**UNIVERSIDAD DEL QUINDÍO  
FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**INGENIERIA DE SOTWARE III**

**Paulo Cesar Alvis Ramos, Diego Alejandro Sánchez Parra, Leidy Andrea Raigoza Palacio**

# patrones de diseño

Los patrones de diseño (Design Patterns) son soluciones a los problemas de diseño de software que se encuentran reiteradas veces en el desarrollo de aplicaciones. Estos patrones abarcan los diseños reusables y las interacciones de los objetos.

Como dice Erich Gamma: “Un patrón es una manera de capturar diseños recurrentes, de tal manera que otros la puedan adquirir y utilizar fácilmente este conocimiento y experiencia”.

En 1994, Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides publicaron un libro titulado: “Design Patterns - Elements of Reusable Object-Oriented Software”, con el cual introducen este concepto. Estos autores son conocidos como La Banda de los Cuatro (GoF Gang of Four) y nos muestran los 23 patrones generales, que son considerados como la base para todos los patrones existentes y que se clasifican en tres grupos: creacional, estructural y del comportamiento.

A continuación, se expondrán tres de los patrones de diseño aplicables a nuestro proyecto, teniendo en cuenta que en su mayoría se implementa bajo el ambiente proporcionado por el IDE Unity.

## Patrón de comportamiento: estrategia

Este patrón de comportamiento logra hacer intercambiables un conjunto de algoritmos. Permite cambiar el algoritmo sin importar el cliente que lo implemente.

Para el producto en particular es ventajoso este patrón ya que, de manera general, presenta situaciones en los que una misma operación o conjunto de operaciones pueden realizarse de formas distintas. Como por ejemplo cuando el usuario interactúa con los distintos solidos pausando la animación interna, ensanchando tamaños y cambiando de escenario por medio de punto de control.

De esta manera definimos en nuestro proyecto el patrón estrategia en todas las interacciones con el usuario, así:

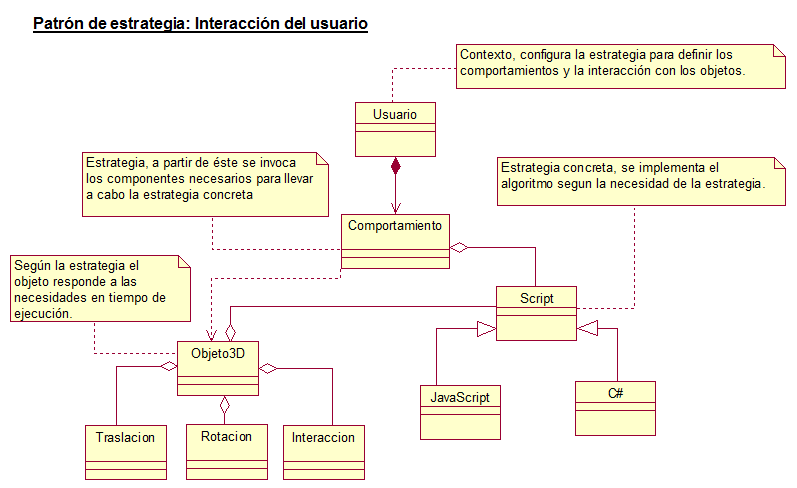


Figura : Modelo conceptual de patrón estrategia.

## Patrón de comportamiento: estado

El objetivo fundamental del patrón de comportamiento estado (Pattern State) es permitir que un objeto cambie su comportamiento en función de su estado interno. De esta forma da la sensación de que el objeto cambia de clase.

Para nuestro proyecto se utiliza este patrón sobre el controlador de primera persona (First Person Controller) que es el objeto que contiene el personaje que navega por la aplicación e interactúa con él.

La ventaja principal del uso del patrón estado en el producto, radica en que define y encapsula los estados como un conjunto de características el cual permitirá que el objeto tenga un comportamiento concreto.

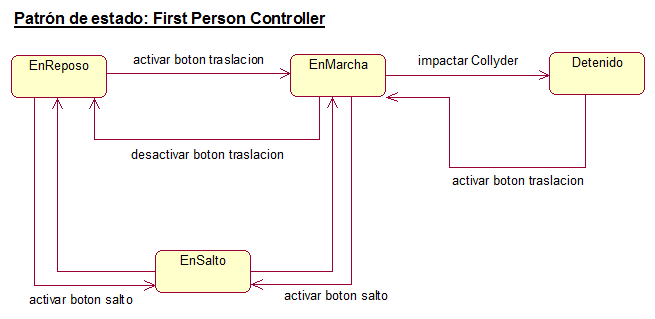


Figura : Modelo conceptual de patrón estado.

## Patrón de comportamiento: cadena de responsabilidad

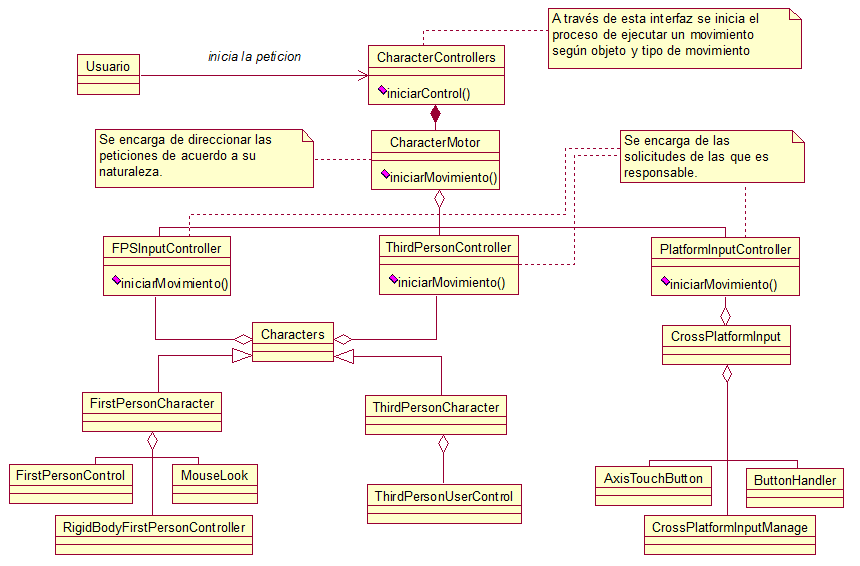
El patrón cadena de responsabilidad consiste en crear una cadena de objetos receptores para una solicitud. Este patrón desacopla el remitente y el destinatario de una solicitud basada en el tipo de petición. En este modelo, normalmente cada receptor contiene una referencia a otro receptor. Si un objeto no puede manejar la solicitud la pasa al siguiente receptor y así sucesivamente.

Figura : Modelo conceptual de patrón cadena de responsabilidad.

# Patrones de movimiento en unity

Para el desarrollo del “producto mínimo viable basado en realidad virtual como herramienta de apoyo en la enseñanza de cálculo multivariado y vectorial para estudiantes de Ingeniería de Sistemas y Computación de la Universidad del Quindío”, se tiene en cuenta cierta secuencia o patrón de movimiento para construir objetos en 3d con la finalidad de que sean dinámicas y prácticas. Cada uno de estos movimientos programados en el lenguaje de javascript y/o C#.

Entre los patrones de movimiento se encuentra:

* El movimiento del personaje principal (First Person Controller) que se mueve en una plataforma a través de una serie de coordenadas de las distintas escenas diseñadas.
* Objetos 3D que se mueven en las diferentes escenas por medio de eventos.
* Cortes de los objetos 3D por un plano.

Para la implementación de cada uno de los patrones de movimiento es necesario crear scripts de comportamiento, los cuales serán los encargados de implementar las acciones a realizar por cada objeto, los cuales como práctica de reconocimiento tendrán la palabra “Patron\_movimiento”. (AEV)

# patrones de diseño de interacción

Los patrones de diseño de interacción tienen el objetivo de resolver problemas de usabilidad y de experiencia de usuario, ya que están basados en el “Diseño centrado en el usuario”. Por lo tanto, los conceptos aplicados en el producto mínimo viable basado en realidad virtual como herramienta de apoyo en la enseñanza de cálculo multivariado y vectorial para estudiantes de Ingeniería de Sistemas y Computación de la Universidad del Quindío”.

## Simplicidad

El producto estará centrado en la simplicidad visual, con elementos y / objetos 3D mínimos, y la función de cada uno de los elementos estará bien definida lo cual permitirá que el usuario intuya ciertos comportamientos a seguir, como avanzar, observar, señalar. Cada uno de los objetos 3D se clasificara según la temática de los mismos, de tal forma que cada escena tendrá una familia de objetos 3D.

## Consistencia.

Teniendo en cuenta que para la representación gráfica de los objetos 3D es necesario varias interfaces gráficas, cada una de estas interfaces serán consistentes entre si, es decir se utilizará la misma gama de colores, los mismos iconos para representar acciones, de tal forma que se pueda llevar un comportamiento especifico por parte del usuario.

## Navegación intuitiva

La navegación intuitiva es fundamental en el producto ya que la navegación entre contenidos debe ser fácilmente comprendida por cada uno de los usuarios, por lo tanto se espera que el usuario reconozca a primera vista cada uno de los paneles, botones y demás componentes para ir de una escena a otra, y así mismo para interactuar con cada uno de los objetos 3D. Toda esta gama de componentes permitirá al usuario “intuir” los contenidos de objetos 3D de la escena siguiente, ya que una navegación intuitiva permite lograr un uso fluido.

## Patrón de composición

En el producto se implementara el patrón de composición el cual permitirá maximizar la reutilización de todos los componentes propuestos, es decir aplicados a cada uno de los objetos 3D, lo cual al utilizar los componentes indivisibles y complejos en diferentes contextos concederá que a partir de un objeto 3D en donde es posible observar los diferentes conceptos matemáticos se le aplique una animación que permita obtener dos o más objetos 3D y cada uno de manera independiente genere más conceptos matemáticos en tiempo real, lo cual tendrá un tiempo de ejecución activado por un evento. Es importante aclarar que el patrón de composición para generar nuevos objetos estar dato por el mismo concepto matemático ya establecido, es decir cada objeto 3D generara nuevos objetos 3D propios de la familia que pertenece, es decir a partir de una esfera puedo obtener un elipsoide. (Peredo Valderrama, Peredo Valderrama, &amp; Anaya Rivera, 2014)

# referencias

AEV, A. d. (n.d.). Unity3D: Sistema de patrones de movimiento. Retrieved Octubre 5 , 2016, from Unity3D: Sistema de patrones de movimiento: http://aev.org.es/unity3d-sistema-de-patrones-de-movimiento/

https://docs.unity3d.com/es/current/Manual/InstantiatingPrefabs.html

https://www.google.com.co/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=patrones+de+dise%C3%B1o+de+interaccion%2Bunity

AEV, A. d. (s.f.). Unity3D: Sistema de patrones de movimiento. Recuperado el 5 de octubre de 2016, de Unity3D: Sistema de patrones de movimiento: http://aev.org.es/unity3d-sistema-de-patrones- de-movimiento/

Peredo Valderrama, I., Peredo Valderrama, R., & amp; Anaya Rivera, K. (2014). Interacción de modelos 3d con realidad aumentada. SISTEMAS, CIBERNÉTICA E INFORMÁTICA, 61-66.

https://www.facebook.com/groups/InterActionforo/