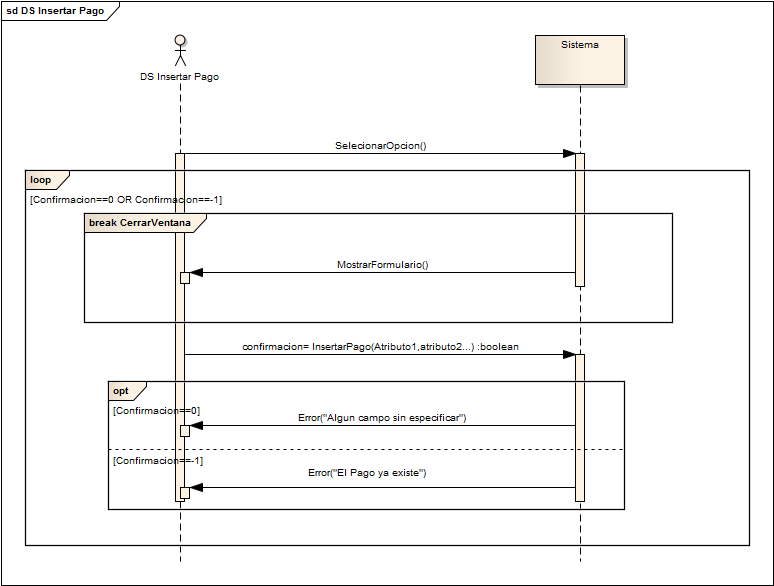


#### *Diagrama de secuencia: Modificar Instalaciones*

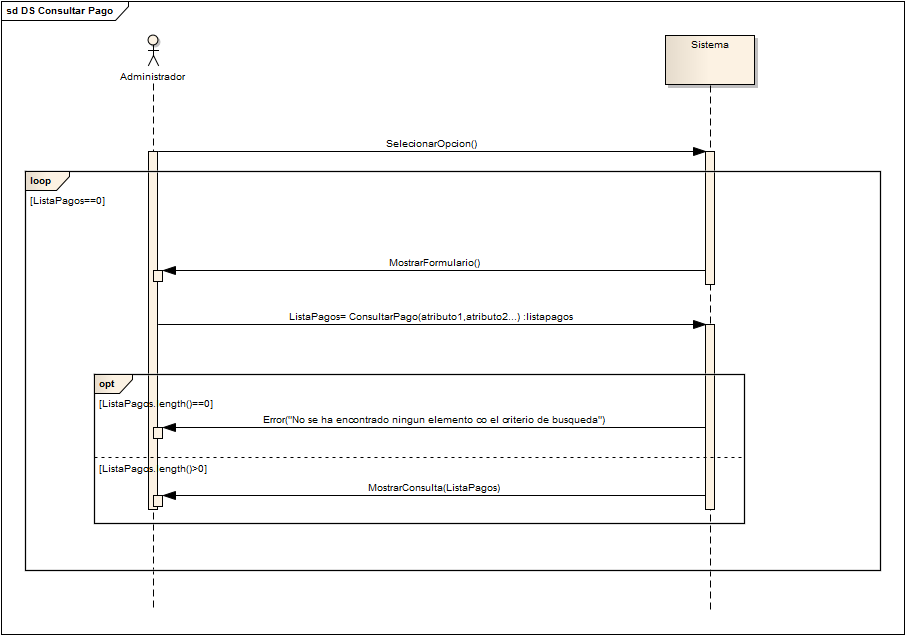


### Gestión de Pagos

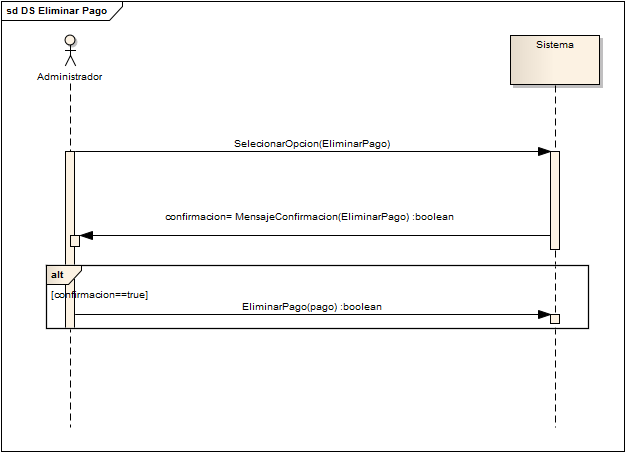
#### *Diagrama de secuencia: Insertar Pago*



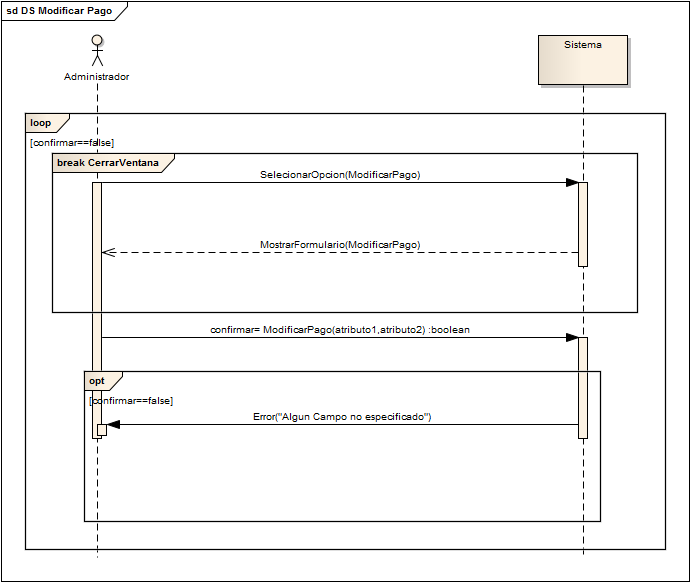
#### *Diagrama de secuencia: Consultar Pago*



#### *Diagrama de secuencia: Eliminar Pago*

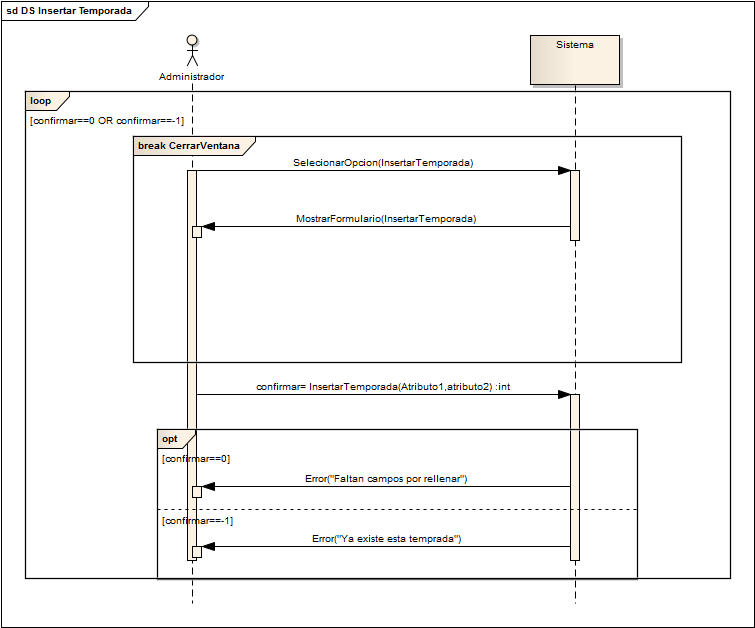


#### *Diagrama de secuencia: Modificar Pago*

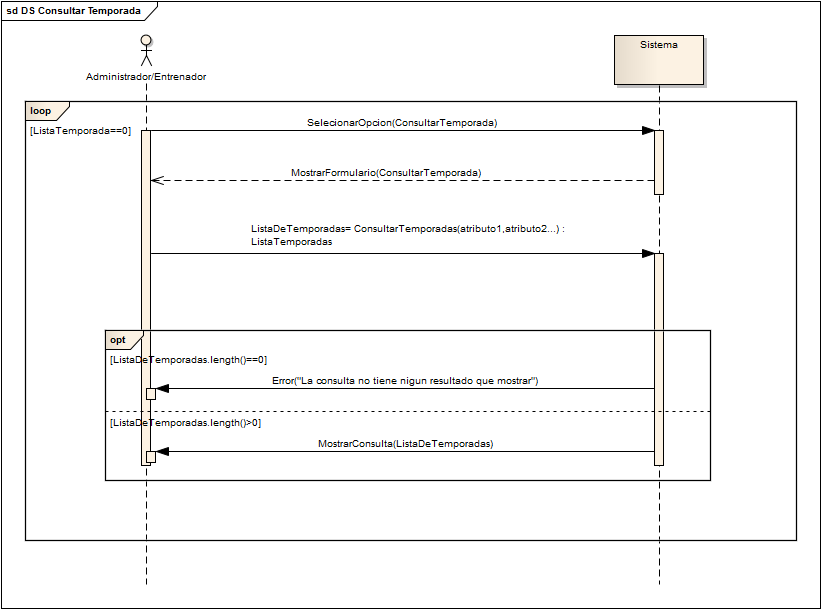


### Gestión de Temporadas

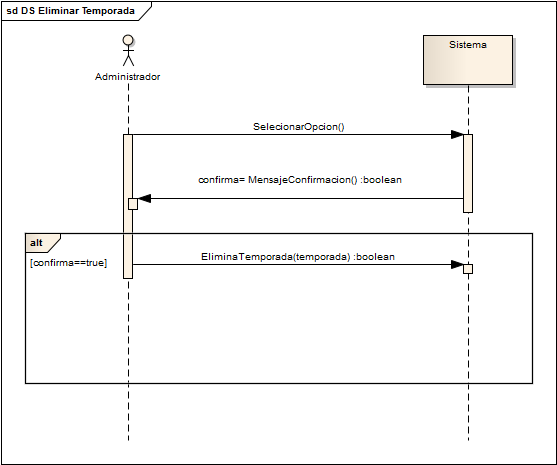
#### *Diagrama de secuencia: Insertar Temporada*



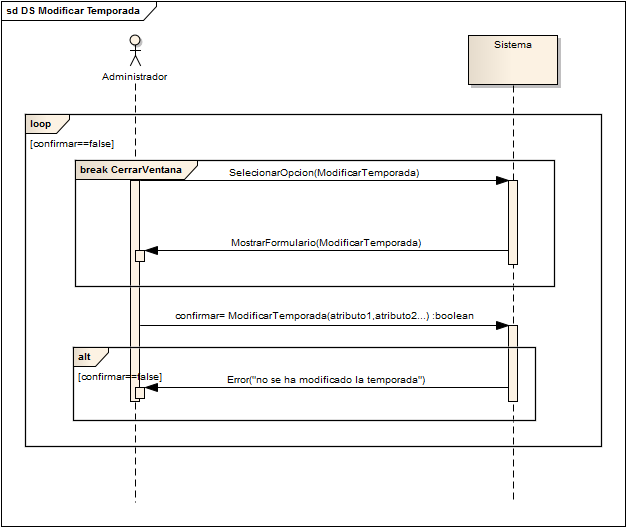
#### *Diagrama de secuencia: Consultar Temporada*



#### *Diagrama de secuencia: Eliminar Temporada*



#### *Diagrama de secuencia: Modificar Temporada*



# **Análisis**

## Identificar clases, atributos y relaciones

En este punto se buscan las abstracciones más significativas que identifiquen los aspectos claves del dominio del problema. Las clases conceptuales deben interrelacionarse entre sí a través de las relaciones para satisfacer las necesidades de información o de comportamiento que demandan los casos de uso y así comprender mejor el modelo.

A continuación se detallan las clases, atributos y relaciones entre las clases que se han obtenido para esta segunda iteración del problema planteado.

### Clase: Actividad

Atributos:

* descripcion (string): Descripción breve de la actividad para consultas por el sistema.
* fechaFin (date): Fecha fin de la actividad para la temporada.
* fechaInicio (date): Fecha inicio de la actividad para la temporada.
* nombre (string): Nombre identificativo de la actividad.
* precioNoSocio (float): Es el precio que tendrán que abonar los alumnos que no sean socios y que deseen participar en la actividad.
* precioSocio (float): Es el precio que tendrán que abonar los alumnos que si sean socios y que deseen participar en la actividad.
* idActividad (int): Número identificador de la actividad.

Relaciones:

* Asociación con la clase Alumno.
* Asociación con la clase Temporada.

### Clase: Categoría

Atributos:

* descripción (String): Breve descripción de la categoría del alumno.
* edadMaxima (int): La edad máxima posible a la que se puede acceder a la categoría.
* idCategoria (int): Número identificador de la categoría.
* tipo (String): Nombre de la categoría del alumno.

Relaciones:

* Asociación con la clase Equipo.
* Asociación con la clase Grupo.

### Clase: Equipo

Atributos:

* idEquipo (int): Número identificador del equipo.
* sexo(Sexo): Este campo corresponde con el sexo que tendrán los integrantes del equipo, que pertenece al tipo enumerado Sexo en donde se establecen todos los valores posibles que puede tomar el campo.

Relaciones:

* Asociación con la clase Categoría.
* Asociación con la clase Entrenador.
* Asociación con la clase Fundación.
* Asociación con la clase Alumno.
* Asociación con la clase Temporada.

### Clase: Grupo Entrenamiento

Atributos:

* idGrupo (int): Número identificador del alumno.
* Sexo(Sexo): Este campo corresponde con el sexo que tendrán los integrantes del equipo, que pertenece al tipo enumerado Sexo en donde se establecen todos los valores posibles que puede tomar el campo.

Relaciones:

* Asociación con la clase Entrenador.
* Asociación con la clase Categoría.
* Asociación con la clase Alumno.
* Asociación con la clase Temporada.

### Clase: Pago Temporada (clase de asociación entre las clases Temporada y Alumno)

Atributos:

* idPagoTemporada (int): Número identificador del pago por temporada.

Relaciones:

* Asociación con la clase Cuota precio (composición de esta última)

### Clase: Pago Actividad (clase de asociación entre Alumno y Actividad)

Atributos:

* idPagoActividad (int): Número identificador del pago de la actividad.
* recibo (string): Ruta relativa o completa del archivo que contiene la imagen del recibo/pago/domiciliación del pago de la actividad.

Relaciones:

* Asociación con la clase Cuota precio (composición).

### Clase: Cuota Precio

Atributos:

* idCuotaPrecio (int): Número identificador del precio de la cuota.
* importe (float): Valor numérico del importe de la mensualidad del alumno.
* pagado (boolean): Valor booleano que establece si la mensualidad ha sido pagada o no por el alumno. Se modificará una vez se haya recibido del banco el documento con los pagos del mes.
* fechaPago (date): Fecha en la que se realizó el pago.

Relaciones:

* Asociación con la clase Pago Temporada.
* Asociación con la clase Pago Actividad.

### Clase: Temporada

Atributos:

* curso (string): año correspondiente al curo de la temporada (ejemplo: 2012-2013 sería 12/13).
* idTemporada (int): Número identificador de la temporada.

Relaciones:

* Asociación con la clase Equipo.
* Asociación con la clase Grupo.
* Asociación con la clase Alumno.
* Asociación con la clase Temporada.

### Clase: Instalación

Atributos:

* capacidadEquipos (int): Indica el número de equipos que pueden entrenar a la vez en esa instalación.
* localizacion (String): Indica la calle y zona donde está ubicada la instalación.
* nombre (String): Es el nombre de la instalación.
* idInstalación (int): Número identificador de la instalación.

Relaciones:

* Asociación con la clase Actividad.
* Asociación con la clase GrupoEntrenamiento.

### Clase: Horario (clase de asociación entre Instalación y GrupoEntrenamiento)

Atributos:

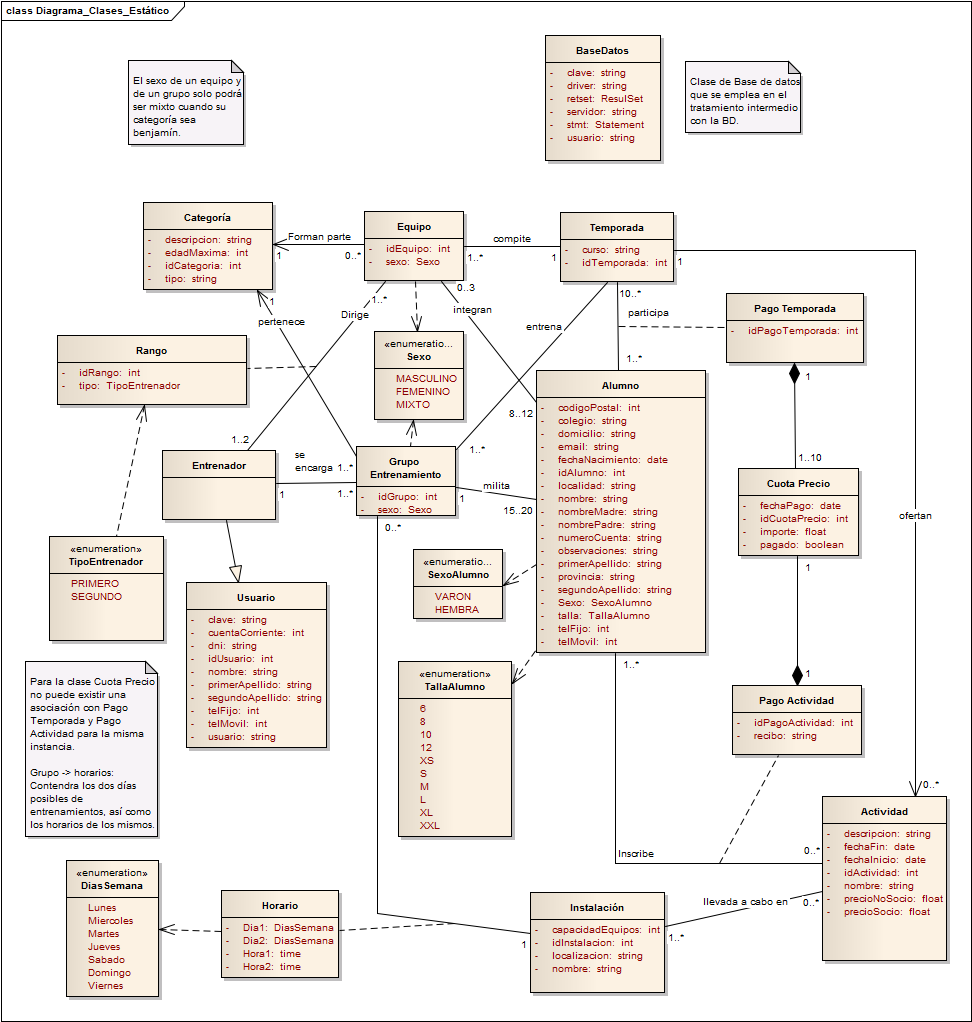
* Dia1 (DiasSemana): Este campo corresponde con un día de la semana en el que el grupo entrenará en la instalación, que pertenece al tipo enumerado DiasSemana en donde se establecen todos los valores posibles que puede tomar el campo.
* Hora1 (time): Este campo indica la hora en la que el grupo entrenará en la instalación el día1.
* Dia2 (DiasSemana): Este campo corresponde con el otro día de la semana en el que el grupo entrenará en la instalación, que pertenece al tipo enumerado DiasSemana en donde se establecen todos los valores posibles que puede tomar el campo.
* Hora2 (time): Este campo indica la hora en la que el grupo entrenará en la instalación el día2.

## Modelado estático – Diagrama de clases

El modelo estático proporciona mecanismos para describir y representar las interrelaciones estructurales que se establecen entre las clases conceptuales de un modo visual y compacto.

* Proporciona una estructura estática de las clases conceptuales.
* Al ser un modelo de análisis sólo muestra las clases conceptuales extraídas del dominio de aplicación (dominio del problema) y sus relaciones, no componentes software pertenecientes a la solución.
* La reducida barrera que propugna la orientación a objetos entre el problema y la solución, posibilita que el modelo estático inspire la construcción del modelo de diseño.
* Definen un vocabulario común entre clientes y desarrolladores.
* Una vez que el modelo es estable, la descripción de cada clase será tan detallada como sea posible.
* Elimina la vaguedad en la definición de las clases del dominio del problema.

A continuación se describe el diagrama de clases obtenido para el problema especificado.



## Modelado del comportamiento externo – Contratos

Describen el comportamiento detallado del sistema en función de los cambios de estado de los objetos cuando se invoca una operación del sistema.

Es un documento que describe lo que una operación se propone lograr, sin decir cómo se conseguirá.

* Define la especificación de una operación sin entrar en su implementación.
* Suele redactarse con un estilo declarativo.

### Contratos de los casos de uso de la Gestión de Actividades.

#### Contratos de Caso de uso: Insertar Actividad.

|  |  |
| --- | --- |
| **Nombre** | InsertarAlumnos(listaAlumnos: lista) |
| **Responsabilidades** | Introduce los alumnos que van a disfrutar de la actividad. |
| **Tipo** | Sistema |
| **Referencias cruzadas** | - |
| **Notas** | - |
| **Excepciones** | - |
| **Salida** | Devuelve si la operación ha sido efectuada correctamente. |
| **Precondición** | Ninguno de los campos puede ser nulo. |
| **Postcondición** | Los alumnos de la lista se han insertado en la actividad en cuestión. |

|  |  |
| --- | --- |
| **Nombre** | InsertarInstalacion(listaInstalaciones: lista) |
| **Responsabilidades** | Introduce las instalaciones donde va a desarrollarse la actividad. |
| **Tipo** | Sistema |
| **Referencias cruzadas** | - |
| **Notas** | - |
| **Excepciones** | - |
| **Salida** | Devuelve si la operación ha sido efectuada correctamente. |
| **Precondición** | Ninguno de los campos puede ser nulo. |
| **Postcondición** | Las instalaciones de la lista se han insertado en la actividad en cuestión. |

|  |  |
| --- | --- |
| **Nombre** | InsertarDatosActividad(atributo1: tipo1, atributo2: tipo2…) |
| **Responsabilidades** | Introduce los datos de la actividad que estamos creando. |
| **Tipo** | Sistema |
| **Referencias cruzadas** | - |
| **Notas** | - |
| **Excepciones** | - |
| **Salida** | Devuelve si la operación ha sido efectuada correctamente. |
| **Precondición** | Ninguno de los campos puede ser nulo. |
| **Postcondición** | La actividad ha sido creada e inicializada correctamente con todos sus atributos, y alumnos e instalaciones, en el caso de que se hayan especificado. |

#### Contratos de Caso de uso: Consultar Actividad.

|  |  |
| --- | --- |
| **Nombre** | consultaActividad(fechaIni: date, nombre: String, temporada: Temporada) |
| **Responsabilidades** | Consultar los datos de una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** |  |
| **Notas** | El sistema muestra las dos formas posibles de elegir una actividad: por su fecha de inicio, mediante el nombre o mediante la temporada. |
| **Excepciones** | - |
| **Salida** | - |
| **Precondición** | Que haya al menos una actividad dada de alta en el sistema. |
| **Postcondición** | El sistema mostrará la lista de actividades que coincidan con la búsqueda. |

### Contratos de Caso de uso: Eliminar Actividad.

|  |  |
| --- | --- |
| **Nombre** | eliminarActividad(actividad: Actividad) |
| **Responsabilidades** | Elimina una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** | Extiende al caso de uso “Consultar Actividades” |
| **Notas** |  |
| **Excepciones** |  |
| **Salida** | - |
| **Precondición** | Que hayamos seleccionado previamente la actividad que deseamos modificar. |
| **Postcondición** | Los datos no serán borrados del sistema si no que se dará de baja la actividad por si estos datos se quieren usar posteriormente. |

### Contratos de Caso de uso: Insertar/Eliminar alumnos Actividad.

|  |  |
| --- | --- |
| **Nombre** | añadirAlumno(alumno: Alumno) |
| **Responsabilidades** | Añade un alumno a una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** | Extiende al caso de uso “Insertar Actividad”, y al caso de uso “Modificar Actividad” |
| **Notas** |  |
| **Excepciones** |  |
| **Salida** | - |
| **Precondición** | Que hayamos seleccionado previamente la actividad que deseamos modificar y que el alumno esté dado de alta en el sistema. |
| **Postcondición** | El sistema guardará que dicho alumno está inscrito en la actividad. |

|  |  |
| --- | --- |
| **Nombre** | eliminarAlumno(alumno: Alumno) |
| **Responsabilidades** | Elimina un alumno de una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** | Extiende al caso de uso “Insertar Actividad”, y al caso de uso “Modificar Actividad” |
| **Notas** |  |
| **Excepciones** |  |
| **Salida** | - |
| **Precondición** | Que hayamos seleccionado previamente la actividad que deseamos modificar y que el alumno esté inscrito en esa actividad. |
| **Postcondición** | El sistema elimina que dicho alumno está inscrito en la actividad. |

### Contratos de Caso de uso: Insertar/Eliminar instalaciones Actividad.

|  |  |
| --- | --- |
| **Nombre** | añadirInstalacion(instalacion: Instalacion) |
| **Responsabilidades** | Añade una instalación a una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** | Extiende al caso de uso “Insertar Actividad”, y al caso de uso “Modificar Actividad” |
| **Notas** |  |
| **Excepciones** |  |
| **Salida** | - |
| **Precondición** | Que hayamos seleccionado previamente la actividad que deseamos modificar y que la instalación esté dada de alta en el sistema. |
| **Postcondición** | El sistema guardará que dicha instalación está asignada en la actividad. |

|  |  |
| --- | --- |
| **Nombre** | eliminar instalación (instalación: Instalación) |
| **Responsabilidades** | Elimina una instalación que estuviese asignada a una actividad existente en el sistema. |
| **Tipo** | Sistema. |
| **Referencias cruzadas** | Extiende al caso de uso “Insertar Actividad”, y al caso de uso “Modificar Actividad” |
| **Notas** |  |
| **Excepciones** |  |
| **Salida** | - |
| **Precondición** | Que hayamos seleccionado previamente la actividad que deseamos modificar y que la instalación este asignada a esa actividad. |
| **Postcondición** | El sistema elimina que dicha instalación está apuntada en la actividad. |