



Pauta de Informe de Proyecto
ILI-236 – Fundamentos de Ingeniería de Software

Profesores: Hernán Astudillo y Marcello Visconti

1	Introducción	3
2	Plan de trabajo	3
3	Análisis	3
3.1	Contexto	3
3.2	Requerimientos.....	3
3.3	Actores.....	3
3.4	Casos de Uso.....	3
3.5	Modelo de Dominio	4
4	Validación	4
5	Diseño.....	4
5.1	Derivación del Modelo de Software.....	4
5.2	Refinamientos	4
5.3	Implantación	5

Preámbulo

El curso requiere entregar 3 informes de progreso, cada uno con la misma ponderación en la nota final. En cada informe se evaluará tanto la *forma* como el *contenido*. Los aspectos de forma incluyen criterios que todo documento profesional debe tener (ausencia de errores ortográficos, claridad de los diagramas, claridad del formato empleado, etc.), y criterios propios de modelado UML (claridad, consistencia entre diagramas, y consistencia de diagramas con otros artefactos). Los aspectos de fondo son completitud y correctitud del modelo con respecto al problema y/o solución examinados.

El propósito de todo informe es convencer a lectores profesionales, posiblemente con conocimiento del dominio del problema, de que se ha hecho un estudio serio del problema y de que se ha identificado una solución razonable. Se sugiere mantener esta filosofía en mente mientras se escribe el informe, porque una explicación simple pero efectiva es más valiosa que una explicación que tiene todo lo pedido pero no logra convencer a su audiencia.

Calendario

Los informes a entregar son:

- ✍ Primer avance (9 de septiembre): casos de uso.
 - ? Identificar el problema, y describirlo con casos de uso de negocio (versión 1), contextualizados y priorizados.
 - ? Entregar secciones 1, 2 y 3.1-3.4 (v1.0).
- ✍ Segundo avance (14 de Octubre): modelo de dominio y validación con prototipo.
 - ? Sintetizar modelo de dominio, refinar y detallar casos de uso, y elaborar prototipo funcional para validar un caso de uso de alta prioridad.
 - ? Entregar secciones 1, 2, 3.1-3.4 (v2.0), 3.5 (v1.0), y 4.
- ✍ Informe final (11 de noviembre): diseño.
 - ? Derivar modelo de software inicial a partir de modelo de dominio, aplicar patrones para aspectos regulares de la solución, refinar partes clave del modelo de software, generar código esqueleto (incompleto) para todo el sistema, y documentar despliegue y dependencias de componentes.
 - ? Entregar secciones 1, 2, 3.1-3.4 (v3.0), 3.5 (v2.0), 4 y 5.

El informe final (completo) debe convencer a los revisores (y a cualquier lector) de que se ha identificado tareas y requerimientos, se ha elaborado un modelo de dominio que representa todas las entidades y procesos relevantes, se ha derivado un modelo de software que explica cómo construir un sistema que haga lo pedido, y se ha comenzado a construir y se podría completar exitosamente si hubiera habido tiempo. No se espera un detalle exhaustivo más allá de lo necesario para convencer, pero por cierto no se espera menos detalle tampoco.

1 INTRODUCCIÓN

PROPÓSITO: EXPLICAR AL LECTOR DE QUÉ TRATA ESTE DOCUMENTO Y SI LE CORRESPONDE LEERLO.

- ✍ Propósito, audiencia (conocimientos esperados), y estructura del documento.
- ✍ Descripción muy breve del problema, aspectos clave identificados, y solución recomendada (todos aspectos detallados más abajo).

2 PLAN DE TRABAJO

PROPÓSITO: CONVENCER A LOS CORRECTORES QUE EL EQUIPO DE TRABAJO SABE LO QUE HACE.

- ✍ **Presentación del equipo de trabajo:** recursos (humanos y técnicos) que tiene el grupo a disposición (miembros, equipos, conocimientos especializados relevantes).
- ✍ **Próximos pasos a tomar:** qué van a hacer para el próximo informe.

3 ANÁLISIS

PROPÓSITO: IDENTIFICAR Y DESCRIBIR LAS TAREAS RELEVANTES DEL PROBLEMA.

3.1 Contexto

PROPÓSITO: DEMOSTRAR QUE HAY UN PROBLEMA Y MERECE RESOLVERSE.

- ✍ Presentación general de la situación actual, con su contexto comercial, organizacional, tecnológico, etc.
- ✍ Explicación del problema existente en la situación actual.

3.2 Requerimientos

- ✍ Lista de funciones del sistema, asignadas a categorías de evidente, oculta o superflua
- ✍ Lista de Atributos del sistema, incluyendo detalles y limitaciones
- ✍ Lista de atributos por funcionalidad, especificando qué atributos debe poseer cada función.

3.3 Actores

PROPÓSITO: PRESENTAR LOS PARTICIPANTES EXTERNOS AL PROBLEMA PERO RELEVANTES.

- ✍ Actores: lista de actores, brevemente descritos.

3.4 Casos de Uso

PROPÓSITO: PRESENTAR Y DESCRIBIR LAS TAREAS DE ALTO NIVEL DEL SISTEMA, RELACIONARLAS CON LOS ACTORES, Y PRIORIZARLAS POR SU RELEVANCIA.

- ✍ **Casos de uso esenciales:** presentación general de los casos de uso identificados, con razonamiento textual (listas) y síntesis gráfica (diagramas de casos de uso).
- ✍ **Para cada caso de uso:** descripción textual (estandarizada) y gráfica (modelo conceptual, diagramas de secuencia de sistema, contratos), e identificación de actores involucrados.
- ✍ **Priorización:** asignación de casos de uso a categorías de esencial, importante y deseable.

3.5 Modelo de Dominio

- ✍ **Entidades reconocidas:** explicación textual breve de entidades que aparezcan en varios casos de uso o que sean sintetizadas a partir de diferentes casos de uso.
- ✍ **Modelo de dominio:** modelo de entidades, agrupadas en paquetes de unidad conceptual y con dependencias (si las hay).
- ✍ **Matriz de rastreabilidad:** una matriz que indique cuáles clases del dominio participan en cuáles casos de uso.

4 VALIDACIÓN

PROPÓSITO: CONVENCER AL CLIENTE O USUARIO QUE SE HA COMPRENDIDO SU PROBLEMA.

- ✍ **Prototipo de validación funcional:** programa simple que muestra cómo un caso de uso de alta prioridad puede ser abordado con las entidades identificadas. Entregar ejecutable y raciocinio empleado en su elección.

5 DISEÑO

PROPÓSITO: DESCRIBIR LA SOLUCIÓN Y RELACIONARLA AL PROBLEMA.

5.1 Derivación del Modelo de Software

PROPÓSITO: MOSTRAR AL INFORMÁTICO QUE EL SISTEMA PROPUESTO DERIVA DEL PROBLEMA DESCRITO.

- ✍ **Modelo de software inicial:** identificación de clases del modelo de dominio que pueden ser usadas como base del software, y elaboración del modelo de software inicial. Entregar casos de uso reales, diagramas de clases, diagramas de interacción (secuencia o colaboración), y diagramas de estados para aquellas situaciones que lo ameriten.

5.2 Refinamientos

PROPÓSITO: MOSTRAR AL INFORMÁTICO CÓMO SE FUE DE SOLUCIÓN GENERAL A DETALLADA.

- ✍ Identificación de lugares de posible refinamiento (p.ej. interacciones, dependencias, etc.).
- ✍ **Para cada lugar de refinamiento:**
 - ? Refinamientos considerados (a pulso o con patrones, según corresponda).

- ? Selección y descripción de una opción: descripción detallada (a nivel software) de la aplicación del refinamiento elegido. Entregar modelos relevantes.

5.3 Implantación

PROPÓSITO: MOSTRAR A LOS INSTALADORES Y ADMINISTRADORES DEL SISTEMA LAS PARTES A ENTREGAR Y SU INTERACCIÓN.

- ✍ **Código fuente completo (parcial):** generación de esqueleto para todo el código fuente del sistema a partir del modelo.
- ✍ **Modelo de implantación:** diagramas de componentes y despliegue.
- ✍ **Dependencias:** indicación de qué productos, componentes o bibliotecas serán requeridas para que el sistema pueda operar normalmente.