

# **Ingeniería del Software**

## **Práctica 4**

### **Diseño Arquitectónico del Sistema**

Jorge Castellano Castellano  
Gabriel Fernández Díaz  
David Morales Sáez

# Índice

- 1. Arquitectura**
- 2. Diagrama de paquetes**
- 3. Diagrama de despliegue**
- 4. Asignación de paquetes a nodos**
- 5. Persistencia de objetos**
- 6. Tecnología usada**

# 1. Arquitectura

El diseño seleccionado para nuestra situación es el de tipo Monolítico. Si bien a nivel general es un sistema que suele descartarse para muchas situaciones y diseños por su inflexibilidad y carácter cerrado, en nuestro caso particular aporta notables ventajas que hacen de él una opción muy recomendable.

Dentro de nuestros objetivos en las aplicaciones multimedia que se desarrollarán, destaca en gran medida el tiempo de respuesta y la fluidez óptima del sistema de cara al usuario. En esta línea aparece la necesidad de aportar a nuestro sistema de una estabilidad y un rendimiento contrastados.

La arquitectura monolítica al ser tan concreta para nuestras necesidades específicas, nos aporta ese rendimiento final requerido.

Cabe destacar que este estilo de trabajo se sigue en la industria actual con ejemplos recientes de grandes producciones como Mass Effect 3 de Bioware, donde se ha empleado el diseño arquitectónico monolítico para garantizar el rendimiento y prestaciones de alta prioridad de un título de primera línea y alcance mundial.

## 2. Diagrama de paquetes

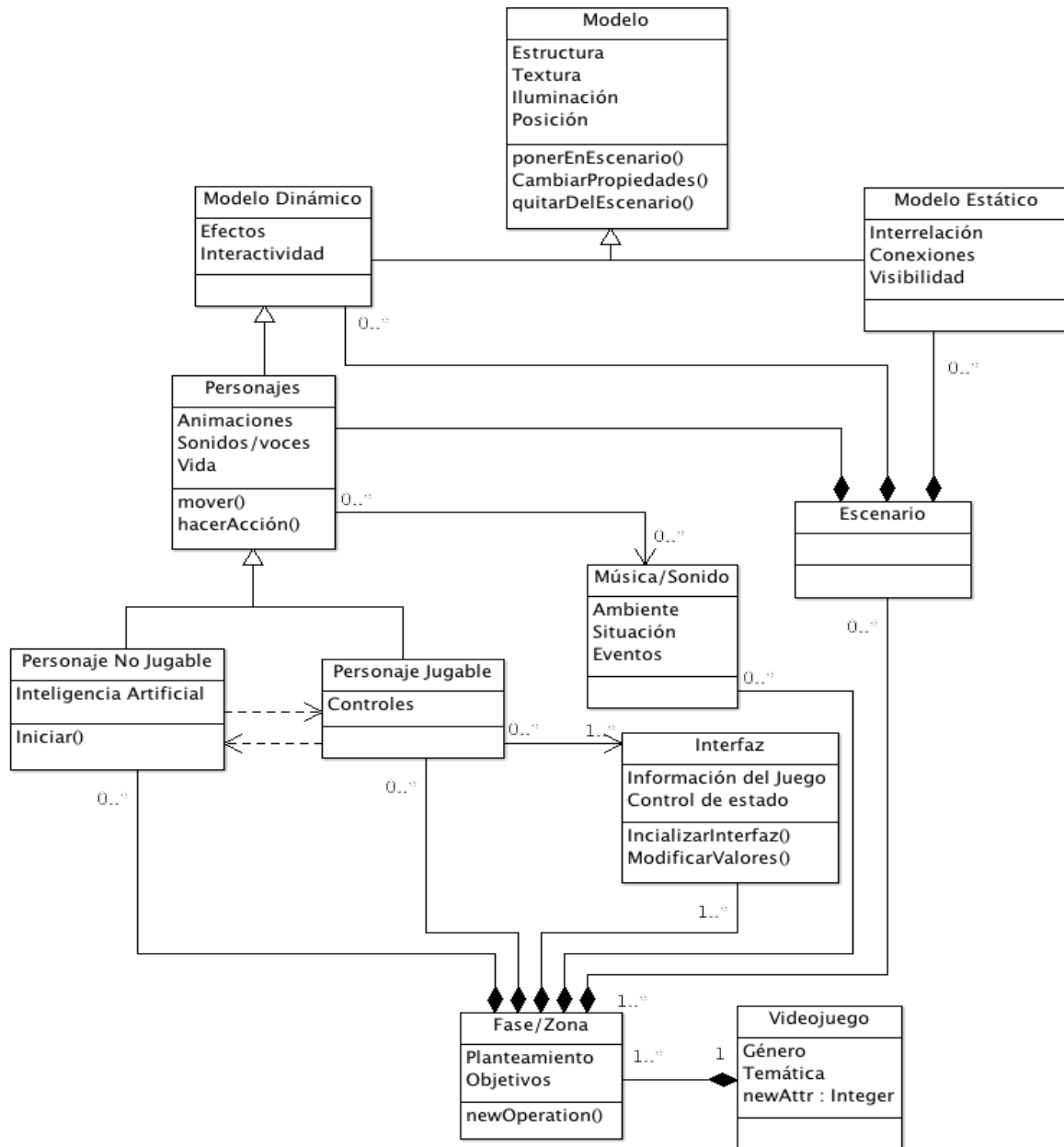
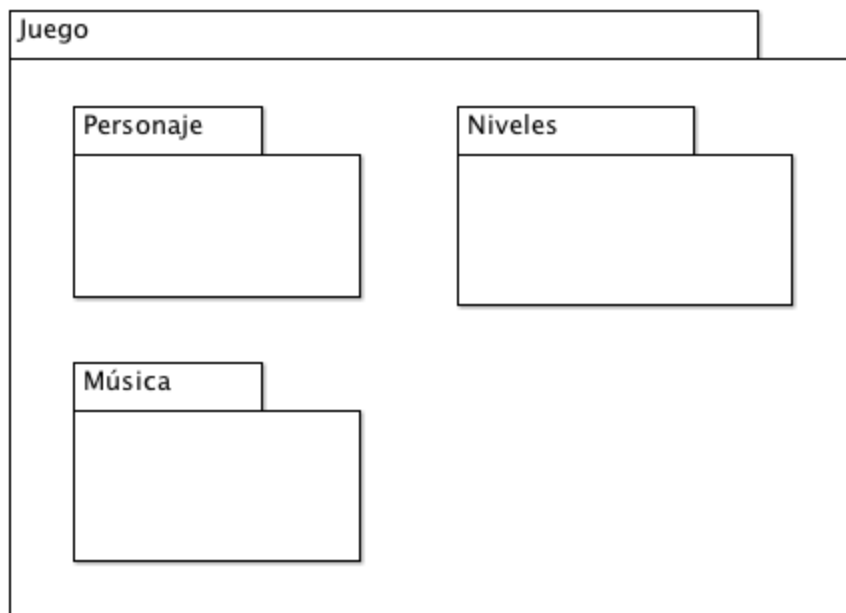


Diagrama de Clases Actualizado

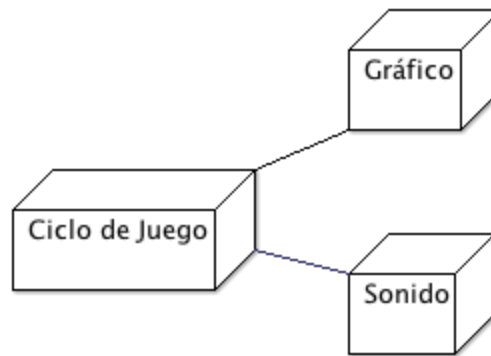


### *Diagrama de Paquetes*

Tras revisar el Diagrama de Clases se ha generado el Diagrama de Paquetes expuesto. El agrupamiento realizado se ha debido a la naturaleza de los elementos presentes en el sistema. De este modo los personajes, por su eminente carácter interactivo, los niveles por componer el entorno de los mismos y la música, por su carácter heterogéneo, han sido los paquetes resultantes, junto al Juego, que los engloba a todos y da cohesión al conjunto.

Como se puede presuponer, en el paquete de música se incluirá la clase Música y Sonido. Por otro lado, en el paquete Personaje, encontraremos las clases Personaje, Personaje Jugable, Personaje No Jugable, Modelo, Modelo Dinámico y Modelo Estático. Por último, en el paquete Niveles, encontraremos las clases asociados a los niveles, es decir, Escenario, Interfaz y Fase/Zona. En el paquete Juego integraremos estos tres subpaquetes con la clase Videojuego.

### 3. Diagrama de despliegue



*Diagrama de Despliegue*

Los nodos principales son:

- Ciclo de Juego: donde se controlarían la entrada/salida de información, la Inteligencia Artificial y más elementos como la física de objetos.
- Gráfico: donde se maneja el entorno visual y el aspecto general del sistema.
- Sonido: para coordinar la música de los escenarios y los sonidos de los distintos elementos de interacción dentro del sistema.

## 4. Asignación de paquetes a nodos

Los paquetes (filas) se desplegarán o no en los nodos (columnas) según la siguiente matriz:

<b>Paquetes /Nodos</b>	<b>Ciclo de Juego</b>	<b>Gráfico</b>	<b>Sonido</b>
<b>Juego</b>	Sí	No	No
<b>Personaje</b>	Sí	Sí	No
<b>Niveles</b>	Sí	No	No
<b>Música</b>	Sí	No	Sí

## 5. Persistencia de objetos

A la hora de afrontar la gestión de los objetos a emplear por nuestro sistema, destacaremos dos casos concretos que definirán los elementos sobre los que se considera una persistencia efectiva entre sesiones, los ficheros de partida guardada y configuración.

En el Fichero de Partida Guardada se almacenarán todos aquellos cambios referentes a la partida y a los elementos que en ella intervienen. Los personajes, niveles, escenario, música y todas las interacciones y variantes dadas durante el juego que les afecten quedaran registradas en este fichero para su posterior recuperación en partidas siguientes del usuario. Por su amplitud y alcance será el fichero más importante a efectos de persistencia.

Para el Fichero de Configuración dispondremos de una serie de valores predefinidos entre los que seleccionar, de modo que solo se almacenarían los cambios efectuados por selección de entre las opciones predefinidas disponibles. Aunque su envergadura sea menor que en el caso de el Fichero de Partida Guardada tendrá una importancia relevante a la hora de adaptar el juego a las necesidades particulares de cada usuario.

Del mismo modo destacar que estos ficheros serán independientes pero compatibles, es decir, debe permitirse la utilización de una partida guardada concreta sin importar las variaciones presentes en el fichero de configuración, por lo que se debe garantizar la independencia y compatibilidad en ejecución de ambos ficheros.



## **6. Tecnología usada**

Para la realización de nuestro proyecto serán necesarios los siguientes recursos tecnológicos:

Lenguaje de programación: Java, muy completo con grandes prestaciones en sectores de gran interés nuestro desarrollo como son la Inteligencia Artificial y los entornos Gráficos.

Plataforma de desarrollo: Eclipse, entorno contrastado de desarrollo con garantías en funcionalidad.

Protocolo: OpenGL, como estándar libre y versátil de tratamiento de gráficos, lo que nos posibilita ventajas en el desarrollo como recurso y en la implementación por su extensión y alcance.

Software de terceros: Windows/Linux/Mac Os, como Sistemas Operativos como plataformas de desarrollo y ejecución final.