

PROYECTO A/D-OO

1 OBJETIVOS

2 METODOLOGÍA

3 ENTREGABLES

4 NORMAS DE ENTREGA

1 OBJETIVOS

El objetivo principal de este proyecto es aplicar las habilidades necesarias para la realización del análisis y el diseño de una aplicación orientada a objetos que cubra las necesidades de información de un caso. Para conseguir este objetivo principal se establecen los siguientes objetivos subordinados:

- Conocer y utilizar el lenguaje de modelado UML
- Aplicar patrones de software y asignar responsabilidades a los componentes software
- Utilizar un proceso de desarrollo iterativo
- Comprender y realizar un conjunto de actividades de análisis y diseño de software en el marco de la metodología Proceso Unificado
- Conocer y utilizar una herramienta CASE
- Comprender la necesidad de analizar de forma estricta los requisitos y el entorno del sistema, entendiendo que el principal objetivo del análisis y el diseño de una aplicación es la resolución de un problema real de información
- Aplicar los métodos, técnicas y habilidades de análisis y diseño de forma coherente y consistente, haciendo encajar todas las piezas en la solución final
- Ser capaces de trabajar en equipo, organizando las tareas y resolviendo los conflictos inherentes a esta forma de trabajo
- Conocer y utilizar una herramienta de gestión de proyectos

2 METODOLOGÍA

Se utilizará un proceso de desarrollo iterativo basado en la metodología Proceso Unificado y soportado por el lenguaje de modelado UML, dos herramientas CASE y una herramienta de gestión de proyecto. Se pueden encontrar ejemplos sobre artefactos y patrones en el libro de Craig Larman ‘UML y patrones’ de la editorial Prentice Hall. El proceso constará de tres iteraciones:

Primera iteración. Se obtendrán los siguientes artefactos:

1.VISIÓN.

Descripción del problema, del entorno de negocio, de actores involucrados y de la solución, incluyendo un diagrama de contexto.

2.REQUISITOS. Lista de requisitos funcionales y no funcionales.

3.CASOS DE USO. Lista de casos de uso que respondan a los requisitos. Para cada caso de uso: una breve descripción, la lista de requisitos cubiertos (funcionales y no funcionales) y su prioridad (alto/medio/bajo). Diagrama de casos de uso que representen todos los casos de uso y sus relaciones (como será muy extenso puede dividirse en varios diagramas, por ejemplo uno por actor).

4.MODELO DE DOMINIO. Diagrama entidad/relación (diagrama conceptual normalizado del modelo de datos) que representa las entidades, atributos y asociaciones del entorno del problema y su solución. Descripción de elementos no obvios: entidades, atributos, relaciones y cardinalidades máximas y mínimas.

5.MODELADO DE CASOS DE USO. Seleccionar del artefacto 3, según la prioridad más alta, un mínimo de 8 casos de uso (que incluyan los casos de uso que cubrirán las funciones indicadas como importantes en el documento Enunciado_del_problema). Desarrollar cada uno de los casos de uso seleccionados, mediante una especificación según plantilla, incluyendo aspecto de interfaz y un diagrama de actividad.

Segunda iteración. Se obtendrán los siguientes artefactos:

ARTEFACTOS 1, 2, 3 y 4 revisados.

ARTEFACTO 5 revisado y completado con el desarrollo del total de casos de uso (o al menos de 30 casos). En el desarrollo de estos nuevos casos de uso no es necesario incluir aspecto de interfaz ni diagrama de actividad.

6.MODELO DE DATOS. Diagrama de tablas (para una base de datos relacional) y script de creación de la base de datos obtenido automáticamente del diagrama de tablas. Descripción, si es el caso, de otros elementos de almacenamiento de datos, por ejemplo ficheros de texto, xml, etc.

7.MODELO DE DISEÑO.

7.1.Diagrama de clases de diseño: diagrama de clases de diseño que representa la vista estática de las definiciones de las clases.

7.2.Modelo de objetos de software y sus colaboraciones (los 8 casos de uso desarrollados, hasta completarlos, entre la segunda y la tercera iteración, al menos 3 en la segunda iteración). Para cada caso de uso:

-*Diagrama de secuencia del sistema.*

-*Diagrama de colaboración de diseño*, para el escenario principal del caso de uso. Muestra el flujo de mensajes entre los objetos (invocación de métodos) para realizar el escenario. Es necesario diseñar toda la funcionalidad aunque se podrán utilizar framework o paquetes siempre que se indique, por ejemplo se podrá suponer que se utilizan un framework para gestionar la persistencia y que el sistema se comunica con él a través de una fachada, incluyendo solo la fachada en el diagrama. Cada mensaje en los diagramas de secuencia que represente un método no obvio (por ejemplo, selecciona información utilizando criterios, genera información combinando o elaborando otra o realiza cálculos) se representará mediante un diagrama de actividad y/o una descripción detallada de su funcionamiento y/o pseudocódigo.

Tercera iteración. Se obtendrán los siguientes artefactos:

ARTEFACTO 7 revisado y completado con los diagramas de secuencia y colaboración hasta completar un mínimo de 8 casos de uso.

8.MODELADO DE LA ARQUITECTURA. Diagrama despliegue que muestre nodos físicos. Para aquellos nodos que contengan componentes, un diagrama de componentes para cada uno, que muestre la estructura de paquetes, ficheros, etc. del nodo.

9.PLAN DE IMPLANTACIÓN. Incluye el diseño de la migración y/o carga inicial de datos, el manual de explotación y el manual de usuario (este último no tendrá que ser completo, se incluirá el índice detallado completo y el desarrollo de los apartados necesarios para incluir las funciones indicadas como importantes en el documento Enunciado_del_problema).

Se utilizará como soporte la herramienta CASE Modelio. Para el diagrama Entidad/Relación, el Diagrama de tablas y la generación de base de datos se utilizará la herramienta TOAD. **No se admiten otras herramientas CASE.**

Durante la realización del diseño se deberán aplicar, al menos, tres patrones de diseño y el patrón de arquitectura Modelo-Vista-Controlador. El análisis y el diseño presentado deben cubrir toda la funcionalidad solicitada en el enunciado de la práctica, considerándose favorablemente aquellas soluciones que aporten mayor funcionalidad.

Cada grupo realizará la gestión interna del mismo, llevando a cabo una reunión inicial en cada iteración, en la que se definen tareas a realizar, tiempo estimado y persona/s del grupo responsable de la tarea. Semanalmente se realizará una reunión de seguimiento de las tareas realizadas y de planificación del trabajo semanal, actualizando el tiempo real utilizado en cada tarea. También se pueden incluir, eliminar o modificar tareas si es necesario. El seguimiento y control del proyecto se llevará a cabo a través de la herramienta JIRA a la que tiene acceso el profesor, que podrá consultarla en cualquier momento por lo que esta información deberá estar siempre actualizada.

3 ENTREGABLES

Se considera un entregable la versión final de cada uno de los 9 artefactos desarrollados durante el análisis y el diseño de la aplicación. Además se consideran entregables estos dos informes:

10.APLICACIÓN DE PATRONES. Explicación gráfica y textual de la forma en que se han aplicado patrones en el proyecto realizado, indicando para cada patrón aplicado su nombre, el problema que resuelve y la solución aplicada. El problema resuelto y como se ha aplicado el patrón para su solución deben ser detallados mediante elementos, diagramas y/o modelos del proyecto. Por ejemplo, si se ha aplicado el patrón Experto para asignar un método determinado a una clase, se deberá indicar el método, las posibles clases candidatas para asignarlo y porque, en base al patrón, se ha elegido una de ellas

y se han descartado las otras, incluyendo el diagrama en el que se ve la asignación.

11. INFORME DE GESTIÓN DEL PROYECTO. Informes obtenidos de la herramienta JIRA según plantilla. Explicación de la gestión de versiones de los entregables y de los ficheros de los proyecto de Modelio y de TOAD (acceso a información compartida).

Cada entregable se entregará como un documento de texto (word o pdf) en el que se incluirán las tablas, figuras, diagramas, etc. copiados de las herramientas según sea necesario (excepto en los casos indicados en la plantilla). Se utilizarán las plantillas disponibles.

4 NORMAS DE ENTREGA

El proyecto se elaborará obligatoriamente por equipos constituidos por cuatro/cinco personas.

La entrega se realizará a través de Aula Virtual en un fichero zip con los entregables en word o pdf, **un único fichero** de proyecto de Modelio (un zip de la carpeta del proyecto) y **un único fichero** de proyecto de TOAD (**no se admitirán entregas con varios ficheros de TOAD o varios ficheros de Modelio**). El nombre del fichero zip será M o T (si el grupo es de mañana o de tarde), la hora de inicio de la clase de prácticas, un guion y el número de grupo (ejemplos: M12-1.zip, M8-3.zip, T14-2.zip). El nombre de los ficheros word o pdf será el número de entregable (por ejemplo 1.docx o 1.pdf para Visión).

En el caso de entregas en Evaluación continua se añadirá al fichero zip, un documento de texto (word o pdf) denominado Modificaciones, en el que se resuman las modificaciones realizadas a los artefacto entregados en iteraciones anteriores.

Se suspende la práctica si falta algún entregable o no sigue el formato o las instrucciones de la plantilla correspondiente, si no se envía un único proyecto de Toad en un único fichero y un único proyecto de Modelio en un único fichero, ambos con los elementos indicados en las instrucciones de las plantillas, o si no se ha utilizado Jira para la gestión del proyecto, incluyendo taras, responsables, tiempos estimados y reales de cada una de las iteraciones.

A criterio del profesor, en base al proyecto entregado, los alumnos seleccionados deberán exponer y defender su proyecto el día de la evaluación final.

La actualización, con foto, del perfil de cada alumno en Aula Virtual es obligatoria antes de octubre. No se evaluará ninguna iteración del proyecto si falta esta información de alguna de las personas del grupo.

Este documento es válido para todas las convocatorias del presente curso.

