

PROYECTO A/DOO

- 1 OBJETIVOS
- 2 METODOLOGÍA
- 3 ENTREGABLES
- 4 NORMAS DE ENTREGA
- 5 ANEXO - Comentarios a los artefactos

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

2 METODOLOGÍA

Se utilizará un proceso de desarrollo iterativo basado en la metodología Proceso Unificado y soportado por el lenguaje de modelado UML y una herramienta CASE. El proceso constará de tres iteraciones:

Primera iteración. Se obtendrán los siguientes artefactos:

1. VISION.

1.1.Descripción del problema: enunciado del problema, entorno de negocio y alternativas de solución.

1.2.Descripción del personal involucrado: participantes del problema y sus objetivos (de cada participante).

1.3.Descripción de la solución: diagrama de contexto, características y beneficios.

2.REQUISITOS. Lista de requisitos funcionales y no funcionales (facilidad de uso, fiabilidad, rendimiento, soporte, interfaces, gestión de errores, seguridad, restricciones software y hardware).

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 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 entidades, relaciones y atributos.

5.MODELADO DE CASOS DE USO. Seleccionar del artefacto 3, según la prioridad más alta, 12 casos de uso. Desarrollar cada uno de los casos de uso seleccionados, mediante una especificación según plantilla, incluyendo aspecto de interfaz y un diagrama de actividad.

6.PLAN DE PRUEBAS. Definición de un plan de pruebas que incluya estrategia y procedimientos de prueba (unitarias, de integración y de sistema) Desarrollar los casos de prueba de sistema necesarios para probar los 12 casos de uso seleccionados en el artefacto 5 y todos los requisitos no funcionales, incluyendo pruebas de esfuerzo o estrés.

7.ANÁLISIS DE CONSISTENCIA.

7.1.Tabla que relacione los requisitos (artefacto 2) con el párrafo del enunciado del que proceden. Explicación textual del origen de los requisitos no procedentes del enunciado.

7.2.Tabla de doble entrada (requisito / caso de prueba) que relacione para cada requisito (artefacto 2) los casos de prueba del sistema involucrados (artefacto 6) y viceversa.

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. En el desarrollo de estos nuevos casos de uso no es necesario incluir aspecto de interfaz ni diagrama de actividad.

ARTEFACTO 6 revisado. Incluir un ejemplo de prueba unitaria y un ejemplo de prueba de integración. No es necesario desarrollar las pruebas de sistema de los nuevos casos de uso modelados.

ARTEFACTO 7 revisado con la versión actualizada de los artefactos. Además:

7.3.Verificar que todas las clases y métodos del modelo de clases de diseño (artefacto 8.1) se utilizan en el modelo de colaboración de objetos software,

como objetos y mensajes, y viceversa (artefacto 8.2.2). Al haber utilizado Rational como herramienta CASE, basta indicar de forma detallada el tipo de control realizado utilizando la herramienta.

Hacer un diagrama de clases a partir del modelado. Ir añadiendo las cosas de diseño a este diagrama de clases. Habrá que añadir una fachada de la bbdd.

8.MODELO DE DISEÑO.

8.1.*Modelo de clases de diseño*: diagrama de clases de diseño que representa la vista estática de las definiciones de las clases.

8.2.*Modelo de objetos de software y sus colaboraciones* (seleccionados entre los casos de uso desarrollados, hasta completar, entre la segunda y la tercera iteración, un **mínimo de 8 casos de uso**). La selección de los casos de uso se hará teniendo en cuenta la mayor complejidad en el modelado del caso y la variedad de procedimiento, por ejemplo, si existen procesos batch se debe incluir alguno:

8.2.1. *Diagrama de secuencia del sistema*, uno para cada caso de uso.

8.2.2. *Diagramas de colaboración de diseño*, uno para cada escenario de cada caso de uso (obligatorio el escenario principal y aquellos que representen una funcionalidad compleja o importante). Cada diagrama muestra el flujo de mensajes entre los objetos (invocación de métodos) para realizar un escenario de un caso de uso. 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, utiliza los atributos de otras clases, 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.

A parte del diagrama entidad relacion, se hace automáticamente supuestamente con el Toad

9.**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.

Tercera iteración. Se obtendrán los siguientes artefactos:

ARTEFACTO 7 revisado con la versión final de los artefactos.

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

10.MODELADO DE LA ARQUITECTURA. Diagrama de componentes y de despliegue.

11.PLAN DE IMPLANTACIÓN. Incluye el diseño de la migración y/o carga inicial de datos, la estrategia de implantación (formación, documentación, infraestructura e instalación), el manual de explotación y el manual de usuario (para este último solo el índice completo y el desarrollo de uno de sus apartados).

Se utilizará como soporte la herramienta CASE Rational Modeler. 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 11 artefactos desarrollados durante el análisis y el diseño de la aplicación. Además se consideran entregables estos dos informes:

12.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.

13.INFORME DE GESTIÓN DEL PROYECTO. Informes obtenidos de la herramienta JIRA que muestren la división de trabajo del proyecto, las horas estimadas y reales por artefacto y por persona y que incluya los campos Comentario de las épicas, tareas y subtareas. Explicación de la gestión de versiones de los entregables (acceso a información compartida) y de los ficheros de los proyecto de Rational y de TOAD.

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.

4 NORMAS DE ENTREGA

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

La entrega se realizará a través de Aula Virtual en un fichero zip con los entregables en word o rtf, el fichero de proyecto de Rational y el fichero del proyecto de TOAD. 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.doc o 1.pdf para Visión).

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.

5 ANEXO - Comentarios a los artefactos

Se pueden encontrar ejemplos sobre artefactos y patrones en el libro de Craig Larman 'UML y patrones' de la editorial Prentice Hall.

Diagrama de contexto. Un único diagrama de casos de uso en el que se ven los subsistemas como casos de uso y su relación con los actores (personas o sistemas de información). Un subsistema es una agrupación lógica de casos de uso. Se describe brevemente el flujo de información de cada relación (puede ser un nombre o una frase). El número ideal para sistemas medios es tres o cuatro, en sistemas sencillos pueden no utilizarse subsistemas y representar solamente al sistema completo.

Descripción detallada de un caso de uso. Debe contener:

Identificativo y Nombre

Descripción breve

Actor principal y otros actores involucrados

Requisitos funcionales cubiertos

Precondiciones:

Condiciones para terminar con éxito el caso: Situaciones, datos condiciones, etc. importantes que hay que probar en algún momento de la ejecución del caso de uso; Situaciones, datos condiciones, etc. no sometidas a prueba, pero básicas para éxito del caso de uso.

Postcondiciones:

Situación del sistema después de acabar con éxito el caso de uso. Describir:
Creación y eliminación de entidades; Modificación de atributos;
Asociaciones entre entidades formadas o canceladas.

Escenario principal:

Actor	Sistema
1.- Paso 1.	2.-Paso 2.
3.- Paso 3.	4.- Paso 4.
...	...

Escenarios excepcionales o alternativos:

En paso N: si no se ha podido x, entonces xxxxx

En paso N: si y , entonces xxxxx

En cualquier momentos: si z, entonces xxxxx

Requisitos no funcionales involucrados

Frecuencia (si el caso de uso tiene una frecuencia definida)

Diagrama de actividad (solo para los casos de uso más prioritarios)

Descripción de Interfaz (solo para los casos de uso más prioritarios):

Lista y descripción de pantallas que aparecen a lo largo del desarrollo del caso. Representación de las pantallas más significativas para servir como prototipos. Pautas generales de diseño de la interfaz (teclas de función, menú, iconos, ayuda on-line, mensajes de error)

Plan de pruebas. Definición del plan (recursos, entorno de pruebas, herramientas, tiempos) y tipos de prueba. En este proyecto no es necesario definir casos de prueba de cada prueba unitaria ni de integración, pero si **indicar el tipo de elementos sobre los que se realizarán las pruebas unitarias y de integración y un ejemplo de cada uno de estos tipos de pruebas.** Las pruebas de sistema sí deben planificarse completas para los 12 casos de uso seleccionados en el artefacto 5 y todos los requisitos no funcionales. **No olvidar planificar pruebas de esfuerzo.** La planificación de cada prueba incluirá más de un caso de prueba. Se indicará el formato del documento de planificación de las pruebas y se rellenará para cada prueba planificada. Un ejemplo de formato puede ser este:

Identificativo de prueba	Elemento/s de prueba (RNF o CU)	Número del caso / Número total	Objetivo	Entrada	Salida esperada

Análisis de consistencia, artefacto 7.2. Ejemplo de tabla entre los requisitos y las pruebas sistema. En horizontal se representan todos requisitos y en vertical todas las pruebas. La intersección se marca cuando la prueba se utilice para probar el requisito.

	Requisito 1	...	Requisito n
Identificativo de Prueba 1			
...			
Identificativo de Prueba n			

Manual de explotación. El objetivo de este documento es recoger el conjunto de tareas que se deberán realizar para la correcta explotación del sistema. Se deberán indicar, entre otros aspectos, las tareas programadas (gestión de backups, mantenimiento de logs, etc.), procedimiento de paradas programadas, monitorización y gestión de la capacidad. Esta es la información que puede ir en un manual de explotación:

ÍNDICE	DESCRIPCIÓN
1. CALENDARIO DE OPERACIONES A REALIZAR.	Calendario resumen en el que se especifica en qué momento debe realizar cada una de las actividades que se describen a continuación.
2. DISPOSITIVOS DE ALMACENAMIENTO SECUNDARIO A UTILIZAR.	En este punto se le da información sobre los dispositivos de almacenamiento secundario que se deberán utilizar para hacer copias de seguridad o salvados de la base de datos u otra información del Sistema.
3. REALIZACIÓN DE COPIAS DE SEGURIDAD.	En este punto se le informa de cómo realizar, paso a paso, las Copias de Seguridad. También se le indican los momentos más adecuados para realizarlas y los requisitos que se deben cumplir para que éstas puedan llevarse a cabo y resulten útiles.
4. CLASIFICACIÓN Y ACCESO DE COPIAS DE SEGURIDAD.	Información para administrar las Copias de Seguridad de forma que si es necesario acudir a ellas sigan un orden cronológico y estén debidamente etiquetadas e identificadas.
5. MONITORIZACIÓN Y GESTIÓN DE LA CAPACIDAD.	Descripción detallada de aquellos recursos del Sistema que deben ser monitorizados. Se indicarán las necesidades de monitorización existentes para cada recurso, los umbrales esperados y las previsiones estimadas de crecimiento del Sistema en términos de almacenamiento, capacidad de procesamiento, tráfico de red y cualquier otro recurso.
6. EMISIÓN DE INFORMES A PETICIÓN.	Listado de posibles informes a petición y usuarios autorizados.
7. ESTABLECIMIENTO DE PUNTOS DE RESTAURACIÓN DEL SISTEMA.	Información para que el operario conozca los pasos a seguir para establecer puntos de restauración del sistema que permitan deshacer los cambios realizados en el Sistema desde la última vez que el equipo funcionaba correctamente.
8. SALVADO DE LA BASE DE DATOS.	Enumeración de las tareas a realizar y las normas que se deben cumplir para realizar el salvado de la base de datos.
9. TAREAS DE MANTENIMIENTO DE LOS EQUIPOS.	Enumeración de las tareas de mantenimiento y revisiones periódicas que los equipos precisan.
10. RESPONSABLES DIRECTOS DE CADA FUNCIONALIDAD DEL SISTEMA.	Listado de los responsables (y modos de contactar con ellos) de cada proceso que se debe lanzar y que le ha podido dar problemas. Aparece un responsable funcional (usuario de la propia empresa) y un responsable técnico un único responsable de mantenimiento de la propia empresa o de la empresa con la que lo tenemos contratado)
11. ACTUACIONES ANTE SITUACIONES DE RIESGO.	Incluye aquellas actuaciones destinadas a restaurar el servicio ante situaciones anormales como caídas del Sistema, corte de suministro eléctrico, pérdida de datos, fallos de seguridad, etc. (planes de contingencia). Todas aquellas personas relacionadas con el Sistema que se puedan ver afectadas por un arranque, re arranque o parada del Sistema, deben ser avisadas a través de una lista de correo.
12. ROTACIÓN DE LOGS. PERIODICIDAD.	Entre las tareas programadas del Sistema encontramos la rotación de Logs. En este apartado se describirá la política de rotación de logs y la periodicidad de esta rotación. Quedarán definidas las condiciones de rotación (tamaño de logs, intervalo de tiempo, etc.), el procedimiento de rotación y la política de almacenamiento de logs antiguos, así como cualquier otra información que pueda resultar de interés.