

# Diagrama de Clases



English

## Entradas

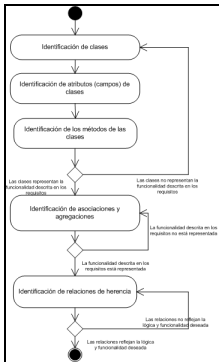
- Diagrama de Casos de Uso (empleando UML)
  - ◆ ¿Se puede emplear gestión de la configuración?: **Si**
- Especificación de Requisitos
  - ◆ ¿Se puede emplear gestión de la configuración?: **Si**

## Salidas

- Diagrama de de Clases (empleando UML)
  - ◆ ¿Se puede emplear gestión de la configuración?: **Si**

## Solución

## Proceso



## Tiempo de Desarrollo

- ◆ Para adquirir el conocimiento necesario para desarrollar el producto software:
- ◆ Para crear el Patrón de Producto:
- ◆ Para aplicar el Patrón de Producto:

## Video Explicación

- ◆ No aplica



## Patrones Relacionados

- Diagrama de Casos de Uso
- Diagrama de Colaboracion
- Diagrama de Secuencia
- Especificacion de Requisitos

## Controladores de Calidad

- Ninguno

## Plantillas

-  Formato Diagrama de Clases
-  Plantilla Diagrama de Clases

## Ejemplos

-  Ejemplo de un diagrama de clases



## Herramientas de Soporte

- Argo UML (Open Source Project)
- Dia (GNOME)
- Rational Software Modeler (IBM)
- StarUML (Open Source Project)
- Visual Paradigm for UML (Visual Paradigm)



## Contexto Inicial

Se cuenta con la descripción del sistema desde la perspectiva del usuario (por ejemplo en un diagrama de casos de uso) y se desea crear un diagrama que describa la estructura de un sistema software, mostrando los componentes o piezas de software que se encargarán del funcionamiento del sistema.



## Contexto Resultante

Se obtiene un diagrama que representa mediante clases los componente lógicos que se encargarán del funcionamiento del sistema.



## Problema

Se necesita crear una representación de los componentes (clases y objetos) que implementarán el funcionamiento del sistema. Dicha representación debe mostrar lo que podrá hacer el sistema así como la forma en la que será construido.



## Restricciones (*Forces*)

- **Tipo de Organización:** PyMEs, Grandes Empresas.
- **Tipo de Sistema:** Aplica a todos los tipos de sistemas.
- **Paradigma de Programación:** Orientado a Objetos.



## Roles

- Analista
- Jefe de Proyecto



## Lecciones Aprendidas

- Beneficios de utilizar este patrón
  - ◆ La visión estática del sistema software que se obtiene al utilizar este patrón está basada en UML, lo que facilita la descripción de los objetos o entidades del mundo real que interactúan con el sistema.



## Nivel de Madurez

- No aplica



## Conocimientos y Habilidades Básicos



### Conocimientos

- Conocimientos intermedios de UML.
- Interpretación de diagramas de casos de uso.



### Habilidades

- Capacidad de abstracción.
- Capacidad de Análisis.
- Visión holística o sistémica de problemas.



## Recursos de Información

- Amescua A., et al. (2003). *Análisis y Diseño Estructurado y Orientado a Objetos del Sistema Informáticos*. McGraw Hill/Interamericana de España, S.A.U.
  - Ferré Grau, X & Sanchez-Segura, M. (2004). *Desarrollo Orientado a Objetos con UML*. Recuperado el 2009-11-26 de <http://www.clikear.com/manuales/uml/index.aspx>.
  - Fowler, M. (2004). *UML distilled: a brief guide to the standard object modelling language*. Addison-Wesley.
  - Jacobson, I. & Böoch, G.(1999). *The unified software development process*. Addison-Wesley.
  - Rumbaugh, J. & Jacobson, I. (2005). *The unified modeling language reference manual*. Addison-Wesley.
  - Rumbaugh, J. (2005). *Object-Oriented Modelling and Design*. Prentice Hall.
-