



SUBEQUIPO DE DISEÑO - DISEÑO

Jesús Manuel Contreras Siles
Diego Muñoz Ríos



24 DE MAYO DE 2013

UNIVERSIDAD DE GRANADA

Contenido

Control de Versiones	2
Descomposición del sistema en subsistemas de diseño para obtener la arquitectura del sistema	3
Establecer la arquitectura del sistema	3
Subsistemas funcionales	3
Requerimientos no funcionales	3
Objetivos de diseño	4
Determinación de la arquitectura	4
Diagrama de arquitectura software de tres capas cerradas	5
Obtener Diagrama de Despliegue de Diseño	6
Diagrama de despliegue de diseño	7
Modelar diagrama de componentes	8
Subsistema Estadísticas Alumno.....	9
Subsistema estadísticas equipo.....	9
Subsistema Estadísticas entrenador.....	10
Subsistema estadísticas partido	10
Diagrama de componentes	12
Modelo del comportamiento externo – Diagrama de secuencia de las operaciones del sistema	13
Consultar estadística de jugador.....	13
Consultar clasificación	14
Consultar estadística	15
Consultar estadística de entrenador	16
Consultar estadística de temporada.....	17
Consultar partido	18
Modificar partido	19
Introducir partido.....	20
Eliminar partido.....	21
Introducir estadística de jugador del partido.....	22
Consultar estadística de jugador del partido	23
Anexo control de Versiones	24

Control de Versiones

Fecha	Versión	Descripción
24/05/2013	1.1	Entrega diseño

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 GestiónPartidos

Requerimientos no funcionales

Existen algunas restricciones y requisitos no funcionales que ya se detallaron en la etapa de análisis:

- No requiere un conocimiento específico para la utilización del software.
- La aplicación tendrá manual de uso.
- La base de datos estará implementada en el lenguaje objeto relacional como MySQL.
- La aplicación estará realizada en el lenguaje de programación Java.
- Se creará 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 desplegará sobre una distribución Microsoft Windows.

- El administrador se encargará de tareas de mantenimiento.
- Cuando el sistema detecte una anomalía mostrará al usuario el mensaje de error pertinente y abortará 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/yyyy.
- El control de asistencia no es controlado, pero se puede incluir en la sección de observaciones de los alumnos.
- Durante el partido, el segundo entrenador introducirá las estadísticas de los alumnos en tiempo real, pero no contará con conexión a internet, por lo que la información se guardará en un fichero XML temporal.

Objetivos de diseño

A partir de los requerimientos no funcionales se obtienen los siguientes objetivos de diseño:

- **Facilidad de uso:** Utilizando un entorno gráfico intuitivo, basado en ventanas, adaptable a los posibles tipos de usuarios y con un manual de usuario.
- El sistema debe facilitar la incorporación de la tecnología MySQL para el manejo de la persistencia.
- **Seguridad:** Debe haber mecanismos de control de acceso para todos los usuarios del sistema.
- **Fiabilidad:** se espera que el sistema responda de forma satisfactoria.

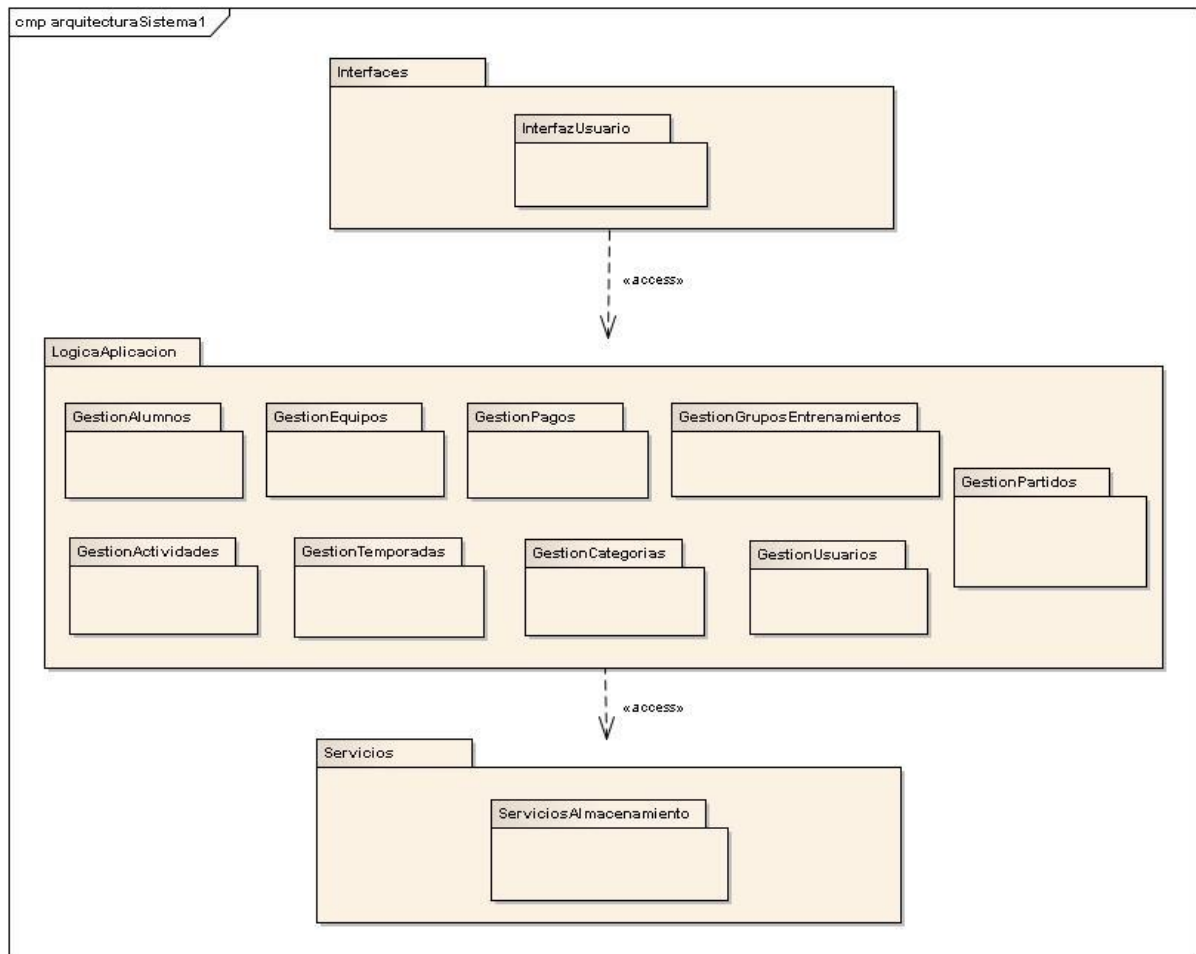
Determinación de la arquitectura

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 de 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 sistema software en el hardware. Además se especifican los componentes y las conexiones entre los distintos nodos.

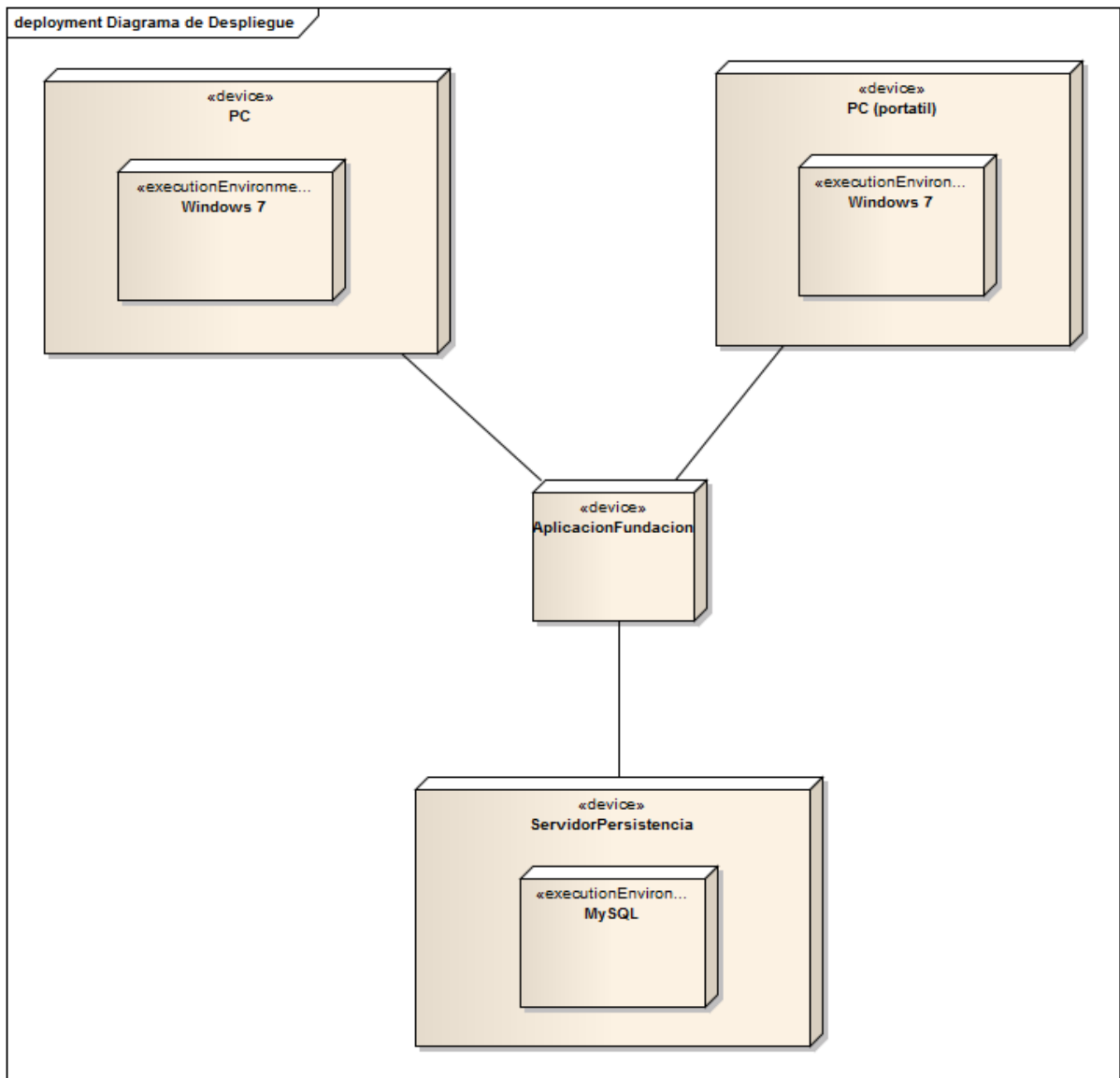
Se tienen los siguientes nodos, los cuales pueden ser dispositivos físicos o entornos de ejecución:

- PC, que es el ordenador que hay en la oficina.
- PC (portátil), 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 <<Execution Environment>>, 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.

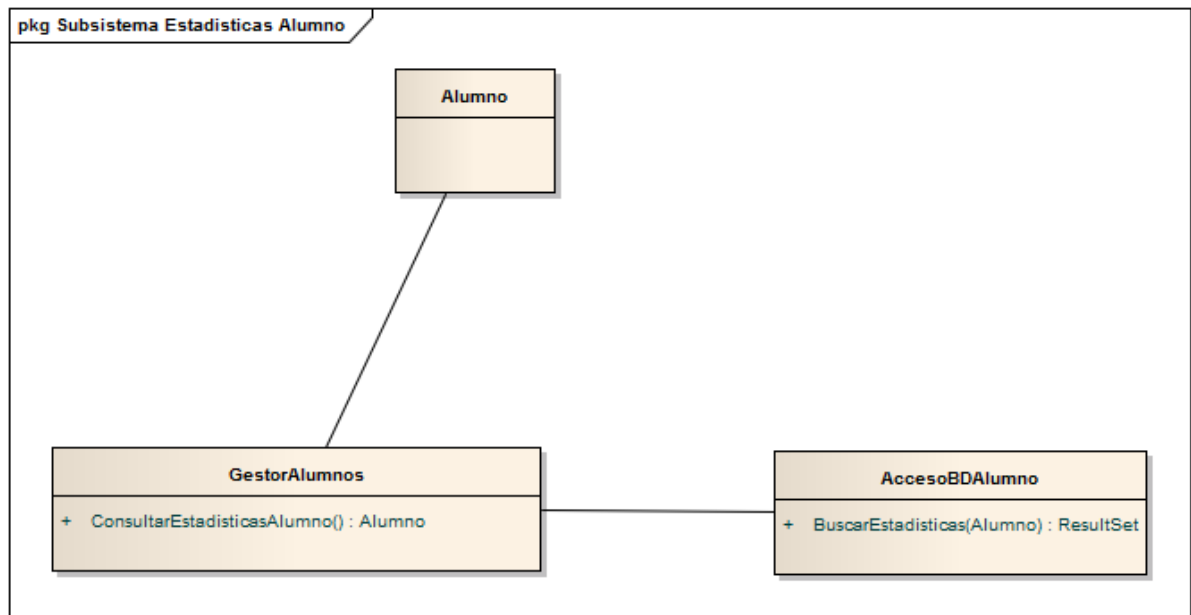
Las interfaces facilitan la sustitución y reutilización de componentes:

- 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 implementan dicha interfaz.
- Un componente puede utilizarse en un contexto determinado si y solo si todas sus interfaces de importación son suministradas por otros componentes.

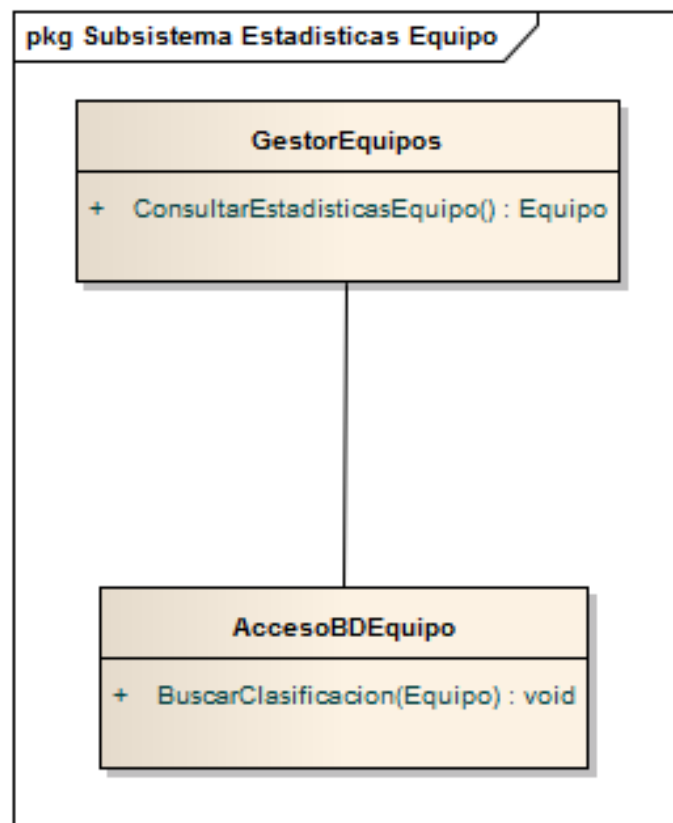
Un puerto de componente agrupa un conjunto semánticamente cohesivo de interfaces proporcionadas y requeridas.

Aclarado todo sobre los diagramas de componentes se muestran los siguientes diagramas obtenidos

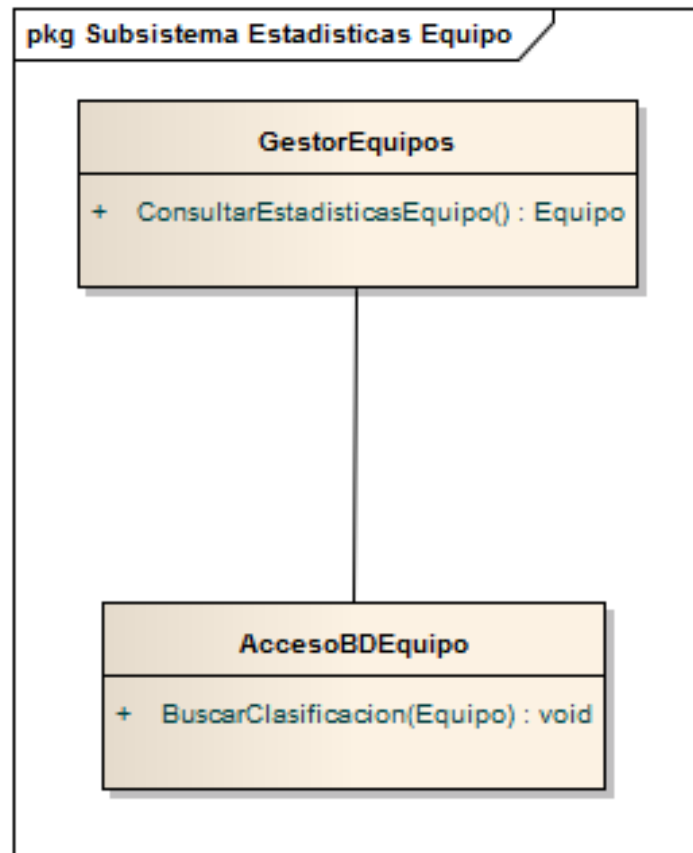
Subsistema Estadísticas Alumno



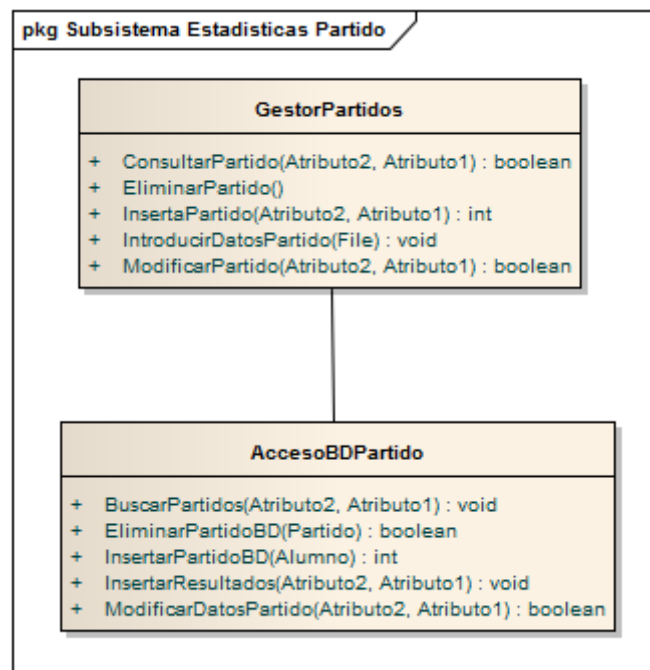
Subsistema estadísticas equipo



Subsistema Estadísticas entrenador



Subsistema estadísticas partido



Subsistema estadísticas temporada

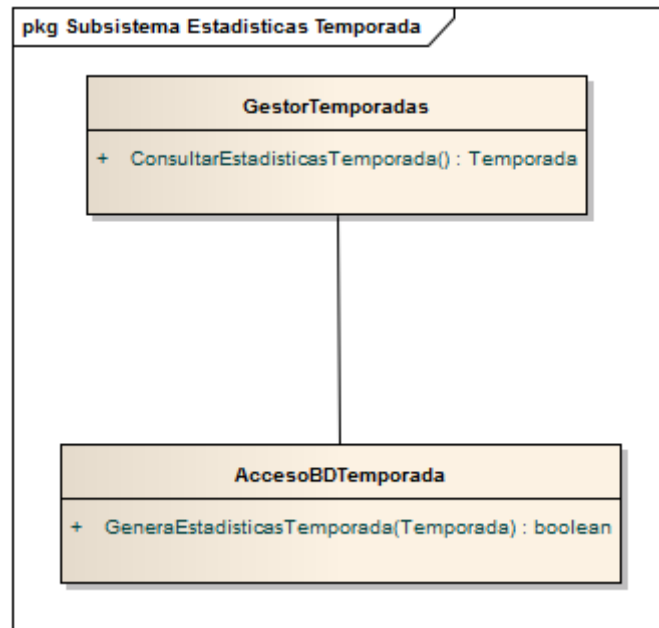
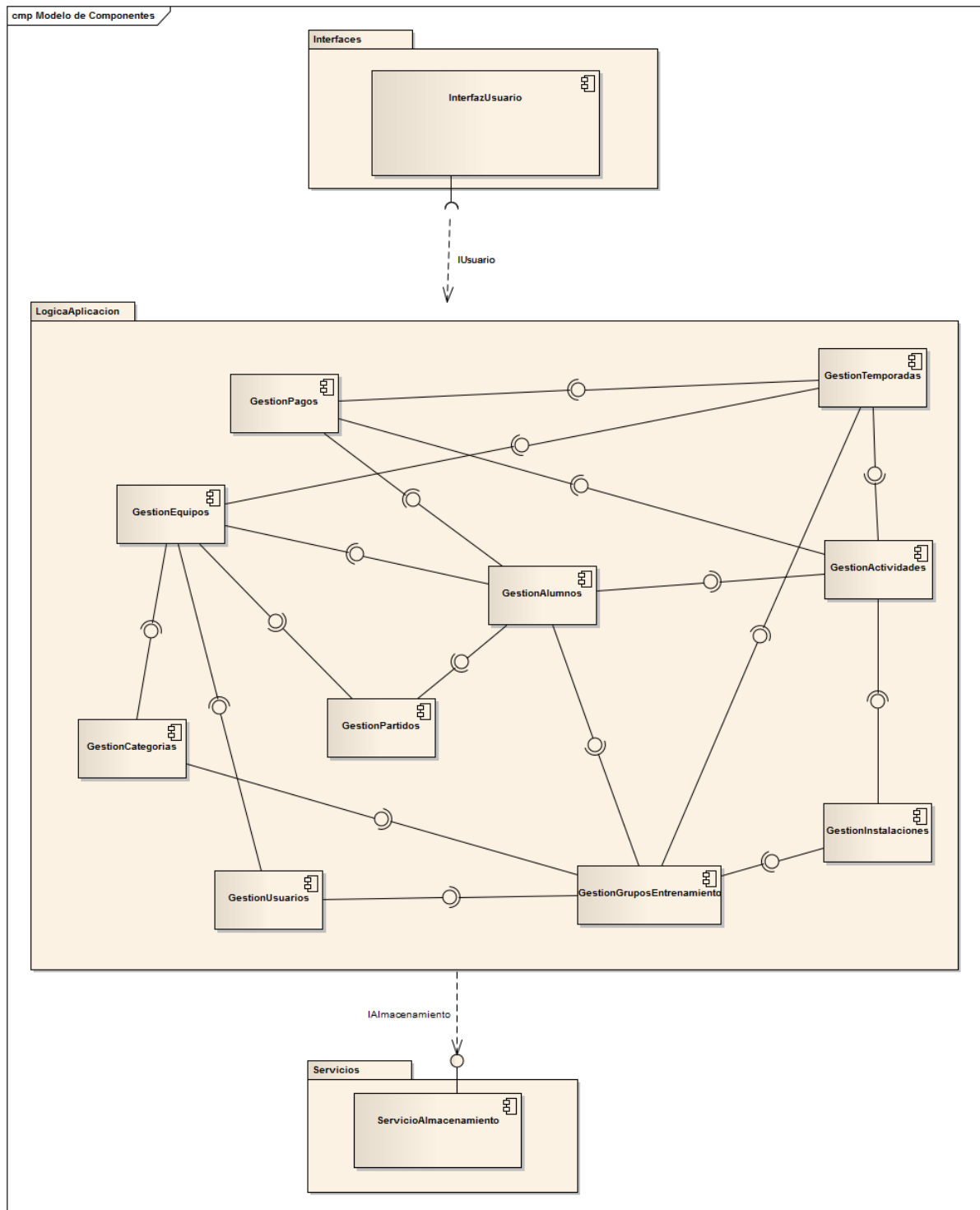


Diagrama de componentes



Modelo del comportamiento externo – Diagrama de secuencia de las operaciones del sistema

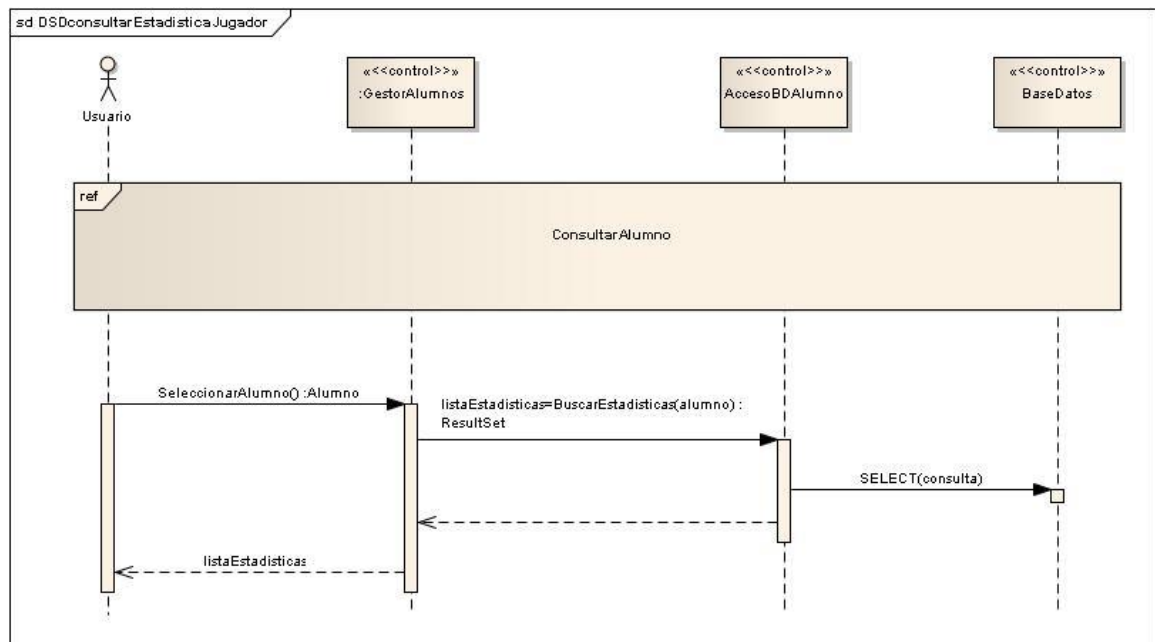
Es un documento que describe lo que una operación se propone lograr, sin decir cómo se conseguirá.

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

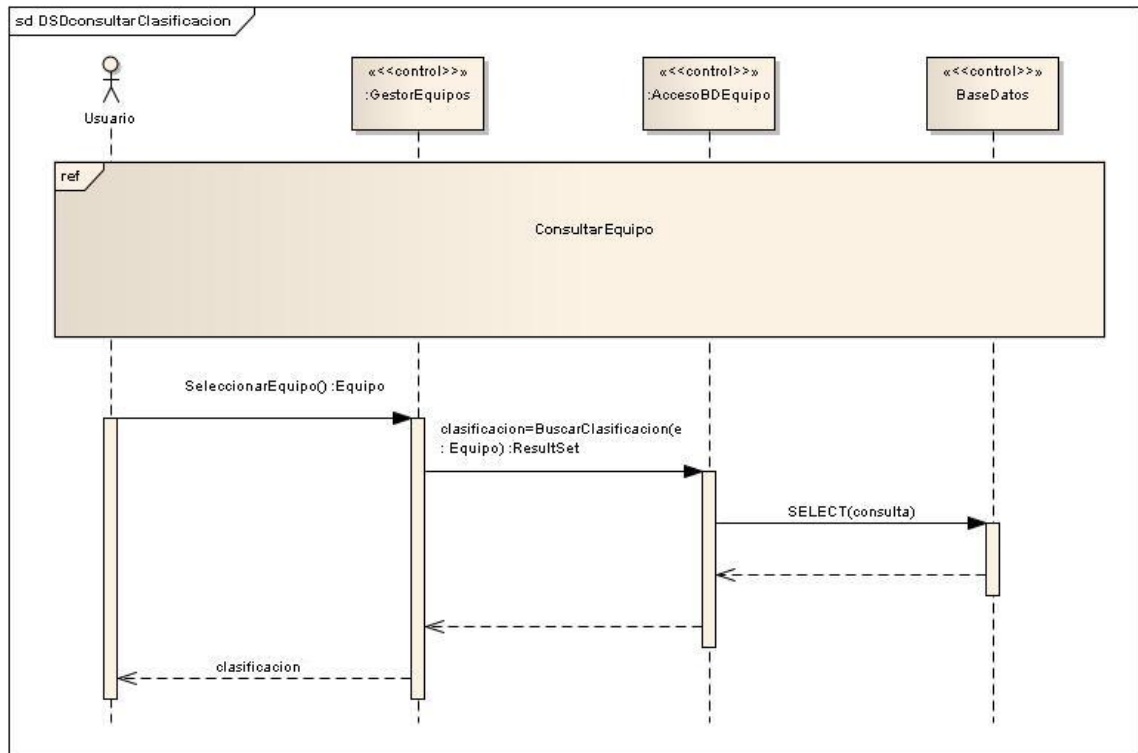
En este caso no serán diagramas de secuencia del sistema como tales, sino las operaciones que definen los contratos de los diagramas de secuencia obtenidos.

A continuación se describen los diagramas de secuencia de las operaciones de los contratos anteriores.

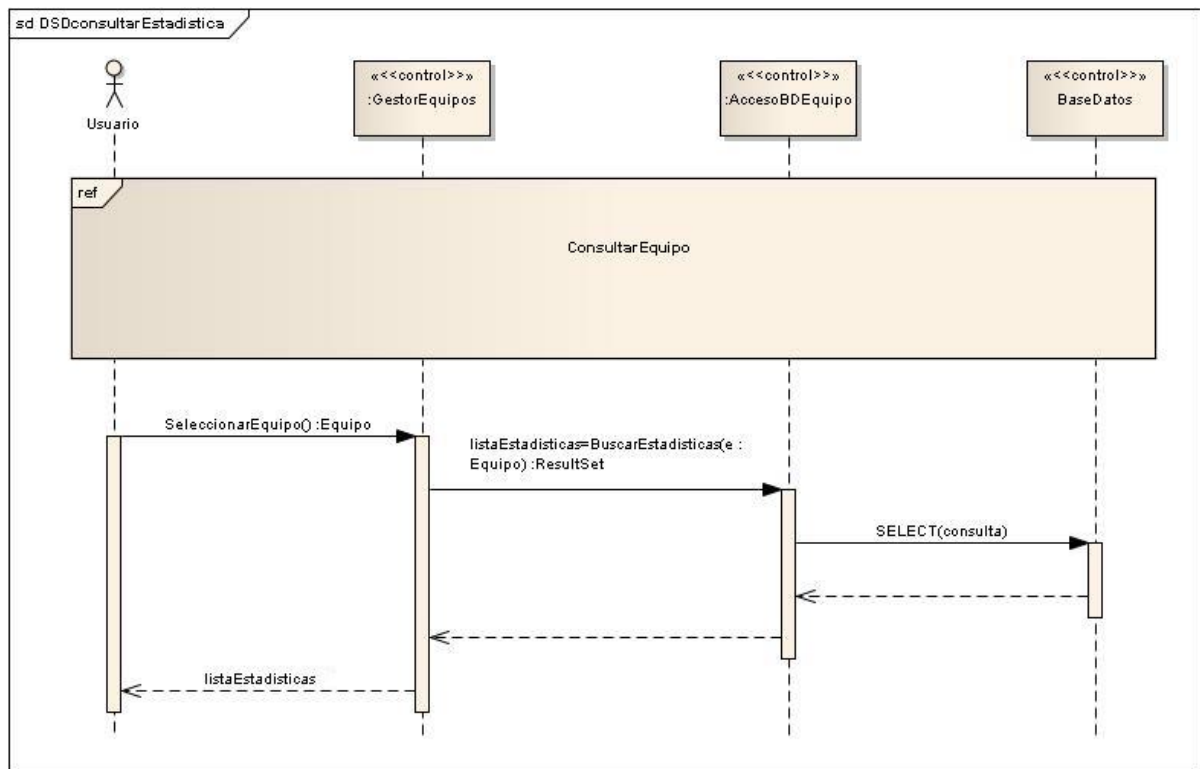
Consultar estadística de jugador



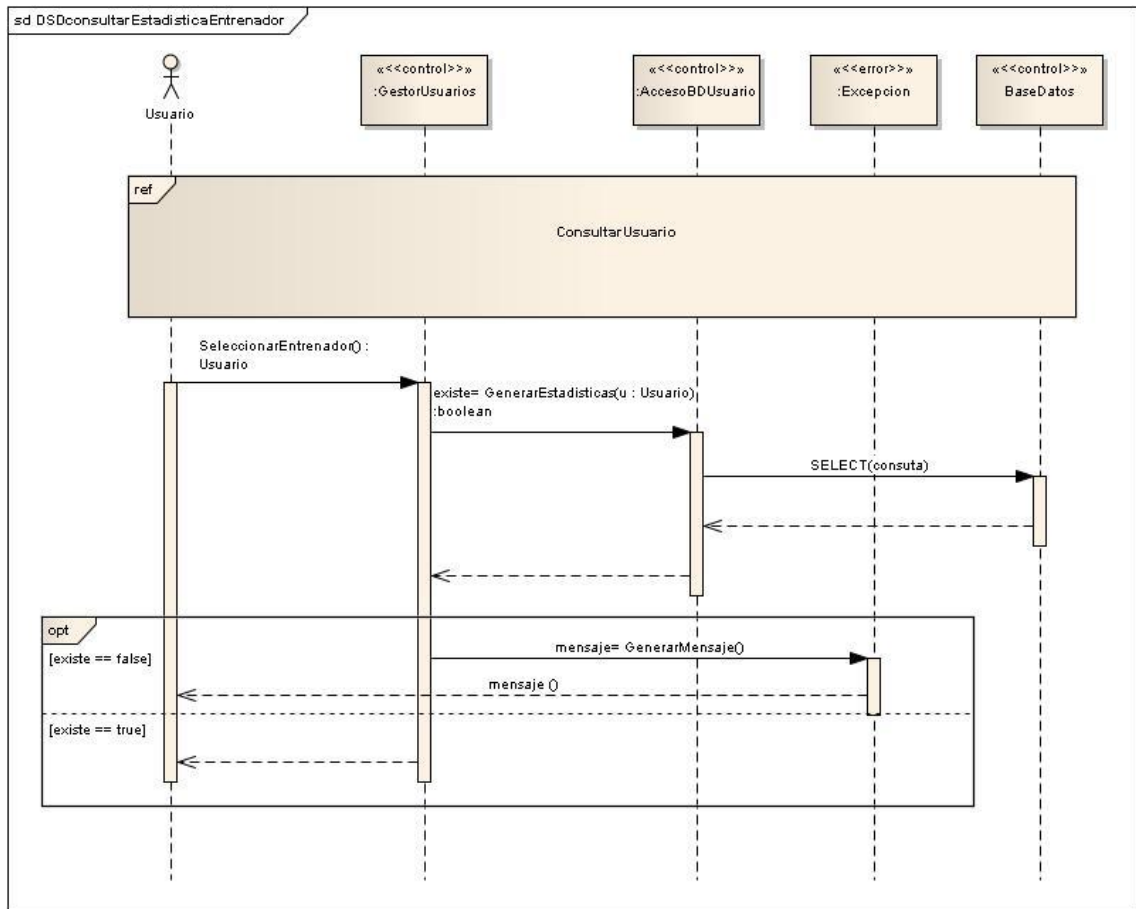
Consultar clasificación



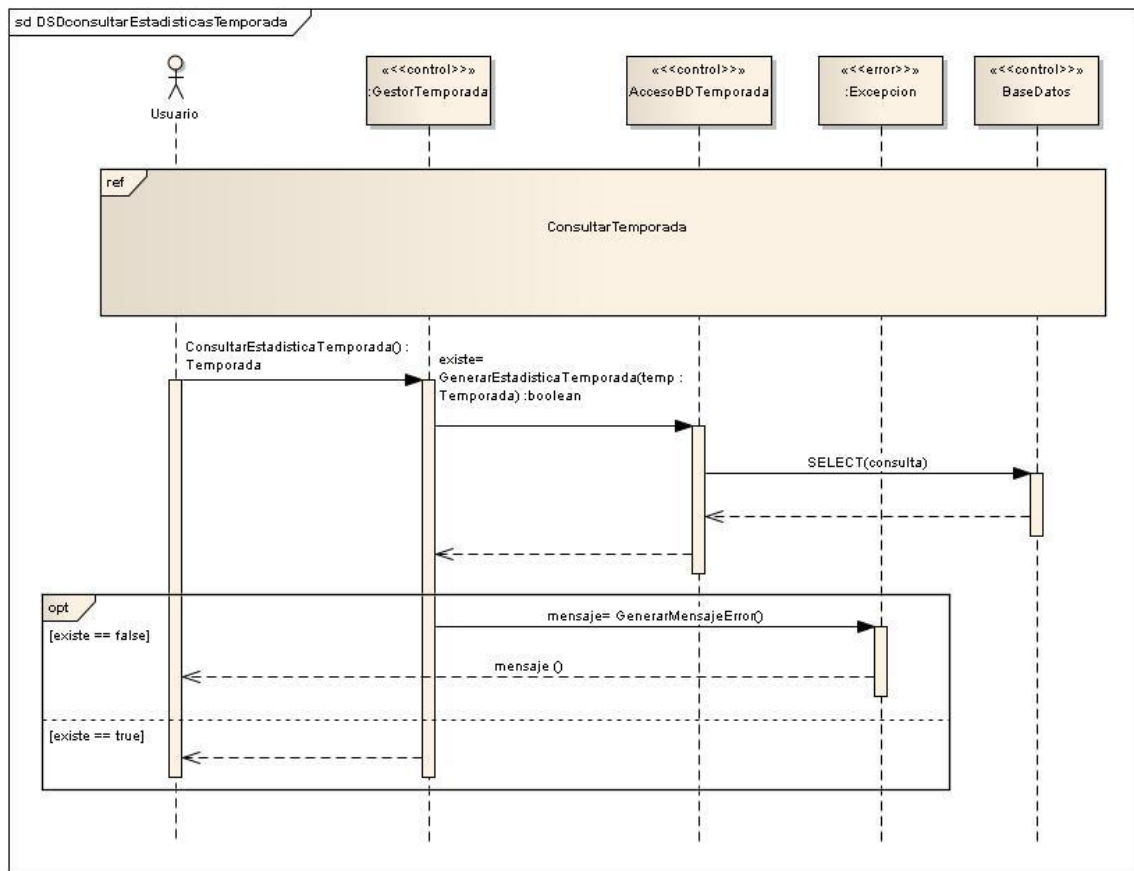
Consultar estadística



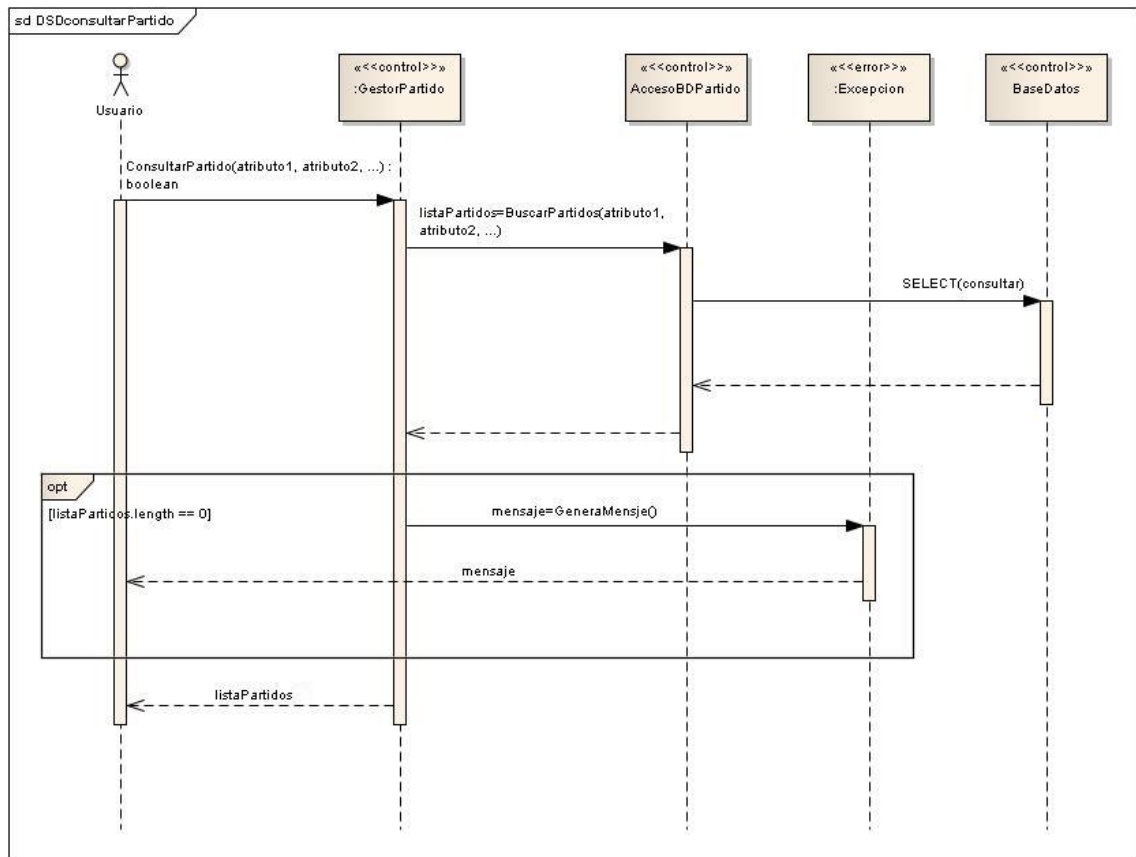
Consultar estadística de entrenador



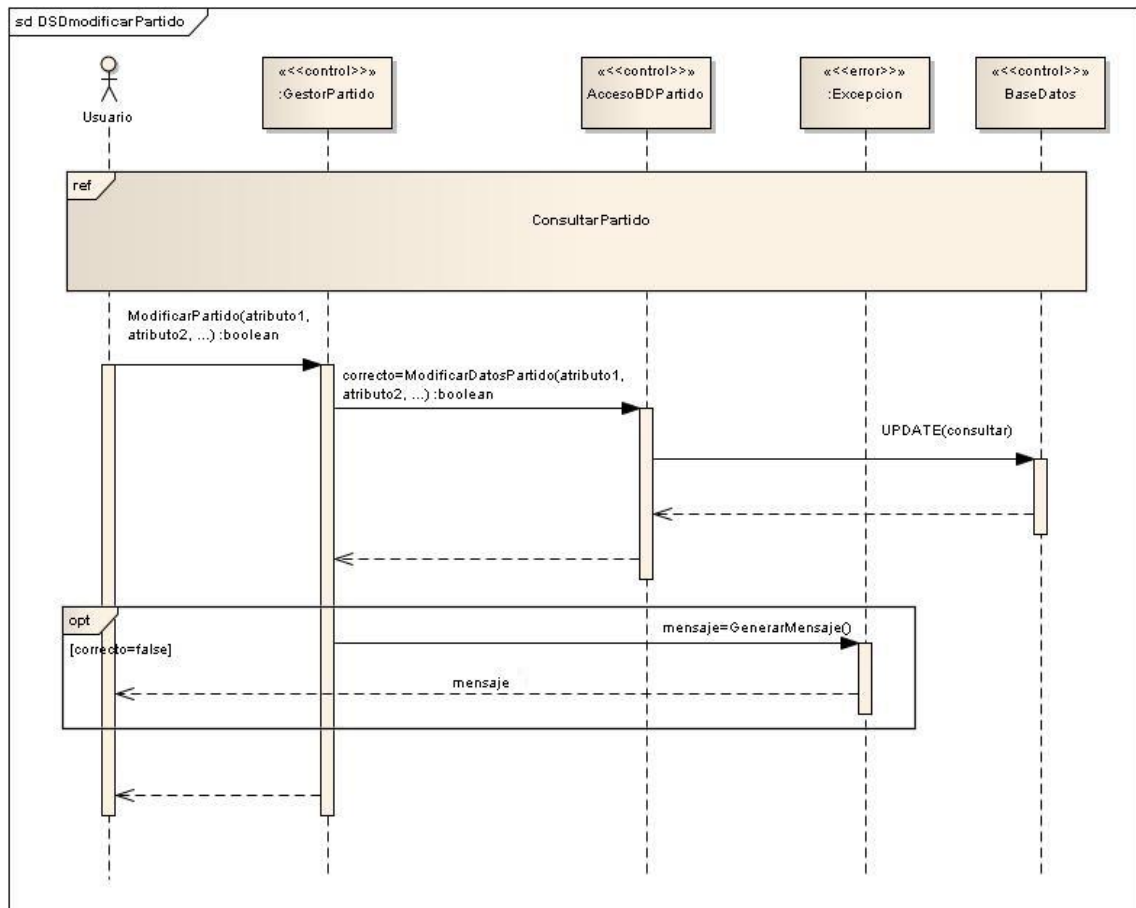
Consultar estadística de temporada



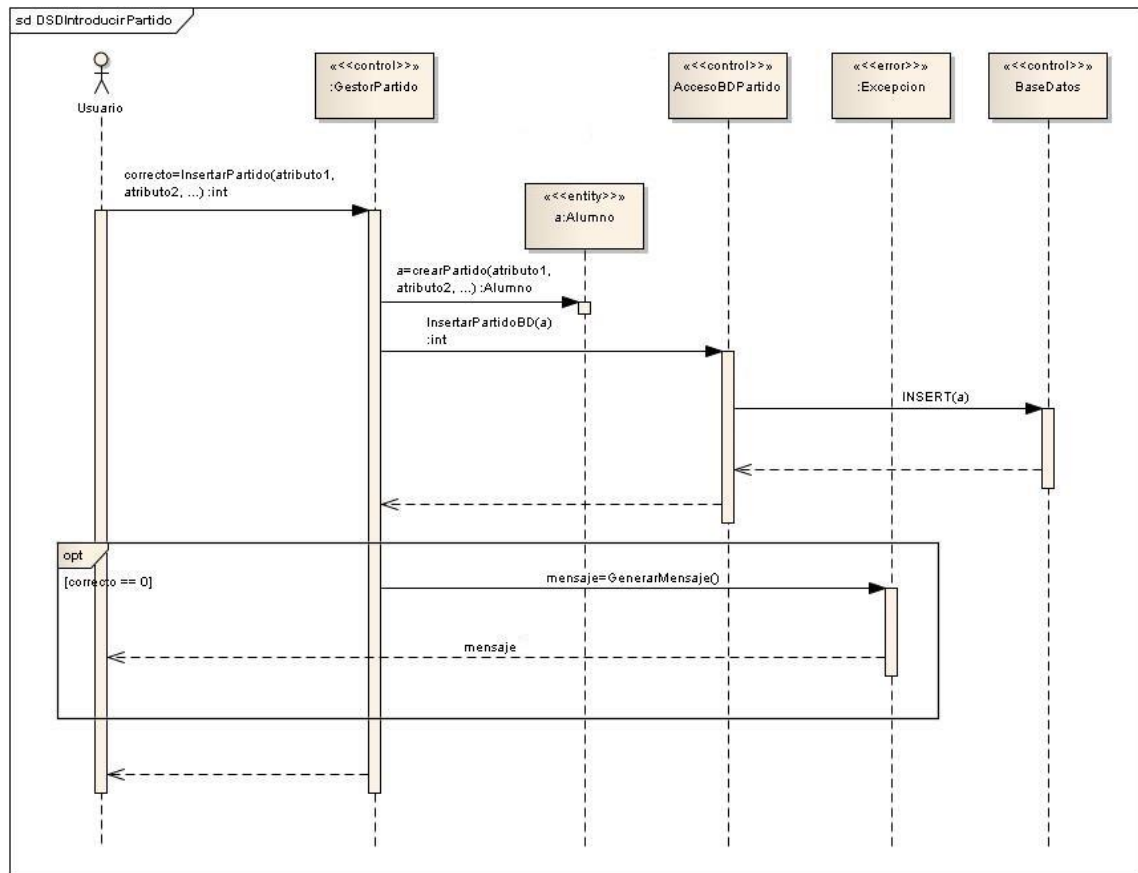
Consultar partido



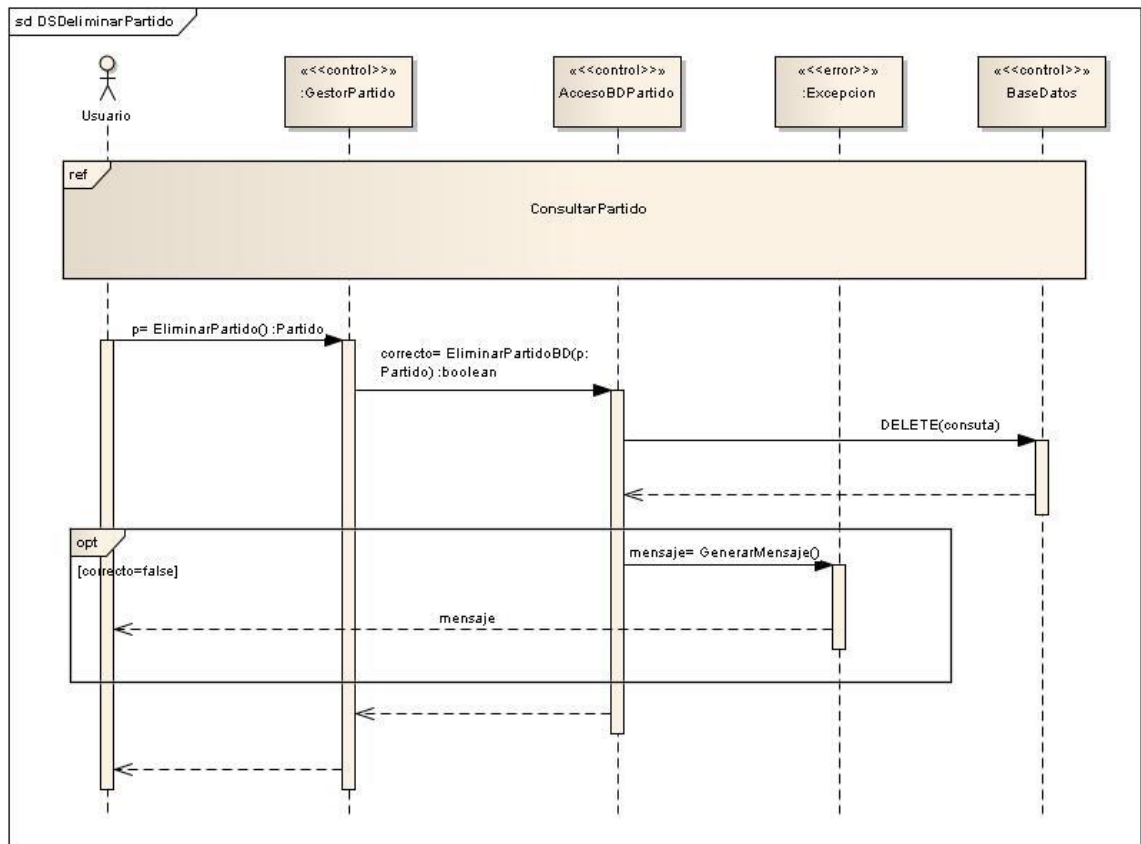
Modificar partido



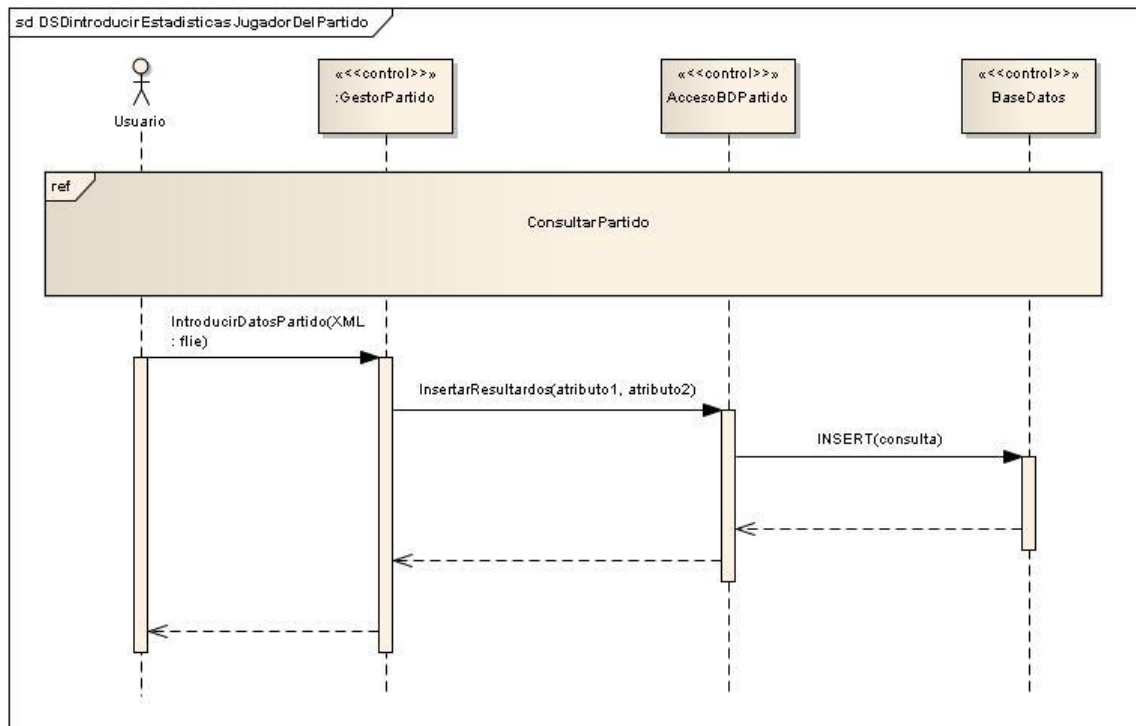
Introducir partido



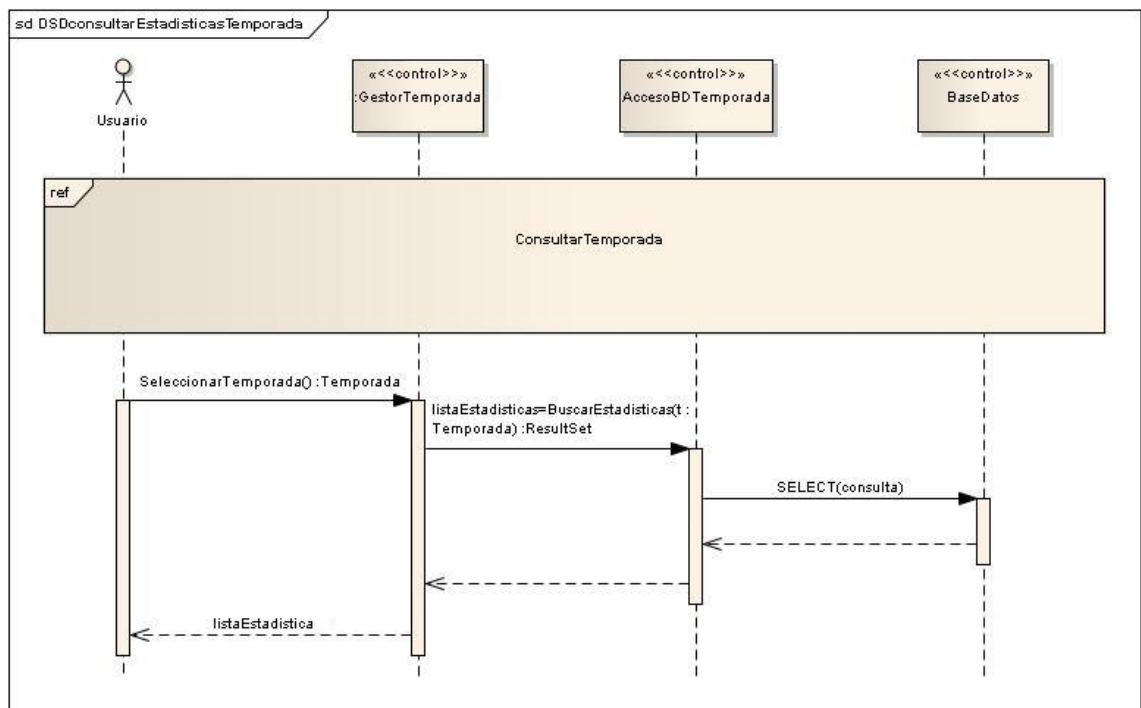
Eliminar partido



Introducir estadística de jugador del partido



Consultar estadística de jugador del partido



Anexo control de Versiones

Fecha: 27/3/2013

Versión: 1.1

- Diagramas de Paquetes Arquitectura. v.1.1.
- Diagrama de Despliegue de Diseño. v.1.1.
- Diagramas de Componentes. v.1.1.
- Diagramas de Clases. v.1.1
- Diagramas de Secuencia de Diseño. v.1.1.