**Descomposición del sistema en subsistemas de diseño para obtener la arquitectura del sistema.**

**Establecer la arquitectura del sistema.**

Para llevar a cabo la fase de diseño hay que establecer la arquitectura del sistema teniendo en cuenta los siguientes puntos:

* Objetivos de diseño del sistema.
* Estilos arquitectónicos posibles.
* Documentación de las etapas anteriores del proceso de desarrollo.

**Subsistemas funcionales.**

En el modelado de requisitos se han detectado los siguientes subsistemas funcionales:

* Subsistema GestionAlumnos.
* Subsistema GestionUsuarios.
* Subsistema GestiónPagos.
* Subsistema GestiónGruposEntrenamiento.
* Subsistema GestiónEquipos.
* Subsistema GestiónActividades.
* Subsistema GestiónTemporadas.
* Subsistema GestiónCategorías.
* Subsistema GestiónInstalaciones.

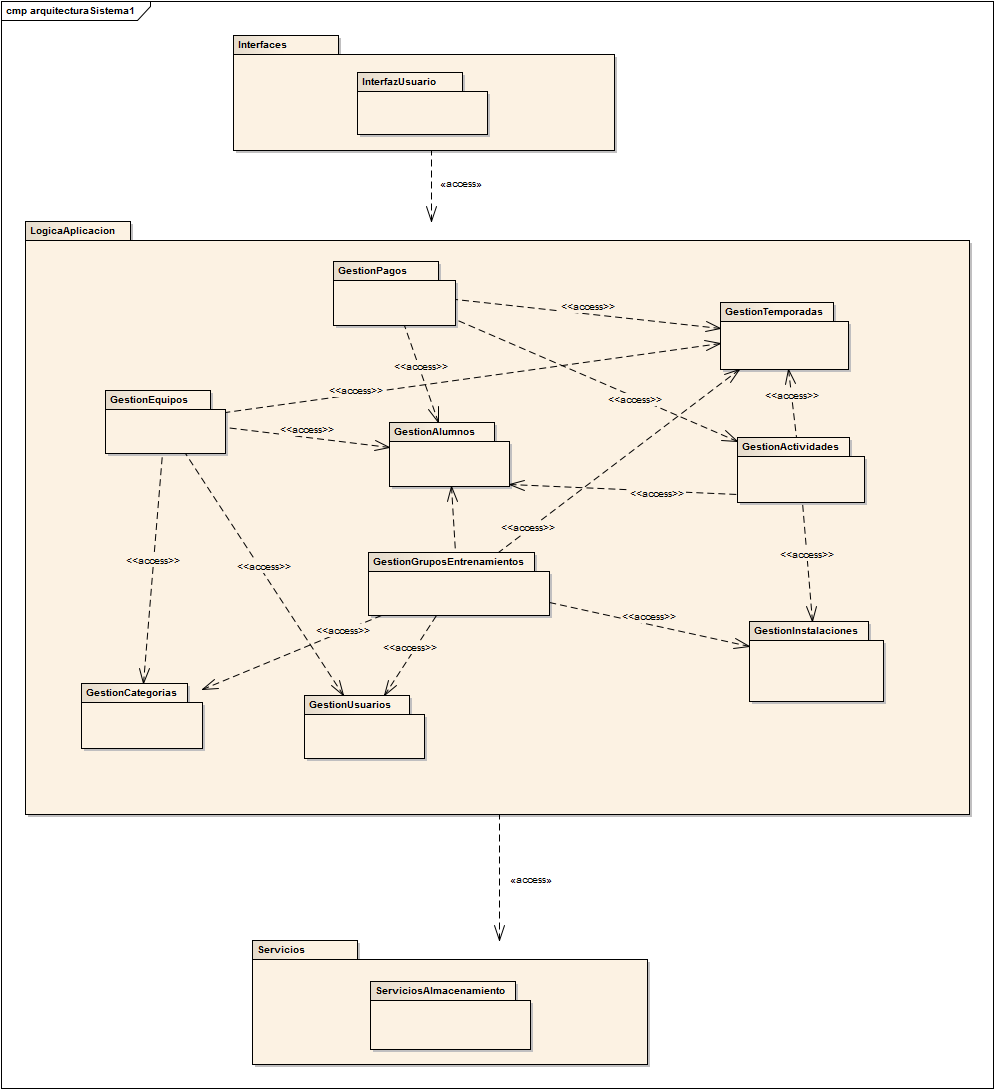
**Determinación de la arquitectura.**

Después de analizar los objetivos de diseño y los estilos arquitectónicos para el sistema se decide:

Utilizar una arquitectura software basada en capas cerradas que separe la aplicación en tres niveles ya que esto mejora la estructura del sistema y fomenta la flexibilidad en cuanto a sustitución de servicios o interfaz de usuario.

* Interfaz de usuario.
* Lógica de aplicación.
* Servicios

### Diagrama arquitectura software de tres capas cerradas



**Obtener Diagrama de Despliegue de Diseño.**

Para obtener el diagrama de despliegue de diseño se determinan los posibles nodos del sistema, con los cuales se especifica el hardware físico sobre el que se ejecuta el sistema de software y cómo se ubica el software en el hardware. Además se especifican los componentes y las conexiones entre los distintos nodos.

Se tienen los siguientes nodos, los cuales pueden ser dispositivos físicos o entornos de ejecución:

* PC, que es el ordenador que hay en la oficina.
* PC (Portatil), que es un portátil que pueden usar los entrenadores durante los partidos.
* Aplicación Fundación, que es la aplicación sobre la que se trabaja.
* Servidor de Persistencia, que es la tecnología que se usará para almacenamiento de datos.

Dentro de los nodos anteriores se anidan los siguientes nodos con el prototipo <<ExecutionEnvironment>> ,lo cual representa un tipo de entorno de ejecución para software.

* Execution Environment Windows 7.
* Execution Environment MySQL.

### Diagrama de despliegue de diseño



**Modelar Diagrama de componentes.**

El diagrama de paquetes está estructurado en agrupaciones lógicas, pero es necesario definir subsistemas, es decir, cerrar las partes reutilizables del sistema en componentes. Para ello es necesario añadir interfaces.

Con el diagrama de componentes se va a definir la implementación del sistema a partir de los módulos de software y su interrelación.

Con cada componente se va a representar una parte modular de un sistema que encapsula sus contenidos y cuya manifestación es reemplazable.

Cada componente puede:

* Tener atributos y operaciones.
* Representar a cualquier cosa desde clases sencillas a aplicaciones, subsistemas y sistemas.
* Tener interfaces proporcionadas, requeridas y puertos, es decir, servicios que ofrece un componente a otro o que necesita de otro.

Las interfaces permiten a cada componente comunicarse con otros componentes. Estas son una colección de operaciones que se utilizan para especificar un servicio de un componente.

Las interfaces facilitan la sustitución y reutilización de componentes:

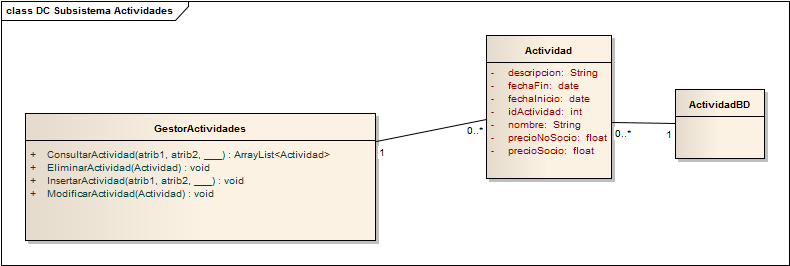
* Un componente puede exportar su interfaz e importar interfaces de más de un componente.
* Un componente que utiliza una interfaz determinada funcionará adecuadamente independientemente del componente que realice la interfaz.
* El componente que realiza la interfaz es siempre sustituible por un componente o conjunto de componentes que implementen dicha interfaz.
* Un componente puede utilizarse en un contexto determinado si y solo si todas sus interfaces de importación son suministradas por otros componentes.

Un puerto de componente agrupa un conjunto semánticamente cohesivo de interfaces proporcionadas y requeridas.

El diagrama de paquetes está estructurado en agrupaciones lógicas, pero necesitamos definir subsistemas, es decir, cerrar partes del sistema en componentes.

## Descomposición en Subsistemas

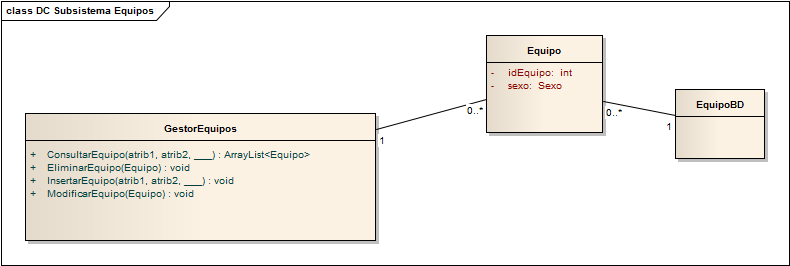
### Subsistema Gestión de Actividades



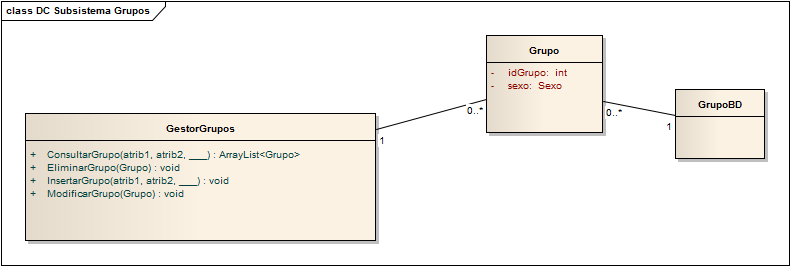
### Subsistema Gestión de Categorías

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagrama de Componentes\Diagrama de componentes de Subsistemas\DC Subsistema Categorias.png

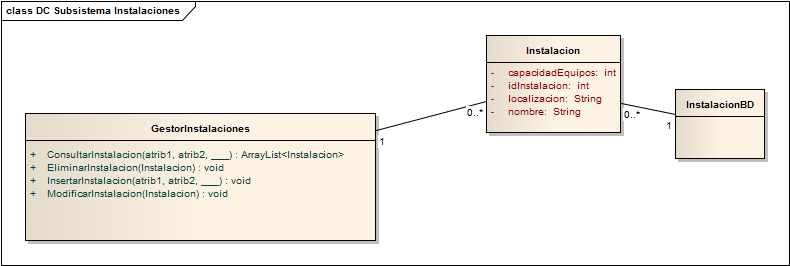
### Subsistema Gestión de Equipos



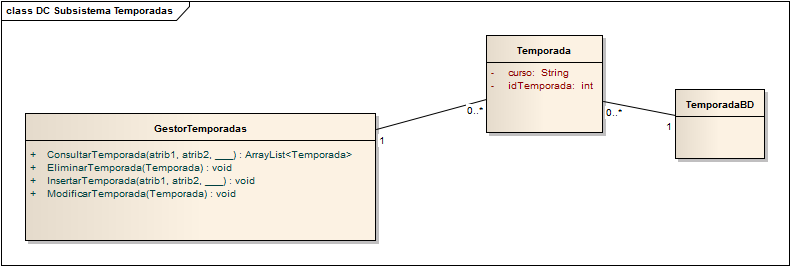
### Subsistema Gestión de Grupos



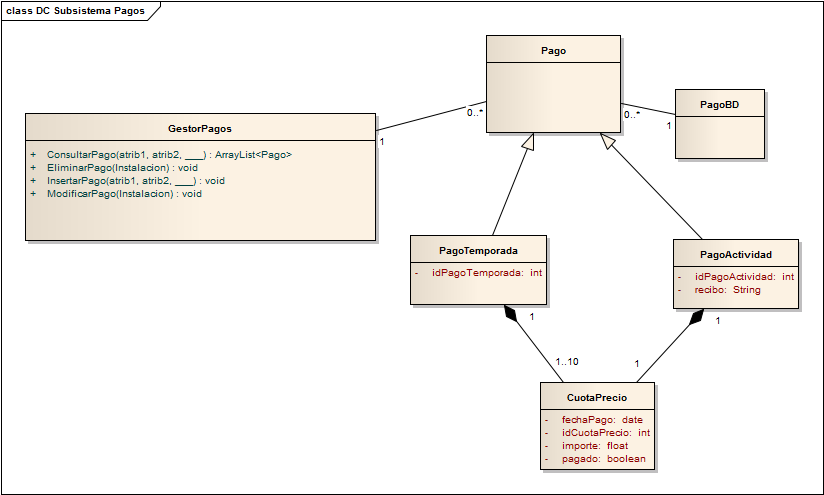
### Subsistema Gestión de Instalaciones



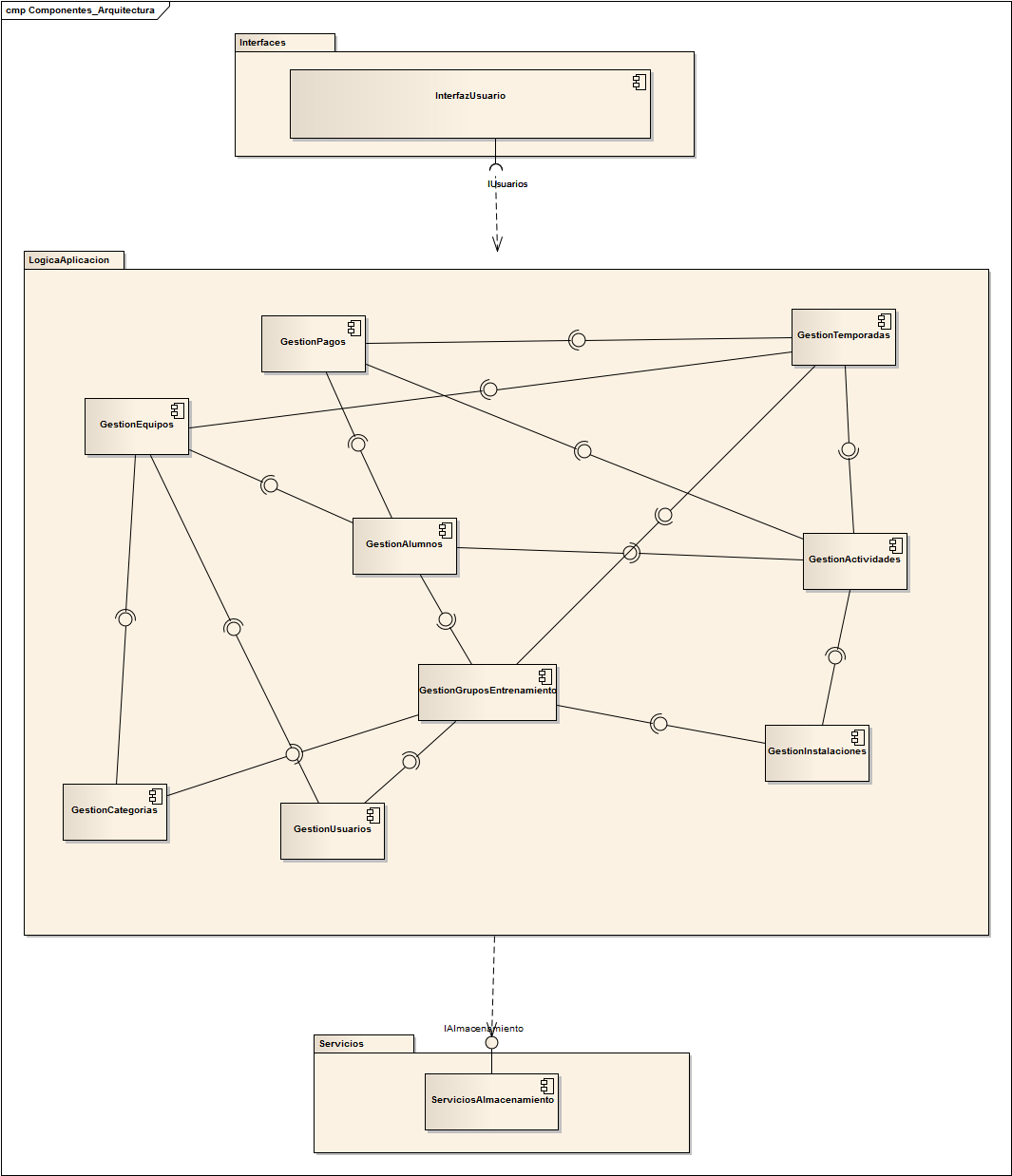
### Subsistema Gestión de Temporadas



### Subsistema Gestión de Pagos



### Arquitectura: Diagrama de Componentes

****

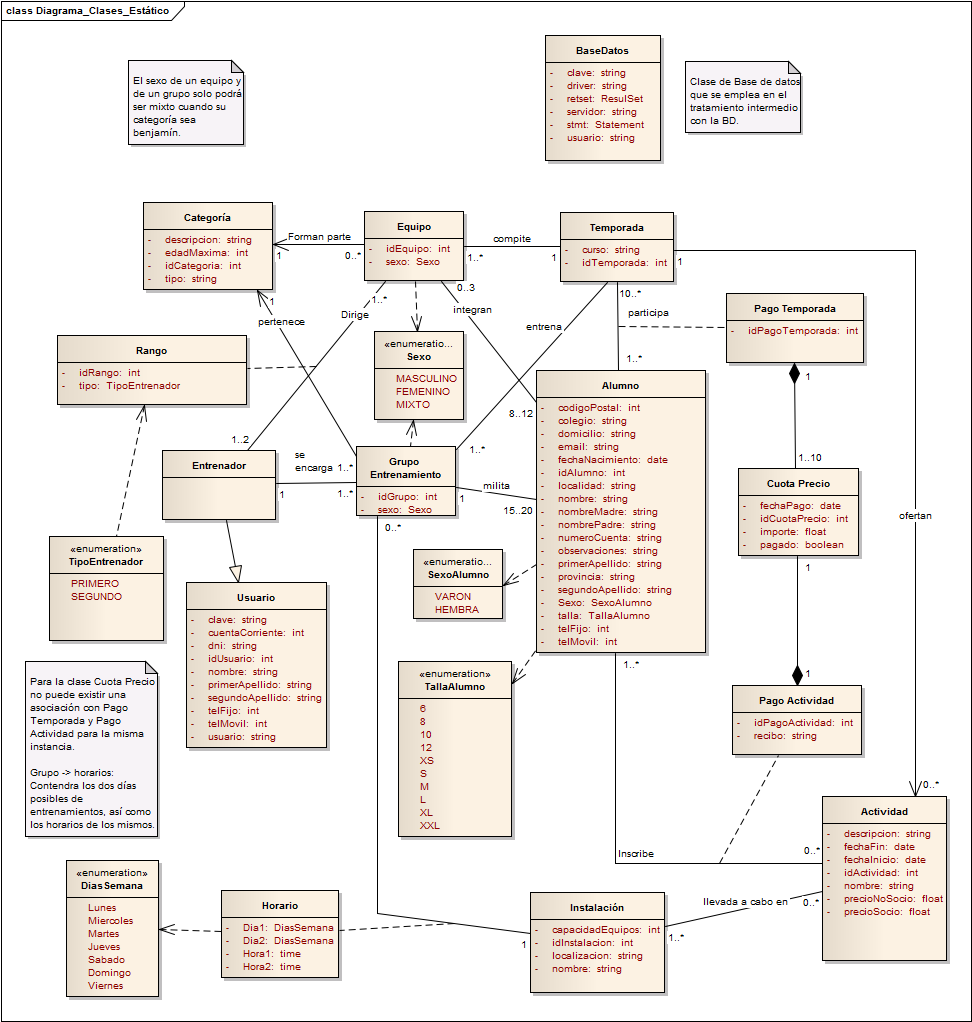
**Encajar el Diagrama de Clases (obtenido anteriormente) en la arquitectura obtenida en el apartado anterior**

**Diagrama de análisis**

Al terminar la etapa de diseño, se ha refinado suficientemente el diagrama de clases y las relaciones entre estas. También se conocen mejor los mensajes que se intercambian los objetos para realizar las tareas necesarias.

El modelo estático de análisis puede utilizarse para definir los paquetes de la capa de lógica de aplicación.

### Diagrama de clases estático.



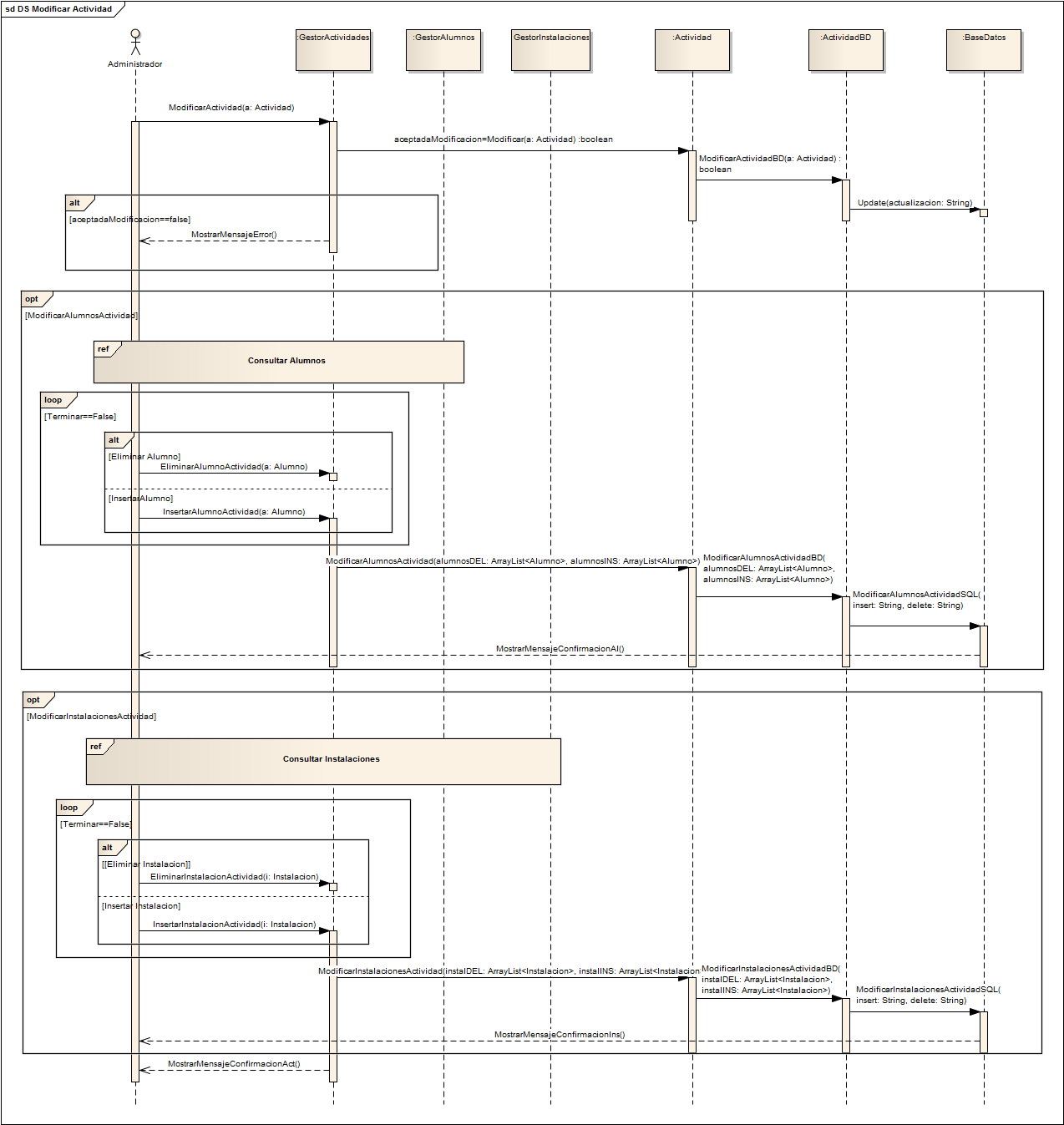
**Diagramas de Secuencia del Diseño.**

En esta etapa, las clases tienen ya definidas las operaciones. Además en estos diagramas se incluyen mensajes con las consultas a los objetos de control de la BD y se muestran las entidades creadas. A continuación se presentan algunos Diagramas de Secuencia de Diseño.

### DSD Gestión de Actividades

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionActividades\DS Insertar Actividad.pngDSD Insertar Actividad.

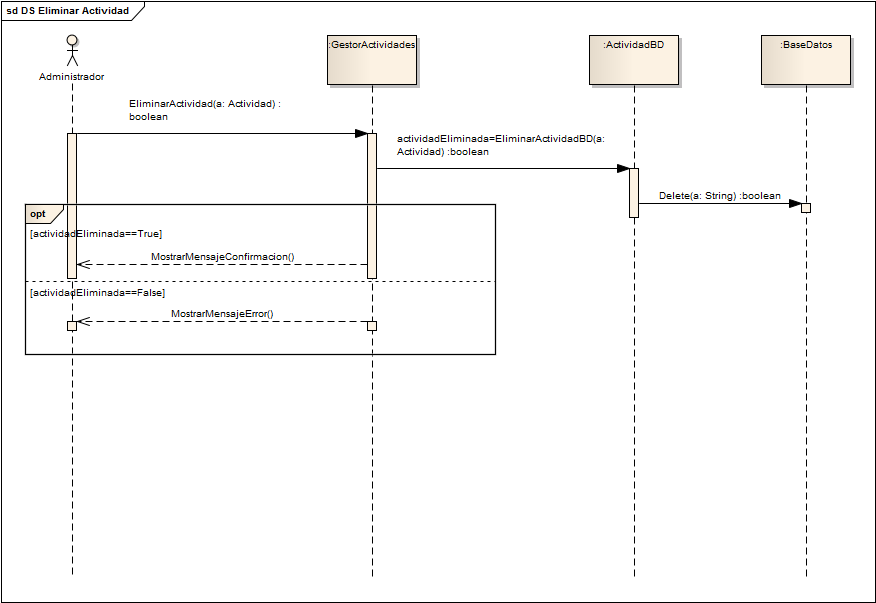
### DSD Modificar Actividad.



### DSD Consultar Actividad.

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionActividades\DS Consultar Actividad.png

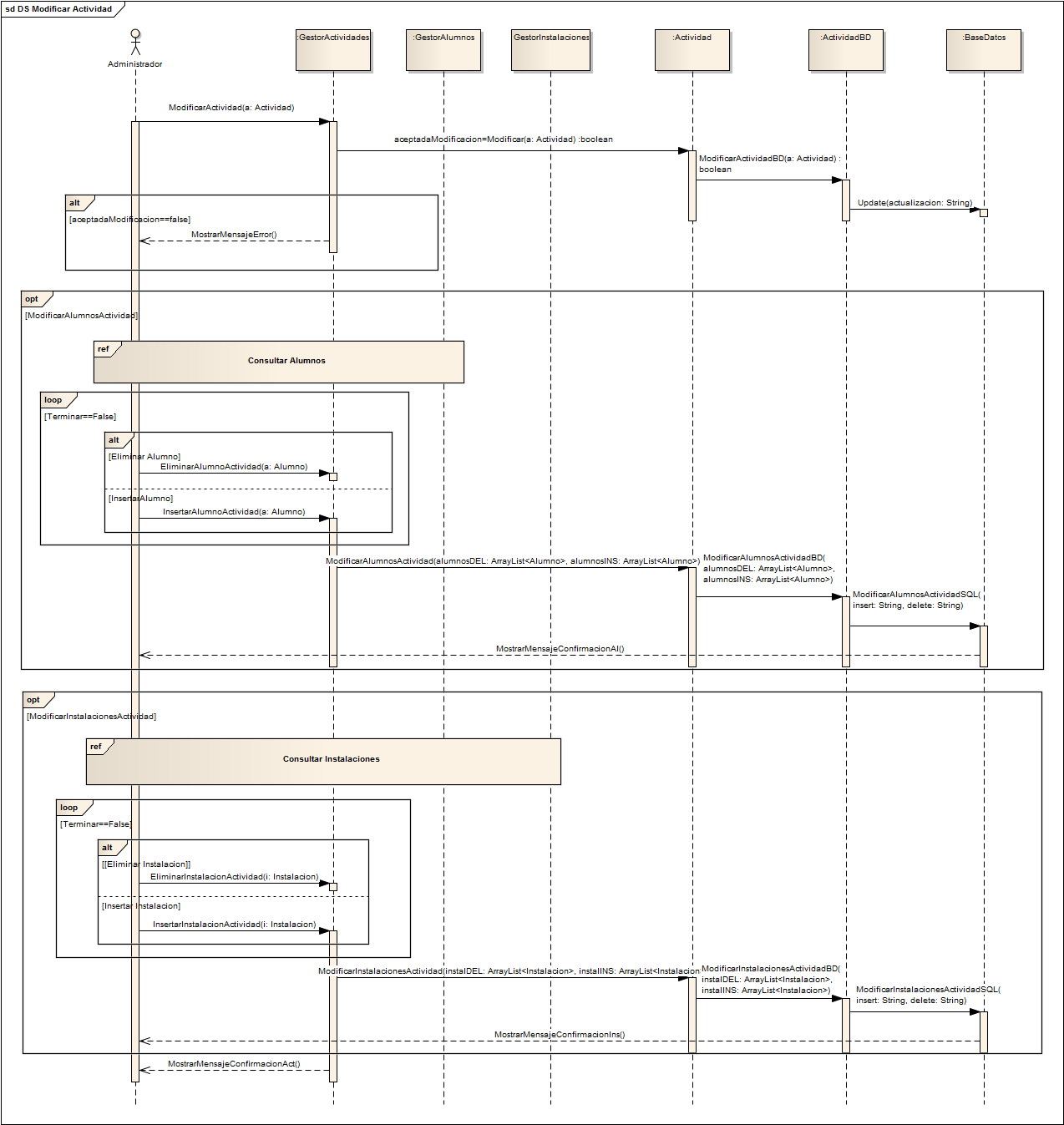
### DSD Eliminar Actividad.



### DSD Gestión de Actividades

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionActividades\DS Insertar Actividad.pngDSD Insertar Actividad.

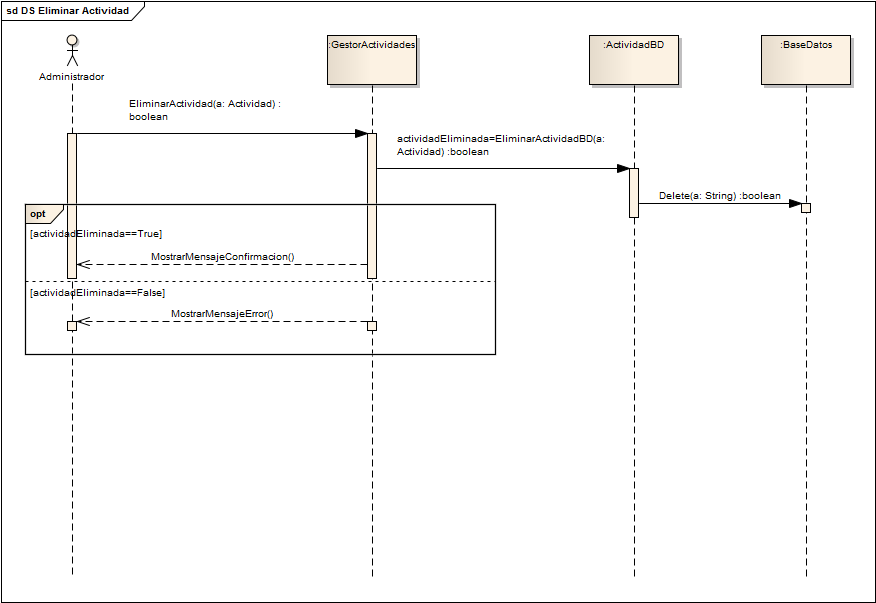
### DSD Modificar Actividad.



### DSD Consultar Actividad.

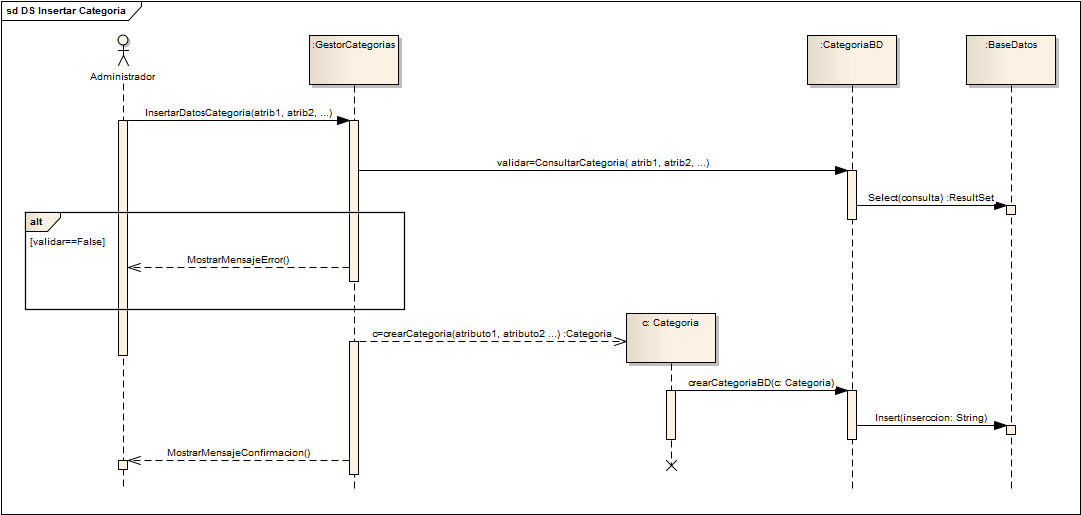
### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionActividades\DS Consultar Actividad.png

### DSD Eliminar Actividad.

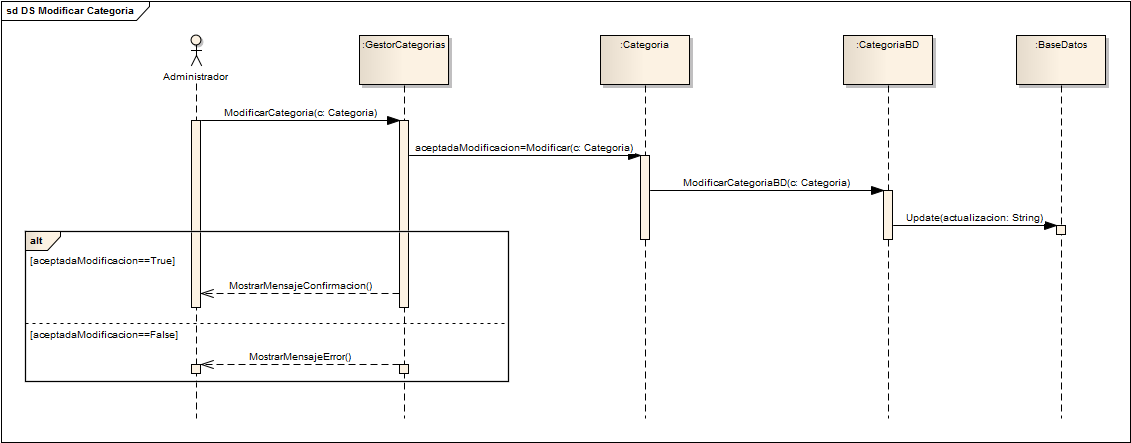


### DSD Gestión de Categorías

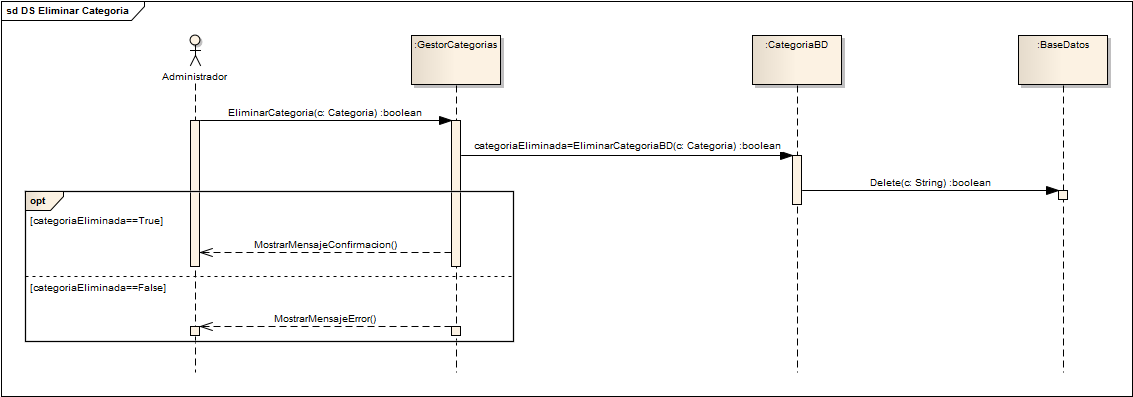
### DSD Insertar Categoría.



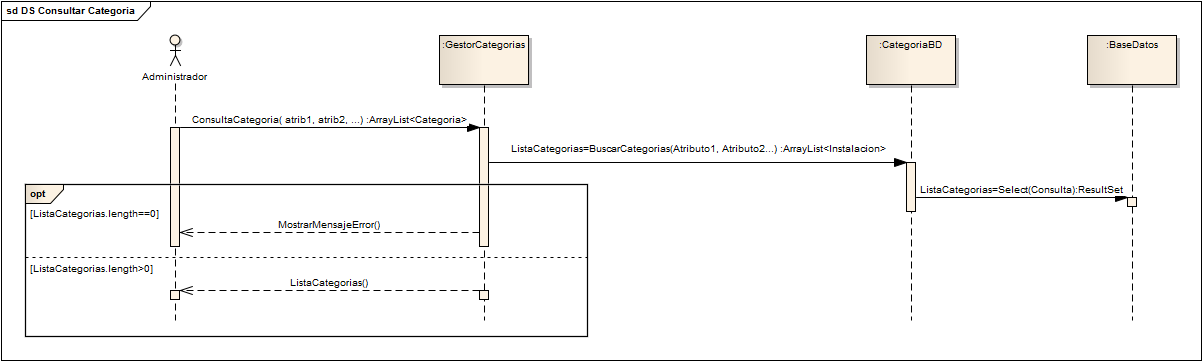
### DSD Modificar Categoría.



### DSD Eliminar Categoría.

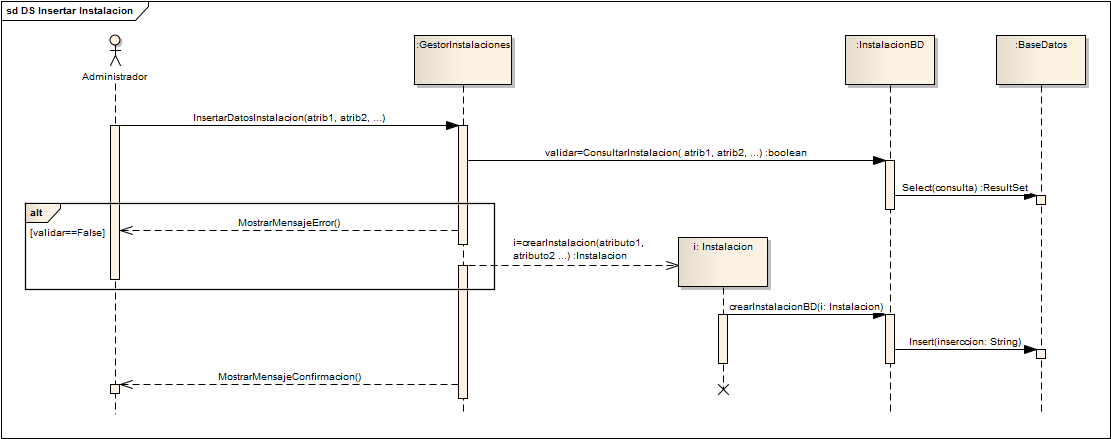


### DSD Consultar Categoría.



### DSD Gestión de Instalaciones

### DSD Insertar Instalación.



### DSD Modificar Instalación.

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionInstalaciones\DS Modificar Instalacion.png

### DSD Eliminar Instalación.

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionInstalaciones\DS Eliminar Instalacion.png

### DSD Consultar Instalación.

### C:\Users\Francisco\Documents\GitHub\Proyecto\docs\Diseño\Iteracion 2\Diagramas De Secuencia\Diagramas de Secuencia Diseño\DS GestionInstalaciones\DS Consultar Instalacion.png