Proyecto de Taller V

Ingeniería de Software Orientada a Objetos

Informe Principal
junio de 2001

Instituto de Computación
Facultad de Ingeniería
Universidad de la República

Autores: Natalia Dimu
Alejandro Friss de Kereki
Tutores: Ing. Rodolfo Paiz
Ing. Andrés Vignaga

Resumen

En 1999 se propuso el proyecto *Estudio de Herramientas para el Desarrollo Orientado a Objetos*, que entre otras cosas produjo una propuesta metodológica que abarca aspectos de análisis, diseño e implementación orientada a objetos. En dicho trabajo se hizo particular énfasis en la aplicación del *UML* como estándar de notación, quedando pendiente completar la metodología con temas como gestión, calidad y verificación. Esto motivó la propuesta de un nuevo proyecto en el año 2000 que continuara con la línea de trabajo comenzada el año anterior. Ese proyecto es que se presenta en este trabajo, y se denomina *Ingeniería de Software Orientada a Objetos.*

El resultado es una propuesta metodológica completa que abarca absolutamente todos los aspectos cubiertos por el trabajo anterior y que da una serie de pautas y consejos para realizar las tareas relacionadas con la gestión, calidad y verificación, sobre cada actividad de esta metodología.

Palabras clave: Ingeniería de software, proceso de desarrollo, gestión, calidad, verificación, orientación a objetos, casos de uso.

Resumen de Contenido

Contenido

Parte I

Introducción

Capítulo 1
Presentación

En este documento se presenta el proyecto de Taller V *Ingeniería de Software Orientada a Objetos*. En él se plantean los objetivos perseguidos, se explica el abordaje del proyecto, se sintetizan los elementos producidos y se exponen las conclusiones.

1.1 Objetivos del Proyecto

Los objetivos de este proyecto son:

- Estudiar metodologías de desarrollo orientadas a objetos, concentrándose en las áreas de verificación, calidad y gestión de proyectos, las que muchas veces no son exploradas especialmente.
- Definir una metodología que complemente la obtenida en el Taller V de 1999 *Herramientas para el Desarrollo Orientado a Objetos*, aportándole las actividades correspondientes a las áreas anteriores, que no formaban parte de la misma. Con esto se pretende ampliar el alcance de la metodología anterior, la que ya había resultado exitosa.

1.2 Alcance del Proyecto

Para el desarrollo de este proyecto se determinaron las siguientes restricciones:

- Sólo se estudiará metodologías que empleen el paradigma de orientación a objetos o contemplen la posibilidad de utilizarlo.
- Para el complemento de la metodología iniciada en el taller anterior, en general se considerará a la misma como una entidad cerrada, o sea que la nueva metodología deberá adaptarse a la anterior. Sólo será posible alterarla sujeta a la aprobación de su autor y luego de discutir la conveniencia de los cambios.
- Las actividades principales, sobre las que se concentrará la atención, son la verificación, calidad y gestión del proyecto. También es posible incluir referencias a otras actividades, sin necesidad de profundizar en ellas.
- No se contemplará la definición de un proceso de mejora continua.

1.3 Logros

- Conocimiento detallado de la metodología presentada en el taller anterior.
- Conocimiento del estado del arte en cuanto a metodologías para el desarrollo orientado a objetos.
- Conocimiento de un amplio espectro de opiniones acerca de gestión, calidad y verificación.
- Descripción de una metodología que complementa la propuesta en el mencionado taller
- Documentación completa de los resultados.

1.4 Desarrollo del Proyecto

En el desarrollo del proyecto, el primer paso consistió en lograr un conocimiento profundo de la metodología propuesta en el taller anterior.

El siguiente paso fue la investigación y determinación del estado del arte en cuanto a metodologías de desarrollo orientadas a objetos, haciendo hincapié en las actividades de gestión de proyectos, verificación y calidad.

Una vez que se obtuvo una visión relativamente amplia de los temas referentes a este proyecto, se pasó a la etapa de decidir cuál sería la forma de presentación de la metodología.

El último paso fue la concepción concreta de la metodología que abarcara la propuesta realizada en 1999. Durante el desarrollo de esta etapa surgió de manera lateral un tema de investigación que motivó la realización del reporte técnico *Detección de Relaciones entre Casos de Uso*, en conjunto con uno de los tutores del proyecto. Dicho reporte técnico se encuentra en consideración de ser publicado en la conferencia CLEI 2001.

1.5 Documentación del Proyecto

Se describe en esta sección la manera en que están organizados los documentos del Taller 2000 y cuál es su estructura. También se especificará qué es lo que se debe tener en cuenta al momento de leerlos.

1.5.1 Organización de Documentos

La documentación del proyecto está compuesta por tres partes: un documento metodológico, que contiene una descripción detallada de la metodología propuesta; un reporte técnico llamado “Detección de Relaciones entre Casos de Uso” que desarrolla un aspecto de la propuesta metodológica; y un informe principal, que es este documento.

La figura 1-1 muestra un diagrama de componentes que permite visualizar esta estructura.

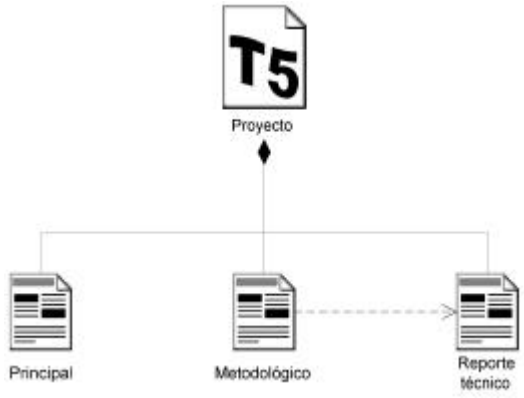

Fig. 1-1: Documentos del proyecto

1.5.2 Recomendaciones para la Lectura

Las siguientes recomendaciones son útiles para una fácil lectura de los documentos:

1. Estar familiarizado con la notación de UML

2. Lectura completa de este documento. Con ello se obtiene:

   - Una visión general del proyecto.
   - Conceptos fundamentales usados en el mismo.
   - Descripción de la metodología.

3. Lectura del documento metodológico. Con ello se obtiene:

   - Una visión detallada de la metodología, integrada con la parte correspondiente al taller anterior.

4. Lectura del informe técnico *Detección de Relaciones entre Casos de Uso*. Con ello se obtiene:

   - Conocimiento detallado de una de las técnicas utilizada en la metodología, que fue propuesta por los autores de este taller y desarrollada en conjunto con el autor del taller anterior y tutor del actual.

1.5.3 Estructura de los Documentos

El documento metodológico y este informe están estructurados de una misma manera. Tienen una primera parte que corresponde a una introducción, en la que se hace una presentación del documento y se explica cómo está organizado. El contenido central puede estar formado por una o más partes. En caso de ser necesario, una parte final contiene los apéndices, el glosario y el índice alfabético.

Para facilitar la lectura, los temas están en general organizados en capítulos cortos y de manera esquemática.

Cabe aclarar que el reporte técnico tiene un formato especial, distinto al de los otros documentos en cuanto a sus partes. Esto obedece a que este documento presenta características

esencialmente distintas y además debía cumplir con el formato esperado para el envío a una conferencia internacional.

Si bien el documento metodológico presenta la metodología total, incorporando también los elementos desarrollados en el taller anterior de forma integrada, es necesario distinguir claramente lo generado en ambas versiones, como es lógico tratándose de un proyecto de grado.

En el documento metodológico, las partes que corresponden con el primer proyecto están indicadas con una línea vertical a la derecha de los párrafos, como en este ejemplo:

Este párrafo fue generado en la primera etapa, como lo indica la línea vertical que se encuentra a la derecha.

Éste pertenece a la segunda.

Capítulo 2
Organización del Documento

Este documento es el informe principal del proyecto de Taller V *Ingeniería de Software Orientada a Objetos* y contiene información general del proyecto, como ser una presentación, los objetivos y las conclusiones, así como capítulos dedicados a los temas específicos tratados durante su desarrollo.

Este documento está organizado de la siguiente forma: está dividido en tres partes, la primera corresponde a una introducción, la segunda a las actividades desarrolladas en el proyecto y la tercera a los apéndices.

La segunda parte contiene información acerca de las actividades desarrolladas en el proyecto: el capítulo 4 está dedicado a presentar la definición de los conceptos centrales. El capítulo 5 presenta un resumen de la metodología propuesta. El capítulo 6 está dedicado a las conclusiones a las que se llegó luego de completado el proyecto y al planteo de los trabajos futuros.

La tercera parte contiene dos apéndices. El primero presenta un resumen de la metodología desarrollada en el taller anterior, cuyo conocimiento es esencial para comprender las tareas propuestas en la nueva versión. Este resumen está tomado del informe principal de dicho taller.
El segundo apéndice presenta un resumen de las propuestas que de alguna forma proporcionan las bases de la metodología.

Parte II

El Proyecto

Capítulo 3
Conceptos Principales

En el desarrollo de cualquier tema, es esencial definir correctamente los principales conceptos usados, sobre todo cuando los mismos admiten generalmente muchas definiciones. Para la presente propuesta, estos conceptos son los de verificación, calidad y gestión.

3.1 Verificación

En forma genérica, la verificación es el conjunto de tareas que pretenden asegurar que un cierto producto cumple la función que se espera de él.

En las propuestas metodológicas, es frecuente que la verificación se vea limitada a los productos de software.

En el contexto de la presente propuesta, el significado de este concepto se amplía de forma considerable y cualquier producto generado puede ser objeto de verificación. En este caso, el software es sólo un producto más, aunque por supuesto que se reconoce su importancia. Esta importancia es la que sugiere que su verificación reciba un nombre particular, y se la conoce por *testeo* (anglicismo por *testing*).

Así como se define precisamente los productos que abarca, es necesario definir el objetivo de la verificación limitando su alcance, lo cual se realiza a continuación.

Anteriormente, se aclaró que la verificación se vincula con la función que se espera de un cierto producto, por lo que no abarca otras cualidades importantes, como su comprensibilidad, mantenibilidad, entre otras.
Por ejemplo, en un diagrama de colaboración no se verifica si la asignación de responsabilidades a los objetos es la más conveniente o si hay posibilidad de mejorar el diseño mediante el agregado de una nueva clase. Lo único que se debe verificar es que mediante el intercambio de mensajes se pueda realizar correctamente la operación para la que se crea el diagrama.

Todas las otras cualidades entran bajo la órbita del área de calidad, que se presenta a continuación.

3.2 Calidad

Las tareas de calidad suelen dividirse en tres grandes actividades: definición, aseguramiento y control. La definición refiere a enunciar los aspectos que se consideran importantes, el

aseguramiento a la descripción de los procedimientos que se aplicarán para cumplir con esos aspectos y el control a las formas en que se comprobará que esos procedimientos se cumplan y los objetivos se logren.

Además, puede establecerse otra clasificación según se oriente a los productos o al proceso.

En este trabajo, el término calidad se aplica a todas estas actividades.

La propuesta cubre los aspectos de definición y aseguramiento de la calidad ya que, donde corresponde, se indica los aspectos importantes a considerar y la forma de comprobarlos para cada producto o proceso. Por lo tanto, quedan sentadas las bases para que el responsable de calidad pueda cumplir correctamente con el control.

Para ilustrar la aplicación del concepto de calidad, puede resaltarse que en la propuesta se brinda especial atención a que el usuario quede satisfecho con el proceso de desarrollo y el producto final, y al chequeo de todos los factores que pueden provocar algún riesgo al proyecto. En muchos casos, se sugiere realizar esto mediante reuniones con los involucrados.

En general, también la toma de mediciones es responsabilidad de calidad (mediciones de cumplimiento, de avance, estimaciones, etc.).

La definición de las actividades se orienta a alcanzar un proceso de desarrollo con un nivel de madurez que se considere claramente repetible.

3.3 Gestión

El concepto de gestión que se maneja en la propuesta coincide en buena medida con una definición usual.

Las tarea principal de gestión es la planificación y seguimiento de todas las actividades. La planificación implica también la asignación de recursos, mientras que el seguimiento se vincula en buena medida con la evaluación de los riesgos detectados.

Al igual que en el caso de la calidad, en la propuesta se trata de brindar la mayor facilidad para esto, aunque en general no es posible llegar a dar detalles debido a las variantes que puede presentar un proyecto. Sin embargo, lo que se busca es simplificar la tarea mediante el enunciado de las distintas responsabilidades en cada actividad y sugerencias o pautas para su cumplimiento.

Capítulo 4
Metodología

Esta metodología se caracteriza por un enfoque orientado a objetos, siguiendo un modelo iterativo e incremental de desarrollo.

La construcción del sistema se divide en partes y cada una de ellas es desarrollada por completo de a una por vez.
Para esto, previo a la construcción se realiza la planificación y elaboración del problema, en la que se determina la división que condicionará la etapa de construcción. El criterio de partición del problema a usar son los casos de uso.
De esta manera se definen en primer lugar dos etapas: una de planificación y elaboración, y otra de construcción.
Las tareas básicas en este contexto constituyen la parte propia del taller anterior y para un mayor detalle, se puede consultar el apéndice A.

En este capítulo se describe la metodología propuesta, limitada a los aspectos agregados en este trabajo.

4.1 Características y Etapas

El complemento a la metodología está levemente estructurado de acuerdo a la propuesta *Unified Process* (UP) de Ivar Jacobson, Grady Booch y James Rumbaugh, pero también complementado con elementos de otras propuestas, entre las que se puede destacar *Object-Oriented Software Process* (OOSP); además de aportes puntuales tomados de muy diversas fuentes.

La forma de presentación sigue los lineamientos del taller anterior pero hubo necesidad de extrapolar etapas y actividades dentro de éstas, ya que los temas involucrados no tenian un contexto previamente definido.

La principal característica es un seguimiento constante de todo el proceso de desarrollo, desde el comienzo mismo del mismo y desde los puntos de vista de la verificación, calidad y gestión. Esto obedece a una intención de, por un lado, reconocer y atacar inmediatamente los riesgos que se puedan producir, y por el otro, lograr una eficaz administración de los recursos.

Otra característica a resaltar es que la metodología cubre todos los aspectos relativos a un proceso de desarrollo, aunque algunos son deliberadamente pasados a un segundo plano por ser conocidos y prácticamente independientes de la metodología utilizada, como es el caso de la gestión de configuraciones del software (SCM).

Para el proceso de desarrollo se definen cuatro etapas, cada una de las cuales puede estar formada por varias actividades.

La primera es una etapa de inicio donde se realizan las tareas de preparación, fundamentalmente vinculadas con la gestión. Posteriormente se inicia el desarrollo en sí, que se divide en dos etapas. La primera es la de planificación y elaboración, y luego se pasa a la etapa de construcción. Finalmente, se llega a una etapa donde se realizan actividades posteriores, vinculadas con la revisión de todo el proceso de desarrollo y la actividad de soporte y mantenimiento.

Estas etapas se muestran en la figura 4-1.

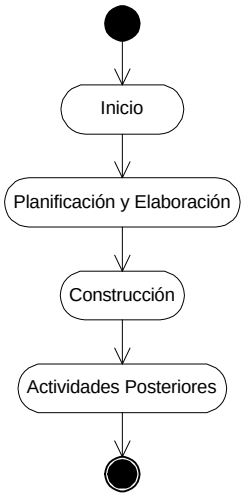

Figura 4-1: Etapas del proceso de desarrollo

4.2 Inicio

En esta etapa se realizan las tareas previas al comienzo del proyecto, y está compuesta por una sola actividad, como se muestra en la figura 4-2.

Figura 4-2: Actividad de inicio

4.2.1 Preparación del Proyecto

Antes de enfrentar un proyecto, hay varios aspectos que se deben planear, como la estructura del equipo de desarrollo. Además hay algunas tareas que ya deben ser realizadas, como la contratación (si es necesaria) y asignación del personal necesario para comenzar.

4.3 Planificación y Elaboración

En esta etapa se analiza toda la realidad del problema, por lo que hay una importante tarea de validación a realizar. Por otro lado, se hace necesario realizar estimaciones del esfuerzo que podrá demandar el desarrollo.

La figura 4-3 muestra las actividades que componen esta etapa, aunque es necesario aclarar que, a pesar de lo que se indica en el diagrama, las mismas no son necesariamente secuenciales, puede haber cierto grado de concurrencia.

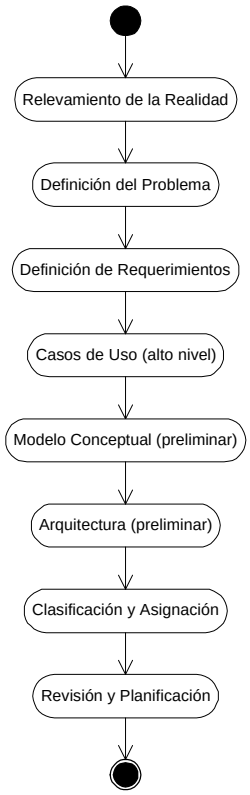

Figura 4-3: Actividades de planificación y elaboración

A continuación se describen las actividades dentro de la etapa de planificación y elaboración.

4.3.1 Relevamiento de la Realidad

Desde el primer momento se comienza la verificación de todas las actividades y todos los productos, lo que en este caso se realiza de forma conjunta con el cliente pero con una guía definida. Esto último tiene el efecto de aprovechar al máximo el –muchas veces escaso– tiempo del cliente y del equipo de desarrollo.

4.3.2 Definición del Problema

En esta actividad sigue la coordinación con el cliente. Además, es posible realizar las primeras estimaciones del esfuerzo que podría insumir el proyecto, aunque condicionadas a la experiencia anterior.

4.3.3 Definición de Requerimientos

A partir de la definición de los requerimientos, sería posible realizar una estimación del costo del proyecto por medio de diversos métodos, entre ellos la medición de los puntos de función, bastante difundida. Sin embargo, en este punto no se sugiere especialmente su utilización porque es relativamente compleja y no está demostrado que se adapte bien a metodologías con las características de ésta (iterativa y guiada por casos de uso).

4.3.4 Casos de Uso de Alto Nivel

Tomando como base la definición de requerimientos, es posible definir los casos de uso mediante los requerimientos que involucran. Esto es posible porque un caso de uso es un proceso completo y de valor para el usuario, por lo que está formado por varios sub-procesos que coinciden con los requerimientos funcionales.
Esta forma de definir los casos de uso abre nuevas posibilidades para su verificación y análisis. En particular, definiendo los casos de uso de esta manera es posible realizar un análisis completo para detectar las relaciones que pueden existir entre ellos, lo que aumenta el grado de entendimiento y permite organizar mejor el trabajo.
En la metodología no era necesario profundizar en los detalles del estudio que se podría realizar, pero éste fue incluido en el reporte técnico generado, “Detección de relaciones entre casos de uso”.

A partir de la definición de los casos de uso, es posible realizar una estimación del esfuerzo que demandará el proyecto, medido directamente en horas-hombre.

4.3.5 Modelo Conceptual Preliminar

Aún siendo un modelo preliminar, es posible detectar ciertos detalles que podrían representar problemas en el futuro, ya que en definitiva este modelo servirá de base para el diseño.

4.3.6 Arquitectura Preliminar

La arquitectura es considerada como un factor que puede afectar profundamente el desarrollo del proyecto, tanto de forma positiva (dividiendo la complejidad del sistema) como de forma negativa (obligando a un gran rediseño posterior si la elección no es adecuada). Por esto, se recomienda la realización de una revisión formal donde se evalúe la decisión tomada.

4.3.7 Clasificación y Asignación de Casos de Uso

En esta actividad se determina el orden en que irán siendo atacados los casos de uso, de acuerdo a criterios establecidos. Estos criterios tienden a desarrollar primero los casos de uso que pueden presentar cierta dificultad, complejidad e impacto considerable sobre la arquitectura.
A partir de la asignación, se estima también la duración y el esfuerzo de cada ciclo.

4.3.8 Revisión y Planificación

Esta actividad consiste en dos actividades. Por un lado, un repaso de lo realizado durante toda la etapa de Planificación y elaboración, con el propósito de determinar el estado actual del proyecto y, en última instancia, llegar a decidir sobre su continuidad. Por otro lado, planificación de la etapa siguiente, la de Construcción.

4.4 Construcción

En esta etapa aparecen nuevas actividades vinculadas con el diseño y la implementación, y también comienza a trabajar a pleno todo el equipo.

Las distintas actividades de cada uno de los ciclos de desarrollo se muestran en la figura 4-4. Los andariveles organizan las actividades según sean de análisis, diseño o implementación.

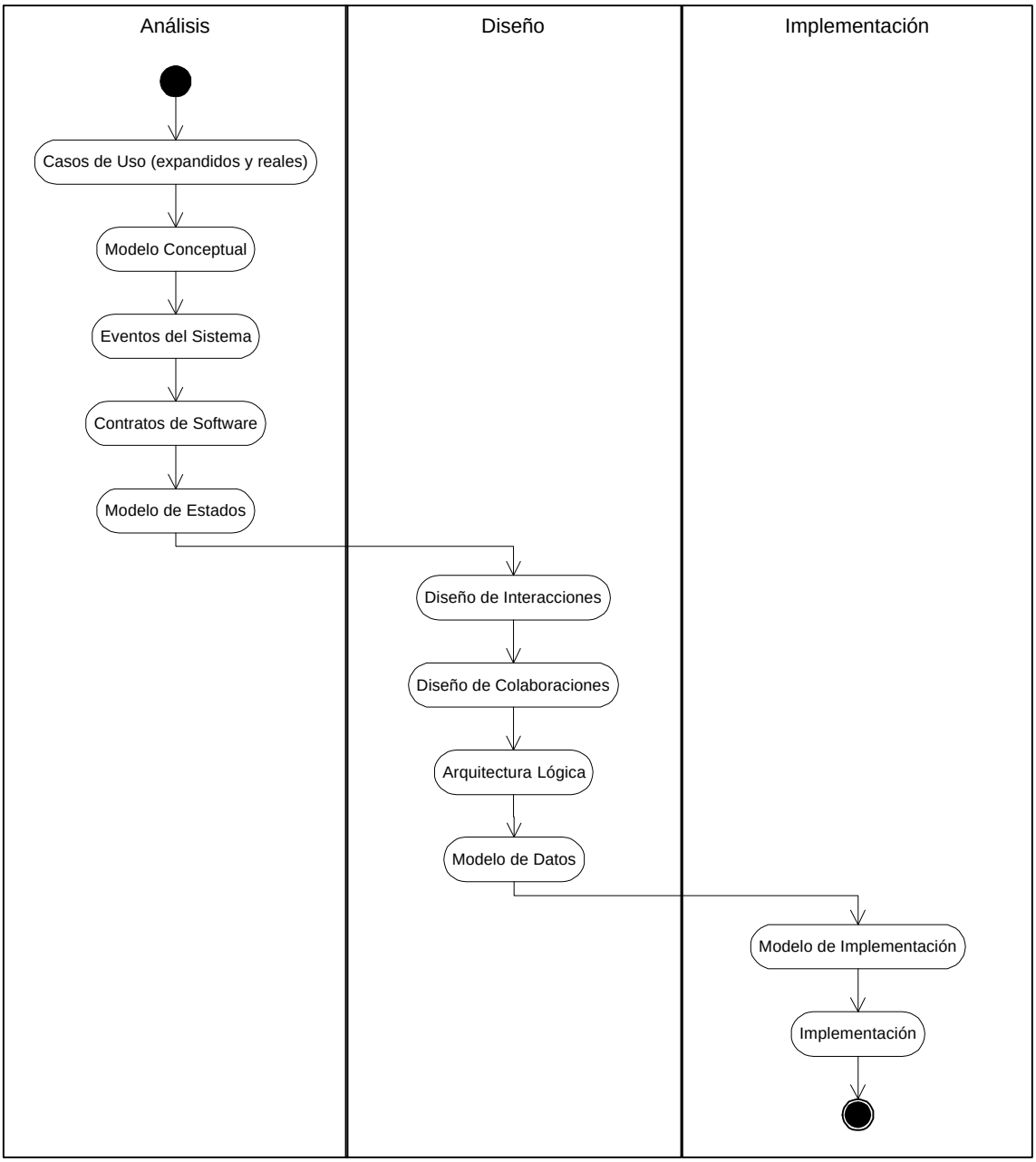

Fig. 4-4: Actividades de un ciclo de desarrollo

A continuación se describen las actividades dentro de un ciclo de desarrollo.

4.4.1 Casos de Uso Expandidos y Reales

Los casos de uso reales definen la interfaz de usuario, y en realidad es una actividad vinculada al diseño. Sin embargo, se adelanta hasta este punto porque la amigabilidad de la interfaz es posiblemente el factor que más influye en la satisfacción del cliente, y es conveniente que se vaya definiendo junto con él y se obtenga su aprobación.

4.4.2 Modelo Conceptual

Por consistir en un refinamiento del modelo conceptual preliminar, todas las actividades se vinculan con las anteriores. En particular, la planificación de recursos de alguna forma es complementaria a la anterior, ya que depende del nivel de detalle alcanzado en la primera instancia.

4.4.3 Eventos del Sistema

Los eventos del sistema representan las operaciones que se van a desarrollar, por lo que debe controlarse que no haya olvidos ni errores en su definición. Por otro lado, y en previsión de que se recurra al usuario para consultas futuras, debe comprobarse que estos eventos puedan llegar a ser comprendidos por él.

4.4.4 Contratos de Software

La tarea de definición de los contratos es relativamente compleja y costosa, por lo que una asignación adecuada de personal es particularmente útil en esta instancia. En particular, es conveniente que la persona que realiza el contrato sea la misma que identificó el evento correspondiente, aprovechando su conocimiento del mismo.

4.4.5 Modelo de Estados

Generalmente se realizan diagramas de estados para aquellos conceptos cuyo comportamiento ante ciertas operaciones no está bien comprendido o definido, por lo que hay que recurrir al cliente o al usuario del sistema para clarificar esta situación.

4.4.6 Diseño de Interacciones

Esta actividad es central en el diseño, ya que aparecen los objetos con sus responsabilidades. Además de permitir el cumplimiento de la funcionalidad esperada, el diseño debe tener otras características deseables, como: facilidad para comprenderlo, modificarlo y reutilizarlo.

4.4.7 Diseño de Colaboraciones

En esta actividad aparece el diagrama de clases donde se muestra las clases con sus atributos y métodos, por lo que su verificación allana el camino para la implementación.
Por otra parte, puede ser posible comenzar lentamente las tareas correspondientes al siguiente ciclo de desarrollo, ya que los responsables de las tareas de análisis prácticamente finalizaron sus actividades por el ciclo actual.

4.4.8 Arquitectura Lógica

En el caso normal, esta actividad es relativamente simple ya que básicamente consiste en ubicar los nuevos elementos en la arquitectura definida previamente, pero de cualquier manera exige una atención particular por la posibilidad de detección de problemas.

4.4.9 Modelo de Datos

Frecuentemente, el principal factor que determina la robustez y performance del sistema es la forma de almacenamiento de sus datos, por lo que es una actividad importante desde el punto de vista de la verificación y calidad.

4.4.10 Modelo de Implementación

El principal aspecto a controlar en esta instancia es que la organización del código fuente y los ejecutables generados guarden vinculación con la arquitectura.

4.4.11 Implementación

Esta actividad consiste en generar el código de la aplicación, definiendo las clases, sus atributos y métodos. La verificación de esta actividad es lo que se denomina propiamente *testeo*, y es una parte importante de la tarea total de verificación.
Por otro lado, al llegar al fin del ciclo de desarrollo, es momento de tomar mediciones y eventualmente realizar ajustes para los siguientes.

4.5 Actividades Posteriores

Una vez que se finaliza propiamente el desarrollo, quedan básicamente dos actividades por hacer vinculadas con el proyecto: evaluar todo el proceso y realizar el mantenimiento.

La figura 4-5 muestra estas actividades.

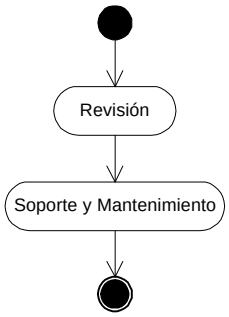

Fig. 4-5: Actividades posteriores

4.5.1 Revisión

Esta actividad consiste en realizar una evaluación lo más completa de lo ocurrido en el proyecto, buscando rescatar los aspectos positivos y reconociendo claramente los negativos. Con esto se busca obtener experiencia muy valiosa para los próximos proyectos.

4.5.2 Soporte y Mantenimiento

Esta actividad pretende aumentar el aprovechamiento del software por parte del usuario, lo que se realiza básicamente de dos formas: ayudar al cliente (soporte) y corregir los defectos del software o ajustar su funcionamiento (mantenimiento).

Capítulo 5
Conclusiones y Trabajos Futuros

5.1 Conclusiones

Concluido el proyecto, corresponde resaltar:

- Los logros (detallados en la sección 1.3) coinciden ampliamente con los objetivos planteados originalmente.
- La utilización de un lenguaje completo, en este caso *UML*, en todas las etapas del proceso, facilita las tareas vinculadas con la verificación.
- Se describe de forma detallada y precisa las tareas a realizar en cada actividad.
- La calidad comienza a controlarse desde el comienzo mismo del proceso de desarrollo, y además esto no implica un gran esfuerzo adicional.
- Pueden faltar elementos de evaluación porque no fue posible afianzar las ideas mediante el seguimiento de un caso de estudio.

5.2 Trabajos Futuros

- Consolidar la propuesta metodológica con un proyecto que la aplique en su totalidad.
- Definición precisa del nivel de la metodología en algún modelo de proceso formal, por ejemplo *CMM*.
- Adaptar especialmente la metodología para el empleo de herramientas *CASE*.
- Desarrollo de una herramienta automatizada para detectar relaciones entre casos de uso.

Parte III

Apéndices

Apéndice A
Resumen de la Metodología Anterior

En este apéndice se describe la metodología anterior *Estudio de Herramientas para el Desarrollo Orientado a Objetos* de Andrés Vignaga.
Esta es la metodología a partir de la cual surgió la propuesta metodológica de este Taller V *Ingeniería de Software Orientada a Objetos*.

Las siguientes secciones presentan a modo de resumen y sin entrar en detalles la metodología propuesta en el taller pasado. Por una descripción más profunda, se puede consultar la documentación correspondiente al taller anterior.

A.1 Características y Etapas

Esta metodología en esencia se centra en actividades de análisis, diseño e implementación orientada a objetos, siguiendo un modelo iterativo e incremental de desarrollo.

La construcción del sistema se divide en partes y cada una de ellas es desarrollada por completo de a una por vez.
Para esto, previo a la construcción se realiza la planificación y elaboración del problema, en la que se determina la división que condicionará la etapa de construcción. El criterio de partición del problema a usar son los casos de uso.

De esta manera se definen dos etapas: una primera de planificación y elaboración, y otra de construcción.

En la etapa de planificación y elaboración se identifican todos los casos de uso del sistema y en la etapa de construcción se implementa el comportamiento de los casos de uso, generalmente uno en cada ciclo de desarrollo

Corresponde resaltar también que, donde es posible, se utiliza el lenguaje UML para la especificación de las actividades.

A.2 Planificación y Elaboración

El objetivo de esta etapa es obtener un conocimiento de la realidad del problema, definir qué es lo que se debe hacer y planificar la etapa siguiente.

La figura A-1 presenta un cuadro donde se muestran las distintas actividades que componen esta etapa.

Para cada actividad se indica su nombre, una breve descripción y en los casos que corresponde se mencionan los elementos de UML usados para su especificación.

Actividad	Descripción
Relevamiento de la Realidad	Investigación del dominio del problema, para obtener una descripción de la realidad. Es posible representar de forma particular los procesos fundamentales.
Definición del Problema	Determinación clara y precisa del problema a resolver.
Definición de Requerimientos	Especificación de los requerimientos funcionales del producto a desarrollar.
Casos de Uso (alto nivel)	Identificación de los actores y casos de usos del sistema, realizando una especificación de los mismos en alto nivel.
Modelo Conceptual (preliminar)	Identificación de los conceptos relevantes del dominio de la aplicación junto con sus relaciones.
Arquitectura (preliminar)	Identificación de subsistemas de menor complejidad y tamaño. Definición de alto nivel de la arquitectura lógica del sistema.
Clasificación y Asignación	Ordenación de los casos de usos y asignación de ellos a los distintos ciclos de desarrollo.

Fig. A-1: Actividades de planificación y elaboración

A.3 Construcción

En este punto es donde se produce la iteración en el proceso de desarrollo. Consiste en realizar un ciclo de desarrollo completo para cada caso de uso identificado en la etapa anterior. Qué caso de uso considerar en cada ciclo se estableció en la etapa anterior en la actividad de Clasificación y asignación.

Para cada ciclo de desarrollo hay actividades vinculadas con el análisis, el diseño y la implementación.

Las actividades de análisis profundizan el estudio del caso de uso y las operaciones del sistema, limitándose a investigar *qué* deben realizar estas operaciones, y no *cómo* lo deben realizar.
Las actividades de diseño crean una solución basada en objetos que intercambian mensajes y realizan las operaciones definidas en la etapa anterior.
La actividad de implementación organiza el código fuente y los ejecutables.

A continuación se muestran las actividades a desarrollar en cada uno de los ciclos de desarrollo.

La figura A-2 presenta un cuadro donde se muestran las distintas actividades vinculadas con el análisis.

Para cada actividad se indica su nombre, una breve descripción y los elementos de *UML* usados para su especificación.

Actividad	Descripción
Casos de Uso Expandidos	Especificación detallada de los casos de uso.
Modelo Conceptual	Refinamiento del modelo conceptual preliminar, mostrando en detalle los conceptos relevantes para el caso de uso a tratar y sus relaciones.
Eventos del Sistema	Identificación de los eventos generados por los actores sobre el sistema durante el transcurso del caso de uso tratado.
Contratos de Software	Especificación de las operaciones internas al sistema que se disparan con la ocurrencia de eventos del sistema. Para cada operación se indica el estado del sistema antes y después de su ejecución (precondiciones y postcondiciones).
Modelo de Estados	Especificación de los estados por los que pasan los elementos durante su ciclo de vida y los eventos que disparan las transiciones entre estos.

Fig. A-2: Actividades de análisis dentro de un ciclo de desarrollo

La figura A-3 presenta un cuadro donde se muestran las distintas actividades vinculadas con el diseño dentro de cada ciclo de desarrollo.

Para cada actividad se indica su nombre, una breve descripción y los elementos de *UML* usados para su especificación.

Actividad	Descripción
Diseño de Interacciones	Determinación de qué entidades de software colaboran (y en que forma) para resolver las operaciones del sistema. Asignación de responsabilidades.
Diseño de Colaboraciones	Modelado de las clases de objetos (junto con sus relaciones) que participan en la resolución de cada operación del sistema.
Arquitectura Lógica	Refinamiento de la arquitectura propuesta en la etapa anterior, representando también los nuevos elementos que se generaron en el diseño.
Arquitectura Física	Determinación de la topología de la aplicación, o sea la distribución de elementos en los componentes físicos.
Modelo de Datos	Identificación de los objetos persistentes y determinación de la estructura lógica y física de la base de datos en que se almacenarán. También se determina la forma de comunicación (*framework* de persistencia).

Fig. A-3: Actividades de diseño dentro de un ciclo de desarrollo

La figura A-4 presenta un cuadro donde se muestran la actividad vinculada con la implementación dentro de cada ciclo de desarrollo.

Para ella se indica su nombre, una breve descripción y el elemento de *UML* usado para su especificación.

Actividad	Descripción
Modelo de Implementación	Organización del código fuente implementado. Configuración de ejecutables y otros elementos necesarios para la ejecución (bibliotecas, por ejemplo).

Fig. A-4: Actividades de diseño dentro de un ciclo de desarrollo

Apéndice B
Resumen de las Principales Metodologías Estudiadas

Este apéndice pretende describir a grandes rasgos las tres metodologías de desarrollo orientado a objetos que fueron encontradas por los autores y de las cuales se tomaron aportes.

B.1 Unified Process

Si bien no se tomaron muchos aportes de las actividades específicas de este proceso de desarrollo, esta metodología, propuesta por Booch, Rumbaugh y Jacobson, fue la principal fuente de aportes en cuanto a la estructura de la metodología propuesta debido a sus características principales.

B.1.1 Características Principales

El Unified Process (en adelante: UP) presenta tres principios fundamentales:

- El desarrollo se apoya en los casos de uso, ya que son esenciales para la correcta captura de los requerimientos del sistema y también sirven como guía para el desarrollo.
- El software se desarrolla de forma iterativa, permitiendo un *feedback* permanente que enriquece el mismo proceso y elimina gran parte de sus riesgos. La iteración se realiza fundamentalmente sobre los casos de uso.
- Modelado preciso de la arquitectura del sistema, que contribuye de forma esencial a la claridad, el entendimiento y por tanto a la facilidad de desarrollo, mantenimiento y calidad general del software. Se sugiere -aunque no se obliga a- realizar el modelado en el lenguaje UML.

Estos tres principios fundamentales resultan de gran importancia para el proyecto, ya que son los mismos principios que guían la metodología del proyecto del Taller 1999. De esto se desprende que no hay diferencias notorias en cuanto a la filosofía de los métodos.

Otros principios esenciales son: gestión de requerimientos, verificación continua de la calidad y control apropiado de los cambios.

Por otra parte, el UP presenta dos conceptos esenciales: fases y flujos de trabajo. Las fases corresponden con la división en el tiempo, es decir los cortes transversales al proceso. Los flujos de trabajo corresponden con los agrupamientos de actividades que se van realizando a lo largo de todas las fases, o sea que representan los cortes longitudinales.

B.1.2 Fases

El UP identifica cuatro fases principales, dentro de las cuales se realizan varias iteraciones. Cada fase finaliza con un *hito* o *mojón* del proceso, que consiste en una gran revisión de lo producido en la fase y una profunda planificación de lo que se va a realizar en la siguiente.

Las fases son: *comienzo*, *elaboración*, *construcción* y *transición*.

B.1.2.1 Fase de Comienzo

La fase de comienzo implica la idea del proyecto y la definición del alcance. Fundamentalmente, se trata de realizar un análisis de factibilidad que, con su correspondiente evaluación, permita moverse con cierta tranquilidad a las siguientes fases, aunque esta factibilidad seguirá siendo analizada principalmente durante la fase de elaboración (en las otras fases, ya debería haber una gran seguridad de que el plan es posible, ya que los costos empiezan a trepar y se incrementan los riesgos).

Por otra parte, ocurre que el trabajo de la fase de comienzo se centra en el modelado del negocio (si se realiza) y el análisis de requerimientos, ya que es vital conocer cuanto antes el alcance del proyecto. Esta actividad es indudablemente la tarea más importante de esta primera fase y debe conseguir identificar aproximadamente el 50% de los casos de uso y describir o analizar alrededor del 10%, en particular aquellos que se reconozcan como más importantes, críticos o peligrosos.

Al fin de la fase se realiza la primera revisión del proyecto, ya que se alcanza el primer mojón: Evaluación de objetivos.

B.1.2.2 Fase de Elaboración

El concepto de elaboración refiere a la planificación de recursos y especificación de características y diseño.

En esta fase se debe lograr la captación casi total de los requerimientos, la identificación de los riesgos, incorporación de nuevos integrantes (necesarios para la fase de construcción) y –fundamentalmente– la generación de una arquitectura que parezca estable y firme para pasar a la construcción, lo que puede realizarse durante más de una iteración.

En resumidas cuentas, el mayor esfuerzo de esta fase está puesto en dos grandes actividades: por un lado, el análisis de requerimientos, y por el otro el de análisis y diseño.

Al final de esta fase se realiza la segunda evaluación de importancia, donde se considera lo principalmente el establecimiento de una propuesta firme de arquitectura y la factibilidad del proyecto.

B.1.2.3 Fase de Construcción

La etapa de construcción es la que efectivamente realiza el producto, llegando al punto de estar listo para la entrega al cliente.

Esta fase se suele caracterizar por realizar un gran número de iteraciones, y en cada una se ataca un conjunto de casos de uso, realizando una mini-cascada para desarrollarlo por completo. Esto significa que la arquitectura se sigue refinando a medida que los casos de uso le imponen ciertos cambios.

Al finalizar esta fase se realiza una tercera revisión, en la que se evalúa si la primera versión del producto está lista para ser distribuida.

B.1.2.4 Fase de Transición

En la fase de transición se siguen todos los pasos necesarios para que el producto llegue efectivamente al cliente y logre su aprobación. Como ejemplo de estos posibles pasos, se puede resaltar la propia manufactura del producto, la distribución, capacitación, etc.

En esta fase se somete la versión inicial del producto a un testeo beta o a un testeo de aceptación. Adicionalmente, se puede someter a testeo alfa o a validación de terceros.

La evaluación final comprende dos partes, ya que se debe considerar que se está terminando la fase pero también el ciclo de desarrollo. En la misma debe verificarse que se logró la aceptación del usuario y por el otro que se cumplieron con las metas y objetivos establecidos para el proyecto.

B.1.3 Flujos de Trabajo

Existen dos clases de flujos de trabajo: los principales y los de soporte. Como su nombre lo indica, los principales son aquellos que resultan centrales en la realización de un proyecto, o más precisamente son aquellos que dependen únicamente de ellos mismos. Estos flujos incluyen: modelado del negocio, análisis de requerimientos, análisis y diseño, implementación, testeo y despliegue.
Los flujos de soporte son aquellos que son necesarios como soporte de los otros, es decir que no tienen un valor en sí mismos, pero contribuyen de forma esencial al éxito de un proyecto. Estos son: configuración del software, gestión y entorno.

B.1.3.1 Modelado del Negocio

Esta actividad trata de realizar un modelado de requerimientos mediante el empleo de casos de uso pero desde el punto de vista del negocio global, o sea que sin considerar las particularidades de los diferentes actores o distintos componentes.

B.1.3.2 Análisis de Requerimientos

Este flujo de trabajo resulta de vital importancia para el desarrollo del proyecto, ya que la correcta captación de los requerimientos es absolutamente imprescindible para realizar un sistema que realice lo esperado.
El UP propone sólo la realización de modelo de casos de uso, considerándolo enteramente suficiente para realizar esta actividad, aunque no descarta el empleo previo de la captación y descripción habitual de los requerimientos, sobre todo a través de la norma IEEE 830.

B.1.3.3 Análisis y Diseño

A grandes rasgos, puede decirse que la tarea de análisis intenta desenterrar todo lo que puede ser importante para el sistema. Asimismo, se realiza un modelado conceptual, reconociendo las entidades que forman parte del sistema y las relaciones entre ellas.
Este análisis se realiza principalmente en la fase de elaboración, ya que requiere un cierto conocimiento de la realidad del proyecto.
En la etapa de diseño, todas las actividades se apoyan en el concepto fundamental de a*rquitectura* a nivel físico y lógico.

B.1.3.4 Implementación

A partir de la arquitectura del sistema, y atacando cada vez un caso de uso en particular o un conjunto medianamente reducido de ellos, es posible crear el código fuente, scripts, y todo lo que sea necesario para llevar a cabo los mismos. La idea es generar las clases y componentes identificados en el diseño.

B.1.3.5 Testeo

Tal como se mencionó previamente, una característica importante del UP es la de buscar permanentemente la calidad a través de un testeo intenso y constante de todo lo producido.

Por otra parte y a grandes rasgos, el testeo se puede dividir en dos modalidades: testeo de integración y testeo de sistema. El testeo de integración busca evaluar la integración de dos o más componentes, frecuentemente dentro de una versión generada. Por el contrario, el testeo de sistema busca evaluar el sistema en su totalidad, generalmente buscando casos extremos, distintas configuraciones, etc.

B.1.3.6 Despliegue

El propósito de este flujo de trabajo es producir una versión final del software (release) y hacerla llegar a los usuarios.
Cubre varias actividades, como ser:

- Armar la versión final
- Empacar el software
- Distribuir el software
- Instalar el software
- Asistir y asesorar a los usuarios

B.1.3.7 Gestión de Configuraciones del Software

La gestión de configuraciones del software se encarga de mantener las versiones de los distintos productos y las dependencias entre ellos. Además, se encarga de analizar el impacto potencial de los cambios y su seguimiento en caso de que se concrete.

B.1.3. 8 Gestión

La gestión del proyecto se encarga de balancear los objetivos de los distintos involucrados, manejar el riesgo y superar las dificultades para llegar a realizar exitosamente un producto que satisfaga al cliente (el que encarga el software) y al usuario. Esto se logra a través de una cuidadosa planificación que se realiza con la base de un plan de proyecto y planes particulares para cada iteración.

B.1.3.9 Entorno

Este flujo de trabajo pretende proveer al equipo de desarrollo de las condiciones necesarias para realizar su tarea, como por ejemplo: instalar el software necesario, dar guías para realizar ciertos procedimientos, etc.

B.2 Process Patterns

Los patrones de proceso no son un proceso de desarrollo en sí mismo. En cambio, un patrón de procesos se define como una colección de técnicas generales, acciones y/o tareas (actividades) para desarrollar software orientado a objetos.

B.2.1 Tipos de Patrones de Proceso

Un dato importante es que el enfoque de un patrón de procesos puede verse desde un muy alto nivel hasta un nivel más detallado.
Básicamente existen tres tipos de patrones de procesos, en orden creciente de escala éstos son: patrones de tarea, de etapa y de fase.

- Patrones de proceso de tareas: Este tipo de patrones describe de manera detallada los pasos para realizar una tarea específica, como ser por ejemplo, una revisión técnica.
- Patrones de proceso de etapas: Estos patrones describen los pasos, realizados usualmente de manera iterativa, de una etapa del proyecto. Una etapa del proyecto es una forma de más alto nivel de un patrón de procesos de etapa, la cual está compuesta por varios patrones de proceso de tareas.
- Patrones de proceso de fases: Este tipo de patrones de proceso describe la interacción entre patrones de proceso de etapas de una fase de un proyecto. Un patrón de proceso de fase se compone de dos o más patrones de proceso de etapas.

B.2.2 Elementos de un patrón de proceso

Existen distintos formatos para documentar un patrón de procesos, pero en general se está de acuerdo en que ciertos elementos esenciales deben estar presentes. Sin tener en cuenta el formato particular que se emplee, los siguientes elementos deben ser fácilmente reconocibles en todo patrón de procesos: nombre, problema, contexto, fuerzas, solución, ejemplos, contexto resultante, razón, patrones relacionados y usos conocidos.

B.2.3 El OOSP

El Object Oriented Software process (OOSP) es un proceso de desarrollo de software basado en patrones de proceso.
Está formado por cuatro fases seriales donde cada una está compuesta por etapas iterativas dentro de cada fase.
Los patrones de procesos en la forma de OOSP han sido usados para formar un proceso de desarrollo de proyectos de gran escala usando tecnología orientada a objetos.

B.3 OPEN Process

Object-oriented Process, Enviroment, and Notation (OPEN) es una metodología de tercera generación que abarca el ciclo de vida completo, es de dominio público y enfocado al proceso. Está diseñada principalmente para el desarrollo de aplicaciones orientadas a objetos y basadas en componentes.

OPEN está definido como un marco de trabajo, conocido como OPF (Open Process Framework). Es un metamodelo de proceso del cual se puede generar un modelo de proceso específico para cada organización. Cada una de estas instancias es creada eligiendo actividades, tareas y técnicas específicas (tres de los principales meta-niveles) y configuraciones específicas para cada proceso.

B.3.1 Características Técnicas

Aunque existen varias meta-clases en el OPF, las mismas recaen básicamente en cinco grupos: unidades de trabajo, productos de trabajo, productores de trabajo, soporte de etapas y soporte de lenguajes. Esto conforma un conjunto de componentes, los cuales pueden combinarse para generar las distintas instancias específicas de OPEN. La organización desarrolladora elige qué componentes poner y cómo colocarlos. De esta manera se puede crear un proceso más bien iterativo incremental (IIP) o más bien orientado a un desarrollo en cascada.
De esta forma el hincapié de OPEN está puesto sobre la interacción entre productores (típicamente personas), lo que ellos hacen (unidades de trabajo) y lo que ellos producen (productos de trabajo).

Más allá de estos componentes, las etapas y los lenguajes proveen un soporte adicional. Existen varios tipos de etapas como lo son las fases, ciclos de vida y mojones que son usados en el proceso de desarrollo. Por otro lado los lenguajes, ya sean naturales, de modelado o de codificación son necesarios para su uso como herramientas las cuales sirven en la documentación del proyecto. OPEN soporta UML, OML y cualquier otra notación OO.

B.3.2 Componentes del Marco de Trabajo

El OPF provee de una librería de componentes predefinidos que pueden ser instanciados, adaptados y extendidos para satisfacer las distintas necesidades de los distintos proyectos.
Como se vio antes, los mismos son: productos de trabajo, lenguajes, productores, unidades de trabajo, etapas.

B.3.2.1 Productos de trabajo

El objetivo primario de cualquier proceso de software es el de desarrollar uno o más productos relacionados. Cualquier unidad significativa obtenida en el proceso de desarrollo se llama *producto de trabajo*. Uno o más *productores* desarrollan un *producto de trabajo* durante una o más *unidades de trabajo.*

B.3.2.2 Lenguajes

Un lenguaje es un medio de documentar una unidad de trabajo. Por ej. modelos de casos de uso son documentados con UML u OML.

B.3.2.3 Productores

Un productor es cualquier entidad que produzca, ya sea directa o indirectamente, versiones de uno o más productos de trabajo. OPEN distingue entre productores directos, aquellas personas o roles representados por personas junto a las herramientas que ellas utilizan, y productores indirectos. Productores indirectos consisten en grupos de personas u organizaciones

B.3.2.4 Unidades de Trabajo

El OPF define como unidad de trabajo una operación realizada por un productor durante el desarrollo y del cual se obtiene una unidad de producto.
El OPF provee las siguientes clases predefinidas de unidades de trabajo: tarea, técnica, ejecución de una tarea y actividad.

B.3.2.5 Etapas

Una etapa es formalmente, una duración o punto de tiempo identificado. OPEN hace una clasificación de etapas según este criterio: existen etapas con duración que intuitivamente se puede decir que son aquellas que comprenden un período de tiempo relativamente considerable; en cambio una etapa instantánea tiene una duración muy corta y sirve para crear una "marca" en el proceso.
Etapas con duración: ciclo, fase, flujo de trabajo, proyecto, construcción, release y despliegue.
Etapa instantánea: mojón.

B.3.3 Uso del Marco de Trabajo

El marco de trabajo OPEN no puede ser usado "as is". Primero debe ser instanciado para crear un proceso de desarrollo que se adecue a las necesidades de cada organización. El proceso requiere entonces unas guías de proceso:

- Instanciar la librería de clases para crear los componentes
- Escoger los mejores componentes
- Adaptar los componentes en detalle
- Extender la librería de componentes