

Desarrollo Web Entorno Desarrollo

Fase de Diseño de la Solución

Diseño de las posibles soluciones

Es una actividad orientada a modelar alguna de las posibles soluciones que se pueden realizar al problema, con la finalidad de resumir, previo a cualquier ejecución, las posibilidades que se tienen para abordar el problema.

Esta fase suele confundirse con el fraccionamiento del problema, ya que realmente cuando se fracciona un problema, se está acotando el alcance de dicho problema, haciéndolo cada vez más sencillo de resolver.

Hay diversas fases a la hora de diseñar las posibles soluciones y hacen que se acote mucho esos modelos de implantación.

Fase de Reutilización

En la fase de reutilización, tal y como indica su nombre, consiste en analizar si ya se tiene el problema o parte del mismo ya resuelto. Para ello, solo será necesario realizar una breve búsqueda en internet de los diversos problemas que se tienen para poder reutilizar al máximo el contenido que ya se expone en las redes.

Esta fase es una de las más importantes, debido a su simplicidad y rapidez para resolver los problemas que hay que abordar, como puede ser la estimación en tiempo, alcance y coste de la resolución.

Fase de Modelado: UMLs

Según Wikipedia: El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional, Rational Unified Process o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se representa la realidad de una utilización en un requerimiento. Mientras que programación estructurada es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. Sólo se van a explicar los tres diagramas más importantes:

Diagrama de clases

El diagrama de clases es un tipo de diagrama de estructura estática que describe la arquitectura de un sistema mostrando las clases del mismo, sus atributos, operaciones (o métodos), y las relaciones entre las clases.

Este tipo de UML proporciona mecanismos uniformes para representar los miembros de la clase, como atributos y métodos, así como información adicional sobre ellos:

Visibilidad

Para especificar la visibilidad de un miembro de la clase se ha de colocar uno de los siguientes símbolos delante de ese miembro:

- Público: +
- Privado: -

Ámbitos

Se especifican dos tipos diferentes, clasificadores, identificados por estar subrayados, e instancias.

- Clasificadores, también denominados como estáticos en muchos lenguajes. En Java son las clases. Contiene lo siguiente:
 - Los valores de los atributos son los mismos en todas las instancias.
 - La invocación de métodos no afecta al estado de las instancias.
- Instancias, es la propia instancia de la clase:
 - Los valores de los atributos pueden variar entre instancias.
 - La invocación de métodos puede afectar al estado de las instancias (los valores de los atributos pueden cambiar).

Relaciones a nivel de clase

- Enlace: Es la relación más básica entre objetos.
- Asociación: Representa a una familia de enlaces. Es la línea que une dos Clases.
- Agregación: Es una asociación más específica representada por indicarle un nombre a la asociación. Por ejemplo, Un profesor “imparte clases en” la escuela.
- Composición: La representación en UML de una relación de composición es mostrada con una figura de diamante relleno del lado de la clase contenedora, es decir, al final de la línea que conecta la clase contenido con la clase contenedor.
- Diferencia entre Agregación y Composición:
 - Relación de composición:
 - Cuando se intenta representar un todo y sus partes.
 - Un motor es parte de un coche.
 - Cuando se elimina el contenedor, el contenido también es eliminado.
 - Si eliminamos una universidad, eliminamos igualmente sus departamentos.
 - Relación de Agregación:
 - Cuando representamos las relaciones en un software o BBDD.
 - El modelo de motor MTR01 es parte del coche MC01. Como tal, el motor MTR01 puede ser parte de cualquier otro modelo de coche, es decir, si eliminamos el coche MC01, no es necesario eliminar el motor pues podemos usarlo en otro modelo.
 - Cuando el contenedor es eliminado, el contenido usualmente no es destruido.
 - Un profesor tiene estudiantes, cuando el profesor muere, los estudiantes no mueren con él/ella.

Ejemplo práctico

Problema -> Se quiere desarrollar un sistema de información para la Universidad de Oriente según la descripción siguiente.

La **Universidad** se caracteriza mediante su *nombre* y la *ciudad* donde se sitúa. En la **Universidad** están vinculados dos tipos de **Persona**: **Trabajadores**, que la Universidad emplea, y **Estudiantes**, que estudian en la **Universidad**. Cada **Persona** tiene una *CI* y un *nombre*. Los **Trabajadores** pertenecen a dos grupos: **PDI** y **PAS**. Cada **Trabajador** tiene asociada una *fecha de inicio* de su contrato. Cada miembro del **PDI** también tiene una *categoría*, mientras que cada miembro del **PAS** tiene un *puesto*. Los miembros del **PDI** pueden o no ser **Doctores**. Las actividades que desarrolla el **PDI** son *investigar* y *enseñar*, mientras que la actividad que desarrolla el **PAS** es *administrar*. La **Universidad** se compone de un conjunto de **Departamentos**, cada uno de los cuales tiene un *nombre* y un conjunto de **Trabajadores** adscrito. Un **Trabajador** no puede estar adscrito a más de un **Departamento**. Un **PDI** está adscrito obligatoriamente a un **Departamento**, mientras que un **PAS**, no. Cada **Departamento** está dirigido por un **Doctor**. Un **Estudiante** puede ser bien **Estudiante de grado**, de una determinada titulación, bien **Estudiante de Doctorado**, de un determinado programa de Doctorado. Un **Estudiante de grado** puede también colaborar con un **Departamento** como becario y puede realizar un PFC dirigido por un miembro del **PDI**. Un **Estudiante de Doctorado** realiza una tesis dirigida por un **Doctor**. Puede suponer que un **Estudiante** no puede estudiar en más de una **Universidad** y que un **Trabajador** no puede ser empleado por más de una **Universidad**.

Proporcione un modelo de esta descripción en forma de un diagrama de clases UML utilizando para nombres de clases únicamente las palabras que aparecen en negrita en la descripción anterior. Las palabras que aparecen en cursiva proporcionan pistas para la definición de los otros elementos del modelo.

No hace falta proporcionar información de tipado para las propiedades que pueda definir. Para más puntuación, añada a su modelo los elementos necesarios para tomar en cuenta lo siguiente:

Otras limitaciones:

- Una **Persona** puede ser a la vez **Trabajador** y **Estudiante**
- Un **Estudiante** no puede ser a la vez **Estudiante de grado** y **Estudiante de Doctorado**
- Los únicos tipos de **Trabajador** que existen son **PDI** y **PAS**
- Un **Trabajador** no puede ser a la vez **PDI** y **PAS**.

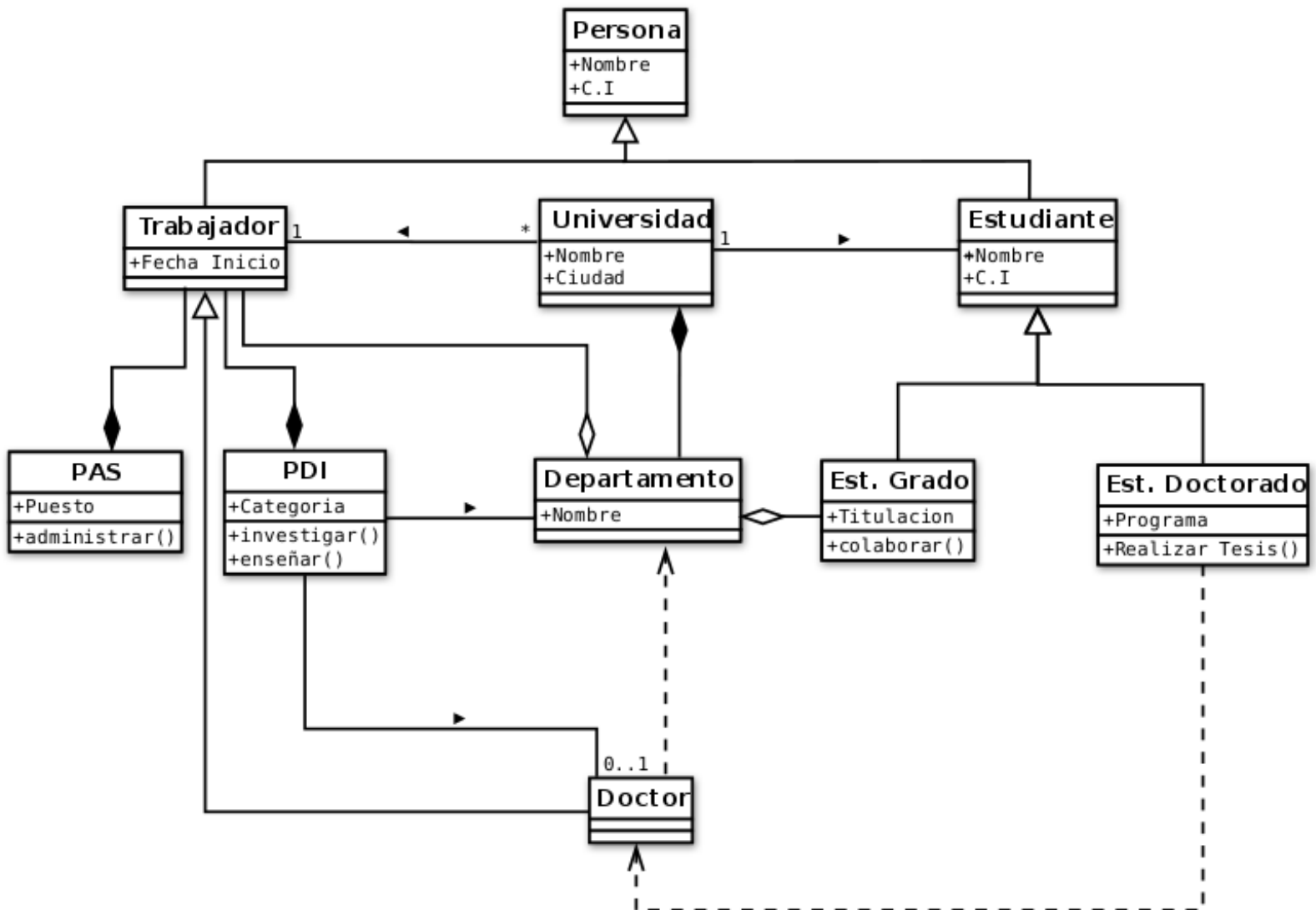


Diagrama de componentes o flujos

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos.

Debido a que los diagramas de componentes son más parecidos a los diagramas de casos de usos, estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Diagrama de despliegue

Un diagrama de despliegue modela la disposición física de los artefactos del software en nodos. Muestra la arquitectura del sistema así como el despliegue de los artefactos de software a los objetivos de despliegue.

Los elementos usados por este tipo de diagramas son nodos y asociaciones.

Los diferentes tipos de sistemas de despliegues se pueden clasificar en:

- **Sistemas empotrados:** Un sistema empotrado es una colección de hardware con una gran cantidad de software que interactúa con el mundo físico.
- **Sistemas cliente-servidor:** Los sistemas cliente-servidor son un extremo del espectro de los sistemas distribuidos y requieren tomar decisiones sobre la conectividad de red de los clientes a los servidores y sobre la distribución física de los componentes software del sistema a través de nodos.
- **Sistemas completamente distribuidos:** En el otro extremo se encuentran aquellos sistemas que son amplia o totalmente distribuidos y que normalmente incluyen varios niveles de servidores. Tales sistemas contienen a menudo varias versiones de componentes software, alguno de los cuales pueden incluso migrar de un nodo a otro. El diseño de tales sistemas requiere tomar decisiones que permitan un cambio continuo de la topología del sistema.

Fase de Planificación

Una buena planificación es fundamental para simplificar la ejecución del trabajo, reducir los tiempos, tener los materiales/herramientas necesarias a punto, realizar tareas en paralelo en los trabajos en equipo, etc...

Hace falta elaborar los siguientes documentos: Hoja de procesos, presupuesto y planning.

Hoja de Procesos

Es una tabla en la que se descompone y simplifica el trabajo en una serie de tareas simples y ordenadas, especificándolas, indicando los materiales y herramientas implicados, la duración estimada y los roles de cada operación.

Se puede realizar a mano, en un procesador de textos o en una hoja de cálculo, excel por ejemplo. Se ahorrará tiempo si se emplean plantillas ya diseñadas.

Presupuesto

Es una tabla que recoge una relación valorada de todos los costes del proyecto: los materiales que se compraron, la cantidad y precio de cada uno y coste total.

Se suele realizar en una hoja de cálculo.

Planning

El planning es una representación gráfica de la organización temporal del proyecto. Se indican las tareas (de forma más sencilla que en la hoja de procesos), se especifica la duración y las relaciones que tiene con otras tareas.

El diagrama más empleado es el diagrama de Gantt o de barras. En él cada tarea o actividad tiene asignado un tiempo que se corresponde con una barra de longitud escalada de acuerdo con su duración. Las barras de las tareas o actividades se ponen todas en una escala temporal, de proyecto (empezando con día 1, semana 1...) o ya con fechas de calendario (aunque esto

último no es recomendable). Las distintas tareas se “vinculan” unas a otras con relaciones lógicas como “empezar la tarea B cuando acabe la A”; “empezar la tarea B a la vez que la A”, “terminar al mismo tiempo las tareas A y B, etc...”

Con este diagrama conseguimos controlar a simple vista la organización total del proyecto y la situación en determinada fecha. Existen programas sencillos que nos ayudan a crear diagramas de Gantt o de barras. Un ejemplo es GanttProject, un software libre muy usado. Mediante estos programas y aplicaciones podemos asignar recursos a las tareas, e introducir datos reales de principio y fin de tareas de forma que nos superpone al diagrama original otro denominado “de seguimiento”. Este “diagrama de seguimiento” es muy útil para ayudar a la toma de decisiones en la ejecución de tareas.