ENTREGA SUBEQUIPO DE DISEÑO – DISEÑO

Moisés Gautier Gómez
Francisco Javier Gómez del Olmo
Julio Ros Martínez

27 DE MARZO DE 2013

Universidad de Granada

Contenido

Control de Versiones.

- 1º Control de versión

Fecha	Versión	Descripción
27/3/2013	1.1	Entrega diseño 27-3-2013

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 GestionGruposEntrenamiento.
- Subsistema GestionEquipos.
- Subsistema GestionUsuarios.
- Subsistema GestionActividades.
- Subsistema GestionTemporadas.
- Subsistema GestionCategorias.
- Subsistema GestionPagos.

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 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.

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.

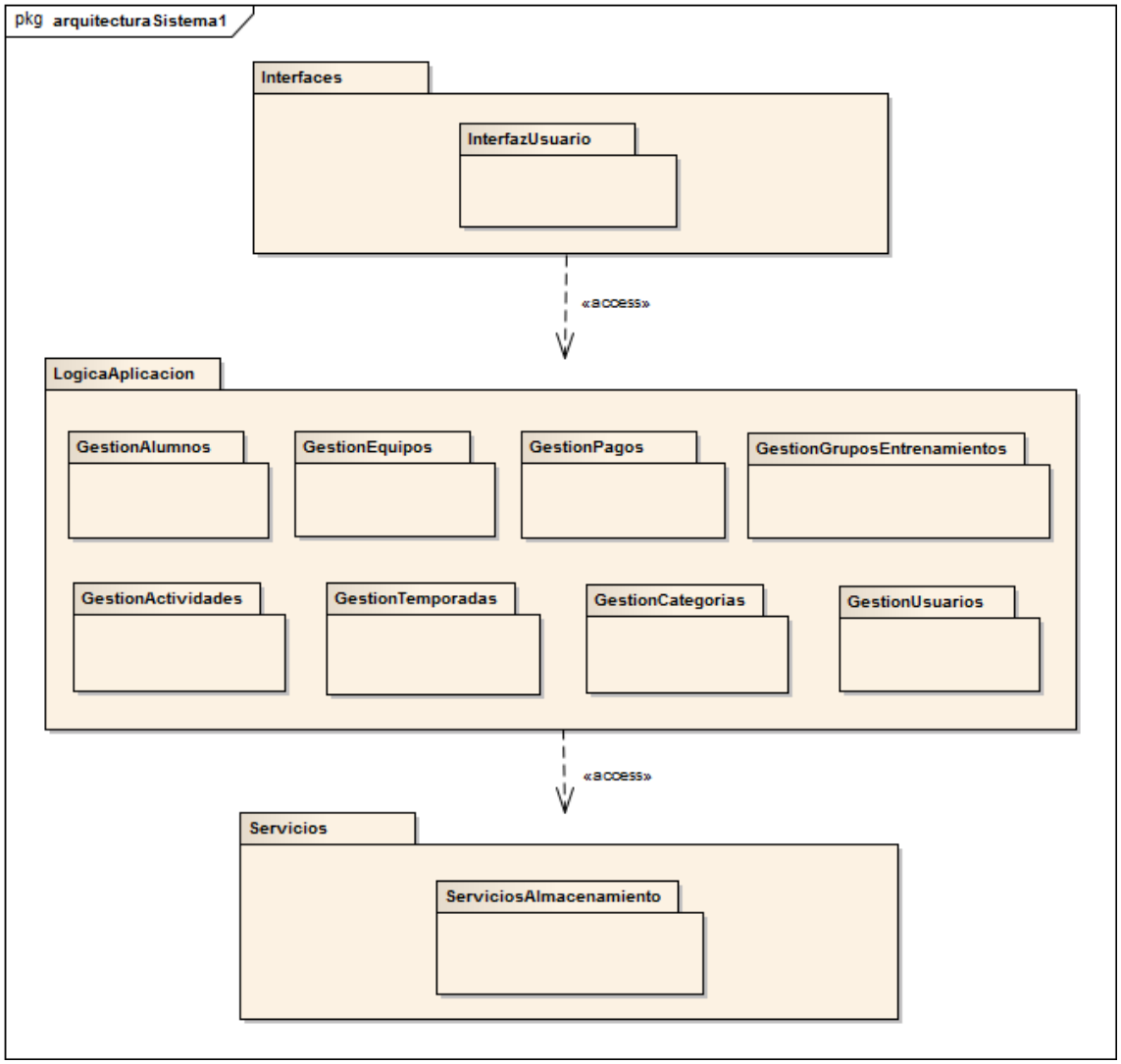
Determinación de la arquitectura.

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.

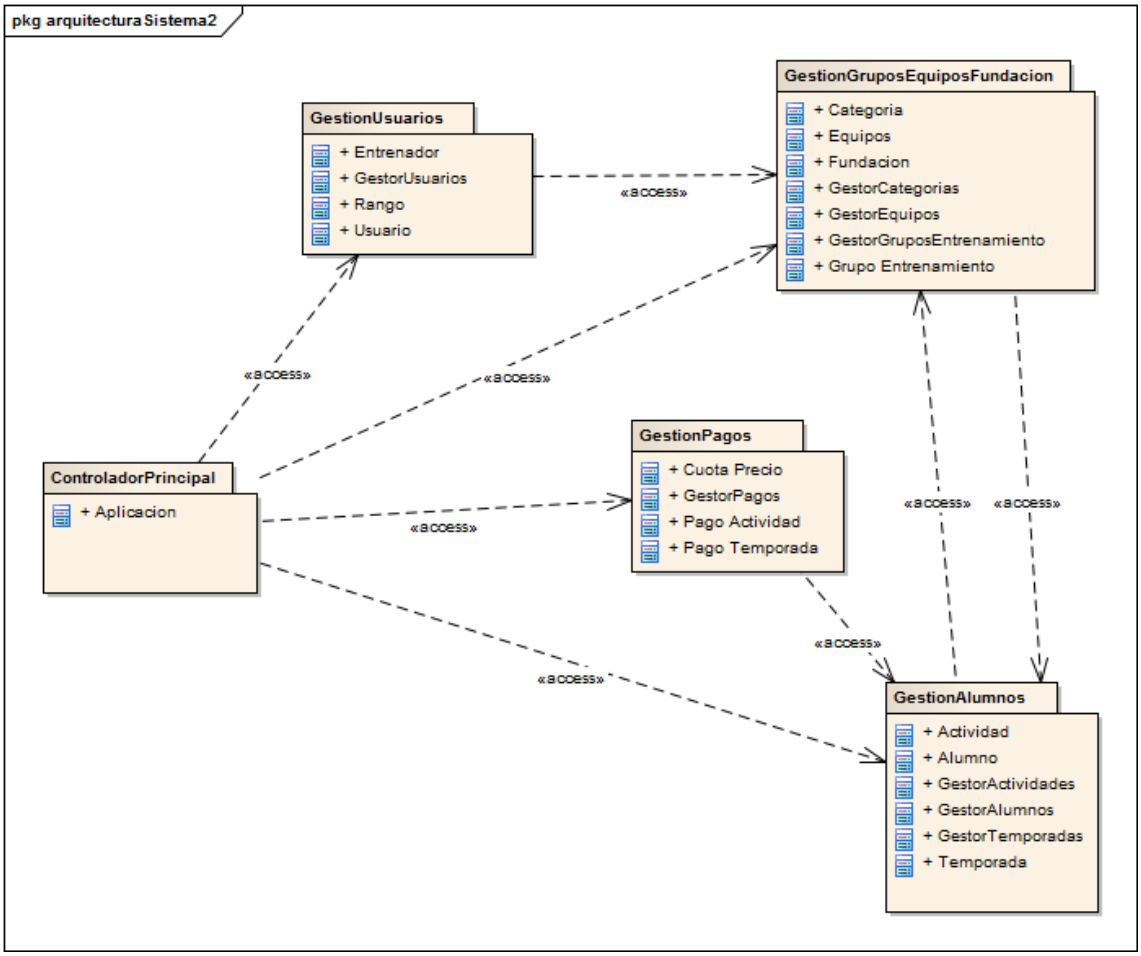
Diagrama arquitectura software de tres capas cerradas.



Modelar la arquitectura :Diagramas de paquetes.

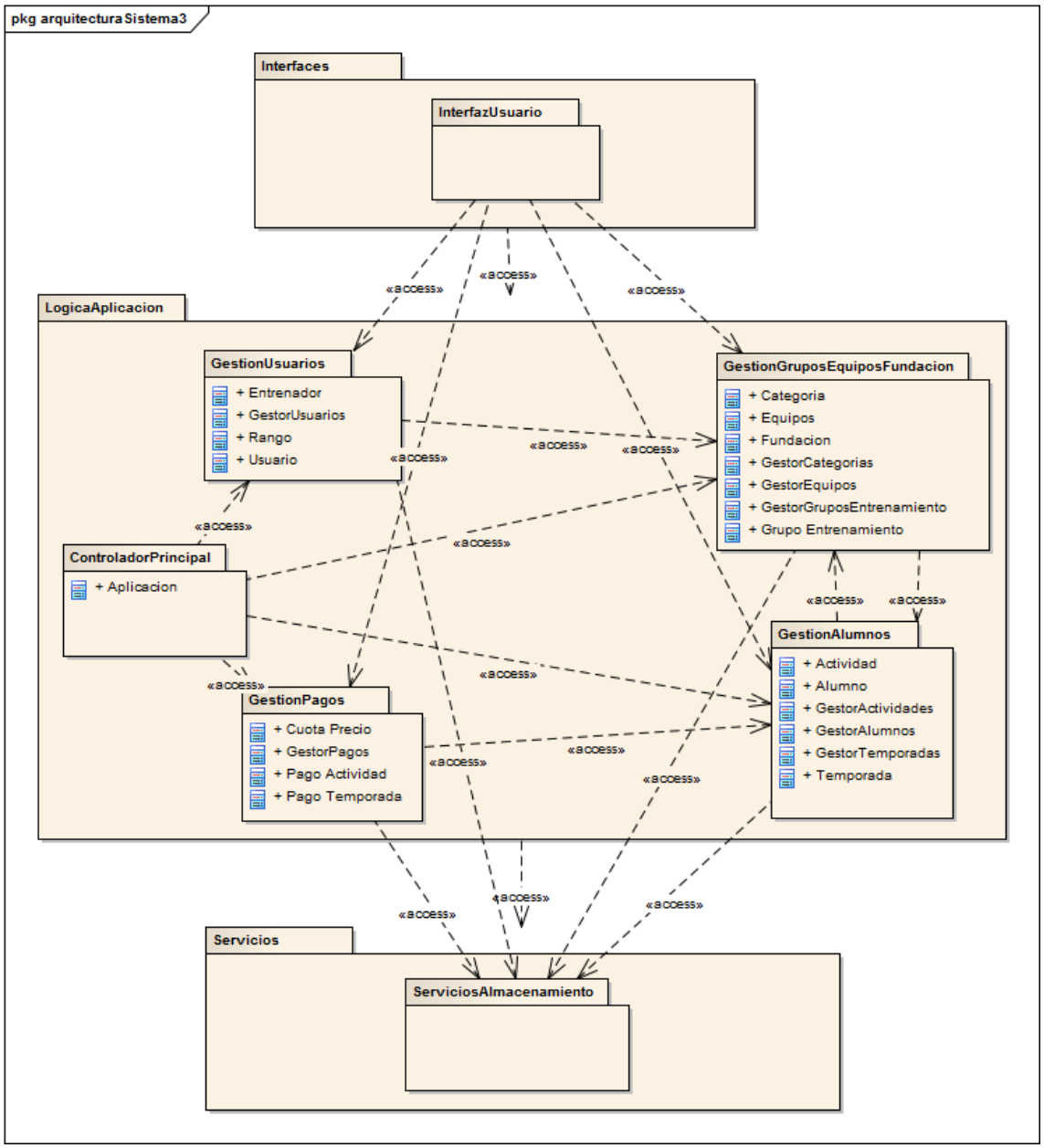
Una descomposición como la expuesta anteriormente nos lleva al diagrama de paquetes siguiente.

Diagrama de paquetes de la lógica de aplicación.



A partir de la estructuración refinada de los subsistemas de la capa de lógica de aplicación, se ha diseñado una nueva arquitectura software del sistema.

Diagrama de nueva arquitectura software del sistema.



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.

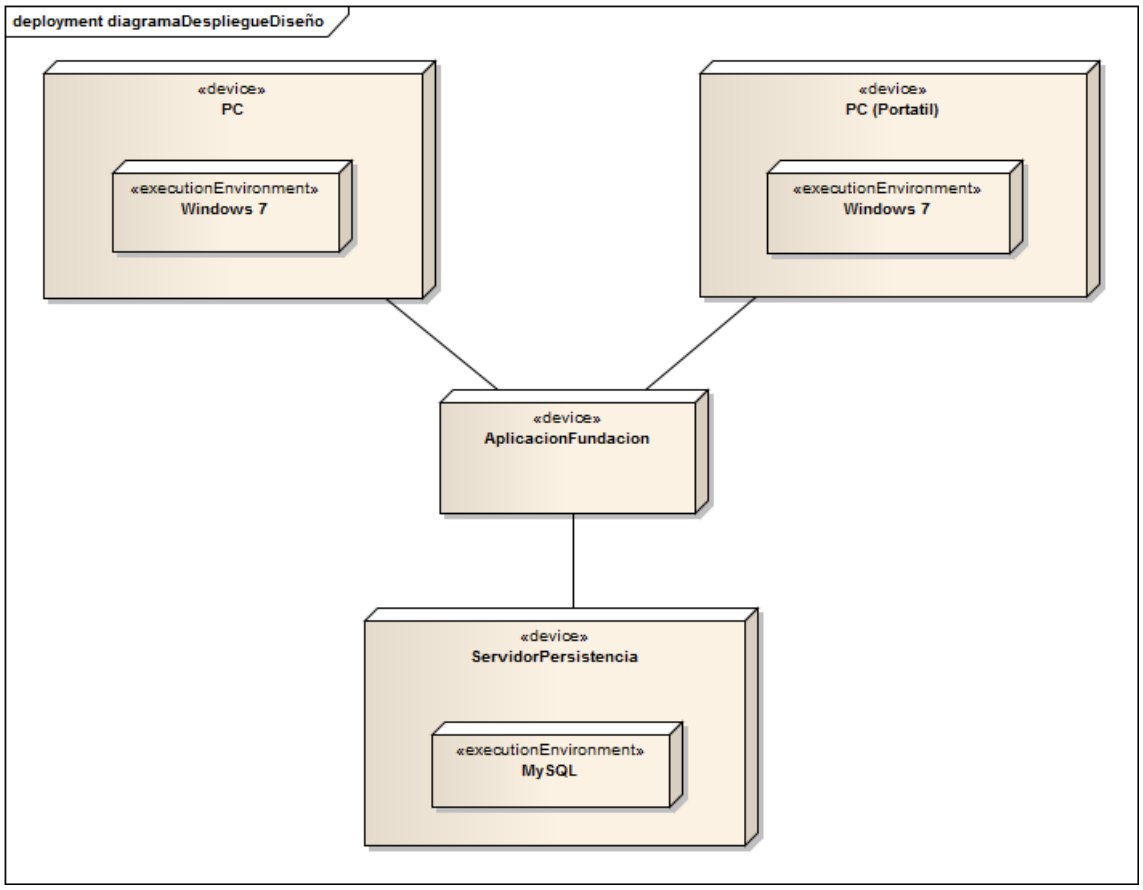
Se tienen los siguientes nodos, los cuales pueden ser dispositivos físicos o entornos de ejecución:

- 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 <<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.

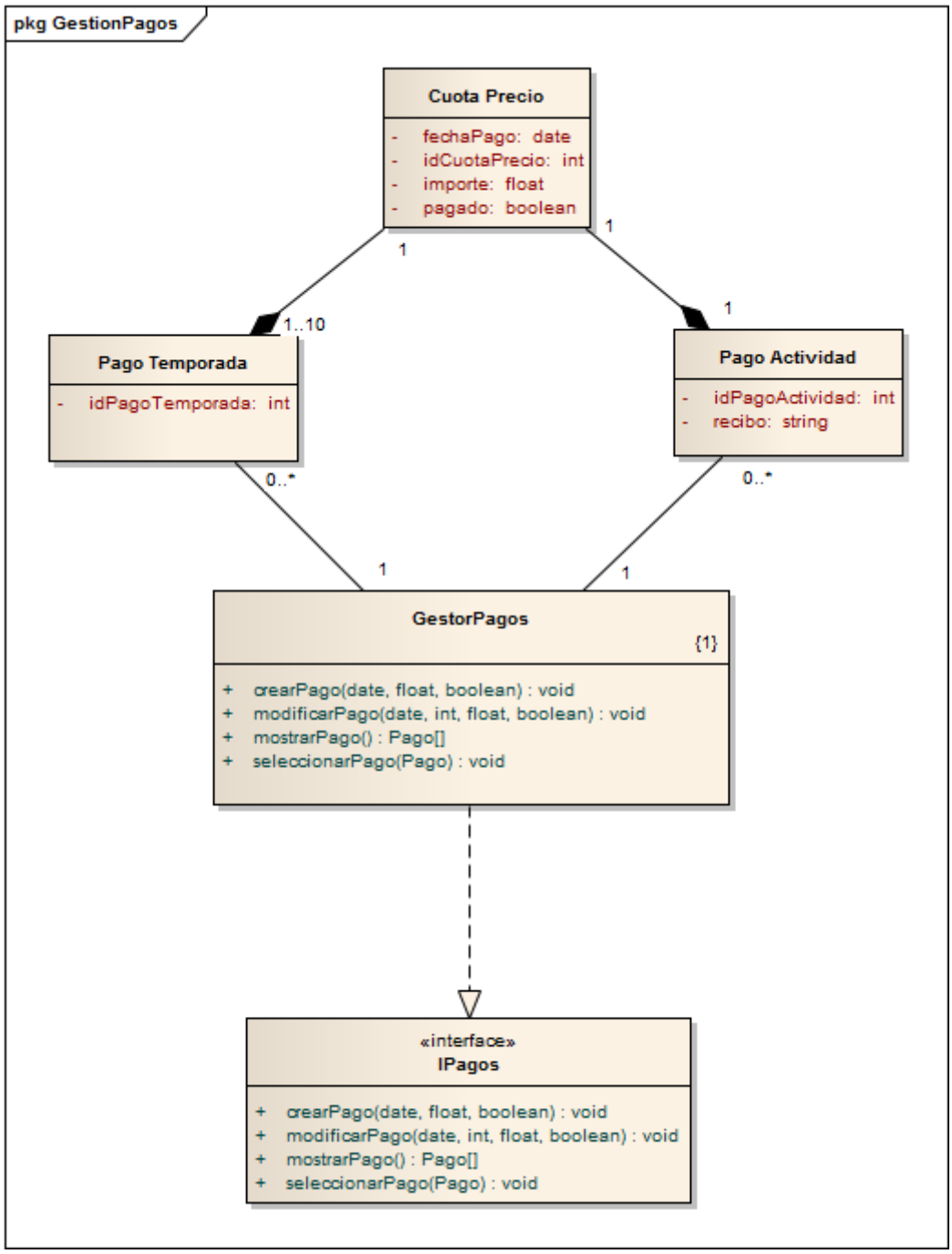
Las interfaces facilitan la sustitución y reutilización de componentes:

- 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 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.

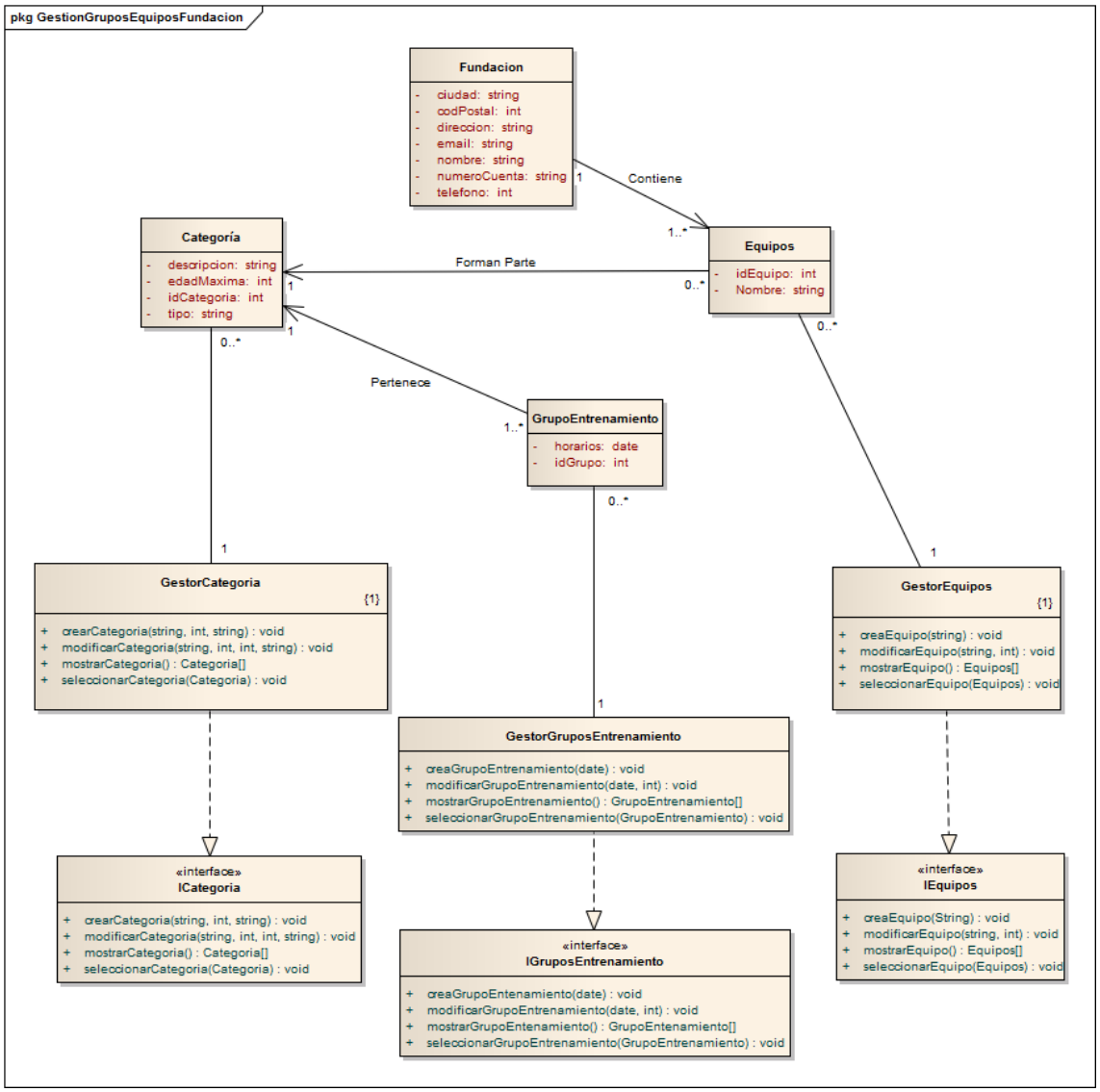
Un puerto de componente agrupa un conjunto semánticamente cohesivo de interfaces proporcionadas y requeridas.

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

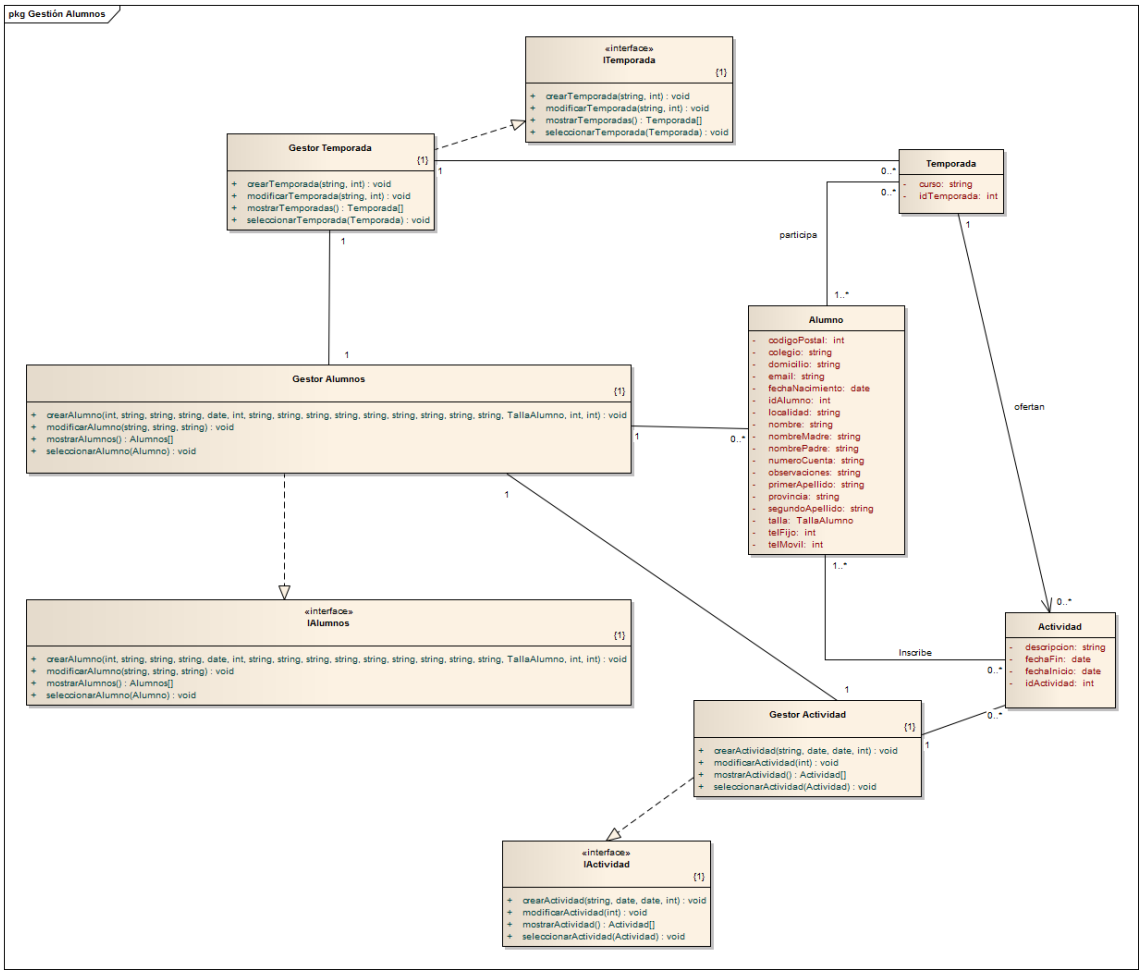
Subsistema Gestión de Pagos.



Subsistema Gestión de Grupos Equipos Fundación.



Subsistema Gestión de alumnos.



Subsistema Gestión de usuarios.

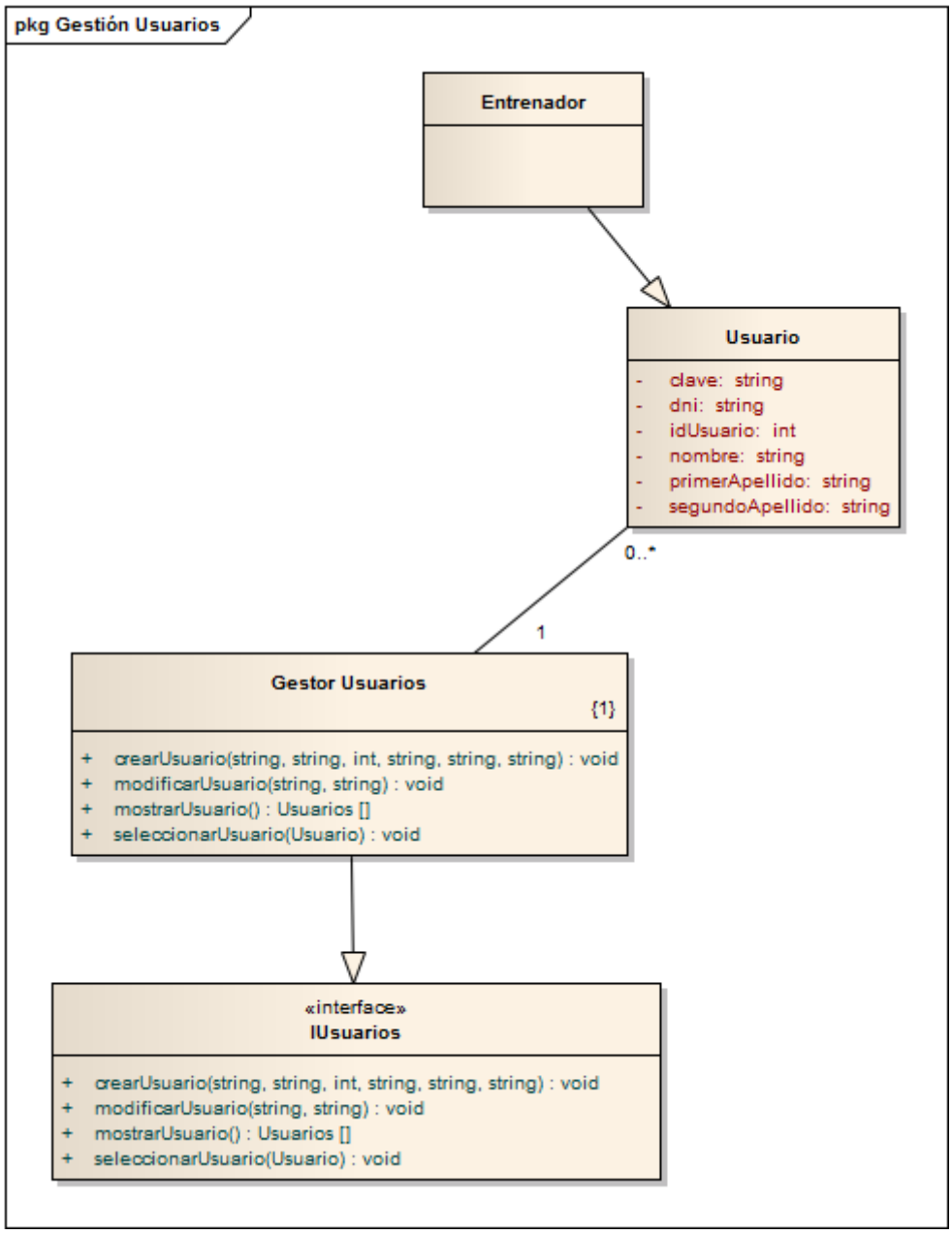
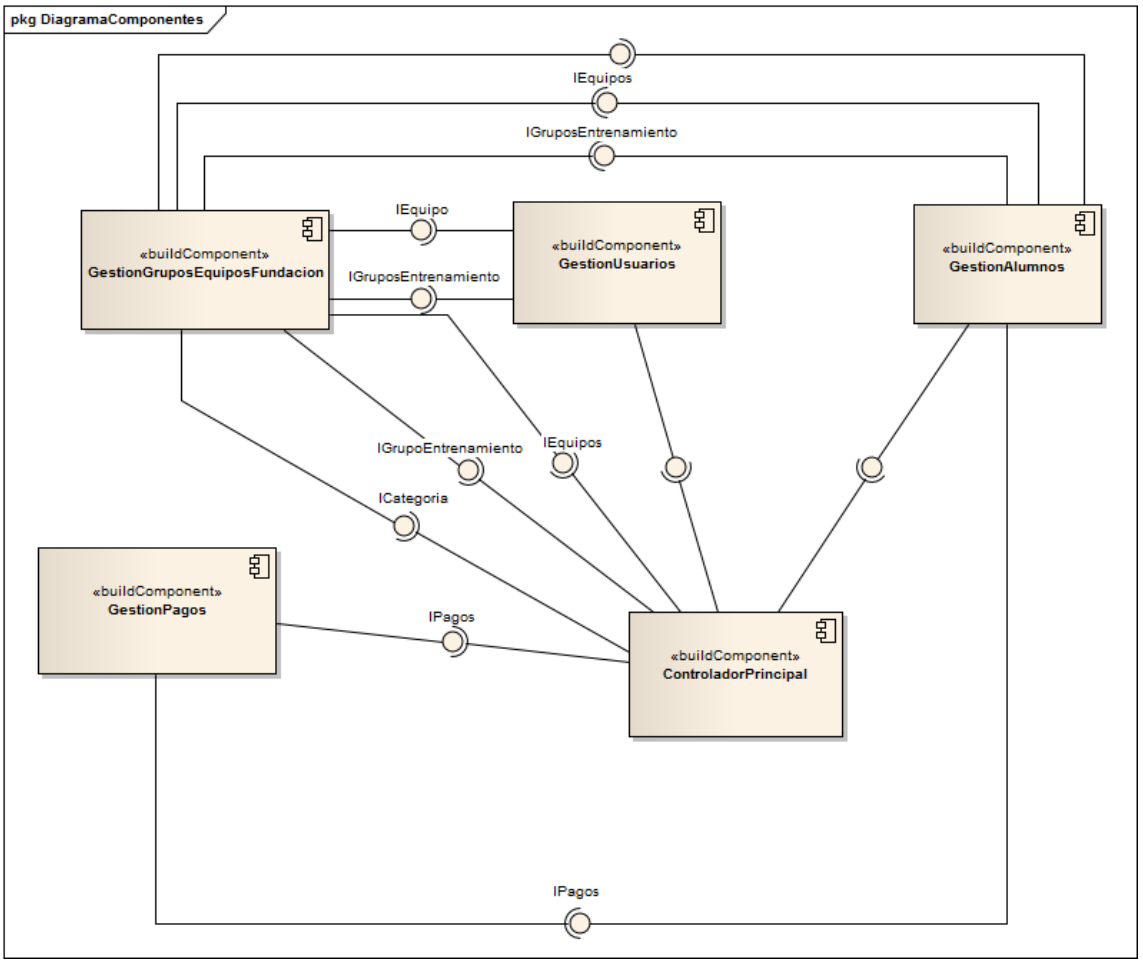
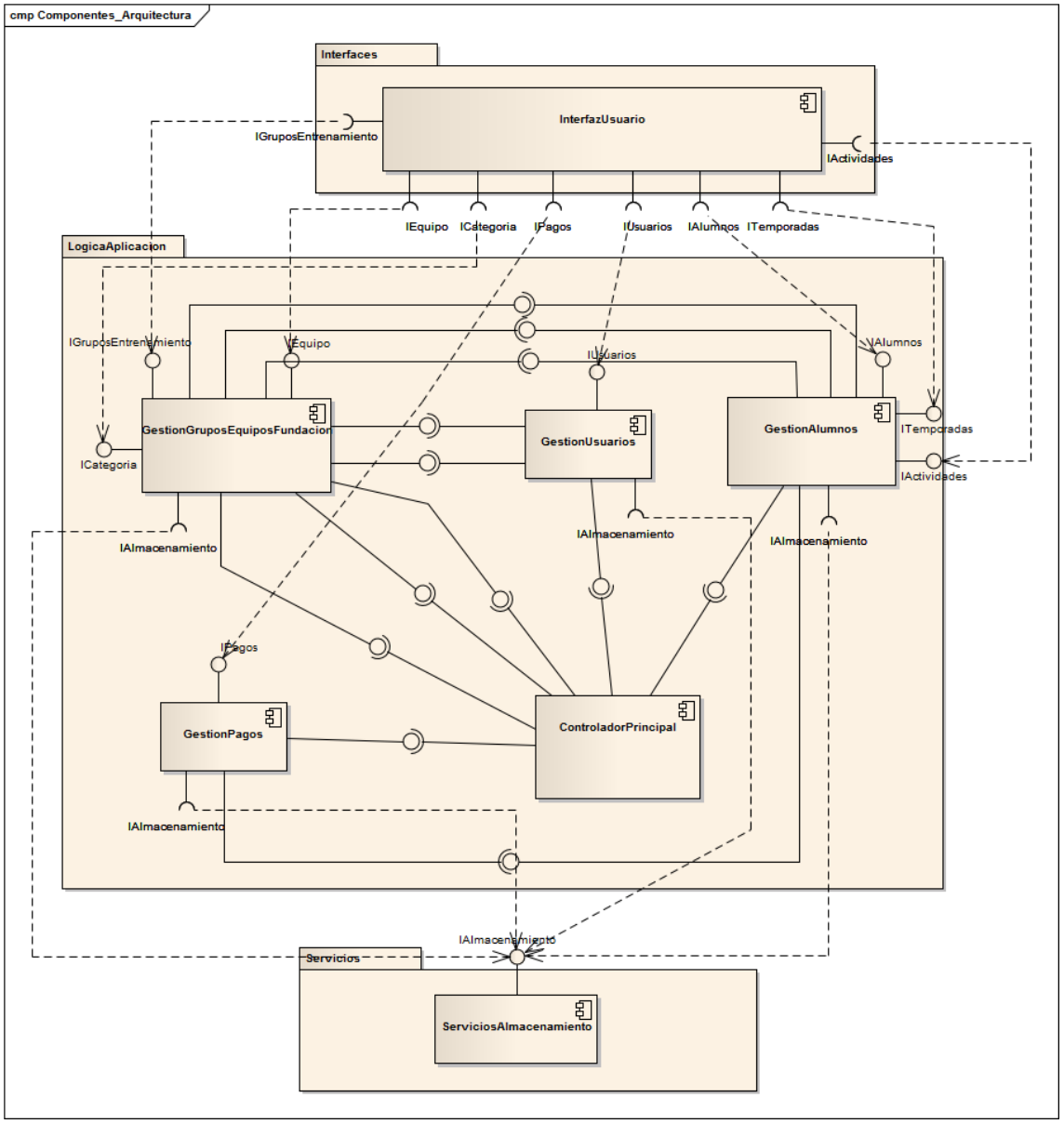


Diagrama de componentes.



Componentes arquitectura.



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.

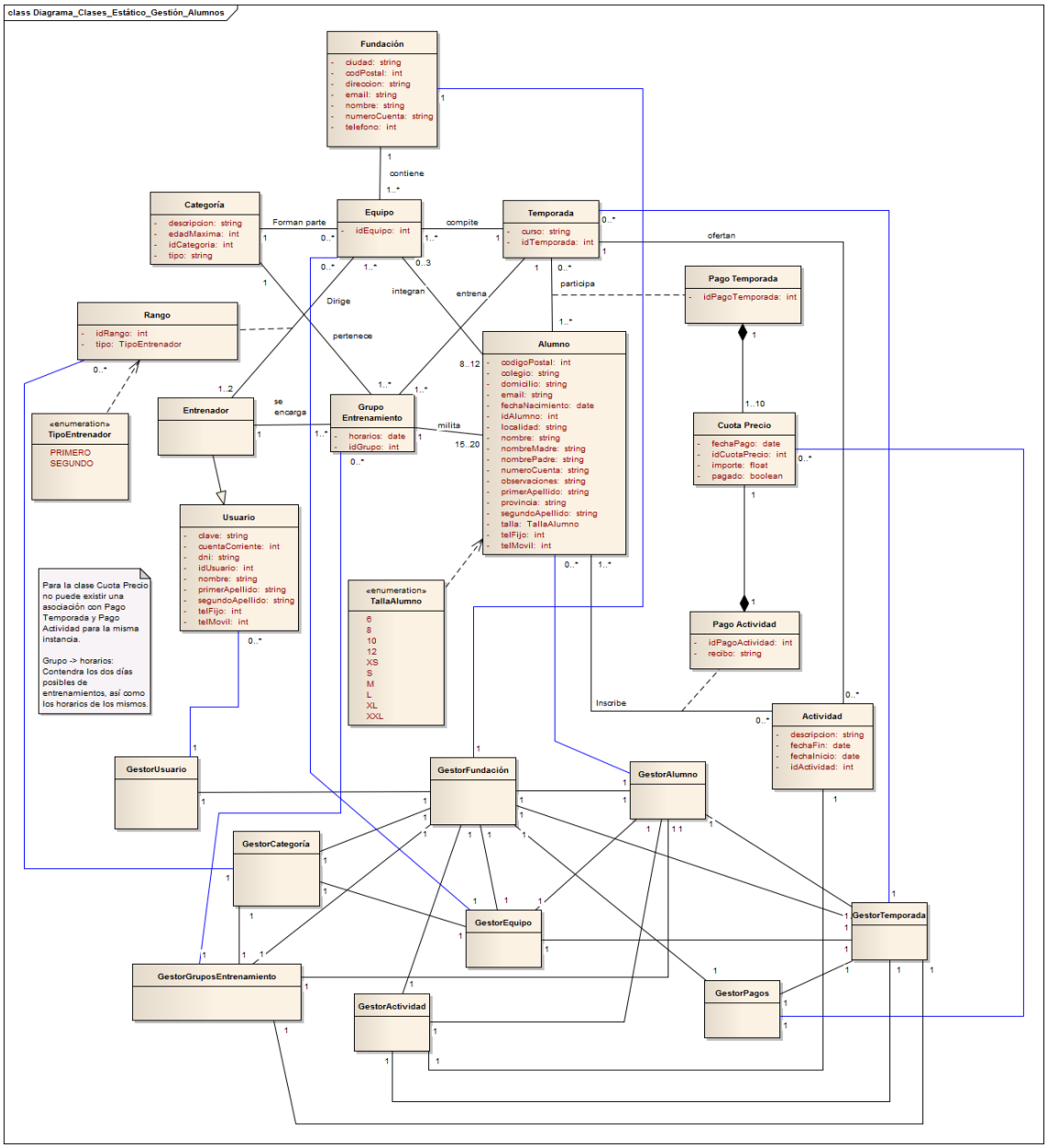
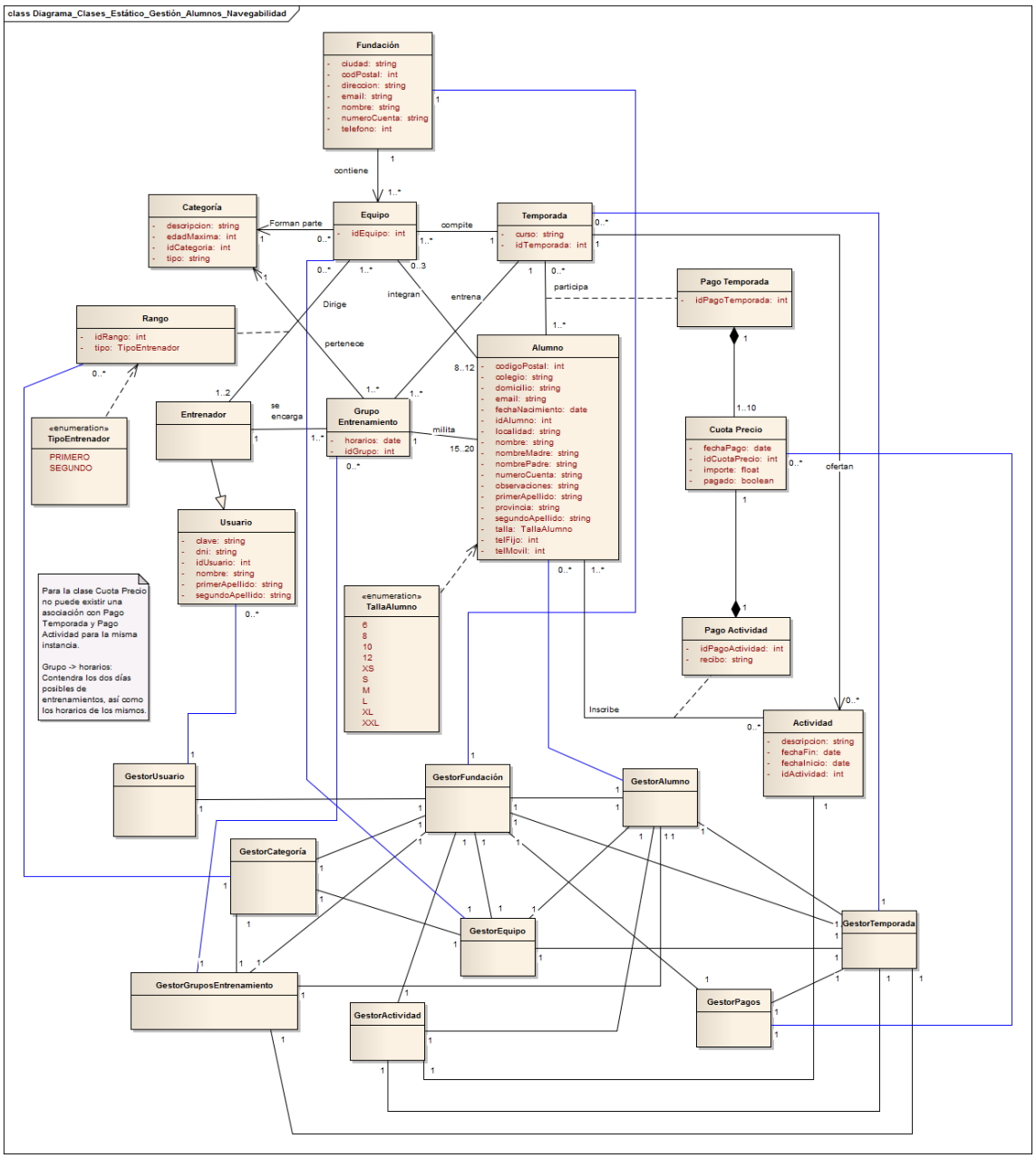


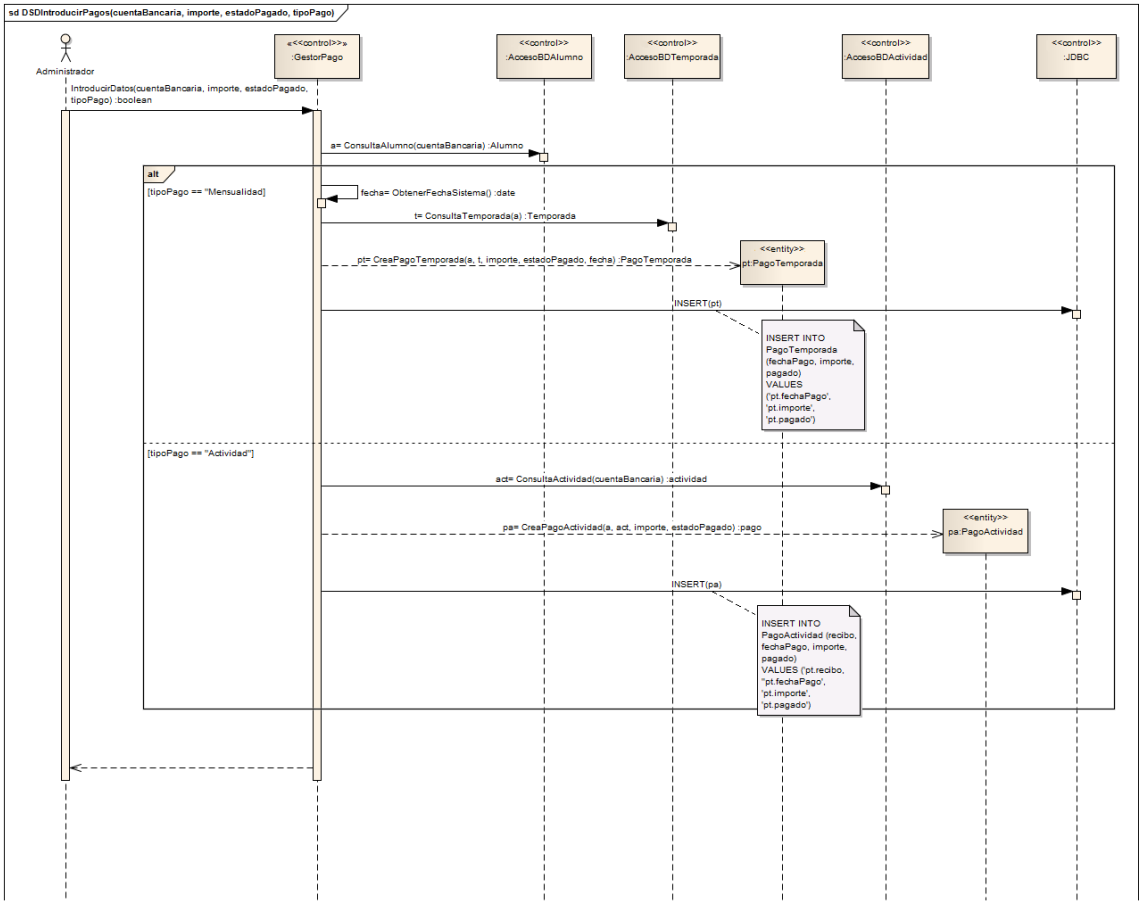
Diagrama de clases estático con navegabilidad.



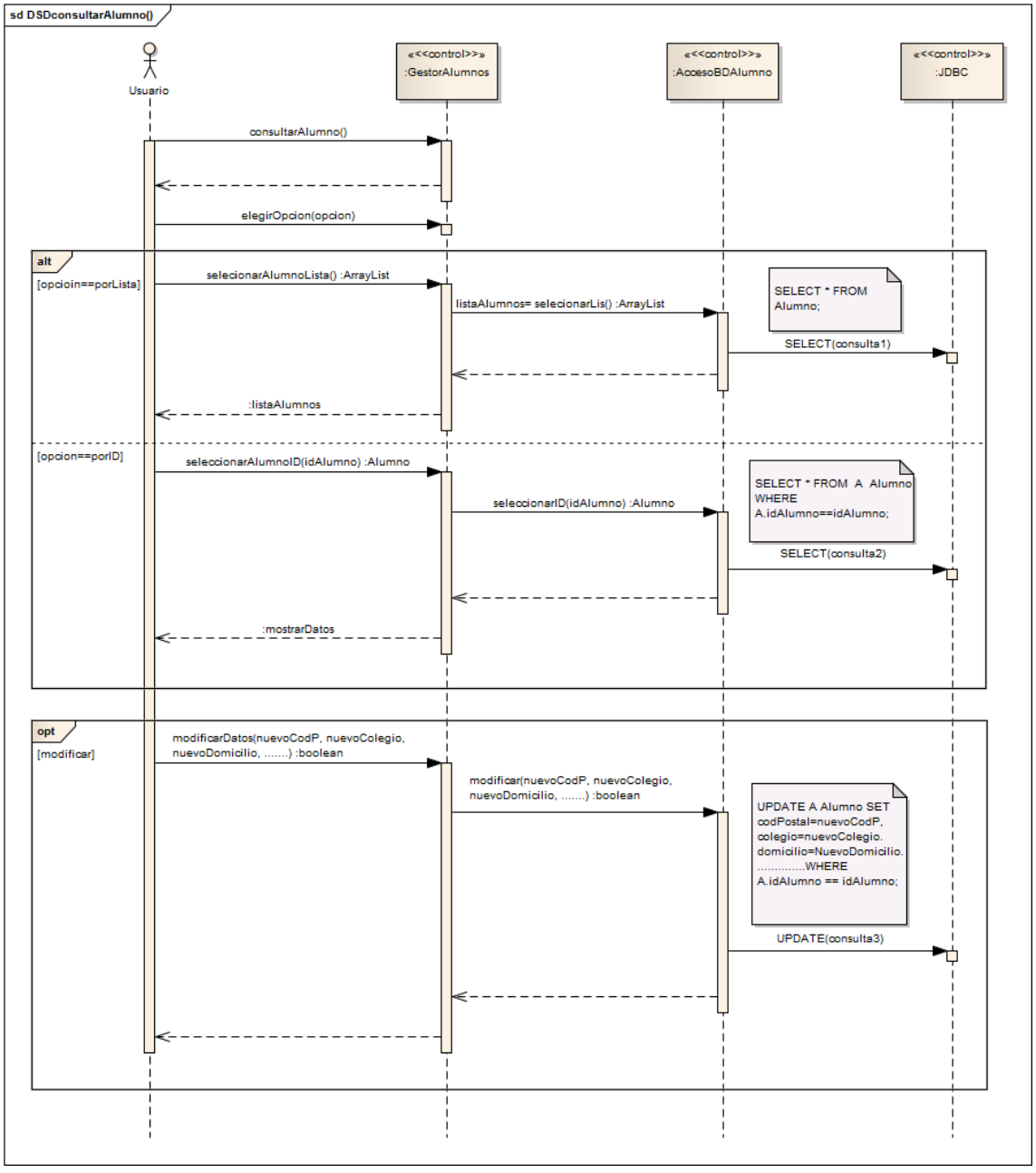
Diagramas de Secuencia del Diseño.

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.

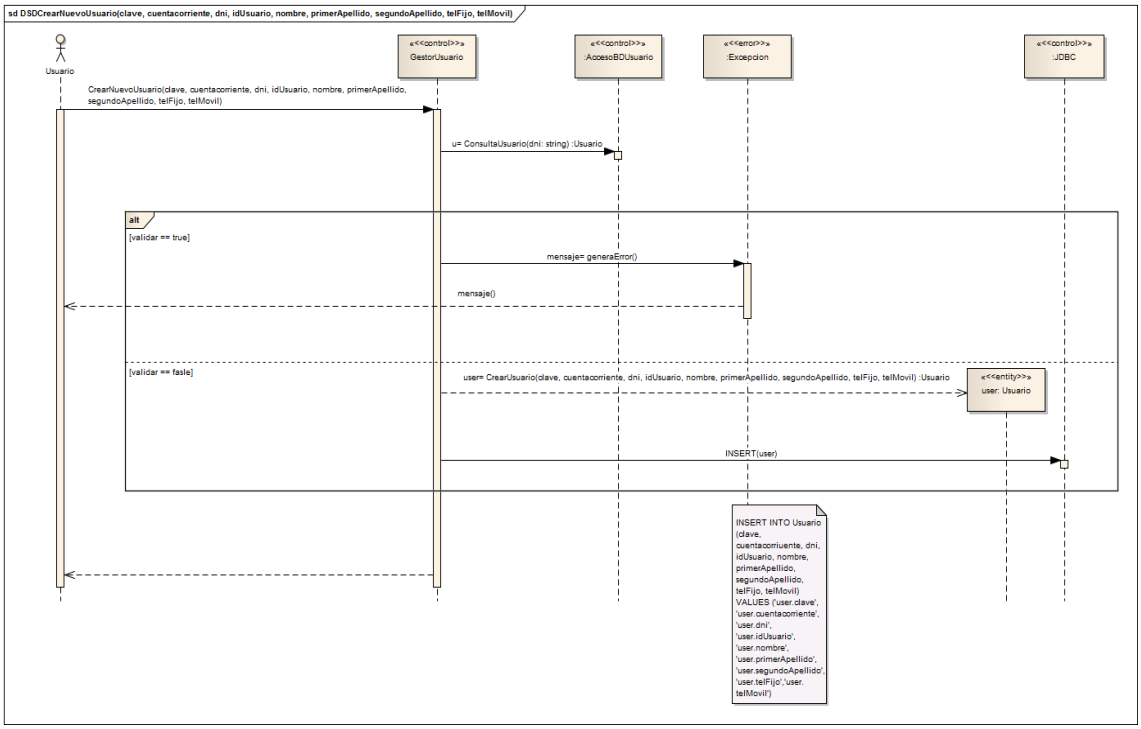
DSD Introducir Pagos.



DSD Consultar Alumno.



DSD Crear Nuevo Usuario.



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.