



Integración de una Herramienta CASE en un Entorno Académico Colaborativo para la Enseñanza de Ingeniería de Software

Autor: Ing. Nicolás Battaglia

Director: Dr. Carlos G. Neil

Co-Directores:

Dr. Alejandro Fernández

Esp. Gabriela Milanese

Tesis presentada para obtener el título de
Magister en Tecnología Informática

Noviembre, 2017.

Resumen:

La Ingeniería de Software (**IS**) propone un marco de trabajo sustentado en modelos, que permiten una representación abstracta de un sistema informático. Esta labor es sumamente compleja, principalmente, para quienes están comenzando su aprendizaje y no cuentan con la experiencia suficiente en el campo profesional.

El proceso de enseñanza y aprendizaje del modelado de software resulta aún más complejo si no se dispone de las herramientas e infraestructura necesarias. Sumado a esto, la necesidad creciente de realizar un trabajo grupal para avanzar en la resolución de problemas que plantean los sistemas cada vez más complejos; por otro lado, la creciente evolución en materia de Tecnologías de la Información y Comunicación (**TIC**) propician un entorno ideal para contextualizar el trabajo colaborativo incluso, también, enfocado en la enseñanza y aprendizaje de modelado de software.

Las actuales herramientas de modelado que se utilizan para asistir a la **IS** no reúnen las características necesarias que requiere el proceso de enseñanza y aprendizaje de esta disciplina. Es sumamente importante que el cuerpo docente y los alumnos dispongan de un entorno en el que puedan interactuar y colaborar durante el proceso de modelado como, así también, durante el proceso de evaluación y seguimiento de actividades relacionadas.

El objetivo de este trabajo es diseñar e implementar un modelo de aprendizaje colaborativo asistido por computadora integrado con una herramienta de modelado de software, para utilizar durante el proceso de enseñanza, aprendizaje y evaluación de esta actividad, como parte intrínseca del proceso propuesto por la **IS**.

Palabras claves:

Enseñanza y Aprendizaje, Evaluación Colaborativa, Herramientas CASE, Ingeniería de Software, Trabajo Colaborativo

Dedicatoria

A mi abuelo. Dedico y agradezco por guiarme en este camino.

Dedico de manera especial a toda mi familia, que me acompaño y me contuvo durante este largo proceso. En especial a Mama, Papa, Memi, Maru, Jesi, Pilar, Macarena y Catalina.

Agradecimientos

Quiero agradecer inicialmente a mi director Carlos Neil, y a mis codirectores Alejandro Fernández y Gabriela Milanese.

Principalmente a Carlos quien me acompaño desde el principio, brindándome todo su apoyo y plena confianza, guiarme y aconsejarme siempre.

A Alejandro y Gabriela ya que sin ellos hubiese sido imposible realizar este trabajo multidisciplinario.

Mi agradecimiento también va dirigido a Marcelo De Vincenzi, quien día a día renueva su confianza en lo académico y profesional y me brinda su constante apoyo en las actividades que realizo en la Universidad Abierta Interamericana.

Extiendo mi agradecimiento a quienes participaron en el proyecto de investigación que culminó exitosamente con esta tesis. A mis amigos, colegas y alumnos que participaron y colaboraron para culminar un ciclo exitoso de investigación y trabajo.

Reconocimientos

A todos mis amigos y colegas, que me ayudan a mejorar día a día como persona y profesional.

Imposible nombrar a todos.

Jurado

Dra. Claudia Pons

Universidad Abierta Interamericana

Dr. Fernando Asteasuain

Universidad Abierta Interamericana

Dr. José Matías Rivero

Universidad Nacional de La Plata

Índice General

Capítulo 1	1
Introducción	1
1.1. Problemas y Soluciones.....	3
1.1.1. Trabajo Multidisciplinario	4
1.1.2. CSCL para el Modelado de Software.....	4
1.1.3. Seguimiento, Evaluación y Autoevaluación de la IS en Entornos Colaborativos	5
1.2. Nuestra Propuesta:	5
1.2.1. uCASE-CL	6
1.2.2. UAI Case	6
1.3. Objetivos de la Tesis.....	7
1.4. Contribuciones Principales.....	8
1.4.1. Trabajos Presentados Vinculados con la Tesis.....	8
1.5. Estructura General de la Tesis.....	10
1.5.1. Capítulos.....	10
1.5.2. Anexos.....	12
Capítulo 2	13
Trabajo Colaborativo e Innovación Tecnológica	13
2.1. Introducción	13
2.2. Trabajo Colaborativo.....	14
2.2.1. Ventajas y Desventajas del Uso de Entornos Colaborativos.....	16
2.2.2. Taxonomías de Groupware	17
2.2.2.1. Matriz Espacio-Temporal	18
2.2.2.2. Modelo de las 3C.....	18
2.2.2.3. Del Dominio.....	19
2.2.2.3.1. Sistemas de Mensajería	20
2.2.2.3.2. Editores Grupales.....	20
2.2.2.3.3. Sistemas de Conferencia.....	21
2.2.2.3.4. Sistemas de Apoyo a la Toma de Decisiones	23
2.2.3. Awareness	25
2.2.4. Ingeniería de Colaboración	28
2.2.4.1. La Capa de Proceso	29
2.2.4.2. Patrones de Colaboración.....	30
2.2.4.3. Thinklets	32
2.3. Innovación Tecnológica.....	35
2.3.4. Innovación Tecnológica en la Educación	37

Resumen.....	39
Capítulo 3	40
Aprendizaje en Entornos Virtuales Colaborativos	40
3.1. Introducción	40
3.2. Concepto de Aprendizaje Colaborativo	42
3.2.1. Teoría del Constructivismo Social	43
3.2.2. La Interacción Social y su Influencia en el Desarrollo	46
3.2.2.1. La Zona de Desarrollo Próximo	48
3.3. Los Entornos Virtuales de Enseñanza y Aprendizaje	50
3.3.1 Cambios en los Roles del Docente y del Alumno	62
3.3.2 Variables Críticas en los Entornos Virtuales de Enseñanza	63
3.4. Aprendizaje Colaborativo Asistido por Computadora (CSCL)	64
3.5. Evaluación en Entornos Virtuales de Enseñanza	70
3.5.1. Autoevaluación	73
3.5.2. Evaluación Enciclopédica	74
3.5.3. Evaluación Colaborativa	75
3.6. Resumen.....	76
Capítulo 4	77
Herramientas CASE	77
4.1. Introducción	77
4.2. Concepto de herramientas CASE	78
4.2.1. Características	79
4.3. Modelado UML.....	81
4.3.1. UML	82
4.4. Caracterización de una Herramienta CASE para Modelado	84
4.4.1. Método de Evaluación	85
4.4.2. Nuestra Propuesta de Evaluación	89
4.4.2.1. Definición de Criterios de Evaluación	90
4.4.2.1.1. Extensivo al Proceso Completo de Desarrollo	92
4.4.2.1.2. Adaptabilidad y Flexibilidad	93
4.4.2.1.3. Amigable para el Usuario	94
4.4.2.1.4. Producir Documentación	94
4.4.2.1.5. Trabajo Online y Colaborativo	95
4.4.2.1.6. Manejo de OCL.....	97
4.4.2.1.7. Ingeniería Inversa y Generación de Código	98
4.4.2.1.8. Apoyo Metodológico.....	98
4.4.2.2. Ficha de Evaluación	99

4.4.3. Proceso de Evaluación de Herramientas CASE	100
4.4.3.1. Puntuación de las Herramientas.....	100
4.4.3.2. Características Deseables.....	102
4.5. Resumen.....	104
Capítulo 5	105
uCASE-CL: Enseñanza y Aprendizaje Colaborativo de la IS en Entornos Virtuales Ubicuos.....	105
5.1. Introducción	105
5.2. Enseñanza del Modelado UML Utilizando Herramientas CASE.....	106
5.2.1. Enseñanza de UML.....	107
5.2.2. Ventajas y Desventajas de Utilizar Herramientas CASE.....	109
5.4. Nuestra propuesta: uCASE-CL.....	110
5.5. Implementación del modelo	118
5.5.1. Dimensión Tecnológica	118
5.5.2. Dimensión Pedagógica	119
5.5.3. Dimensión Psicológica.....	119
5.6. Resumen.....	121
Capítulo 6	123
UAI Case: Implementación del modelo uCASE-CL	123
6.1. Introducción	123
6.3. UAI Case: Nuestra Propuesta	123
6.3.1. Contexto.....	124
6.3.2. Características Principales.....	124
6.3.3. Dimensión Tecnológica	125
6.3.4. Dimensión Pedagógica	126
6.3.4.1. Modelado UML colaborativo libre.....	126
6.3.4.2. Modelado UML colaborativo coordinado.....	126
6.3.4.3. Evaluación colaborativa de modelos UML.....	126
6.3.4.4. Autoevaluación colaborativa de grupos de trabajo utilizando rubricas.....	126
6.3.5. Dimensión Psicológica.....	127
6.3.5.1. Modelado UML colaborativo coordinado.....	127
6.3.5.1.1. Descripción.....	127
6.3.5.1.2. Evaluación	128
6.3.5.1.3. Interacción y Transformación	129
6.3.5.2. Evaluación colaborativa de modelos UML.....	131
6.3.5.2.1. Descripción.....	131

6.3.5.2.2. Evaluación	131
6.3.5.2.3. Interacción y Transformación	133
6.3.5.3. Autoevaluación colaborativa de grupos de trabajo utilizando rubricas.....	134
6.3.5.3.1. Descripción.....	134
6.3.5.3.2. Evaluación	135
6.3.5.3.3. Interacción y Transformación	136
6.4. Resumen.....	137
Capítulo 7	138
Trabajos Relacionados	138
7.1. Introducción	138
7.2. Resumen de Trabajos Relacionados	138
7.3. Comparación de nuestra Propuesta vs Trabajos Relacionados.....	143
7.3.1. Uso de Herramientas CASE en la Educación	144
7.3.2. Enseñanza y Aprendizaje de la IS en entornos Virtuales Colaborativos	144
7.4. Publicaciones Vinculadas a la Tesis.....	145
7.5. Resumen.....	146
Capítulo 8	147
Conclusiones y Trabajos Futuros.....	147
8.1. Introducción	147
8.2. Resumen e Interpretación Final.....	147
8.3. Contribuciones Principales.....	149
8.4. Trabajos Futuros	149
Anexo I.....	151
Revisión, Pruebas y Puntuación de Diferentes Herramientas CASE.....	151
I.1. Introducción	151
I.2. Análisis de Herramientas	151
I.2.1. Enterprise Architect	151
I.2.2. Gliffy	154
I.2.3. yEd	156
I.2.4. ArgoUML	158
I.2.5. StarUML.....	160
I.2.6. VisualParadigm.....	162
I.3. Resumen.....	165
Anexo II.....	166
Metodología para Diseñar Actividades Colaborativas.....	166

II.1. Introducción	166
II.2. Propuesta	166
II.2. Resumen.....	169
Anexo III.....	170
UAI Case: Prototipo	170
III.1. Introducción	170
III.2. Plataforma Tecnológica Ubicua	170
III.2.1. Cliente (Frontend)	171
III.2.2. Servidor (Backend)	173
III.2.2.1. Arquitectura	173
III.2.2.2. Seguridad	174
III.2.2.3 Trabajo en Tiempo Real	175
III.2.3. Servidor CASE	175
III.2.4. DBAAS.....	176
III.3. Plataforma CSCL	176
III.3.1. Trabajo Colaborativo.....	177
III.3.1.1. Espacio de Trabajo para Grupos de Alumnos	177
III.3.1.2. Espacio de Trabajo para Grupos de Docentes	178
III.3.1.3. Espacio de Trabajo para un Curso.....	179
III.3.1.4. Sesiones de Chat.	180
III.3.1.3. Tareas (ToDo).....	181
III.3.1.4. Mensajería Interna.....	182
III.4. Herramienta CASE	184
III.4.1. Características Principales.....	184
III.4.3. Actividades Colaborativas	185
III.4.4. Evaluación de UAI Case	186
Acrónimos	188
Referencias.....	190

Índice de Figuras

Figura 1.1. Conceptos principales	2
Figura 2.1. Taxonomía de las 3C	19
Figura 2.2. Aplicación MIRC para Mensajería Instantánea	20
Figura 2.3. Editor de texto Google DOC con 6 colaboradores	21
Figura 2.4. Editor Realttimeboard	21
Figura 2.5. Google Hangouts	22
Figura 2.6. Skype	23
Figura 2.7. MeetingSphere (Brainstorming)	24
Figura 2.8. MeetingSphere (Ranking)	24
Figura 2.9. MeetingSphere (Discussion)	25
Figura 3.1. Red de Aprendizaje	58
Figura 3.2. Variables Críticas de la Formación en Red	63
Figura 3.3. Modelo CSCL	66
Figura 3.4. Ejemplo de rúbrica.	74
Figura 4.1. Capas de la IS (Extraído de [PRE10])	79
Figura 4.2. Framework RUP (Extraída de [IBM17])	88
Figura 4.3. Desarrollo Iterativo e Incremental con RUP (Extraída de [KRU04])	88
Figura 4.4. Tendencia de búsqueda entre 01/2004 y 01/2017	96
Figura 4.5. Tendencia de búsqueda entre 01/2012 y 01/2017	96
Figura 4.6. Tendencia de búsqueda entre 01/2016 y 01/2017	96
Figura 4.7. Tendencia en Google sobre "ONLINE UML TOOL"	97
Figura 5.1. Diagrama de clases en la vista de análisis.....	108
Figura 5.2. Diagrama de clases en la vista de diseño.....	109
Figura 5.3. Modelo uCSCL de Coto y Collazos [CCR16]	112
Figura 5.4. Modelo uCSCL	113
Figura 5.5. Modelo U-CSCL propuesto en [CCR16]	114
Figura 5.6. Enfoque general uCASE-CL.....	115
Figura 5.7. Modelo UCASE-CL detallado	117
Figura 6.1. Modelo de despliegue de UAI Case	125
Figura 6.2. Casos de interacción para el proceso de modelado colaborativo.....	129
Figura 6.3. DS de dinámica grupal para el proceso de modelado cooperativo.	130
Figura 6.4. Casos de interacción para el proceso de evaluación colaborativa.	133
Figura 6.5. DS de dinámica grupal para el proceso de evaluación colaborativa.	133
Figura 6.6. DOC para el proceso de evaluación colaborativa	134
Figura 6.7. DS de dinámica grupal para el proceso de autoevaluación colaborativa. ...	136
Figura 6.8. DOC para autoevaluación colaborativa	137
Figura I.1. Logo Enterprise Architect.....	151
Figura I.2. Interfaz de usuario de Enterprise Architect	154
Figura I.3. Logo Gliffy.....	154
Figura I.4. Interfaz de usuario de Gliffy.....	156
Figura I.5. yEd	156
Figura I.6. Interfaz de usuario de yED	158
Figura I.7. ArgoUML	158

Figura I.8. Interfaz de usuario de ArgoUML	160
Figura I.9. Logo StarUML	160
Figura I.10. Interfaz de usuario de StarUML	162
Figura I.11. Logo Visual Paradigm	162
Figura I.12. Interfaz de usuario Visual Paradigm	164
Figura II.1. Modelo para el diseño de actividades colaborativas.....	168
Figura III.1. Despliegue de UAI Case.....	171
Figura III.2. Arquitectura Servidor UAI Case	173
Figura III.3. Autenticación	175
Figura III.4. Autorización	175
Figura III.5. Espacio de Trabajo del Grupo de Alumnos	177
Figura III.6. Proyectos Colaborativos del Grupo de Trabajo	178
Figura III.7. Espacio de Trabajo para un Grupo de Docentes.....	179
Figura III.8. Espacio de Trabajo del Curso desde la Perspectiva el Alumno.....	179
Figura III.9. Espacio de Trabajo del Curso desde la Perspectiva de Docente.....	180
Figura III.10. Salón de chat en UAI Case.....	181
Figura III.11. Lista de Tareas Pendientes.....	182
Figura III.12. Registro de Eventos sobre una Lista de Tareas	182
Figura III.13. Redacción de un Mensaje Interno	183
Figura III.14. Información Contextual sobre Mensaje Recibido.....	183
Figura III.15. Información Contextual sobre Mensaje Leído.....	184
Figura III.16. Centro de Mensajería.....	184

Índice de Tablas

Tabla 2.1. Matriz Espacio-Temporal.....	18
Tabla 2.2. Elementos de awareness relativo al presente	27
Tabla 2.3. Elementos de awareness relativo al pasado	28
Tabla 2.4. Patrones de Colaboración	31
Tabla 2.5. Ejemplo de thinklet RichRelation	33
Tabla 2.6. Ejemplo Thinklet LeafHopper	35
Tabla 3.1. Características y Comparación entre Formación presencial y virtual.....	54
Tabla 3.2. Ventajas y Desventajas de la Educación Virtual.....	55
Tabla 4.1. Taxonomía del estándar UML	84
Tabla 4.2. Ficha de Evaluación para Herramientas CASE.....	100
Tabla 4.3. Resumen comparativo sobre Herramientas de Modelado.....	102
Tabla 6.1. Tabla CCD para el proceso de evaluación colaborativa.	132
Tabla 6.2. Tabla CCD para el proceso de autoevaluación colaborativa.....	136
Tabla I.1. Evaluación de Enterprise Architect	153
Tabla I.2. Evaluación Giffy	156
Tabla I.3. Evaluación yED	158
Tabla I.4. Evaluación ArgoUML	160
Tabla I.5. Evaluación StarUML.....	162
Tabla I.6. Evaluación Visual Paradigm.....	164
Tabla II.1. Relación entre MDAC Y TA	169
Tabla III.1. Tabla de evaluación UAI Case.....	186

Capítulo 1

Introducción

La innovación constante de las Tecnologías de Información y Comunicación (**TIC**) ha modificado notablemente diferentes aspectos sociales y culturales, y la educación es uno de los ámbitos que más impacto y aportes recibió de esta disrupción tecnológica. La posibilidad de contar con dispositivos que permiten acceder a la información en cualquier momento del día, que poseen herramientas tales como *chats*, *foros*, *blogs* y *el acceso a redes* que promueven la interacción entre personas, posibilita un cambio de enfoque en las metodologías de enseñanza y aprendizaje [LAV12].

Tal es así que el trabajo colaborativo, las **TIC** y la enseñanza se integran para crear entornos de aprendizaje colaborativo asistido por computadora (**CSCL**). Esta integración se basa en el impacto social del trabajo colaborativo y las **TIC** en la educación. Por un lado, el trabajo colaborativo apoyados en el concepto de Zona de Desarrollo Próximo (**ZDP**) propuesto por Vygotski a principios del siglo XX y, por el otro lado, las **TIC** en el marco de las propuestas *de e-learning* o *blended learning*.

El desarrollo creciente de las **TIC** junto con el concepto de trabajo colaborativo, conforman los entornos de Trabajo Colaborativo Asistido por Computadora (*Computer Supported Collaborative Work*, **CSCW**). Este concepto, integrado en entornos de enseñanza y aprendizaje colaborativo, da origen a los entornos de Aprendizaje Colaborativos Asistidos por Computadora (*Computer-Supported Collaborative Learning*, **CSCL**). Ver figura 1.1.

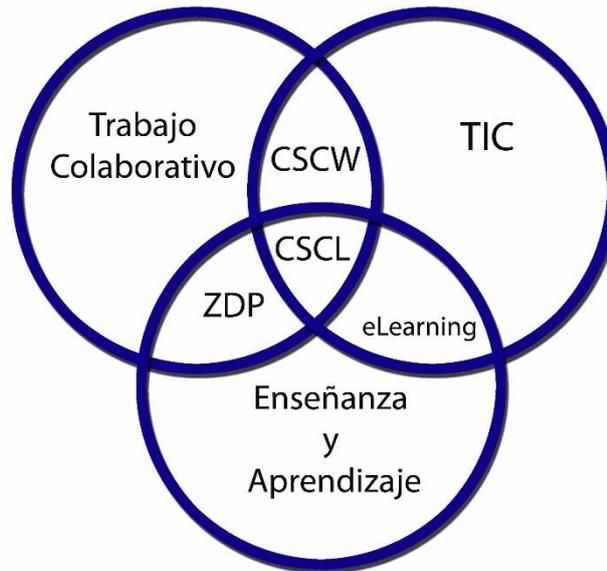


Figura 1.1. Conceptos principales

El estudio teórico de la Ingeniería de Software (**IS**) no es suficiente para comprender y resolver los problemas de cooperación y colaboración que surgen durante el desarrollo de un proyecto informático. Los estudiantes suelen centrar su esfuerzo en aspectos técnicos y asumen que los inconvenientes vinculados al trabajo en equipo no impactarán en el proyecto [BOU05]. En virtud de resolver el problema planteado previamente, Daniele et al. [DAN10] proponen, para mejorar los aspectos comunicacionales, la integración de las plataformas **CSCL** con entornos especializados en la resolución de problemas prácticos y técnicos en un proyecto de desarrollo de software; además, esta propuesta potencia el aprendizaje de la **IS** y permite el desarrollo de competencias profesionales y sistemáticas en el desarrollo de software.

Por otro lado, la **IS** plantea áreas de conocimiento específico que requieren del uso de herramientas complejas tal como sucede, por ejemplo, durante el modelado de software mediante el uso de herramientas específicas denominadas **CASE**. Sin bien la **IS** tradicionalmente ha apoyado las iniciativas de modelado con herramientas que permiten la edición de diferentes tipos de diagramas (incluida la gama de diagramas **UML**) [ZAP07], el uso de **UML** en el modelado de software puede resultar ambiguo en algunos casos, sobre todo cuando éste está vinculado a un proceso de enseñanza y aprendizaje [FLI05]. El desafío de los docentes es múltiple: a la complejidad que implica la ambigüedad en la

interpretación de los modelos se le suma la dificultad en la enseñanza del uso de herramientas profesionales (costosas en algunos casos) que, además, poseen funcionalidades ajenas al proceso de enseñanza, como ser, por ejemplo, aquellas que permiten la evaluación de resultados en problemas de modelado de software.

De la misma manera que sucede en los procesos presenciales o tradicionales de enseñanza, en los entornos **CSCL** la evaluación cumple un papel muy importante. Según el estudio realizado en un relevamiento previo [BAT16c], en el modelo de enseñanza y aprendizaje de la **IS** mediado por tecnología, los estudiantes cumplen un papel de mayor protagonismo y responsabilidad y, en consecuencia, los docentes asumen un nuevo rol cuando es el alumno quien adquiere mayor autonomía en su proceso de aprendizaje, a través de entornos virtuales.

En este contexto, la evaluación es entendida como un proceso que promueve el aprendizaje con una finalidad formativa, más que como un proceso de control de resultados. Por consiguiente, un proceso específico de enseñanza y aprendizaje en un entorno virtual colaborativo requiere también de un proceso de evaluación, autoevaluación y seguimiento acorde al paradigma de aprendizaje colaborativo planteado por el concepto **CSCL** [BAT17].

En esta tesis nos enfocaremos en la enseñanza y el aprendizaje del modelado de software y utilizaremos el estándar **UML** como lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Por otro lado, de ahora en más utilizaremos el termino "**CASE**" para referirnos al conjunto de herramientas que brindan soporte a cualquier actividad dentro de la **IS**, incluso las actividades de modelado.

1.1. Problemas y Soluciones

A continuación, se detallarán los principales problemas identificados, con un breve resumen del mismo y la solución propuesta.

1.1.1. Trabajo Multidisciplinario

El diseño, funcionalidad y comportamiento de un sistema deberá estar íntimamente relacionado con el tipo de usuario que lo utilice [HAS04]. Esta afirmación también aplica a las plataformas **CSCL** en las que la interacción entre personas tiene un rol fundamental. Para lograr esto, es necesario el trabajo interdisciplinario entre los campos de la tecnología, la psicología y la pedagogía, además de requerir un estudio pormenorizado del perfil del alumno y/o tutor que interactuará con el sistema [BAT17]. Por esta razón, integrar cuestiones de colaboración e interacción entre alumnos y docentes deberá realizarse y fundamentarse sobre la base de los conceptos definidos por la psicología de las interacciones sociales y los aspectos pedagógicos vinculados a las actividades académicas.

- Problema: No contar con un modelo que formalice el trabajo interdisciplinario entre la psicología, la pedagogía y la tecnología para el diseño de soluciones de aprendizaje colaborativo.
- Solución: Diseñar un modelo de integración que involucre las tres áreas indicadas (tecnología, psicología y pedagogía) para crear una solución de aprendizaje colaborativo asistido por computadora

1.1.2. CSCL para el Modelado de Software

Las plataformas **CSCL** promueven un gran avance en materia de enseñanza y aprendizaje ya que facilitan la tarea del docente para que los alumnos puedan acceder al material educativo en cualquier momento y lugar [FIL10]. Sin embargo, hay áreas de conocimiento que requieren de recursos externos para que ciertos temas puedan ser enseñados y evaluados como, por ejemplo, el uso de herramientas de modelado de software. La enseñanza de la **IS** requiere, además, esfuerzos extras para generar un entorno propicio de colaboración entre docentes y alumnos para poder realizar evaluaciones y seguimiento de trabajos [NEI16].

- Problema: La carencia de herramientas de modelado con características destinadas al aprendizaje integradas con plataformas **CSCL**.
- Solución: Diseñar una herramienta de modelado con las características necesarias para utilizarla durante el proceso de enseñanza y aprendizaje de la **IS**

1.1.3. Seguimiento, Evaluación y Autoevaluación de la IS en Entornos Colaborativos

La integración de un entorno de enseñanza asistido por computadora con una herramienta de modelado de software en el marco de trabajo colaborativo, facilita que los alumnos puedan construir su conocimiento centrado en el concepto de Zona de Desarrollo Próximo [VIG97]. Sin embargo, estos entornos **CSCL** integrados deberían, entre otras cosas, brindar la posibilidad de realizar el seguimiento del aprendizaje, además de la evaluación y autoevaluación de trabajos, siempre considerando que estas actividades deberán realizarse en entornos colaborativos.

- Problema: Carencia de entornos **CSCL** con herramientas de evaluación y autoevaluación de actividades relacionadas al modelado de software.
- Solución: Integrar en una herramienta de modelado de software las funcionalidades necesarias para realizar la evaluación y autoevaluación colaborativa relativas al proceso de enseñanza y aprendizaje de la **IS**.

1.2. Nuestra Propuesta:

El objetivo de la tesis es proponer una solución, en la enseñanza de la **IS** durante el proceso de modelado, a problemas tales como la carencia de herramientas informáticas con funcionalidades destinadas a la enseñanza y aprendizaje del modelado de software.

La propuesta incluye el diseño y desarrollo de una solución que integrará un entorno colaborativo de enseñanza asistido por computadora con una herramienta de modelado **UML**. Este recurso estará integrado al ámbito académico presencial para el apoyo a la enseñanza, seguimiento y evaluación de trabajos relacionados a la **IS**. En resumen, proponemos, por un lado, la definición de un modelo conceptual; por otro lado, la formalización de los procesos pedagógicos más importantes y, por último, su implementación en una solución informática que pueda ser utilizado en el ámbito académico.

1.2.1. uCASE-CL

La enseñanza de la **IS** en entornos virtuales colaborativos depende de un trabajo multidisciplinario que involucra la pedagogía, la psicología, la informática y las TIC. A estos pilares, que conforman las plataformas **CSCL**, es necesario agregarle uno más: una herramienta **CASE** de modelado, con las características necesarias para ser utilizada durante el proceso de enseñanza y aprendizaje [BAT17].

En virtud de lo anteriormente expresado, planteamos el modelo **uCASE-CL** [BAT17] [BAT17b] que tiene como finalidad definir todos los bloques funcionales y técnicos requeridos para que la solución propuesta tenga una base conceptual que permita:

- Comprender el dominio donde se utiliza (en este caso el modelado de software)
- Adaptarse a los medios que brindan acceso de forma ubicua a la información
- Definir aspectos tecnológicos
- Formalizar los procesos colaborativos de aprendizaje, evaluación y autoevaluación.

1.2.2. UAI Case

UAI Case [BAT17d] es la herramienta que materializa los conceptos planteados por el modelo **uCASE-CL** indicado anteriormente. Esta herramienta cuenta con un conjunto de características deseables que servirán como base para el desarrollo de una solución acorde a los objetivos planteados en esta tesis, a saber:

- Ser amigable y sencilla de aprender ya será destinada a alumnos universitarios de los primeros años.
- Mantener constantemente información contextual sobre el proceso de desarrollo, indicando roles y actividades inherentes a cada actividad desarrollada.
- Soportar **UML** de forma integral, desde aspectos de diseño y notacionales, hasta validación de modelos y especificación detallada de clases, casos de uso, objetos, etc. Las validaciones de los modelos **UML** deberán estar acompañados con mensajes personalizados en cada caso, mostrando el error y la forma correcta de hacerlo.

- Permitir interacción *online* y colaborativa para que múltiples usuarios trabajen juntos o separados, en el mismo tiempo o en momentos diferentes, sin que esta característica afecte negativamente el proceso de enseñanza y aprendizaje. Por tal razón, la herramienta debe disponer de información contextual acorde a esto.
- Mantener versiones y revisiones del trabajo realizado, con la información de quién lo hizo, qué hizo, cuándo y qué observaciones tuvieron los participantes del grupo de trabajo
- Facilitar una completa integración con otras herramientas de modelado utilizadas por la industria del software
- Mantener compatibilidad con otros formatos estándares tales como, **XMI**, imágenes, PDF, etc.
- Tener documentación de uso y manuales de usuario accesibles, fáciles de leer, con videos y ejemplos prácticos.
- Permitir la generación de documentación de forma personalizada y compatible con los formatos más comunes, tales como DOC, RTF, PDF, etc.
- Incluir funcionalidades para utilizar durante el proceso de enseñanza y aprendizaje, tales como evaluación, autoevaluación y seguimiento de trabajos.

1.3. Objetivos de la Tesis

Los objetivos **principales** de la tesis son los siguientes:

- Identificar problemas relacionados a la enseñanza y aprendizaje de la **IS**, en base al análisis de la bibliográfica vinculada con esta temática.
- Definir un modelo conceptual para la integración de una herramienta de modelado **UML** en un entorno ubicuo de aprendizaje colaborativo asistido por computadora.
- Diseñar técnicas de evaluación y autoevaluación colaborativas para utilizar durante el proceso de enseñanza y aprendizaje.

- Desarrollar un prototipo funcional que implemente el modelo conceptual propuesto.

1.4. Contribuciones Principales

Las contribuciones principales de la tesis son:

- Un modelo conceptual para representar la integración de un entorno de aprendizaje colaborativo con una herramienta de modelado [BAT17b] [BAT17].
- Una técnica para evaluar herramientas de modelado basada en las funcionalidades esperadas [BAT17c] [BAT17e].
- Un proceso de evaluación colaborativa para aplicar durante la enseñanza de las técnicas de modelado durante el proceso de desarrollo de Software [BAT16b] [BAT16c].
- Un entorno colaborativo que permita a los alumnos y docentes, utilizar una herramienta **CASE** integrada durante el proceso de enseñanza y aprendizaje de la ingeniería de software [NEI16] [BAT16] [BAT17d].

1.4.1. Trabajos Presentados Vinculados con la Tesis

A continuación, se listan los trabajos publicados cuya temática está vinculada al de los objetivos planteados en esta tesis:

- Neil C., De Vincenzi M., Battaglia N., Martínez R. (2016). Herramientas Colaborativas Multiplataforma en la Enseñanza de la Ingeniería de Software. In XVIII Workshop de Investigadores en Ciencias de la Computación
- Battaglia, N., Neil, C., De Vincenzi, M., & Martínez, R. (2016). UAI Case: integración de un entorno académico con una herramienta CASE en una plataforma virtual colaborativa. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).

- Battaglia, N., Neil C., Cardacci, D., De Vincenzi M., Martínez R. (2016,). Evaluación y Seguimiento Durante el Proceso de Enseñanza y Aprendizaje del Modelado UML en Entornos Colaborativos. In V Workshop de Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).
- Battaglia, N. Martínez, R. Otero, M., Neil, C., De Vincenzi M. (2016). Autoevaluación Colaborativa por medio de Rubricas en Entornos Virtuales de Enseñanza y Aprendizaje. In I Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), II Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC)
- Battaglia, N., Neil, C., De Vincenzi, M., Martínez, R., González Dana. (2017, junio). uCASE-CL: Aprendizaje Colaborativo de la Ingeniería de Software en Entornos Virtuales Ubicuos. In XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2017).
- Battaglia, N. Martínez, M., Neil, C., De Vincenzi M. (2017). UAI Case: Implementación del modelo uCASE-CL. In II Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), III Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC).
- Battaglia, N., Martínez R., Neil C., De Vincenzi M. (2017, octubre). Una Propuesta de Evaluación de Herramientas CASE para la Enseñanza. In VI Workshop de Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).
- Battaglia, N., Neil C., Martínez R., González, D., De Vincenzi M. (2017). Learning of Software Engineering on Collaborative Virtual Environments. In 7th World Engineering Education Forum (WEEF).
- Battaglia, N., Martínez R. (2017). Propuesta de Metodología de Evaluación para el Análisis de Modelado en Herramientas CASE. CoNaIISI.

1.5. Estructura General de la Tesis

La estructura general de la tesis está compuesta por: 8 capítulos, cada uno de ellos desarrollan los aspectos principales del trabajo; 3 anexos, que describe las técnicas y tecnologías utilizadas para desarrollar el prototipo.

1.5.1. Capítulos

- En el **Capítulo 2**, *Trabajo Colaborativo e Innovación Tecnológica*, detallamos las características principales sobre los conceptos de trabajo colaborativo y software para trabajo en grupo (*groupware*), el concepto de **CSCW** con sus ventajas y desventajas y las distintas taxonomías de *groupware*. Contextualizamos la ingeniería de colaboración y como se relaciona el concepto de patrones de colaboración con los *thinklets*. Presentamos el concepto de *awareness* y se resumen los tipos de indicadores necesarios para contextualizar un sistema *groupware* por medio de esta herramienta. Por último, presentamos el concepto de innovación en **TIC** y su impacto en diversos aspectos sociales, incluyendo a la educación.
- En el **Capítulo 3**, *Aprendizaje en Entornos Colaborativos*, hemos contextualizado el concepto de enseñanza y aprendizaje colaborativo, con sus ventajas y desventajas así como el desarrollo histórico de este concepto hasta el momento en que fue adoptándose dentro del aula. Presentamos el marco teórico que integra aspectos psicológicos y pedagógicos en relación con la interacción social y su influencia en el desarrollo y, en particular, el concepto de “zona de desarrollo próximo”. Por último, presentamos el impacto que las TIC han tenido en los entornos virtuales de enseñanza y aprendizaje creando los entornos uCSCL y cómo estos cambios afectan los métodos de evaluación, abordando el concepto de evaluación colaborativa.
- En el **Capítulo 4**, *Herramientas CASE*, definimos el concepto de herramienta **CASE** y sus principales características. Contextualizamos **UML** y lo relacionamos con el proceso unificado de Rational con el objetivo de entender más en profundidad las herramientas de modelado y el impacto en el proceso de desarrollo de software.

Definimos en base a esto, un conjunto de criterios para poder evaluar y comparar diferentes herramientas. Diseñamos un método para hacer esta evaluación con el objetivo de obtener un índice de aceptación de cada una y poder tener una comparación objetiva. Definimos las características más importantes para una herramienta **CASE** de modelado destinada a la enseñanza y el aprendizaje de la ingeniería de software.

- En el **Capítulo 5**, *Enseñanza y Aprendizaje Colaborativo de la IS en Entornos Virtuales Ubicuos*, realizamos un estudio sobre el estado del arte de las herramientas **CASE** para la enseñanza y de las plataformas **CSCL** para la **IS**, específicamente para modelado UML. Se investigó sobre el modelo **uCSCL** para aprendizaje ubicuo y en base a esto se planteó un modelo conceptual denominado **uCASE-CL** para describir el área de investigación relacionada al aprendizaje colaborativo de la **IS** en entornos virtuales ubicuos.
- En el **Capítulo 6** presentamos *UAI Case* como la implementación del modelo **uCASE-CL** planteado en el capítulo 5. Contextualizamos el proyecto, definimos sus características principales y lo referenciamos a las dimensiones propuestas por nuestro modelo detallando las características de cada actividad que son impactadas por la tecnología, la pedagogía y la psicología.
- En el **Capítulo 7**, analizamos trabajos de investigación vinculados al uso de herramientas **CASE** en la educación de la **IS** en entornos virtuales. En base a esto realizamos una comparación entre nuestra propuesta y los trabajos relacionados. Por último, presentamos los diferentes trabajos que hemos desarrollado vinculados a la temática de esta tesis.
- En el **Capítulo 8**, presentaremos primero un resumen de esta tesis, luego, las contribuciones principales y, por último, las líneas futuras de investigación. En la sección resumen e interpretación final, sintetizaremos la propuesta de la tesis respecto del modelo **uCASE-CL**, mostraremos las diferencias entre nuestro planteo respecto de los trabajos vinculados; en la siguiente sección, contribuciones principales, detallaremos los aportes más significativos de nuestra propuesta; finalmente, en la sección trabajos futuros, presentaremos distintas

líneas de investigación que permitirán continuar con el trabajo desarrollado en la tesis.

1.5.2. Anexos

- En el **Anexo I**, presentamos el análisis detallado de las diferentes herramientas CASE que estudiamos en el **Capítulo 4**, y con el cual obtuvimos el cuadro resumen de aceptación al final del mismo.
- En el **Anexo II**, proponemos una técnica para diseñar y modelar actividades colaborativas requeridas en la dimensión psicológica propuesta como parte del modelo **uCASE-CL**.
- En el **Anexo III**, abordamos el desarrollo del prototipo de *UAI* de forma detallada.

Capítulo 2

Trabajo Colaborativo e Innovación Tecnológica

2.1. Introducción

Desde hace muchos años se estudia la forma en que los seres humanos se comunican y relacionan para poder llevar adelante diversos objetivos. En todo momento es común observar como las personas se organizan y logran de forma coordinada cumplimentar una meta. Esta forma de trabajo, denominada como *trabajo colaborativo* se puede ver en cualquier ámbito: laboral, familiar, incluso y de forma muy marcada, el ámbito académico.

El trabajo colaborativo puede darse en diversos contextos y con personas de distintos niveles de experticia. Suele suceder de forma presencial, en un escritorio, en una reunión, incluso de forma espontánea en una charla informal de café. Puede presentarse en entornos distribuidos, por medio de un teléfono, por medio de una computadora.

En este sentido, los aspectos disruptivos de las Tecnologías de la Información y Comunicación (**TIC**), modificaron notablemente la forma de realizar trabajo colaborativo. La innovación en Información y Comunicación permite explorar y explotar nuevos enfoques con el objetivo de optimizar la forma de llevar adelante soluciones complejas con la colaboración de muchas personas involucradas, incluso de forma distribuida logrando mejores resultados y un uso más óptimo de los recursos.

Es importante destacar que el impacto de la innovación constante en materia de **TIC** afecta de forma beneficiosa a muchos aspectos cotidianos y la forma en que la sociedad se desenvuelve. En resumen, las **TIC** se relaciona con el trabajo colaborativo para crear

sistemas de trabajo colaborativo asistido por computadora (**CSCW**, *Computer Supported Collaborative Work*).

Los contextos colaborativos se expanden también a la educación. Los centros educativos extienden sus propuestas de enseñanza a entornos de *e-learning*, en donde el papel tecnológico es determinante para lograr que en estos ámbitos virtuales de aprendizaje se pueda realizar trabajo colaborativo.

2.2. Trabajo Colaborativo

Desde hace muchos años se habla del trabajo colaborativo como una forma de compartir experiencias e intereses en común que permitan conseguir objetivos. A mediados del siglo XIX Carl Marx ya hablaba sobre trabajo colaborativo y se refería como “múltiples individuos trabajando juntos de una manera planificada, en un mismo proceso o en procesos distintos pero conectados” [NEI16], [BAT16], [SOL13].

El concepto de trabajo colaborativo se introdujo a mediados de siglo por diferentes autores, sobre todo en referencia a actividades que pueden verse como reuniones, charlas de escritorio, charlas telefónicas, etc. Fue recién a finales de los 80’s cuando se comenzó a hablar de trabajo colaborativo como *Computer Supported Collaborative Work (CSCW)*, refiriéndose a múltiples individuos trabajando juntos, con objetivos en común, pero interactuando por medio de plataformas informáticas [BAN89] [GRE87].

Según Bannon, el término **CSCW** se utiliza para el intercambio de ideas en escenarios con perspectivas multidisciplinarias que sirven para generar resultados y aplicaciones realmente útiles [BAN88]. En el mismo autor quien indica que, en la actualidad el término **CSCW** se utiliza tanto para referirse al trabajo cooperativo como también al trabajo colaborativo, pero que existen diferentes apreciaciones. Numerosos autores coinciden con esto e indican que el término preciso es el de trabajo en grupo, cuando múltiples individuos trabajan juntos y relativamente cerca comparten objetivos y poseen comunicación directa. En este sentido, Gert-Jan de Vreede [VRE06] asegura que, al colaborar, las personas podrían lograr mucho más de lo que podría como individuos separados. El *groupware* representa los sistemas informáticos y las tecnologías que soportan a grupos de personas con tareas y metas en común; que proveen una interface

a un ambiente colaborativo; y vuelve a un usuario consciente de ser parte de un equipo [BAN89] [JOH88] [LYN90] [ELL91].

En resumen, cualquiera de las etiquetas que se utilicen (trabajo cooperativo o trabajo colaborativo) utilizando el término **CSCW**, se apoyan en el uso de la informática como medio para realizar el trabajo de forma organizada y que posea todas las características necesarias para evitar la interrupción de la actividad de colaboración.

CSCW debe comprenderse como un esfuerzo para entender la naturaleza y características del trabajo colaborativo con el objetivo de diseñar tecnologías informáticas adecuadas. Es un área de investigación que responde a las siguientes preguntas: ¿Cuáles son las características del trabajo colaborativo frente al trabajo realizado por un individuo de forma individual? ¿Por qué surgen y cuáles son los patrones de colaboración? ¿Qué rol cumple la tecnología informática? [BAN89]. Es el estudio de la forma en que las actividades de trabajo en grupo (colaboración) pueden ser soportadas con sistemas de computadoras. Es un área multidisciplinaria que involucra profesionales de informática, sociología y psicología entre otros.

Los sistemas **CSCW** se organizan en función del grado de necesidad de comunicar que posean las partes involucradas. Así, existen desarrollos de software que responden a los diversos niveles de requerimiento de interacción: Informar, Coordinar, Colaborar y Cooperar [BAI89]:

- Desde el punto de vista de la *comunicación (o información)*, la información circula en un solo sentido y el emisor no necesita conocer al receptor. Entre ellos el contacto es casi nulo. El soporte deberá ser para que el emisor pueda publicar la información y el receptor pueda encontrarla. *Ejemplo: Servidor FTP*
- La *coordinación* representa el primer contacto entre emisor y receptor, y la información comienza a fluir de forma bidireccional. Ellos coordinan el uso de los recursos y la información en el tiempo; no comparten objetos, pero si intereses en común. Esto implica modelar los recursos compartidos, consultar y registrar disponibilidad y uso de los mismos. *Ejemplo: Correo Electrónico.*
- Durante el proceso de *colaboración*, los involucrados participan en un mismo proceso. En este punto, es común encontrar disparidad en las responsabilidades. Esto genera que cada participante sea responsable de una parte y deba ser

evaluado de forma independiente. Es necesario modelar el proceso en relación con las tareas, recursos, tiempos y participantes, informar el estado del proceso, asignar responsabilidades y brindar la posibilidad de integrar resultados individuales. *Ejemplo: GitHUB [GIT17].*

- La *cooperación* representa la consecución de un objetivo común por medio de un plan compartido. En este punto, los objetivos del grupo se priorizan por sobre los de los individuos para dar lugar a una evaluación del equipo. Las interacciones son muy regulares y la toma de decisiones es responsabilidad del equipo. Es necesario brindar acceso concurrente a los recursos compartidos y brindar soporte para reuniones y actividades cara a cara, aún para miembros de un equipo distribuido. *Ejemplo: Soporte para reuniones Electrónicas.*

2.2.1. Ventajas y Desventajas del Uso de Entornos Colaborativos

Los entornos colaborativos presentan ventajas y desventajas que deben ser evaluadas al momento de decidir su implementación, en función del contexto donde se planea llevar adelante la tarea colaborativa [BIB09].

Entre las ventajas podemos observar las siguientes:

- Incremento en el acceso a la información. Los participantes de actividades colaborativas poseen acceso más sencillo a la información importante (por ejemplo, a archivos). Al utilizar herramientas que fomentan el diseño colaborativo de información, como por ejemplo una presentación, mantienen la información más consistente y accesible.
- Fomento del trabajo paralelo multidisciplinario. El trabajo colaborativo suele presentar diversos actores con diversas aptitudes y capacidades. Esto es sumamente beneficioso para las actividades colaborativas, ya que permite mejorar los resultados obtenidos. En muchas oportunidades se arman grupos de trabajo en base a capacidades o experiencias para luego ser combinados para poder obtener el resultado final. El trabajo paralelo distribuido permite, además, realizar otras actividades en simultáneo como, por ejemplo, buscar información sobre el tema en debate.

- Garantiza resultados de colaboración. Todas las actividades colaborativas generan algún resultado concreto. La conversación de un *chat* o los mensajes de un foro, son ejemplos concretos de esto.

Entre las desventajas destacamos las siguientes:

- Discusiones complejas. El primer punto a considerar es que, durante sesiones de remotas, es muy difícil transmitir sensaciones o expresiones. Éstas, muchas veces son parte de la comunicación que se puede llevar adelante en una reunión presencial. Otro aspecto muy importante es que, sin una correcta herramienta de coordinación, por ejemplo, mantener una sesión de chat entre muchos participantes podría generar más problemas que soluciones. Los sistemas colaborativos deben proveer mecanismos que ayuden al usuario a entender el contexto. Estos mecanismos (denominados de awareness) no siempre proveen de la información de forma clara, incluso no pueden informar acerca de sentimientos o sensaciones que son parte intrínseca del proceso comunicacional humano.
- Requiere mayor concentración. Las sesiones de trabajo remoto requieren de mucha más concentración para poder comprender que es lo que quieren expresar los otros participantes. No solo requiere de mayor esfuerzo para comprender, sino también para no tomar decisiones erradas en base a esa información difícil o mal comprendida. Requiere de mucha intuición de los participantes.
- Reducción de la interacción social. Los entornos destinados a trabajo colaborativo no brindan lugar a situaciones ajenas al tema en cuestión. Normalmente estos contextos brindan mejoras notables en el rendimiento, eficiencia y mejora en el tiempo, pero reduce de forma notable el intercambio social.

Comprender las ventajas y las desventajas de los sistemas distribuidos es uno de los factores más importantes para diseñarlos de forma correcta.

2.2.2. Taxonomías de *Groupware*

Una taxonomía nos permite deducir características comunes en cuanto a tecnologías, requerimientos, dificultades, etc. Describimos a continuación cuales son las taxonomías que se utilizan para caracterizar a los sistemas *groupware* [ELL91].

2.2.2.1. Matriz Espacio-Temporal

Las herramientas *groupware* pueden ser diseñadas para un grupo que trabaja en un mismo lugar y momento, lo que denominamos “cara a cara” o bien para un equipo distribuido que trabaja de forma asincrónica, es decir, en momentos diferentes,

A continuación, se presenta una de las taxonomías más comprendidas por el mundo de los desarrolladores. Ésta se centra en identificar la relación espacio-temporal con el fin de definir el ámbito de aplicación para una solución colaborativa. Una de sus grandes desventajas es que una solución difícilmente pueda aplicar a dos celdas de forma directa [GRU94].

	Mismo Tiempo	Diferente Tiempo
Mismo Lugar	Interacciones cara a cara	Interacción Asincrónica
Diferente Lugar	Interacción Sincrónica Distribuida.	Interacción Asincrónica Distribuida.

Tabla 2.1. Matriz Espacio-Temporal

2.2.2.2. Modelo de las 3C

Esta taxonomía muestra la relación y el ámbito de aplicación de las diferentes herramientas colaborativas entre el soporte para comunicación, soporte para coordinación y soporte de cooperación [BOR00].

Taxonomía de las 3C



Figura 2.1. Taxonomía de las 3C

2.2.2.3. Del Dominio

En la actualidad existen diferentes tipos de aplicaciones *groupware* y cada una puede ser utilizada en un ámbito de aplicación específico. Podemos comprender esto como el dominio de aplicación. Acotamos el dominio a un conjunto de aplicaciones específicas que se utilizan tanto en el ámbito académico como en la industria.

Esta clasificación es en referencia al ámbito de aplicación de una aplicación. Los más conocidos (con sus características principales) son:

- Sistemas de mensajería
- Editores grupales
- Sistemas de conferencia
- Sistemas de apoyo a la toma de decisiones (**GDSS**)

2.2.2.3.1. Sistemas de Mensajería

Son los sistemas Groupware más comunes. Los sistemas de mensajería sirven para interconectar personas a través de redes informáticas, soportando el intercambio asincrónico de mensajes de texto entre los participantes de un grupo. Dentro de las aplicaciones más conocidas podemos encontrar el correo electrónico (*e-mail*) o los sistemas de *chat* como los que funcionan bajo el protocolo **IRC** (*Internet Relay Chat*).

Una de las aplicaciones más conocidas para mensajería instantánea es *MIRC* [MIR16]

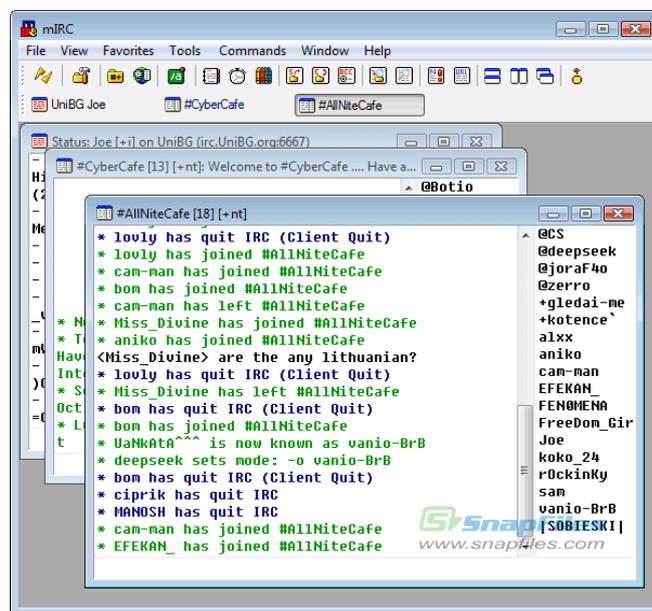


Figura 2.2. Aplicación MIRC para Mensajería Instantánea

2.2.2.3.2. Editores Grupales

Los miembros de un grupo pueden editar de forma conjunta un mismo documento y en tiempo real. Los primeros editores grupales permitían acceso concurrente de lectura y un único acceso a la vez de escritura. El avance en materia de **TIC** permitió mejorar estas limitaciones, ya que hoy en día los editores grupales pueden tener acceso concurrente de escritura. Estas aplicaciones pueden ser pizarras compartidas, editores gráficos, editores para desarrolladores de *software* o herramientas de ofimática.

Un ejemplo muy representativo de este grupo son las herramientas propuestas en la suite de *Google (GSuite)* [GOO17] u otras como *Realtimeboard* [REA17] que permite múltiples soluciones como pizarras compartidas, incluso permite el diseño colaborativo de mapas mentales.

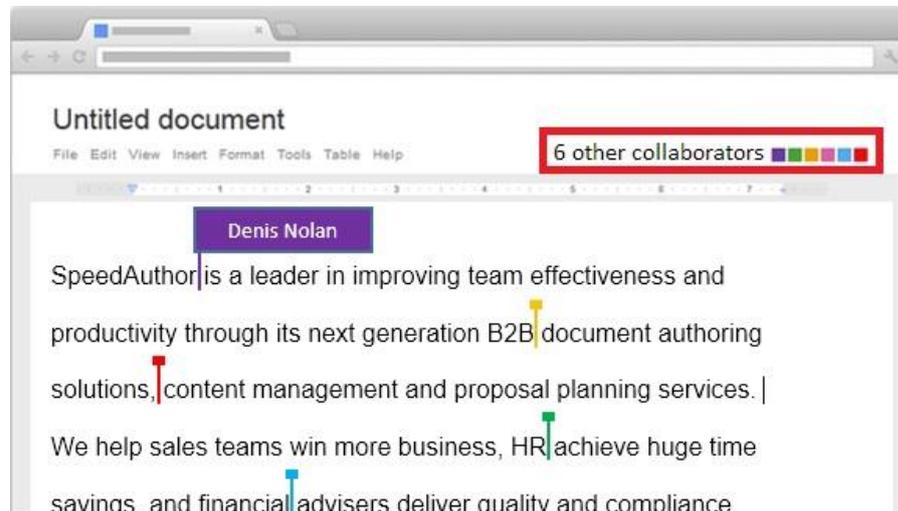


Figura 2.3. Editor de texto Google DOC con 6 colaboradores

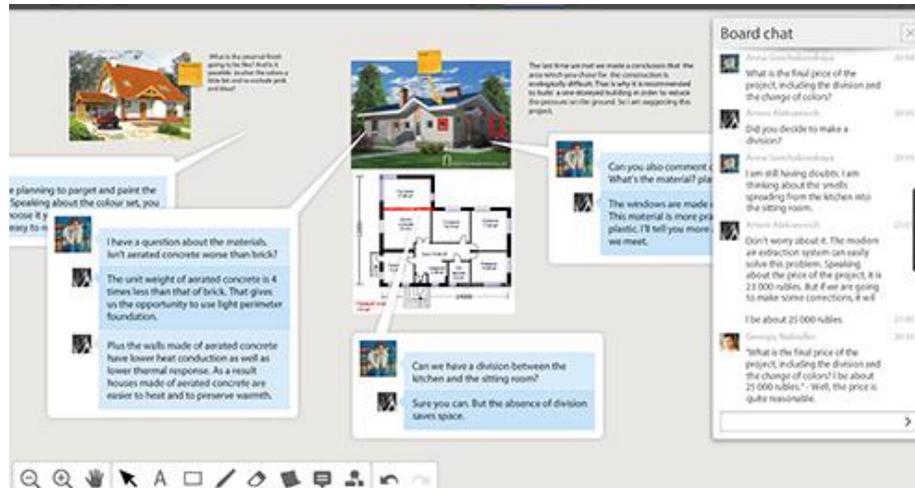


Figura 2.4. Editor Realtimeboard

2.2.2.3.3. Sistemas de Conferencia

Las constantes mejoras en las tecnologías de comunicación permiten realizar videoconferencias de forma muy sencilla. Se basa en enlaces de audio y video en tiempo

real, lo que permite que en simultáneo los participantes de un grupo puedan colaborar para conseguir un objetivo.

Unas de las aplicaciones de conferencia más usada en la actualidad es *Skype* [SKY17]. Por medio de la misma se pueden entablar conversaciones de audio y video incluyendo espacio para *chat*.

En la actualidad existen diferentes aplicaciones que permiten crear enlaces de audio y video en tiempo real, inclusive permiten agregar texto a la conferencia. La tendencia a la ubicuidad hace que las aplicaciones tengan su respectivo cliente para distintos dispositivos móviles

Algunos ejemplos son *Skype* [SKY17] y *Hangouts* [HAN17] de *Google* que también permite compartir pantallas a la conferencia.



Figura 2.5. Google Hangouts

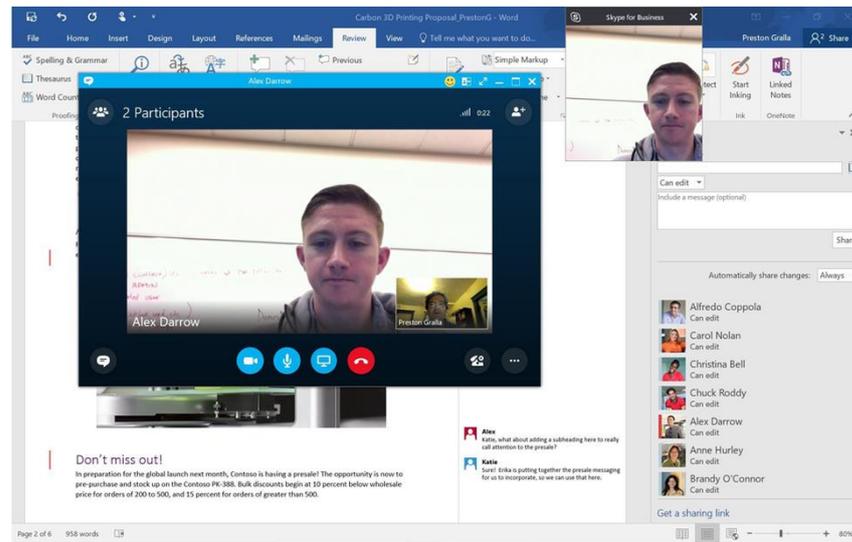


Figura 2.6. Skype

2.2.2.3.4. Sistemas de Apoyo a la Toma de Decisiones

Los sistemas de apoyo a la toma de decisiones en grupo (*Group Decision Support Systems, GDSS*) son también conocidos como Salas de Reuniones (*Meeting Rooms*) proveen mecanismos para explorar problemas no estructurados. Estos sistemas fueron desarrollados para mejorar la eficiencia en las reuniones destinadas a la toma de decisiones.

Algunas herramientas comunes de este grupo son las de *Brainstorming* (Lluvia de ideas) y votación electrónica. Entre las herramientas actuales más destacadas podemos encontrar *MeetingSphere* [MET17] que posee una suite de herramientas destinadas a hacer más productivas las reuniones. Entre las herramientas encontramos una para lluvia de ideas, una para realizar discusiones y otra para realizar votación o *ranking* de tópicos.

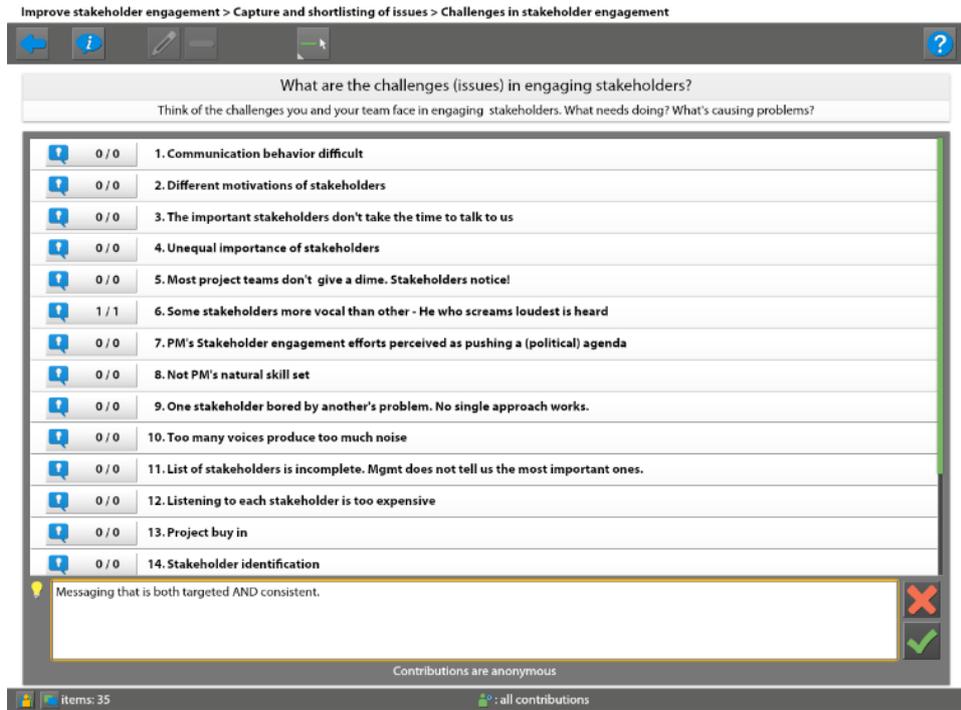


Figura 2.7. MeetingSphere (Brainstorming)

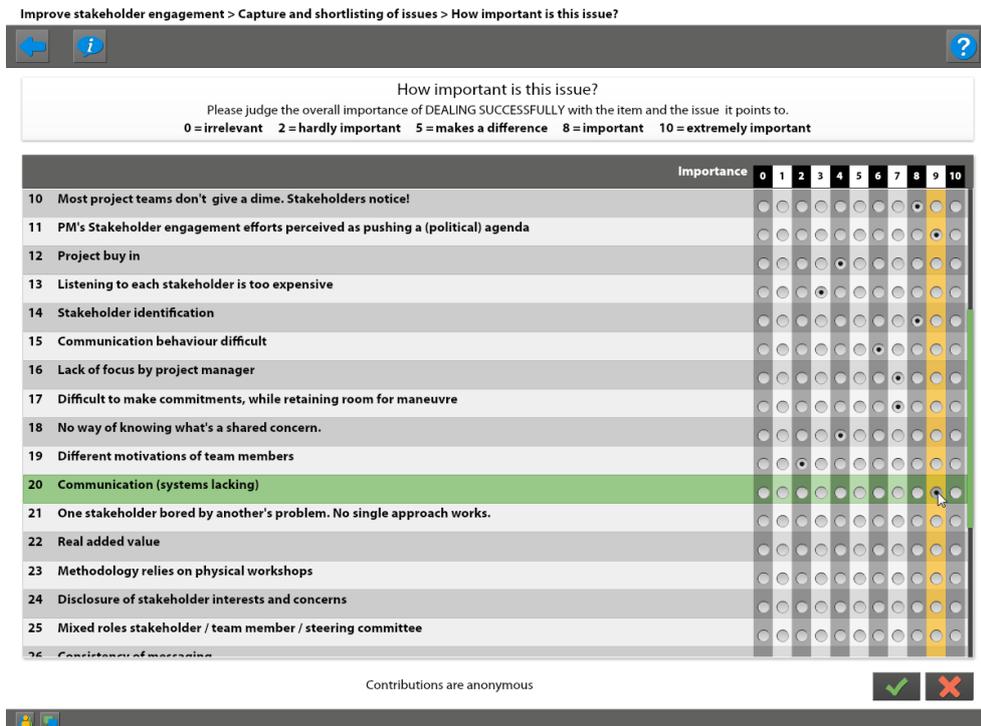


Figura 2.8. MeetingSphere (Ranking)

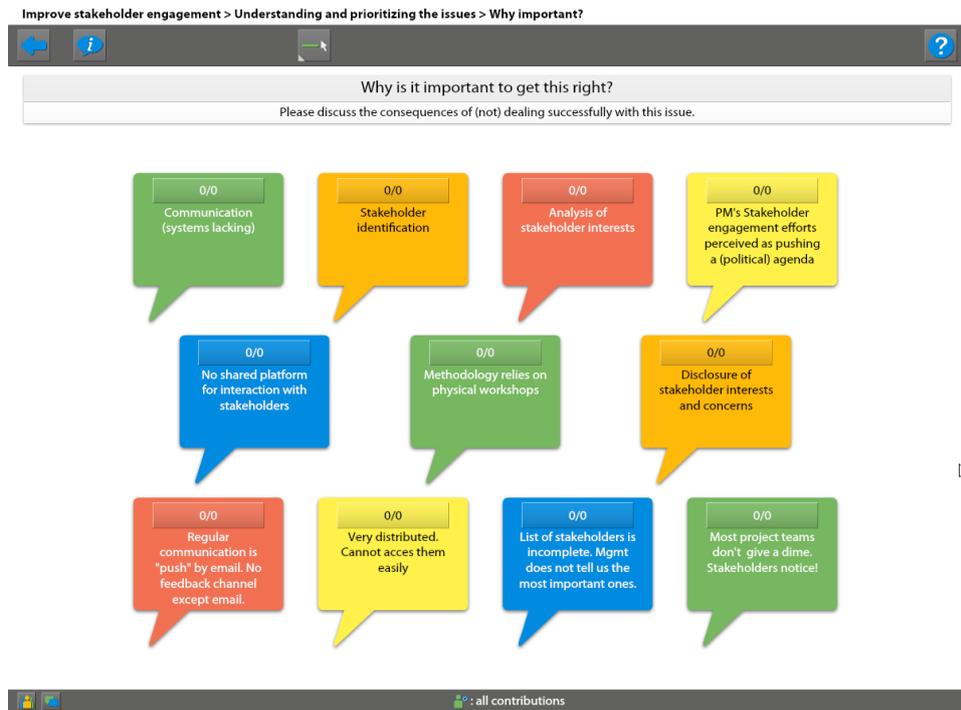


Figura 2.9. MeetingSphere (Discussion)

2.2.3. Awareness

La medida del intercambio de información, conocimiento y la actividad individual y la coordinación, son centrales para una colaboración exitosa. Estos factores son los más importantes a la hora de diseñar un sistema **CSCW** [DOU92].

El mismo autor asevera que el *awareness* es “comprender la actividad de los otros” para poder contextualizar la propia actividad. Es toda aquella información inherente al proceso colaborativo. Representa la información de otros usuarios, las actividades realizadas, el tiempo de participación, su ubicación en el sistema, cambios realizados, objetos que utilizan, entre otras cosas.

Los mecanismos de *awareness* sirven para mejorar la percepción y obtener más claridad con respecto a lo que sucede en cada momento durante una actividad colaborativa. Es importante tener en cuenta que, durante una sesión presencial, los participantes hacen esto de forma natural, cosa que ayuda a comprender en cada momento la forma de participación y manejar la información de forma más eficiente [BIB09] [HER13].

Algunas de sus características más

destacadas son: [GUT02]

- El *awareness* es el conocimiento acerca del estado de un entorno relativo al tiempo y espacio.
- El entorno cambia todo el tiempo, por eso el *awareness* registra este aspecto y deberá mantener informado a los integrantes del grupo.
- Las personas interactúan, exploran el medio ambiente, y el mantenimiento de la información se logra a través de esta interacción.
- El mantenimiento del *awareness* es un objetivo secundario de una tarea, ya que el objetivo principal es completarla.

Existen diferentes tipos de *awareness*, de las cuales destacamos:

- *Awareness* informal. Muestra conocimiento general sobre los miembros del grupo de trabajo, como su ubicación, presencia, foco. Esto permite aprovechar las oportunidades que se presentan para interactuar con otras personas.
- *Awareness* del entorno. Un usuario deberá poder tener una percepción muy amplia de lo que sucede en el entorno y poder responder de forma selectiva a aquellas actividades más relevantes para la tarea en la situación actual y poder predecir el estado de los elementos en un futuro cercano.
- *Awareness* del espacio de trabajo. Representa la comprensión de la interacción de las personas en el entorno de trabajo compartido, con información inherente a lo que suceda pura y exclusivamente en ese contexto. De esta forma las personas podrían participar en tareas que otro esté llevando a cabo en el entorno compartido. Esto incluye información sobre acceso y modificaciones de objetos compartidos.

El modo de *awareness* es el conjunto de eventos que permite describir el estado de la cooperación. Los modos son los siguientes [HER13]:

- Síncrono Acoplado. Responde a lo que está pasando en el ámbito de trabajo (presente)
- Síncrono Desacoplado. Es lo que ha cambiado en el ámbito de trabajo desde el último acceso (pasado).

- **Asíncrono Desacoplado.** Es cualquier cosa de interés que haya cambiado hace poco en otro lugar (pasado)

En la tabla 2.2 se muestran los diferentes elementos de *awareness* para información inherente al Entorno relativo al presente. En la Tabla 2.3 se muestran aquellos elementos de *awareness* relativos al pasado [GUT02].

Categoría	Elemento	Pregunta Específica
Quién	Presencia	¿Hay alguien en el área de trabajo?
	Identidad	¿Quién está participando?
	Autoría	¿Quién está haciendo eso?
Qué	Acción	¿Qué están haciendo?
	Intención	¿Qué objetivo tiene esa acción?
	Artefacto	¿En qué artefacto se está trabajando?
Dónde	Ubicación	¿Dónde están trabajando?
	Mirada	¿Dónde están mirando?
	Vista	¿Dónde pueden ver?
	Alcance	¿Dónde pueden llegar?

Tabla 2.2. Elementos de awareness relativo al presente

Categoría	Elemento	Pregunta Específica
Cómo	Acción Histórica	¿Cómo sucedió esa operación?
	Artefacto Histórico	¿Cómo llego ese artefacto a ese estado?
Dónde	Evento Histórico	¿Cuándo sucedió ese evento?
Quién (Pasado)	Presencia Histórica	¿Quién estuvo? ¿Cuándo?

Dónde (Pasado)	Ubicación Histórica	¿Dónde estuvo determinada persona?
Qué (Pasado)	Acción Histórica	¿Qué estuvo haciendo determinada persona?

Tabla 2.3. Elementos de awareness relativo al pasado

2.2.4. Ingeniería de Colaboración

Cuando dos o más personas mancomunan sus esfuerzos en busca de objetivos compartidos, se produce trabajo colaborativo. El trabajo colaborativo requiere de numerosos procesos colaborativos para poder llegar al objetivo. Para que estos procesos en donde se involucran varias personas puedan llevarse a cabo de la mejor manera, es necesario de un conjunto de herramientas que ayuden a facilitar la interacción humana.

Como cualquier herramienta, las tecnologías de colaboración pueden ser usadas bien o mal. En este sentido, Coto, Collazos y Rivera aseveran que estas herramientas pueden no garantizar el éxito del proyecto si no se conocen bien para que el grupo pueda obtener los resultados esperados. [CCR16].

Gert-Jan de Vreede, en su trabajo titulado “Ingeniería de Colaboración: Diseño de Procesos Repetidos para Tareas Colaborativas de Alto Valor” [VRE06], define que la Ingeniería de Colaboración (**IC**) es un enfoque para diseñar e implementar procesos de trabajo colaborativo recurrentes de alto valor, para ser ejecutada por profesionales sin el apoyo de facilitadores. Como consideración adicional, no todas las empresas poseen los recursos necesarios para acceder a un facilitador constante. En este sentido, muchos equipos podrían no beneficiarse con sus servicios.

Para mejorar las posibilidades de éxito, diversos autores aseguran que los procesos colaborativos deben ser explícitamente diseñados, estructurados y manejados. Esta actividad de diseñar y estructurar los procesos colaborativos es la que define la ingeniería de colaboración, para ser usados de forma repetitiva por un conjunto de personas con objetivos compartidos [BRI06] [KOL07].

La **IC** propone los siguientes roles [KOL06][BRI06]:

- Facilitador, crea un proceso dinámico que involucra las relaciones entre las personas, la tarea y la tecnología. Trata de estructurar las relaciones en el proceso y que el resultado sea positivo.
- Ingeniero en Colaboración, diseña y documenta un proceso de colaboración de tal manera que sea transferible a un practicante, que deberá poder ejecutarlo sin la ayuda del ingeniero.
- Usuario, es un experto de dominio que puede facilitar el entendimiento de un proceso determinado.

La IC, además, se ocupa de generar bloques de conocimiento con las actividades colaborativas, la forma de llevarlas adelante y como pueden ser reutilizadas para potenciar el trabajo grupal.

Vreede proponen dividir la ingeniería de colaboración en cuatro capas bien definidas. La capa de proceso, la capa de patrones de colaboración, la capa de *thinklets* y una capa adicional que representa los fundamentos teóricos sobre lo que se desea afectar con el proceso colaborativo. A fines prácticos, describiremos las tres primeras, ya que están representadas de forma jerárquica.

2.2.4.1. La Capa de Proceso

Esta primera capa de la jerarquía se enfoca en responder la siguiente pregunta ¿qué debería hacer el grupo? No describe como se llevará a cabo, sino qué es lo que debería suceder. Se utiliza principalmente para mantener y comprender los objetivos del proceso.

El modelo propuesto, define un conjunto de fases por las que las diferentes personas del grupo de trabajo pasaran de forma indefinida.

- Entender el problema
- Desarrollar soluciones alternativas
- Evaluar alternativas
- Elegir alternativas
- Hacer un plan
- Tomar acción

- Monitorizar los resultados

Un diseñador de procesos puede usar el modelo como una forma de pensar sobre qué pasos debe tomar un grupo y qué tipo de resultados un grupo debe producir durante su esfuerzo conjunto. Cada uno de las fases definidas tendrá un entregable, por ejemplo

2.2.4.2. Patrones de Colaboración

La **IC** identifica de forma fehaciente un conjunto de Patrones de Colaboración (**PC**) que describen el trabajo colaborativo de un grupo de personas. Estos patrones representan la guía de ejecución de un proceso y la manera en que los participantes de una actividad la llevan adelante de principio a fin. Cada patrón presenta un conjunto de sub patrones que pueden ser relacionados con las actividades colaborativas.

Si bien los **PC** representan la guía de ejecución de un proceso, no presentan una forma detallada para guiar al grupo hacia la meta. Para esto crean los *thinklets*, como técnica de facilitación repetible, transferible y predecible para asistir a un grupo a alcanzar un objetivo [KOL04] [BRI06].

La siguiente tabla resume los **PC** con sus respectivos sub patrones [BRI06].

Nombre	Descripción
Generación	<p>Agregar Contenido al paquete de conceptos del grupo</p> <ul style="list-style-type: none"> • Reunir Recopilar y compartir conceptos conocidos. • Crear Producir y compartir ideas desconocidas por el grupo. • Elaborar Refinar o descomponer conceptos existentes.
Reducción	Mantener foco en los conceptos más importantes o que merecen mayor atención

	<ul style="list-style-type: none"> • Seleccionar Elegir un conjunto de conceptos existentes. • Resumen Derivar conceptos más generales de conceptos específicos en el conjunto existente. • Resumir Capturar la esencia de los conceptos sin eliminarlos.
Clarificación	Lograr el entendimiento común de los conceptos y frases manejados por el grupo por medio de una descripción .
Organización	Comprender mejor las relaciones entre conceptos <ul style="list-style-type: none"> • Clasificar Ordenar conjunto de conceptos por medio de etiquetas. • Estructurar Crear arreglos espaciales entre los conceptos para representar sus relaciones conceptuales.
Evaluación	Mejorar la comprensión del valor relativo de los conceptos bajo consideración <ul style="list-style-type: none"> • Encuesta Evaluar la opinión del grupo con respecto a los conceptos. • Ranking Identifica un orden de preferencia entre conceptos. • Evaluar Especificar y elaborar sobre el valor de los conceptos.

Tabla 2.4. Patrones de Colaboración

2.2.4.3. Thinklets

Un *thinklet* es la unidad más pequeña de capital intelectual necesario para crear un patrón predecible y repetible de pensamiento entre las personas que trabajan juntas en busca de un objetivo.

Los *thinklets* son bloques de construcción que se pueden unir para especificar cómo un determinado **PC** debe realizarse cuando el proceso está en ejecución. Pueden ser usados como unidades conceptuales de construcción de procesos colaborativo, ya que proporcionan el conocimiento necesario sobre las instrucciones o pasos a seguir para llevar adelante el proceso grupal

Una de las grandes ventajas de los *thinklets* es que los diseñadores de procesos colaborativos pueden emplearlos para escoger soluciones conocidas y no invertir esfuerzos en inventar y probar nuevas. Esto puede reducir tanto el esfuerzo como el riesgo durante el desarrollo de procesos grupales [SOL13]. El uso de patrones reduce notablemente la posibilidad de fallo, ya que, al responder a soluciones conocidas y probadas, reduce el esfuerzo y el riesgo al desarrollar.

Los *thinklets* se detallan completamente y son fácilmente modificables. Su sencillez permite ser usados por usuarios poco experimentados. Se han escrito y probado alrededor de 60 *thinklets*.

Las siguientes tablas muestran dos ejemplos resumidos de *thinklets*.

<i>Thinklet</i> : RichRelation	
Grupo	Organización
Escogerlo	<ul style="list-style-type: none"> • Para crear un conjunto de categorías para organizar comentarios en una sesión de <i>Brainstorming</i> • Después de un <i>thinklet</i> de reducción y antes de un thinkLet donde las ideas se agruparán en categorías
No Escogerlo	<ul style="list-style-type: none"> • Para reducir temas importantes. Ayuda a organizar muchas sesiones de <i>Brainstorming</i>, pero cuando no

	signifique reducir ideas en un menos, pero más importantes
Información General	<p>Los participantes participan de la sesión de <i>Brainstorming</i> y encuentran dos ítems relacionados de algún modo. Articulan una relación entre estos dos ítems y si el grupo está de acuerdo se define como etiqueta principal.</p> <ul style="list-style-type: none"> • Entradas Comentarios de la sesión • Salidas Un conjunto de nombres de categorías para organizar los contenidos de la actividad.
Configuración	<ul style="list-style-type: none"> • Colocar una lista de categorías en un contenedor de la herramienta de categorización. • Mostrar el contenido del contenedor. • Preparar para agregar nueva categoría
Pasos	<ol style="list-style-type: none"> 1. Decir “Por favor lea los comentarios de su pantalla. Si encuentra dos que estén relacionados de algún modo, por favor indíqueme cómo”. 2. Agregar nueva categoría con la relación como etiqueta. 3. Continuar con el proceso hasta que los participantes no encuentren más relaciones.

Tabla 2.5. Ejemplo de thinklet RichRelation

Thinklet: LeafHopper	
Grupo	Reducción
Escogerlo	<ul style="list-style-type: none"> • Cuando se sepa de antemano que el equipo va a dar ideas sobre varios temas en simultaneo. • Cuando los participantes tengan diferente nivel de experticia e interés.

	<ul style="list-style-type: none"> • Cuando no sea importante asegurar que todos los participantes contribuyan a cada tema.
No Escogerlo	<ul style="list-style-type: none"> • Cuando sea importante asegurar que cada persona aporte a todos los tópicos.
Información General	<p>Los participantes comienzan con una lista de varios tópicos de discusión. Cada ítem tiene asociada una ventana de comentarios. Cada participante hace su aporte dependiendo del que más le interese y en el cual tenga mayor experiencia.</p> <ul style="list-style-type: none"> • Entradas Una lista de tópicos que serán dirigidos por el equipo. • Salidas Un conjunto de comentarios organizados por tópicos de discusión.
Configuración	<ul style="list-style-type: none"> • Crear una lista con los tópicos de discusión. • Los participantes pueden contribuir con comentarios del tópico. • Los participantes no pueden contribuir con nuevos tópicos. • Las contribuciones podrían ser anónimas.
Pasos	<ol style="list-style-type: none"> 1. Presentar los tópicos al grupo y asegurarse que los participantes lo comprendieron. 2. Informar a los participantes sobre el tipo de ideas con las que se podrá contribuir. 3. Informar como abrir la ventana de discusión para cada tópico. 4. Decir “Comenzar a trabajar con los tópicos que tenga más interés y experticia. Luego, si le sobra tiempo, recorra los otros tópicos para leer y comentar sobre la contribución de otros”.

	5. Decir “Podrías no tener tiempo para trabajar en todo. Por favor, hazlo en los temas más importantes”
--	---------------------------------------------------------------------------------------------------------

Tabla 2.6. Ejemplo Thinklet LeafHopper

2.3. Innovación Tecnológica

Las Tecnologías de Información y Comunicación (**TIC**) representan el fruto de la investigación y desarrollo científicos que, en un proceso continuo de evolución e innovación, modifican directamente la forma de vivir de una sociedad [VAC97].

Se incluyen en el concepto de **TIC** a todos los medios de comunicación social masiva y los medios de comunicación interpersonales con soporte tecnológico como, por ejemplo, el teléfono [GRA00].

La Tecnología es la aplicación de conocimientos científicos para mejorar las actividades humanas. La información son datos que tienen un significado relativo al observador. Estos datos serán transformados en información por un proceso y servirán para la toma de decisiones. En un sistema con participación directa de los seres humanos en el procesamiento de los datos, estos serán transformado por un proceso cognitivo y el resultado será una acción en base a una decisión.

Por otro lado, la Comunicación son los mensajes ente un emisor y un receptor, utilizando un medio en un determinado contexto. Los seres humanos se comunican para expresar, conocer, coordinar grupos, etc.

Los avances tecnológicos generados por la informática y las telecomunicaciones parten de la combinación de estos tres conceptos: las tecnologías que proporcionan información, herramientas para procesarla y los canales de comunicación para transmitirla.

Las **TIC** realizan importantes aportes vinculados al desarrollo de sistemas informáticos que posibilitan procesar, distribuir y almacenar datos, así como generar los canales de comunicación para difundirlos. Estos aportes pueden ser utilizados por la sociedad en diferentes contextos y con distintos objetivos, y entre los principales se pueden mencionar:

- Fácil acceso a una gran fuente de información. Disponer de medios de comunicación como redes inalámbricas o de telefonía móvil que brindan acceso a internet de forma ubicua, permite que la alternativa para obtener información sea enorme.
- Procesamiento rápido y fiable de todo tipo de datos. La innovación tecnológica permitió crear elementos informáticos mucho más veloces. Esto se ve por ejemplo cuando se le delega la responsabilidad de procesar datos a un dispositivo móvil recibiendo solamente datos desde una fuente externa. Otro punto de vista es que los canales de comunicación actuales permiten delegar el procesamiento a equipos informáticos en la nube, dando origen al concepto de “*Cloud Computing*”.
- Canales de comunicación inmediata (sincrónica y asincrónica), Tener inmediatez en las comunicaciones permite explotar al máximo los métodos sincrónicos como *chats* y asincrónicos como los *foros*. La ubicuidad que permiten las TIC impacta de forma positiva en los medios de comunicación.
- Capacidad de almacenamiento. Los avances tecnológicos permiten almacenar grandes cantidades de información en pequeñas unidades transportables como memorias flash. Otro aporte es la informática en la nube, que permite generar independencia de los medios físicos. Tener medios de comunicación rápidos y seguros permite que el almacenamiento de *datos masivos (big data)* se realice en “granjas” de servidores remotos con almacenamiento escalable.
- Automatización de trabajos. El procesamiento automático de datos es una de las características más importantes de la informática. Esto permite automatizar actividades y mejorar en muchos aspectos las tareas cotidianas.
- Interactividad. La interacción entre sistemas a través de las redes, permite extender y conectar sistemas de todo tipo, desde sistemas expertos hasta sistemas autónomos y de control especializado.

La constante innovación tecnológica genera una revolución permanente en la sociedad, modificando la forma de pensar y de actuar de los individuos e impactando en forma directa en los aspectos sociales, culturales y económicos de la vida cotidiana: el trabajo, la salud y la educación entre otros. Las **TIC** forman parte de una cultura tecnológica que

genera cambios muy fuertes en la sociedad actual, configurando *La Sociedad de la Información* [GRA00] [IBA04].

2.3.4. Innovación Tecnológica en la Educación

Los cambios producidos por el impacto de las **TIC** en los procesos educativos, afectan directamente sobre las emociones de la sociedad. Las emociones juegan un papel muy importante y decisivo para realizar tareas complejas y define a la inteligencia emocional como elemento fundamental para obtener éxito en el desempeño, por ejemplo, en actividades colaborativas dentro de equipos de trabajo. La combinación de las estrategias ofrecidas por la inteligencia emocional y el aprendizaje colaborativo, incluyendo la incorporación y uso de nuevas tecnologías a la formación docente, propone superar barreras emocionales para afrontar la informatización del componente primordial del proceso: el individuo [CAL02].

Los avances en **TIC** generan grandes aportes a la educación, por ejemplo, a través de la creación de entornos de aprendizaje asistidos por computadora y el desarrollo de propuestas de enseñanza apoyadas en el aprendizaje colaborativo, brindando así mayor potencial a las propuestas formativas [NEI16].

Basado en estos aspectos, se desarrollan nuevas modalidades de enseñanza como ser, por ejemplo, la educación en línea. Esta modalidad amplía el horizonte más allá de los límites físicos que impone la educación presencial, en donde cada uno de los integrantes aporta conocimientos, estilos y modos de aprender [ROT06].

Esta disrupción producto del impacto de las **TIC**, afecta directamente al ámbito académico y la función que las universidades cumplen en la sociedad. En este sentido, tanto las instituciones de enseñanza, como los docentes y los alumnos deberán cambiar para poder flexibilizarse y adoptar las nuevas concepciones del concepto de enseñanza que, por supuesto, impacta directamente en el proceso de aprendizaje. Es en este aspecto, que se afirma [IBA04] que *“El éxito o el fracaso de las innovaciones educativas depende, en gran parte, de la forma en la que los diferentes actores educativos interpretan, redefinen, filtran y dan forma a los cambios propuestos”*.

El aprendizaje mediado por la tecnología computacional se define como *e-learning*. Si bien *el e-learning* es independiente del grado de presencialidad (puede ser total, medio

o nulo) se vincula con el concepto de educación a distancia el cual apunta a un sistema de enseñanza y aprendizaje que posea un grado de presencialidad medio o nulo [VIT08].

Desde mediados de la década del 80, diversos autores destacan las características de los sistemas de enseñanza a distancia, considerándola como un sistema tecnológico de comunicación masiva. Este sistema sustituye la interacción personal dentro del aula entre el docente y el alumno utilizando una estructura tutorial, con los estudiantes más autónomos.

Las características distintivas de estas plataformas son las siguientes [GAR90] [IBA04]:

- Separación alumno-profesor y cambio en sus prácticas
- La utilización de medios técnicos
- La existencia de una organización de apoyo
- El aprendizaje independiente de los alumnos
- El enfoque tecnológico
- La comunicación masiva

De esta adaptación tecnológica y su integración con entornos de *e-learning*, surge la modalidad mixta de enseñanza, en donde el curso se desarrolla en gran parte de manera tradicional (presencial) para mantener discusiones guiadas por el docente. Otra parte se lleva a cabo en un ambiente virtual para abordar aquellas actividades que requieren de coordinación, cooperación y colaboración para ser llevadas adelante [NEI16] [BAT16] [ROS08].

Según [IBA04] existen cuatro modelos de evolución de los sistemas de enseñanza.

- El modelo de iniciación, no brinda oportunidades de interacción ni se proporcionan recursos extra al material relacionado al curso.
- El modelo estándar, que intenta utilizar las ventajas que proporciona la tecnología para permitir un grado limitado de comunicación e interacción entre docentes y alumnos.
- El modelo evolucionado mejora el estándar al introducir otros elementos complementarios de enseñanza, incluyendo clases en vivo o cuestiones específicas bajo demanda del alumno.

- El modelo radical que, a diferencia del resto de los modelos, ignora el concepto de clase. Aquí los alumnos son organizados en grupos y aprenden de la interacción entre ellos dentro de recursos virtuales existentes con el profesor como guía o tutor.

Resumen

En este capítulo detallamos las características principales sobre los conceptos de trabajo colaborativo y software para trabajo en grupo (*groupware*), el concepto de **CSCW** con sus ventajas y desventajas y las distintas taxonomías de *groupware*. Contextualizamos la ingeniería de colaboración y como se relaciona el concepto de patrones de colaboración con los *thinklets*. Presentamos el concepto de *awareness* y se resumen los tipos de indicadores necesarios para contextualizar un sistema *groupware* por medio de esta herramienta. Por último, presentamos el concepto de innovación en **TIC** y su impacto en diversos aspectos sociales, incluyendo a la educación.

Capítulo 3

Aprendizaje en Entornos Virtuales Colaborativos

3.1. Introducción

Largo ha sido el camino que la humanidad ha ido recorriendo con el correr de los años en busca de una mejor forma de impartir educación. Si bien podría decirse que los primeros pasos han sido rudimentarios y hasta casi inconscientes, la necesidad del hombre de transmitir sus conocimientos ha ido evolucionando hasta los actuales sistemas e instituciones de enseñanza.

Los modelos educativos han ido transformándose junto con la sociedad, adaptándose a los recursos y necesidades de cada época. Lo que probablemente surgió de manera espontánea, derivado de las necesidades básicas de supervivencia, fue tomando cuerpo y forma, como posterior anhelo de mantener viva una cultura, como medio de transmitir conocimientos recolectados y experiencias vividas, hasta nuestros días.

Tal como expresa Marrou en su ensayo sobre la Historia de la Educación en el Período Antiguo [MAR04], la educación es la técnica colectiva mediante la cual una sociedad inicia a su generación joven en los valores y en las técnicas que caracterizan la vida de su civilización; y dado que el educar es producto de la madurez de una sociedad, es entendible que su evolución sea paulatina y vaya fluctuando con el correr del tiempo.

El primer progreso significativo realizado por la sociedad fue la escolarización del proceso educativo, que puede verse en las primeras civilizaciones, donde determinadas artes, ciencias y leyes fueron transmitidas a un reducido grupo de individuos y persistidas por escrito.

La universalización del acceso a la educación, ya no como un privilegio de una minoría selecta, sino como un derecho ganado de cada individuo fue otro de los grandes hitos que marcaron su evolución en su proceso de desarrollo histórico. Otro hecho destacable en este largo trayecto es el giro de perspectiva, propia de la década de los setenta, que pasó a posicionar al alumno en un punto de privilegio: ya no como simple observador, sino con un rol central, activo y protagonista del hecho educativo.

En este sentido, el modelo de enseñanza transmisiva tradicional instalado en las instituciones educativas como herencia cultural de los siglos XIX y XX, ya no puede satisfacer los requerimientos de la actual sociedad. Nuevos desarrollos teóricos emergen y de esta forma, nuevas perspectivas para analizar los procesos educativos.

En los años '80 se produce una crisis de la perspectiva tecnocrática acerca de la enseñanza y surgen nuevas orientaciones como consecuencia de los replanteos epistemológicos en el campo educativo, con la introducción de la psicología cognitiva, del enfoque constructivista que supone un cambio radical frente al planteamiento objetivista de la tecnología educativa tradicional [PON09]. La polémica librada en el campo de la teoría y el consecuente ascenso del paradigma cognoscitivista refutó los principios en que se basaban sus propuestas iniciales.

En un nuevo contexto histórico caracterizado por la globalización, por grandes cambios desde el punto sociológico, tecnológico, político y cultural, que impactan directamente en el conocimiento y su significación social, surge una mirada crítica y se incorporan a su campo de estudio otras perspectivas de análisis provenientes de la filosofía, la lingüística y la sociopolítica.

Los nuevos desarrollos tecnológicos se combinan entonces con teorías y modelos pedagógicos y psicológicos, como la del paradigma socio-constructivista, los modelos de aprendizaje colaborativo entre pares, la teoría de redes y los aprendizajes sobre formación de comunidades de práctica que, durante el siglo XX, van a dar cuenta de otras formas de comprender los procesos de aprendizaje y enseñanza: ya no basados en el modelo docente-emisor y alumno-receptor sino como procesos formativos interactivos y de construcción social del conocimiento.

Estos modelos reciben de manera genérica el nombre de "*learning by doing*": el proceso de enseñanza y aprendizaje consiste en una búsqueda individual y colectiva de significado

y relevancia. El aprendizaje se adentra así en el área de la creatividad, la resolución de problemas y el análisis de situaciones. De este modo, las acciones formativas incorporan las ideas y experiencias de los estudiantes en situación de colaboración. El aprendizaje se plantea como un proceso de construcción social del conocimiento y la enseñanza como una mediación a este proceso [COB16]. Aunque la construcción del conocimiento es personal, tiene su origen en entornos de interacción social, mediante acciones dirigidas a metas socialmente reconocidas y utilizando un sistema de tecnología y conocimiento compartido [LAV91] [GOD96].

Entre tanto, las **TIC** como ya se ha mencionado, han ido evolucionando y han generado cambios profundos en todos los órdenes de la existencia humana al crear nuevas relaciones de interdependencia, de pensamiento y de conocimiento [COZ14].

La conjunción entre la disponibilidad de diversas tecnologías al alcance de grandes mayorías y la amplia difusión en su uso cotidiano, es una tendencia que viene en aumento desde hace algunos años atrás y que, en su acelerado crecimiento, contribuye a aumentar no sólo la cantidad y calidad de la información disponible, sino su modalidad de acceso y distribución, dando lugar a nuevos enfoques educativos y nuevas formas de vinculación entre los roles involucrados en el proceso del aprendizaje.

3.2. Concepto de Aprendizaje Colaborativo

Tal como hemos expresado, diversos han sido los paradigmas provenientes del campo de la psicología y de la educación surgidos a lo largo del siglo XX, que han impactado en mayor o menor medida en el ámbito educativo; todos ellos, surgidos como respuestas a problemáticas sociales, culturales y políticas, propias de cada período histórico.

Analizaremos aquí los aportes del paradigma socio-constructivista que surge a partir de los desarrollos teóricos de la psicología europea de entreguerras cuando autores como Piaget y Vygotsky, entre otros, oponen al auge del conductismo una concepción del sujeto humano radicalmente anti asociacionista, desarrollando la corriente organicista y estructuralista de la psicología cognitiva.

El aprendizaje cooperativo se refiere a un conjunto particular de técnicas de aula que fomentan la interdependencia del alumno como una ruta hacia el desarrollo cognitivo y

social. El aprendizaje colaborativo tiene una base filosófica basada en el "constructivismo social", que ve el aprendizaje como la construcción del conocimiento dentro de un contexto social y que, por lo tanto, alienta la aculturación de los individuos en una comunidad de aprendizaje [OXF97].

3.2.1. Teoría del Constructivismo Social

El constructivismo social surge a fines de los años sesenta definiendo un modelo a partir del cual el conocimiento no sólo se genera en base a la relación ambiente-yo, sino que se nutre de una suma de diversos factores sociales que rodean al individuo. Desde esta perspectiva, el sujeto es una concepción plural, producto de diversas relaciones sociales, donde las realidades se conciben como construcciones históricas y cotidianas, individuales y colectivas [COR08].

El conocimiento se construye a través de la experiencia, que conduce a la creación de esquemas. Los esquemas son modelos mentales que van cambiando, complejizándose y volviéndose más sofisticados a través de los procesos complementarios de asimilación y reestructuración [PIA95].

La propuesta educativa más innovadora de dicho paradigma fue, sin lugar a dudas, el aprendizaje colaborativo, también conocido como aprendizaje entre iguales o entre colegas. El aprendizaje colaborativo centra la educación en un plano social y establece que proceso educativo se basa en la sociabilización de cada individuo dentro de un grupo. De esta forma, el trabajo en equipo se convierte en la herramienta y el medio para alcanzar un objetivo: el de aprender. Como dos caras de una misma moneda, los psicólogos Johnson y Johnson afirman que el resultado en una situación colaborativa es la búsqueda de beneficios tanto individuales como grupales [JOH99]. De esta manera cada individuo, como parte de una unidad, es responsable no sólo de colaborar con sus habilidades y conocimientos, sino también de ayudar a descubrir en el otro sus fortalezas; mientras que con cada aporte individual el equipo se ve beneficiado con una perspectiva más amplia que lo que representaría el único punto de vista de uno de sus integrantes. De estos conceptos se puede deducir que la dinámica del trabajo colaborativo siempre está inmersa en la formación del ser humano, para desarrollar y cultivar sus facultades

como ser sociable y comunicativo, adaptándose a la comunidad a la cual pertenece y sintiéndose parte de dicha sociedad.

Sin embargo, es Delors [DEL96] quien se atreve a otorgarle a este aprendizaje un matiz marcadamente social, declarando la educación como medio de enseñar al individuo “el descubrimiento del otro” y de esta forma inculcar en la persona el respeto y la empatía por el prójimo contribuyendo en la toma de conciencia de las diversidades, las semejanzas y las interdependencias entre grupos humanos.

Uno de los puntos más sobresalientes de la educación colaborativa es como transforma lo que a simple vista podría interpretarse como un obstáculo, en una herramienta potencial de esta metodología. Es precisamente en la heterogeneidad de un grupo de alumnos donde Slavin hace hincapié, transformando la diversidad de sus integrantes, en recursos: donde el equipo se beneficia como consecuencia de los diversos puntos de vista, experiencias previas y estructuras de razonamiento opuestas que enriquecen al conjunto [SLA99].

Cada individuo aporta al grupo sus propios conocimientos y experiencia de vida, enriqueciendo el proceso y por ende a las personas involucradas en él. Los autores Scardamalia y Bereiter resumen esta idea cuando afirman que los estudiantes necesitan aprender profundamente y aprender cómo aprender, cómo formular preguntas y seguir líneas de investigación, de tal forma que ellos puedan construir nuevo conocimiento a partir de lo que conocen. El conocimiento propio que es discutido en grupo, motiva la construcción de nuevo conocimiento [SCA06].

Johnson y Johnson, en su libro “El aprendizaje cooperativo en el aula” [JOH99], respaldan la educación colaborativa, en yuxtaposición con la educación competitiva e individualista, y enuncian los siguientes ítems como logros de este enfoque:

- Mayores esfuerzos por lograr un buen desempeño: esto incluye un rendimiento más elevado y una mayor productividad por parte de todos los alumnos (ya sean de alto, medio o bajo rendimiento), mayor posibilidad de retención a largo plazo, motivación intrínseca para lograr un alto rendimiento, más tiempo dedicado a las tareas, un nivel superior de razonamiento y pensamiento crítico.

- Relaciones más positivas entre los alumnos: esto incluye un incremento del espíritu de equipo, relaciones solidarias y comprometidas, respaldo personal y escolar, valoración de la diversidad y cohesión.
- Mayor salud mental: esto incluye un ajuste psicológico general, fortalecimiento del yo, desarrollo social, integración, autoestima, sentido de la propia identidad y capacidad de enfrentar la adversidad y las tensiones.

Para comprender el aprendizaje colaborativo, es necesario entender cómo se produce. Comprende, además de productos cognitivos logrados a nivel interpersonal e intrapersonal, la valoración y responsabilidad hacia el proceso educativo. Algunas pautas para lograr aprendizaje colaborativo son [CAL02]:

- Estudio pormenorizado de las capacidades, debilidades y posibilidades de cada miembro del equipo.
- Establecimiento de metas conjuntas que incluyan metas individuales.
- Elaboración de un plan de acción con responsabilidades específicas y encuentros para la evaluación del proceso.
- Chequeo constante de progreso del equipo, a nivel individual y grupal.
- Cuidado de relaciones socio-afectivas, sentido de pertenencia, respeto y solidaridad.
- Discusiones progresivas de acuerdo con el objetivo perseguido.

Existen distintas formas de poner en práctica el aprendizaje colaborativo: La interacción entre pares, el tutorio de pares y los grupos colaborativos [TUD93].

- *La interacción entre pares* es la integración de participantes con diferentes niveles de habilidad realizando actividades de forma organizada, con un docente como mediador.
- *El tutorio de pares* involucra a estudiantes en los que se detectó mayor habilidad que ofician de *coach* de sus compañeros, mientras trabajan en forma conjunta. La interacción entre los estudiantes es tan fluida que eleva los niveles de los aprendices y consolida la profundización en el conocimiento.

- *Los grupos colaborativos* están integrados por aprendices de distinto nivel de habilidad, género y procedencia, acumulan puntaje de forma individual y grupal, lo que incentiva y estimula la interdependencia y la preocupación por aprender, ya que el éxito colectivo depende del éxito individual. En este caso el docente debe ser más que un mediador, ya que debe facilitar un proceso grupal efectivo.

3.2.2. La Interacción Social y su Influencia en el Desarrollo

Los entornos de aprendizaje constructivistas apoyan la construcción colaborativa del aprendizaje a través de la negociación social, no de la competición entre los estudiantes para obtener apreciación y conocimiento [JON99].

Las investigaciones llevadas a cabo en relación con los procesos de aprendizaje han demostrado los beneficios obtenidos cuando se emplean ambientes colaborativos, en los que el alumno aprende a construir conocimiento mediante la combinación de experiencias, vivencias, interpretaciones e interacciones con otros integrantes, donde desarrolla el sentimiento de pertenencia a un equipo por el cual se hará responsable, dejando atrás su individualidad en función de un objetivo en común.

Las acciones llevadas a cabo por el individuo, sumadas a la comunicación entre pares, se ven netamente beneficiadas por la garantía y la calidad del conocimiento colectivo generado en la interacción social y en la colaboración, frente al construido individualmente. Es precisamente este aspecto el que enriquece los factores comunicativos y su impacto en la modalidad de enseñanza cooperativa.

Dado que el aprendizaje colaborativo está centrado en el diálogo, la negociación, la palabra y la explicación, genera un ámbito enriquecido y centrado en las relaciones y vivencias de estudiantes y docentes que, combinando el trabajo individualizado con el trabajo en equipo, estimulan el desarrollo de tareas de investigación y generan proyectos comunitarios, evidenciando de esta manera el fruto del trabajo cooperativo.

El papel del estudiante, como parte integrante de un equipo de pertenencia, permite desarrollar su madurez como un ser crítico y autónomo en sus pensamientos, formando su carácter social con el fin de tener una actitud proactiva frente al trabajo grupal, para distinguirse por las ideas que aporta y por las acciones que realiza en beneficio del grupo

al que pertenece. De esta forma, las actividades colaborativas representan la clave en el proceso de aprendizaje donde cada uno se beneficia de la actividad mental de todos; así, las ideas de unos impulsan y alimentan las de los otros.

El proceso de aprendizaje no es un proceso estático, limitado a la adquisición de conceptos y hábitos de manera sistemática, sino que es consecuencia de la interacción entre el aprendizaje y el desarrollo de cada individuo. Distintos trabajos parafrasean a Koffka [KOF25] quien asevera que el desarrollo supera y hasta engloba los procesos de aprendizaje, dando lugar a un progreso evolutivo de maduración, donde el aprender permite crear principios estructurales, aplicables a operaciones distintas de las que le dieron origen.

Vygotski profundiza aún más este concepto cuando toma las teorías de Koffka y las revaloriza incluyendo un nuevo factor: la edad o nivel de desarrollo mental; y es a partir de este último elemento que el autor desarrolla su concepto principal: La Zona de Desarrollo Próximo.

La zona de desarrollo próximo, conocida en las esferas de la psicología simplemente como **ZDP**, es la distancia que existe entre el nivel real de desarrollo (determinado por la capacidad de un individuo de resolver un problema o tarea de forma independiente) y el nivel de desarrollo potencial, a través de la resolución de ese mismo problema o tarea mediante la interacción de un compañero más experimentado o un tutor [VIG97].

De esta manera, queda evidenciado el potencial que representa la interacción social de la figura de un tutor o un par en el proceso de aprendizaje, donde el límite de la creación de conocimiento, se alza aún más si aquello que el alumno no era capaz de resolver o dilucidar por sí solo, puede ser alcanzado mediante la guía o el trabajo en equipo.

Por otro lado, debe tenerse en cuenta que una vez que se crearon los principios estructurales que define Koffka, aquello que en un inicio el individuo no era capaz de resolver por sí sólo, pero sí con ayuda, será resuelto de manera independiente como resultado del proceso de maduración y desarrollo evolutivo. En palabras del propio Vygotski, el aprendizaje despierta una serie de procesos evolutivos internos capaces de operar sólo cuando el niño está en interacción con las personas de su entorno y en cooperación con algún semejante [VIG97].

3.2.2.1. La Zona de Desarrollo Próximo

El aprendizaje colaborativo se fundamenta en una perspectiva teórica de corte socio-constructivista que no se ajusta a otorgar al tutor la parte central del proceso instruccional ni tampoco al alumno, sino que defiende que la actividad constructiva es resultante de la interacción entre ellos [IBA04]. En este punto el aprendiz requiere la acción de un agente mediador para acceder a la Zona de Desarrollo Próximo (**ZDP**). Como plantea Vygotski, la **ZDP** es la distancia que existe entre el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema, y el nivel de desarrollo potencial, determinado a través de la resolución de problemas bajo la guía de un adulto o en colaboración con otro compañero más capaz [VIG97].

Esto es, tratar de identificar los niveles de intervención en el desarrollo de una persona: lo que se puede hacer solo, es decir, sin asistencia de otros y medible por la evaluación convencional, y aquello que se puede aprender y hacer en el futuro gracias a la cooperación de otras personas más capaces o de mayor experiencia, resultando en mayores posibilidades de aprendizaje.

La **ZDP** nos permite entender el papel del agente mediador como guía (alguien más capaz, como un profesor) o la cooperación de un par, como un compañero de equipo cooperativo. Esto último, destacado por:

- El contraste entre diferentes puntos de vista, en relación con una tarea.
- La oportunidad de explicar el propio punto de vista.
- La coordinación entre roles para controlar el trabajo de forma cooperativa.

El proceso de aprendizaje colaborativo brinda la posibilidad de que cada uno de los integrantes del equipo pueda regular sus acciones, organizar sus tareas, definir sus tiempos de aprendizaje, sus obligaciones y su nivel de conversación [GUE04].

La **ZDP** en equipos de trabajo es un factor clave. Se podría entender como la distancia entre lo que un grupo puede realizar por sí mismo en relación a una tarea, guiada por un tutor externo [ILL01]. La coordinación de las **ZDP** individuales está relacionada con la sub división de tareas, es decir, si las tareas son independientes, la **ZDP** también puede serlo.

El uso de las **TIC** y su constante evolución, propician un escenario adecuado para la **ZDP**, en donde las diferentes herramientas provistas, por ejemplo, por un entorno virtual

colaborativo, permiten la comunicación de forma sincrónica y asincrónica, dejando de lado el problema de la distancia. Este espacio virtual fomenta la interacción colaborativa, haciendo de este un contexto factible desde lo tecnológico y ventajoso desde lo educativo, ya que compartir conocimiento y colaborar durante la resolución de un problema ayuda a reducir la brecha entre el desarrollo individual y los conocimientos adquiridos por el desarrollo centrado en el trabajo grupal con un objetivo común.

El entorno tecnológico, los materiales de aprendizaje y las acciones e interacciones se constituyen en mediadores pedagógicos, que intervienen en el hecho educativo, posibilitando los procesos de enseñanza y aprendizaje. La inclusión de las **TIC** transforma la relación educativa al mediar en ella. Tal como lo señala Suárez Guerrero, esta modificación se genera a partir de una estructura básica de acción tecnológica que facilita el procesamiento, gestión y distribución de la información, agregando a la relación educativa nuevas posibilidades y limitaciones [SUA03].

La acción tecnológica no puede ser entendida como un simple dispositivo técnico o como un objeto desprovisto de valoración y significado. La acción mediadora de las tecnologías, configuradas en los entornos virtuales de aprendizaje, promueve determinados y específicos marcos de pensamiento y estructuras de acción externas que favorecen determinados procesos de aprendizaje en los sujetos.

Retomando a Vygotski, el proceso de aprendizaje consiste en una internalización progresiva de instrumentos mediadores y el nivel de desarrollo potencial estaría constituido por lo que el sujeto sería capaz de hacer, con ayuda de otras personas o de instrumentos mediadores. A través de estos instrumentos el sujeto modifica los estímulos, no se limita a responder mecánicamente, sino que actúa sobre ellos, a través de la interacción con el entorno. Este autor distingue dos clases de instrumentos: físicos (herramientas) y simbólicos (sistemas de signos) que la cultura proporciona al sujeto y le permiten regular su actividad y mediar su relación con el mundo, con los hombres y consigo mismo.

Mientras que las herramientas se constituyen en conductores de la influencia humana sobre el objeto de la actividad y, por ende, están externamente orientadas, los signos están internamente orientados y procuran la regulación de los procesos psicológicos,

modifican al sujeto y, a través de éste, al objeto, y dan origen a los significados. Esta doble orientación podemos encontrarla en un mismo instrumento de mediación.

En este marco, concebimos al entorno virtual de aprendizaje como un mediador instrumental que, como herramienta, permite orientar la actividad externa del sujeto y, como sistema de signos (propuestas de enseñanza), le posibilita autorregular su propia actividad modificando sus marcos de pensamiento.

En este sentido, el desarrollo potencial de los estudiantes es el que debe atraer el mayor interés de los educadores, pues de acuerdo con esta teoría de aprendizaje, tienen una importancia relevante los procesos de instrucción externa de mediadores, para su posterior internalización. Esta concepción del aula virtual, los materiales y los medios de comunicación que la conforman como mediadores instrumentales que delinear sistemas de actuación (acción externa) y a la vez generan nuevos marcos de pensamiento (representaciones internas) obliga a pensar qué se pretende que los estudiantes aprendan y cómo, en el momento de diseñar su estructura. “El poder real de la tecnología radica en su capacidad de redefinir y reestructurar de forma fundamental lo que hacemos (...), cómo lo hacemos y cuándo lo hacemos. Llegamos a usar esta tecnología como una herramienta para pensar” [SAL92].

En el modelo de educación tradicional la figura del docente ocupa un rol dominante en el proceso. Esto sumado a las prácticas habituales de ejercitación individual que destacaba los logros individuales y la competencia por sobre los logros grupales y la colaboración para conseguir los objetivos planteados.

La implantación de un sistema de aprendizaje colaborativo (una práctica grupal) no solo traslada la responsabilidad al grupo, sino que brinda a cada participante la posibilidad de proponer diferentes situaciones que converjan en un resultado sinérgico del proceso.

3.3. Los Entornos Virtuales de Enseñanza y Aprendizaje

Las entidades educativas son quienes se encuentran frente al desafío de implementar nuevas propuestas que puedan coexistir con los modelos clásicos de educación presencial. Las TIC son un medio eficaz para avanzar hacia la redefinición de nuevas estrategias que permitan integrar al modelo presencial las tecnologías de educación online colaborativa [ROM10].

Destruídos los límites geográficos y temporales, el sector educativo se encuentra actualmente en un punto estratégico de desvinculación definitiva con los sistemas tradicionales, donde internet brinda la infraestructura necesaria para desarrollar procesos de aprendizaje no presenciales, mediante servicios asincrónicos.

Este cambio revolucionario, que comenzó varias décadas atrás, está desdibujando de forma creciente los límites físicos del aula como espacio de enseñanza convencional. Los nuevos tiempos han generado nuevos actores (Internet, telefonía móvil, realidad aumentada, geolocalización digital) que están modificando nuestra experiencia. S. Bauman sugiere que el tiempo actual – de la cultura digital- es un fluido de producción de información y conocimiento inestable, en permanente cambio, en constante transformación como contraposición a la producción cultural desarrollada en los siglos XIX y XX donde primó la estabilidad e inalterabilidad de lo físico, de lo material [BAU06].

La incorporación en las aulas universitarias de estas nuevas tecnologías supone necesariamente una auténtica revolución metodológica. Los recursos de la web cada vez más participativa, abierta, colaborativa, intuitiva y gratuita ayudan a estos cambios de metodología. En este sentido, las relaciones tradiciones de espacio y tiempo se están rompiendo y en las aulas empiezan a converger con normalidad sistemas de formación mixtos (*blended learning*). Por modalidad mixta se entiende un formato de enseñanza y aprendizaje en el cual una parte del tiempo el curso o asignatura se desarrolla de manera tradicional, en el aula y la otra parte se lleva a cabo en línea [MAS09]. Esta modalidad es una alternativa de la modalidad tradicional de clases teóricas/exámenes programados que se usa en la mayoría de los cursos en el presente. En esta modalidad mixta (presencial/en línea) los estudiantes asisten al aula para tener sesiones de discusión guiadas por el profesor, en base al contenido del curso en la red.

Por supuesto que esta innovación en el ámbito educativo implica un esfuerzo no solo de parte del docente a cargo, sino también del alumno, como detallan a continuación los mismos autores [MAS09].: “Esto requiere acciones educativas relacionadas con el uso, selección, utilización y organización de la información, de manera que el alumno vaya formándose como un maduro ciudadano de la sociedad de la información”.

El apoyo y la orientación que recibirá en cada situación, así como las diferentes posibilidades respecto de la disponibilidad y el acceso a la tecnología son elementos cruciales en la utilización de las TIC para actividades de formación.

El avance y la disponibilidad de los recursos tecnológicos han impactado el ambiente educativo, colaborando en la transformación de sus metodologías y modalidades. García Aretio menciona diferentes etapas o generaciones en el desarrollo de la educación a distancia que han aportado de forma significativa en la revolución de la educación tradicional: la enseñanza por correspondencia, la enseñanza multimedia, la enseñanza telemática, la enseñanza vía Internet (*e-learning*) y la enseñanza ubicua (*u-learning*) [GAR99].

La **enseñanza por correspondencia**, situada entre finales del siglo XIX y principios del siglo XX, fue la primera aproximación a *salir del aula*. Utilizando el correo postal como alternativa, dio inicio a la era de la educación a distancia. Aunque lenta en un principio, permitió evolucionar en el ambiente educativo incursionando por primera vez en las metodologías didácticas, incorporando la figura de un tutor y complementando los clásicos textos con guías de ejercicios y autoevaluación.

Ya a finales de los años sesenta, la educación a distancia se vio nuevamente beneficiada por la difusión de los medios masivos de comunicación, desarrollándose la **etapa de la enseñanza multimedia**. Radio y televisión son las insignias de esta etapa. Recursos audiovisuales como diapositivas, audios y videos, se sumaron al material de estudio, ampliando el material didáctico y transformando la forma de distribución del conocimiento. El texto escrito comienza a ser apoyado por otros recursos audiovisuales.

Cabe destacar que ambas generaciones fueron ricas en diseño y desarrollo de documentación didáctica, pero pobres en la relación entre pares perdiendo, de esta manera, la riqueza aportada por el trabajo en equipo y el aprendizaje colaborativo, beneficios que sólo quedaron vinculados al espacio físico del aula.

Cuando hablamos de **enseñanza telemática**, nos referimos a la formación mediada por las redes de comunicación. La integración de las telecomunicaciones con otros medios educativos mediante la informática define esta etapa. Se apoya en un uso cada vez más generalizado del ordenador y de sistemas multimedia (hipertexto, hipermedia), de teleconferencias y videoconferencias.

Hacia fines de la década del '90 se sitúa el desarrollo de la enseñanza vía Internet (e-learning) **que hace referencia a la cuarta generación**, la cual centra en un modelo de aprendizaje flexible, en el uso del multimedia interactivo y la comunicación mediada por computadora.

Así, se rompe la última de las barreras físicas hacia un mundo integrado donde alumnos, en distintas ubicaciones, interactúan entre sí, en un medio flexible, ágil y de respuesta inmediata. De esta manera, nace el concepto de **campus virtual**, espacio en el que se generan e intercambian contenidos, se gestionan actividades y se establecen vínculos entre alumnos, profesores y propuestas de enseñanza.

Inicialmente el aprendizaje a distancia puso foco en el aprendizaje individual y muy poco énfasis en desarrollar capacidades colaborativas. Con el correr del tiempo, los estudiantes comenzaron a trabajar en grupo y esto fue extendiéndose hasta lograr que las plataformas o entornos de aprendizaje lo consideren en mayor escala. Con aislamiento social e intelectual, los estudiantes pueden no desarrollar y perfeccionar las habilidades cognitivas e interpersonales cada vez más necesarias para las carreras profesionales y empresariales [CON00].

A modo de comparación, mostramos en la siguiente tabla el estudio realizado por Cabero [CAB06b], en el que se observan las características distintivas de la formación virtual.

Formación Virtual	Formación Presencial
Permite que los estudiantes manejen su propio ritmo de aprendizaje.	Parte de una base de conocimiento y el estudiante debe ajustarse a ella.
Basada en el concepto de "Formación en el momento que se la necesita"	Los docentes determinan cuando y como los estudiantes reciben materiales formativos.
Permite la combinación de diferentes materiales (audiovisuales, visuales, auditivos).	El sujeto recibe pasivamente el conocimiento para generar actitudes innovadoras, críticas e investigadoras.

Con una sola aplicación se puede atender a un mayor número de estudiantes.	Se apoya en materiales impresos y en el profesor como fuente de presentación y estructuración de la información.
El conocimiento es un proceso activo de construcción.	Tiende a un modelo lineal de comunicación.
Tiende a reducir el tiempo de formación de las personas.	La comunicación se desarrolla entre el profesor y el estudiante.
Tiende a ser interactiva, tanto entre los participantes en el proceso (profesor y estudiantes) como con los contenidos.	La enseñanza se desarrolla de forma preferentemente grupal.
Tiende a realizarse de forma individual, sin que ello signifique la renuncia a las actividades colaborativas.	Solo puede realizarse en un mismo momento y lugar.
Puede ser utilizada en el tiempo disponible del estudiante.	Se desarrolla en momento fijo en aulas específicas.
Es Flexible.	Tiende a la rigidez temporal.
Su puesta en funcionamiento requiere de recursos no siempre disponibles.	Recursos estructurales y organizativos disponibles.

Tabla 3.1. Características y Comparación entre Formación presencial y virtual.

A continuación, se enuncian las ventajas y desventajas del desarrollo de propuestas de enseñanza en entornos virtuales de formación [CAB06b] [DOR06].

Ventajas	Desventajas
Pone a disposición del alumno un amplio volumen de información.	Requiere más inversión y tiempo del profesor.
Fácil de actualizar la información y contenidos.	Requiere competencias tecnológicas por parte del profesor y el alumno.

Flexibiliza la información independientemente del tiempo y del espacio.	Los estudiantes deben tener habilidades para el aprendizaje autónomo.
Permite la deslocalización del contenido.	Puede disminuir la calidad de la formación si no se realiza el seguimiento adecuado.
Facilita la autonomía del estudiante.	Requiere más trabajo que la enseñanza convencional.
Propicia una formación “ <i>just in time</i> ” y “ <i>just for me</i> ”.	Resistencia al cambio del Sistema tradicional.
Ofrece diferentes herramientas sincrónicas y asincrónicas de comunicación.	Inexistencia de referencias físicas.
Se puede realizar una formación grupal y colaborativa.	Falta de formación del profesorado.
	Problemas de seguridad (informática, de identidad).
	Falta de experiencia en su uso.
	Inexistencia de entornos propios en áreas de conocimiento específico.

Tabla 3.2. Ventajas y Desventajas de la Educación Virtual

La expansión de las tecnologías de la web, entre los años 1998 y 2001 generó una ola de entusiasmo que sacudió el escenario educativo. Una importante cantidad de recursos se movilizó hacia la industria educativa con renovadas promesas de transformación y revolución. Sin embargo, pocos fueron los cambios que se produjeron a pesar de las renovadas promesas que le fueron adjudicadas a la tecnología web, no pudiéndose utilizar esta tecnología para encarnar ideas pedagógicas innovadoras, superadoras del modelo transmisivo en la educación.

A partir del año 2004 se consolida lo que se denominó web 2.0 que presenta como rasgo diferencial la oportunidad para los usuarios de Internet de crear el contenido de la red, de interactuar con otros creando contenido en forma conjunta y participar en las redes sociales en línea. Estas posibilidades que ofrecen las tecnologías 2.0 de facilitar la interacción entre pares se combinan y potencian con las teorías y modelos pedagógicos y psicológicos que durante el siglo XX dieron cuenta de otras formas de comprender los procesos de enseñanza y aprendizaje, que se alejaron del modelo tradicional emisor-receptor.

Como asevera Octavio Islas, las figuras de “emisor” y “receptor”, así como la mayoría de los modelos que ayer permitían explicar el proceso comunicativo, hoy exhiben evidentes limitaciones en sus capacidades explicativas. En su trabajo utiliza el lema es “*anyone, anywhere, anytime*”, es decir, cualquier persona, en cualquier lugar, en cualquier momento, para describir a “la sociedad de la ubicuidad” [ISL08]. Esto lo podemos ver constantemente ante el hecho de disponer de una inmensurable lista de servicios accesibles desde diversos dispositivos, por medio de redes de banda ancha disponibles con tecnologías en constante evolución.

La **enseñanza ubicua** es una transformación del paradigma anterior. Sabemos que la constante expansión de las TIC permitió el paso del aprendizaje electrónico (*e-learning*) hacia lo que hoy se denomina el aprendizaje móvil (*m-learning*) o aprendizaje ubicuo. Este paradigma propone la creación de un ambiente de aprendizaje que le permita a un estudiante aprender, en cualquier momento y lugar.

El **aprendizaje ubicuo** tiene un conjunto de características que lo identifica [YAJ10]:

- La permanencia de la información
- La accesibilidad a la información, que estará disponible siempre que sea de importancia para el alumno.
- La inmediatez, que representa el tiempo de respuesta que perciben los usuarios para recuperar la información.
- La Interactividad que tienen los estudiantes con sus pares y docentes con eficacia y eficiencia.

- La adaptabilidad al contexto, para brindarle a los estudiantes la información adecuada según sus necesidades.

Se debe considerar que ninguna de estas características pone como restricción el medio de comunicación o el dispositivo. Es simplemente una abstracción que permite comprender la independencia de la tecnología para lograr el resultado, es decir, generando una capa tecnológica aislada y omnipresente.

Las tecnologías se conciben a partir de ahora como un nuevo escenario, un entorno de enseñanza, una nueva dimensión que posibilita el desarrollo de procesos de construcción del conocimiento a través del intercambio y la colaboración entre pares, las fuentes de acceso a la información y las intervenciones del docente.

Los entornos virtuales de enseñanza y aprendizaje se constituyen en espacios para realizar acciones formativas en red, brindando más flexibilidad y rompiendo las barreras espacio-temporales.

La elasticidad en cuanto al manejo de los tiempos, la ausencia del requisito de asistencia periódica, la posibilidad de continuar con los estudios desde cualquier lugar en el que el alumno se vea obligado a trasladarse, ya sea por razones laborales o personales; en definitiva, el alto grado de autonomía que caracterizan al sistema, constituye a las plataformas educativas o **LMS** (*Learning Management System*) en una opción apropiada para los tiempos y necesidades de la vida actual, permitiendo compatibilizar las exigencias de capacitación con las limitaciones que imponen las rutinas diarias.

Es importante mencionar que las plataformas de aprendizaje son herramientas que no sólo pueden ayudar a los estudiantes a aprender conceptos nuevos sino también a afianzar y compartir conocimientos y habilidades aumentando, de esta forma, la interacción y la motivación.

Al igual que en el proceso de enseñanza presencial, el estudiante lee textos, debate con sus pares, realiza ejercicios y trabajos prácticos de forma individual o grupal y formula inquietudes al docente. Y de esa misma manera, el lugar espacial, no físico, es planificado, supervisado y dirigido por el profesor.

Cabero propone la siguiente red de aprendizaje en entornos de aprendizaje virtuales, en donde alumnos y tutores están en constante relación con las actividades a realizar y los

contenidos teóricos del modelo. Los roles intervinientes (Alumnos y Tutores) tienen estrecha relación entre ellos y comparten vínculos con los Objetos de Aprendizaje (Contenidos teóricos) y las Actividades a realizar (en grupo, evaluaciones, etc.) [CAB06b].

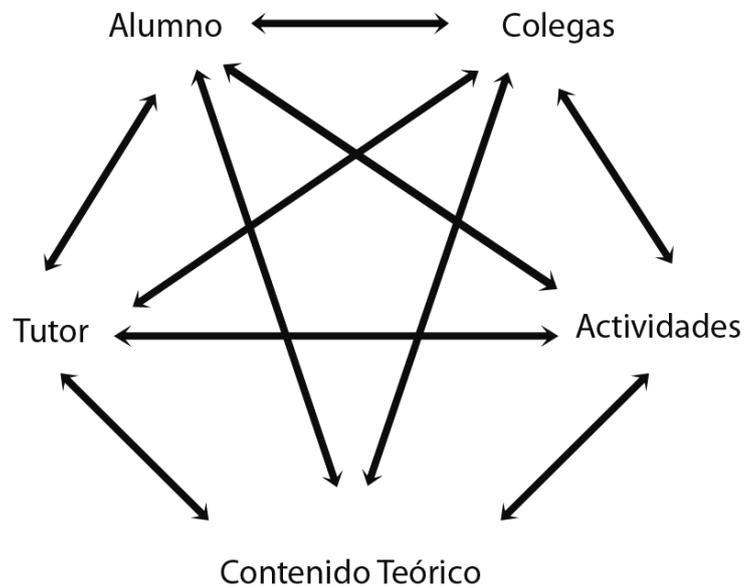


Figura 3.1. Red de Aprendizaje

Los autores españoles M. Area Moreira y J. Adell, establecen cuatro dimensiones presentes en un aula virtual [ARE09]:

- **Dimensión informativa.** Hace referencia al conjunto de recursos compuestos por materiales y elementos que incluyen desde textos y enlaces, hasta presentaciones multimedia, gráficos, mapas conceptuales y videos.
- **Dimensión práctica.** Esta dimensión involucra todas las tareas y actividades que debe desarrollar el alumno y que fueron planificadas por el docente para guiar el camino del aprendizaje; conjugando y balanceando procesos de enseñanza por recepción (cuando las tareas planteadas son de carácter repetitivo) y procesos de aprendizaje constructivo (cuando las tareas a desarrollar se basan en la búsqueda y el análisis de información).
- **Dimensión comunicativa.** Esta dimensión hace referencia a la interacción entre estudiantes y docentes; es el elemento más importante en la educación

colaborativa. Su grado de implementación determinará la motivación, la implicación y el rendimiento de los alumnos y, por ende, puede afirmarse sin lugar a dudas que es directamente proporcional a la calidad de la educación impartida. Sin este factor, el aula se ve notablemente empobrecida, convirtiéndose en un mero repositorio de información carente del estímulo y la fluidez de las relaciones humanas. Herramientas como los foros, las videoconferencias, los chats y la mensajería nutren al aula virtual con la participación social de sus individuos.

- Dimensión tutorial y evaluativa. Hacer referencia a la función del docente en su carácter de tutor y guía en el proceso de creación del conocimiento: motivando al alumno, reforzando su participación, dinamizando actividades y reconociendo esfuerzos o demandando mayor dedicación.

De la misma forma que Area Moreira y Adell describen y pautan las dimensiones presentes en el aula virtual, también las establece Barberá bajo los rótulos de planificación, consulta, comunicación y seguimiento [BAR08]; donde la planificación guarda relación con la dimensión práctica; la consulta, con la dimensión informativa; la comunicación, con la dimensión comunicativa y el seguimiento se correlaciona con la dimensión evaluativa.

De estos autores, se desprende que los ejes que articulan un aula virtual giran en torno a los materiales provistos, los actores del proceso de aprendizaje, las tareas que componen dicho proceso y la comunicación como el factor de impulso para involucrar y motivar al alumno en su propio proceso de aprendizaje y creación de conocimiento. Si se deja de lado, sólo por un instante, el término virtual se verá que dichas dimensiones son las mismas que se materializan en un aula física de la enseñanza tradicional.

Los entornos virtuales requieren entonces de diferentes herramientas de comunicación y colaboración, entre las que se encuentran:

- Herramientas de comunicación asincrónica para crear grupos de discusión sobre un tema determinado o alrededor de una actividad específica, una meta o un proyecto.
- Herramientas de comunicación sincrónica para lograr retroalimentación inmediata y la creación de lazos entre los miembros.

- Herramientas para compartir datos y documentos, con el objetivo de centrar el interés en un tema en particular y definir así una especialidad común.
- Herramientas de edición colaborativa para la edición de un documento por múltiples usuarios.

Independientemente de las particularidades finales de cada entorno, pueden hallarse características comunes que deben prevalecer en toda plataforma educativa. Boneau cita cuatro características fundamentales que deben estar presentes en toda plataforma: interactividad, flexibilidad, escalabilidad y estandarización [BON07].

- Por interactividad se entiende el grado de responsabilidad y el compromiso que siente el usuario como protagonista de su propia educación.
- La flexibilidad hace referencia al conjunto de funcionalidades que otorgan la libertad de adaptar el sistema a los diversos tipos de instituciones, a sus planes de estudio, a su contenido académico y a sus objetivos y pautas pedagógicas.
- Cuando se menciona la escalabilidad, se está hablando de la contemplación del crecimiento futuro de la plataforma. Es decir que siga respondiendo de la misma forma a pesar del incremento del grupo de usuarios que la utilice.
- Finalmente, se entiende por estandarización la posibilidad de incorporar información o funcionalidades de otras plataformas y que éstas se hallen disponibles y funcionando de manera correcta.

Tal como se ha expresado, el desarrollo de las tecnologías de la información y comunicación sumado al advenimiento de las tecnologías 2.0 como facilitadoras y dinamizadoras de la interacción entre pares propicia un ambiente muy adecuado para el desarrollo de propuestas de enseñanza y aprendizaje colaborativo. En este sentido, el trabajo colaborativo puede realizarse como parte del proceso de aprendizaje, utilizando diversas herramientas y recursos tecnológicos virtuales para el intercambio y la cooperación.

El proceso de enseñanza y aprendizaje colaborativo, mediado por las TIC, presenta ventajas de gran relevancia en tanto posibilitan [NEI16] [BAT16] [CAL02]:

- Estimular la comunicación interpersonal para facilitar e incentivar el intercambio de información y discusión entre los implicados en el proceso.

Para esto existen diferentes herramientas de comunicación sincrónica (como chats o sistemas de videoconferencias) o asincrónicas (como foros o listas de discusión).

- Facilitar el trabajo colaborativo ya que permite compartir archivos, editar documentos de forma conjunta, compartir aplicaciones más específicas, calendarios, mapas conceptuales, notas, pizarras y demás herramientas destinadas a la cooperación y colaboración.
- Realizar seguimientos de progreso de forma individual y grupal por medio de trabajos, ejercicios, indicadores de uso y otras métricas específicas a cada aplicación.
- Acceder a la información y contenidos de aprendizaje de forma ubicua. En cualquier momento y lugar el alumno dispondrá de acceso a contenidos, libros, publicaciones, enciclopedias, prácticas de laboratorio y tutoriales para intercambio de recursos e integración de puntos de vista.
- Gestionar y administrar los procesos vinculados a la vida académica de los estudiantes para obtener información valiosa para el seguimiento del proceso educativo.
- Crear ejercicios de evaluación y autoevaluación de forma tal de medir el rendimiento de los alumnos y el desempeño en diferentes áreas, de acuerdo a un contexto determinado.

Las actividades que contienen un componente complejo de acción y toma de decisiones y aquellas que, además, contienen un componente tecnológico importante o bien, que requieren del uso de otros recursos externos, son candidatas para su división en tareas interdependientes, y por su naturaleza están orientadas a ser ejecutadas por más de una persona [ILL01]. El alumno puede ajustar su planificación a sus necesidades y enriquecerse del trabajo colaborativo durante sesiones que pueden ser desde una simple charla en tiempo real, a un complejo proceso específico como puede ser alguno planteado por la Ingeniería de software.

3.3.1 Cambios en los Roles del Docente y del Alumno

En los modelos tradicionales de enseñanza y aprendizaje, el docente o experto en la materia lidera los procesos de enseñanza y aprendizaje, y conduce al alumno por medio del desarrollo de un tema objeto de estudio. El alumno cumple un rol más pasivo en donde su desarrollo está centrado en el docente [VIT08].

Los cambios en el rol del profesor se centran en la función que desempeña en el sistema de enseñanza y aprendizaje. Se suele aceptar que el docente cambia desde el punto de vista de la transmisión del conocimiento a los alumnos para ser un mediador en la construcción del propio conocimiento [ILL01]. Este nuevo rol del docente deja de ser el foco para transferirle esa responsabilidad al alumno, colocándolo en el centro de atención en la que el docente juega un papel decisivo. Esto se plantea en [BAR06] como el paso de la “Enseñanza centrada en el docente” (en donde el docente adquiriría el conocimiento fuera del contexto y lo transmitiría directamente a los alumnos) a la “Enseñanza centrada en el Alumno” (en donde el alumno es muy activo en el proceso de aprender, poniendo mucho énfasis en el proceso y en el aprendizaje por medio de los errores). La institución educativa y el profesor dejan de ser fuente de conocimiento y el profesor pasará a ser la guía de los alumnos, como facilitador de recursos.

Según Salinas, [SAL04] si bien el rol del personal docente también cambia en un ambiente rico en TIC, el profesor deja de ser fuente de todo conocimiento y pasa a actuar como guía de los alumnos, facilitándoles el uso de los recursos y las herramientas que necesitan para explorar y elaborar nuevos conocimientos y destrezas; pasa a actuar como gestor de la pléyade de recursos de aprendizaje y a acentuar su papel de orientador y mediador. A pesar de contar con el apoyo de las herramientas y su correcta aplicación en los sistemas, esto no es suficiente para obtener resultados exitosos, ya que uno de los participantes principales en la enseñanza es el docente, quien dirige el proceso de aprendizaje y debe tener el suficiente conocimiento en las tecnologías para combinarlas junto con las técnicas tradicionales para guiar y orientar a los alumnos.

El rol del alumno debe ser diferente al tradicional, que se enfoca en la adquisición y acumulación de la mayor cantidad de conocimiento posible, aunque se sabe que en un mundo cambiante esto no es eficiente. Los alumnos en contacto con las TIC tienen muchas herramientas para avanzar en esta nueva visión del alumno-usuario. Esto

requiere acciones relacionadas con la selección y utilización de la información y la disponibilidad tecnológica, que son elementos fundamentales en la explotación de TIC para actividades de formación

Tanto el docente como el alumno actúan de distinta manera en una clase presencial como en una virtual. Para esto, las instituciones educativas necesitan involucrarse en los procesos de innovación y apoyarse en las TIC para afrontar el impacto de la era de la información. Sin el apoyo institucional, los proyectos impulsados por docentes entusiastas no suelen ser efectivos, por este motivo es imprescindible que las instituciones se enfrenten a la explotación de las TIC no solo para crear un entorno virtual de enseñanza y aprendizaje, sino para fomentar su uso a los alumnos y capacitar a los docentes en esta perspectiva pedagógica de enseñanza.

3.3.2 Variables Críticas en los Entornos Virtuales de Enseñanza

Cabero [CAB06b] propone ocho variables críticas que garantizan el éxito de las acciones formativas virtuales. Si bien la tecnología es otra variable crítica, no se incluye en el siguiente modelo porque es la que da comienzo a este tipo de acción formativa.

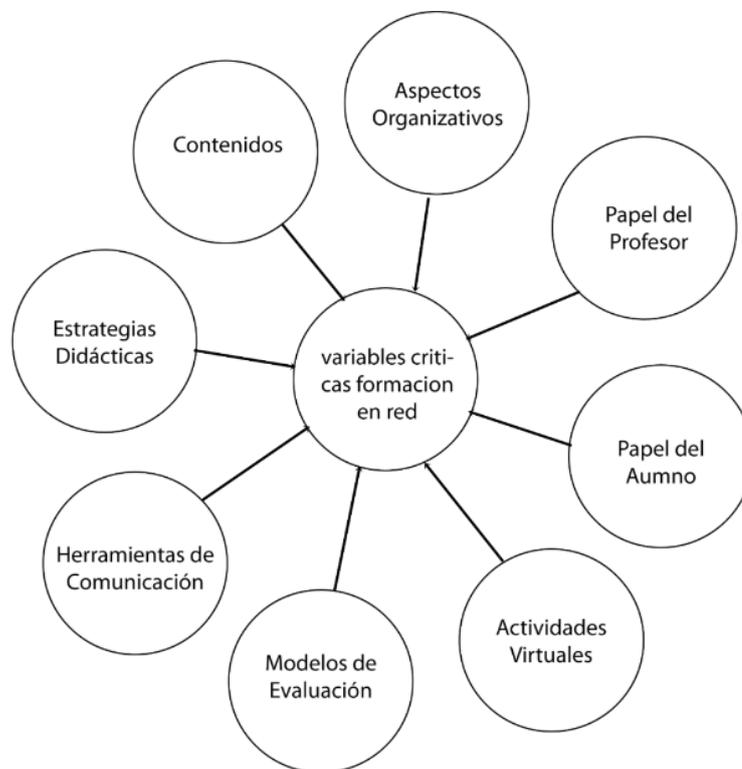


Figura 3.2. Variables Críticas de la Formación en Red

3.4. Aprendizaje Colaborativo Asistido por Computadora (CSCL)

García Peñalvo, en su artículo [PEÑ05] “Estado actual de los sistemas *de e-learning*” define, de manera precisa, un triángulo cuyos vértices se componen por los siguientes elementos: la tecnología (plataforma o entorno tecnológico), los contenidos y los servicios, características imprescindibles que conforman y articulan la propuesta de aprendizaje colaborativo. Y afirma que, alterando el peso de cada uno de dichos componentes, se obtienen varios modelos de plataformas, entre los que se encuentran dos tipos diferenciados: las que se utilizan para impartir y dar seguimiento pedagógico y administrativo a cursos en línea o *Learning Management System (LMS)*; y, por otro lado, las que se utilizan para la gestión de contenidos digitales o *Learning Content Management System (LCMS)*.

Las plataformas educativas, conocidas en su conjunto más completo como **LMS** (*Learning Management System*) o, su definición en castellano, Sistema de Gestión de Aprendizaje, son sistemas complejos que permiten distribuir y monitorear actividades de aprendizaje a través de repositorios.

Entre sus funciones habituales se pueden destacar la configuración y desarrollo de cursos, la gestión de planes de estudios, el seguimiento de prácticas, la elaboración de informes, la evaluación por parte del docente y la autoevaluación, el almacenamiento de documentos y otros recursos y las comunicaciones e intercambios virtuales a través de *chats*, videoconferencias, pizarras electrónicas o mensajería entre alumno-docente y alumno-alumno.

Los ambientes colaborativos o *Computer Support Collaborative Learning (CSCL)* generan innovación en tanto buscan propiciar espacios en los que se fusione el desarrollo de habilidades individuales y grupales, a partir de la discusión entre estudiantes al momento de explorar nuevos conceptos.

De esta forma, se busca que estos ambientes sean fértiles en posibilidades y más que organizadores de información, propicien el aprendizaje y desarrollo personal y social, donde cada miembro del grupo sea responsable tanto de su aprendizaje como del de los restantes miembros del equipo.

El modelo **CSCL** es un enfoque del aprendizaje basado en la psicología de las interacciones sociales, la pedagogía de la enseñanza y la informática. Esto significa, construir el conocimiento a partir de los saberes y las habilidades de otros participantes del proceso, en entornos asistidos por computadora. En este sentido, del Dujo [DEL11] asegura que es importante que durante el proceso de inserción de la tecnología en la educación se construya una perspectiva pedagógica que permita comprender los procesos de interacción social y el impacto en la formación.

Tal como expresa Cristóbal Cobo, las tecnologías entendidas como amplificadoras de capacidades pueden ser detonadoras de nuevos saberes o destrezas. No obstante, sabemos que no hay fórmulas mágicas. Las tecnologías en sí no son disruptivas, lo son la manera en que estas se pueden adaptar y adoptar bajo una u otra forma. Es una realidad que podemos identificar nuevas tecnologías utilizadas para replicar viejas pedagogías, de igual modo que podemos identificar valiosas experiencias pedagógicas con limitada innovación tecnológica [COB16b].

El otro enfoque que conforma el modelo **CSCL** es el psicológico. Lo que examina la psicología es el concepto de actividad. Según Vargas [VAR06] la actividad presupone no sólo las acciones de un solo individuo tomado aisladamente, sino también sus acciones en las condiciones de la actividad de otras personas. En este trabajo asegura que una actividad se compone de una necesidad, un motivo, una finalidad y condiciones para obtener la finalidad. Esta actividad, en este caso, corresponde la enseñanza y el aprendizaje, comprendido como un sistema de relaciones entre individuos organizados. En el mismo sentido, Barros [BAR04] utiliza la Teoría de Actividad (*Activity Theory*) como un enfoque filosófico para “analizar diferentes formas de la práctica humana como procesos de desarrollo, con niveles interrelacionados tanto individuales como sociales”. El autor asegura que la Teoría de la Actividad se puede utilizar cuando se plantean situaciones que involucran a grupos de personas en los que, de alguna forma, se realizan actividades mediadas o facilitadas por un soporte tecnológico, ya que ofrece un marco teórico que ayuda a identificar aspectos sociales y relacionarlos con elementos tecnológicos.

Este autor expresa que cada actividad debe realizar interpretaciones sobre la comunidad, la división del trabajo, las normas de la comunidad, las herramientas y los resultados.

Barros define las siguientes características de la **Teoría de la Actividad**:

- La **TA** identifica los elementos e indica cómo los aspectos sociales influyen en la forma de actuar de los grupos. Para pasar de este plano sociológico (teórico) a uno tecnológico (práctico) es necesario hacer un modelo abstracto con los elementos de la teoría y posteriormente implementarlos en un sistema informático.
- La **TA** se ha mostrado, a lo largo de los últimos años, como un marco teórico fructífero para la descripción, modelado e implementación de sistemas que promuevan el aprendizaje colaborativo.

En la siguiente ilustración se pueden observar los factores que impactan en el modelo **CSCL**, que busca examinar las experiencias de aprendizaje en grupos de alumnos que comparten objetivos, actividades y recursos comunes a través de las nuevas tecnologías en red. Estas tres dimensiones tienen su origen en las variables críticas estudiadas anteriormente y propuestas por Cabero.



Figura 3.3. Modelo CSCL

Dans [DAN09], resalta la evolución sufrida en la Red en estos últimos diez años y destaca las transformaciones que han sufrido las tradicionales plataformas, que han ido mutando en favor de entornos menos direccionales y más participativos. Así el alumno no sólo visualiza y capta información de la red, sino que también, sube y modifica de forma activa el entorno en el cual navega.

De esta manera las plataformas crean un entorno centrado en el alumno en el cual el proceso de aprendizaje se beneficia por un ambiente interactivo, eficiente, fácilmente accesible y completamente distribuido. Redefinir el rol del estudiante de pasivo a un rol netamente activo, modifica también el concepto de aprendizaje que muta del mero almacenamiento de información hacia su reinterpretación cognitiva. En palabras de los propios autores Cabero y Almenara [CAB06], en definitiva, debemos llevar a cabo verdaderas acciones de *e-learning* y no de *e-reading*.

Es en este punto donde se empiezan a vislumbrar de manera clara las contraposiciones que representa el sistema tradicional de enseñanza vs. un sistema colaborativo en línea: mientras que el primero centra su foco en el docente, el segundo transforma al alumno en centro, elevándolo a la categoría de actor primario. El objetivo de distribuir conocimientos de la enseñanza tradicional también se ve afectado, ampliando dicho fin con la acción de captar los contenidos por parte estudiante, ahora protagonista. Otro ítem que se beneficia con esta evolución es el de la creación de los contenidos. Mientras que en el primer caso se construye desde cero, en el segundo, se basa en la reutilización de contenidos, dando lugar a un proceso incremental.

García Peñalvo [PEÑ05], analiza estos sistemas y resalta una perspectiva dual que deben poseer todos los sistemas de *e-learning* que consiste en los aspectos pedagógico y tecnológico.

Mientras que la cualidad tecnológica es fácilmente identificable como aplicaciones desarrolladas en un ambiente de red, el factor pedagógico debe adquirir el mismo grado de importancia que la concientización del diseño e implementación del sistema de software. Esto implica que la plataforma debe estar construida y consolidada de acuerdo a modelos y patrones pedagógicamente definidos, evitando caer en el facilismo de crear plataformas como meros contenedores de contenidos e información en formato digital.

Según [ART05], de acuerdo con el modelo pedagógico centrado en el docente, los estudiantes juegan un rol pasivo y solo adquieren la experiencia en función de los conocimientos brindados por aquel. Es por esto que, en los contextos colaborativos, en los que el alumno juega un rol predominante, es sumamente necesario que el docente pueda participar activamente dentro del proceso de enseñanza.

El origen del aprendizaje colaborativo está dado por el trabajo colaborativo, en especial cuando las actividades de trabajo conjunto son asistidas por una computadora. El concepto de *Computer Supported Cooperative Work* o Trabajo Cooperativo Asistido por Computadora (**CSCW**) se considera antecesor del Aprendizaje Colaborativo Asistido por Computadora o *Computer Supported Collaborative Learning* (**CSCL**). En ambos casos, como consecuencia de la evolución social producto del impacto de las TIC, se propone un cambio relativo a la complejidad del trabajo, debido a la subdivisión de tareas que conlleva su distribución entre diferentes actores, con diferente nivel de desempeño, muchas veces separados por tiempo y distancia.

El aprendizaje colaborativo es habitual en diversas formas de enseñanza. Esto suele plasmarse como trabajo en equipo en forma de proyecto, en donde se busca la completitud de un objetivo. En este punto, [ILL01] establece que el origen de este concepto está en las concepciones no individualistas de la educación y tiene una doble lectura: colaborar para aprender y aprender a colaborar. Es decir, que el trabajo en grupo genere efectos pedagógicos sobre la dinámica, integración y motivación de los estudiantes. Aun así, identifica diversos efectos no deseados como aquellos casos en los que no exista colaboración como tal o bien donde algún miembro de un equipo solo use el trabajo del resto o solo aproveche los resultados para fines propios. Aun así, estas desventajas suelen ser menores que las propias ventajas del modelo.

El modelo **CSCL** se basa en varios enfoques dentro del campo psicológico, como la cognición situada, compartida, distribuida, el enfoque sociocultural, el constructivismo social, la teoría de la actividad y todas sus variantes. Todos estos abordajes teóricos tienen un punto en común: el análisis individualista propuesto por el conductismo queda completamente opacado por las consecuencias cognitivas de las relaciones sociales dentro del sistema mental de cada individuo, que propone Vygotsky en varios de sus trabajos [VIG87], y que ya hemos presentado.

El aprendizaje colaborativo es un proceso en el cual dos o más participantes están unidos para realizar una tarea (simple o compleja), breves en su duración y semejante a un curso académico. Hay aspectos específicos relacionados con el tipo de tarea que definen si ésta es interdependizable. Para esto hay que pensar en el tamaño y granularidad de la tarea de forma tal de evaluar su posible descomposición, como es el caso de tareas muy simples o demasiado complejas.

Una tarea que sea posible descomponer es candidata a ser ejecutada por medio del trabajo colaborativo, en donde cada uno de los participantes llevará a cabo una parte o sub tarea asociada, siendo posible la ejecución concurrente de un grupo de personas o equipo de trabajo. Otro criterio a tener en cuenta es el conjunto de competencias requeridas para cada tarea. Cuando la tarea es compleja, su realización podría llevarse a cabo por medio de un equipo de personas con diferentes capacidades, donde aquellos con mayor experticia en el tema podrían ayudar a otros y ser parte del desarrollo cognitivo propuesto por el aprendizaje colaborativo [VIG79] [ILL01].

La universidad forma profesionales que pueden y deben trabajar en equipo. No solo colaborar para aprender, sino también aprender a colaborar y es por esto que deberá ser prestar especial atención al tipo de actividad, su complejidad, y el grado de dependencia que tiene con un modelo discursivo (discusión verbal) y brindar todo el soporte tecnológico necesario para que todos los participantes de un proceso de colaboración puedan llevar adelante todas las tareas requeridas para lograr el objetivo planteado.

Unas de las características distintivas de los sistemas **CSCL** es la capacidad de trabajar en grupos para la resolución de un problema. Esto supone algunos inconvenientes, como los planteados por Barberá. En este sentido, la autora indica que muchas veces existe cierta oposición al trabajo en grupo, dado que, la sensación de ligadura con el grupo va en contra de la libertad que generan las herramientas, al llevar a cabo los estudios de manera individual, sin depender de otros y a su propio ritmo [BAR06b].

Entendemos que es necesario que la herramienta **CSCL** disponga de un conjunto de formalismos que permitan definir las características del curso. Desde este punto de vista, y sobre la base del trabajo de [ARE09], coincidimos con [ROS08] que los principales factores que intervienen en la enseñanza colaborativa asistida por computadora son:

- La estructura del curso.

- Las actividades de aprendizaje.
- Los materiales didácticos que se ofrecen.
- Las pautas y estrategias de comunicación.
- Las herramientas de colaboración entre alumnos y docentes.
- La evaluación.
- El rol del docente tutor y/o moderador.

El **modelo uCSCL** se basa en el concepto de aprendizaje colaborativo asistido por computadora, pero se apoya en la dimensión tecnológica para permitir que el acceso al conocimiento se haga de forma ubicua, es decir accediendo a él en cualquier momento y lugar [COL13] [CCR16] [YAN15].

3.5. Evaluación en Entornos Virtuales de Enseñanza

Elena Dorrego, en su relevamiento sobre “Educación a Distancia y Evaluación Colaborativa”, asegura que en cualquier modalidad instruccional es necesaria una estrecha relación entre la enseñanza, el aprendizaje y la evaluación [DOR06].

La evaluación es el proceso (o conjunto de procesos) que dirige y da forma al aprendizaje, para obtener y analizar datos para transformarlos en información viable para ser comparada contra un criterio determinado. El objetivo de la evaluación es poder decidir si el objeto evaluado cumple con los criterios esperados y promover otros tipos de aprendizaje más útiles o deseables.

En la educación, la evaluación *formativa* se utiliza para que el alumno pueda comprender sus competencias, obtener información acerca del progreso y obtener retroalimentación del profesor. Son actividades que estimulan la motivación, aumentan la comprensión e informan sobre el progreso. La retroalimentación o *feedback* oportuno, ayuda a proporcionar un refuerzo motivacional y a mejorar las estrategias de aprendizaje para hacerlas más efectivas.

Otro enfoque es la evaluación *sumativa* se utiliza para que el alumno pueda obtener una medida, calificación o grado que indique su desempeño. Su propósito más importante es estimar y registrar los logros del estudiante.

Un tercer enfoque, denominado de evaluación continua, es el que se utiliza en muchos entornos de educación a distancia. Este modelo posee componentes formativos y sumativos y posee las siguientes características:

- Estructura el aprendizaje
- Descompone la carga de evaluación en partes más pequeñas y manejables
- Es motivante
- Facilita a los alumnos una visión sobre sus progresos y competencias

Los propósitos de la evaluación anteriormente mencionados atienden las siguientes necesidades puntuales

- De los estudiantes en relación a su progreso y calificación sobre un nivel de logro.
- De los docentes en relación a conocer si los alumnos alcanzan los resultados esperados y si el dictado del curso es efectivo.
- De las instituciones para evidenciar el logro de metas institucionales, saber si los programas y los docentes son efectivos en la consecución de sus metas y certificar que los alumnos están preparados para ejercer en áreas específicas.
- De la sociedad para conocer a las instituciones y docentes con el objetivo de garantizar su financiamiento

En general, los sistemas de evaluación deben adaptarse a los objetivos de aprendizaje, los contenidos y los destinatarios. El proceso de aprendizaje en línea es muy complejo. Una forma sistemática de reducir esta complejidad es utilizando diferentes métodos de evaluación que ayuden a descomponer el problema [GAR06] [DOR06].

Las **TIC** están cambiando el rumbo de muchos aspectos cotidianos. Estos cambios tienden a explotar el concepto de ubicuidad y centrarse en él para transformar la sociedad. Este movimiento alcanza también a la educación, que si bien fue beneficiada en muchos aspectos hay otros, como el proceso de evaluación que sigue siendo uno de los elementos más controvertidos y difíciles para los docentes, sobre todo cuando el docente pierde protagonismo en entornos donde su presencia no es tan fuerte como dentro del aula. [BAT16c].

Una de las grandes debilidades de los entornos **CSCL** es la forma de poder evaluar el trabajo de los alumnos. Es imprescindible que estos ambientes incluyan evaluaciones formativas para apoyar el aprendizaje. En este sentido Guerra afirma que “Los cambios producidos en las nuevas teorías de aprendizaje que están conduciendo las investigaciones de las últimas décadas sugieren la necesidad de considerar nuevas formas de evaluar” [GUE08]. Con esta afirmación comprendemos que la disrupción de la tecnología en la educación no solo propuso nuevos escenarios para el aprendizaje, también requiere completa atención a la forma de evaluarlo.

Los entornos **CSCL** impactan de forma positiva en el rendimiento estudiantil, pero debe haber una correlación no solo entre el aprendizaje y la colaboración, sino también con la evaluación. Si bien es importante la evaluación individual, hay que comprender que la naturaleza de los entornos **CSCL** es de trabajo en grupo, lo que hace muy importante incluir la evaluación grupal como actividad formativa.

Elena Barbera asegura que, en la evaluación en línea, se apunta a que el mayor responsable del cambio son los instrumentos de evaluación. Las principales ventajas en la utilización de un entorno virtual para llevar a cabo un sistema de evaluación y autoevaluación son [BAR06b]:

- Posibilita un seguimiento individualizado y grupal del aprendizaje del alumno.
- Permite evaluar conocimientos y habilidades.
- Facilita el establecimiento de una evaluación continuada durante el proceso de aprendizaje y reduce el tiempo de su diseño, distribución y desarrollo.
- Agrega una gran flexibilidad temporal y espacial del sistema tanto para la configuración de ejercicios como de su realización. En este sentido puede ser especialmente útil para permitir que el alumno pueda seguir su propio ritmo de aprendizaje.
- Proporciona una respuesta inmediata (retroalimentación) de los resultados del ejercicio

El impacto positivo de las **TIC** en la enseñanza, proporciona nuevas oportunidades de evaluación, entre las que se destacan

- Autoevaluación
- Evaluación Enciclopédica
- Evaluación Colaborativa

3.5.1. Autoevaluación

Otro de los puntos fuertes del proceso de evaluación en línea es la pérdida del control. En este sentido, se comprende que el estudiante asume el control de la evaluación y el profesor pasa a cumplir un rol de guía o tutor. Para esto se diseñaron nuevas formas de evaluación: *la autoevaluación* y la *evaluación por Prueba de Respuesta Objetiva*. La autoevaluación es una técnica que permite al alumno obtener una visión de sus fortalezas y debilidades y saber cuáles son las áreas a mejorar. En este sentido, Raposo propone la utilización de *rubricas* como una herramienta de autoevaluación, que además de mejorar el proceso de autoevaluación, impacta de forma positiva en el proceso de enseñanza y aprendizaje [RAP11].

Las rúbricas son especialmente útiles para realizar una evaluación objetiva y consistente de actividades como trabajos grupales, presentaciones orales, trabajos escritos individuales o prácticas en laboratorio, entre otras. Utilizada como herramienta de evaluación, la rúbrica es útil para el profesor; si se usa como herramienta de trabajo conjunto (evaluación formativa), tanto el profesor como el alumno podrán salir beneficiados.

Una rúbrica es un conjunto de criterios y estándares, generalmente relacionados con objetivos de aprendizaje que se utilizan para evaluar un nivel de desempeño o una tarea, tratándose así de una herramienta de calificación utilizada para realizar evaluaciones objetivas del alumnado. Se agrupan en criterios comunes o específicos de dicho trabajo que están ligados a los objetivos de aprendizaje, para evaluar la actuación de los alumnos en la realización de tareas o bien proyectos [FUR01]. También, es una herramienta para ayudar a la evaluación que puede resultar especialmente útil cuando hay que evaluar aspectos complejos, imprecisos o subjetivos. Consiste en preparar una matriz de

valoración que recoja los elementos que queremos evaluar y fije, para cada uno de ellos, los criterios que usaremos para darles un valor u otro.

Indicadores	A	B	C	PESO	Puntaje
Nombre del caso de uso	El nombre del caso de uso está descrito desde la perspectiva de los objetivos del actor.	El nombre del casos de uso está descrito desde la perspectiva de los objetivos del actor, pero no tiene una vinculación directa con la funcionalidad del sistema	El nombre del casos de uso está descrito desde el punto de vista del sistema	A= 25pts B=15pts C=8pts	15
Descripción del caso de uso	La descripción del caso de uso da una idea general de las características del mismo	La descripción del caso de uso da una idea parcial de las características del mismo	La descripción del caso de uso no permite entender las características del mismo	A= 25pts B=15pts C=8pts	25

Figura 3.4. Ejemplo de rúbrica.

La *prueba de respuesta objetiva* es un tipo de test diseñado para comprobar muchos objetivos de respuesta puntual. Este tipo de *test* tiene algunas limitaciones, como por ejemplo la imposibilidad de que el alumno pueda expresar libremente sus ideas, ya que con preguntas de respuesta abierta requeriría acción de un docente o de algún corrector cognitivo y esto podría resultar muy costoso. Otra limitación es que, al haber una comunicación asincrónica entre el docente y el alumno, la falta de personalización en las respuestas acota mucho la metodología [RAP11] [DOR06] [GAR06] [BAR06b].

Esta técnica es implementada de forma sencilla en muchos entornos virtuales de enseñanza y aprendizaje y ayuda al alumno a tener conocimiento de la asignatura de forma constante. Ambas metodologías de evaluación planteadas son muy útiles no solo para la evaluación individual, sino para la evaluación grupal.

3.5.2. Evaluación Enciclopédica

El acceso ubicuo a la información de distintas fuentes en Internet supone una reducción en la complejidad para elaborar los documentos evaluativos. Si bien el acceso a la información es sencillo y rápido, esto da lugar a información poco fehaciente y alta probabilidad de plagio, generando el riesgo de problemas de instrucción e institución. Dada la importancia de los grandes volúmenes de información y el aporte sustancial a la

evaluación, en esta metodología el docente deja ser un transmisor de información para ser un tutor en la consecución de la misma [BAR06b].

3.5.3. Evaluación Colaborativa

El concepto de colaboración está directamente relacionado a las actividades realizadas por un conjunto de personas con un objetivo en común. En este sentido, y con el enfoque puesto en la educación, el concepto de evaluación colaborativa carece de sentido si es realizada únicamente por el docente.

Según Barberá la colaboración se identifica con las herramientas que suelen ser parte de las plataformas virtuales, como por ejemplo *foros* y *chat* (siendo estos los más emblemáticos). En este trabajo, la autora indica que estos recursos no entran en la actividad docente ni tienen las herramientas para generar indicadores del aprendizaje.

En los espacios colaborativos virtuales, el docente puede generar muchas aportaciones por medio del manejo de grupo. El trabajo en grupo en herramientas **CSCL** permite al docente medir la situación individual de cada uno (por medio de los mecanismos de *awareness*) al mismo tiempo que puede medir el trabajo grupal en cuanto al proceso y al resultado. De esta manera el docente podría, por ejemplo, otorgar una calificación individual y una grupal para un mismo trabajo

La evaluación colaborativa al igual que cualquier otra actividad colaborativa requiere de interacción entre las personas involucradas, incluyendo docentes y alumnos participantes de un grupo. Una forma de medir la participación de las personas en el proceso es, por ejemplo, por medio de la cantidad de posibles mensajes que se emitan. Esta forma de mensurar es meramente cuantitativa. Es decir, no existe forma de medir la calidad de la participación, sino la frecuencia de la misma.

En un proceso evaluativo, Barberá y Ochoa afirman (en sus respectivos trabajos) que la clave en la evaluación colaborativa está en el intercambio de mensajes y en la naturaleza de esta interacción, ya que involucra a docentes y alumnos de la misma manera, pero desde puntos de vista distintos [BAR06b] [OCH08].

Para medir la calidad de las interacciones es necesario enfocarse en el *feedback* o retroalimentación que genere cada una de las mismas. No solo las generadas por el

docente, sino también las que el mismo alumno emite, siendo por demás útiles al resto de los participantes en caso de ser un trabajo grupal

El efecto del *feedback* del profesor es positivo sobre los estudiantes, con el objetivo de promover la discusión de los tópicos de aprendizaje y la formalización del conocimiento. En su trabajo, se proponen tres dimensiones para entender el *feedback* [ROC16].

- En primer lugar, el foco. Es decir, en que el alumno comprenda que la participación en una discusión en línea puede aportar información sobre el contenido del aprendizaje y el contenido de la tarea.
- En segundo lugar, el tipo de *feedback*. Puede ser de verificación, sobre si una producción es correcta o incorrecta, o de elaboración que incluye información para guiar al alumno.
- Otro punto a considerar es la dimensión temporal. Representa el momento en que se realiza el *feedback*, para analizar el ajuste de las ayudas en función de las necesidades de los participantes durante el proceso de instrucción

3.6. Resumen

El objetivo de este capítulo fue contextualizar el concepto de enseñanza y aprendizaje colaborativo, con sus ventajas y desventajas así como el desarrollo histórico de este concepto hasta el momento en que fue adoptándose dentro del aula. Presentamos el marco teórico que integra aspectos psicológicos y pedagógicos en relación con la interacción social y su influencia en el desarrollo y, en particular, el concepto de “zona de desarrollo próximo”. Por último, presentamos el impacto que las TIC han tenido en los entornos virtuales de enseñanza y aprendizaje creando los entornos uCSCL y cómo estos cambios afectan los métodos de evaluación, abordando el concepto de evaluación colaborativa.

Capítulo 4

Herramientas CASE

4.1. Introducción

En el desarrollo de sistemas de información es notorio el interés constante por mejorar productividad y calidad en los productos, ya que impacta directamente en los costos de desarrollo y mantenimiento. Para lograr esto es sumamente necesario establecer un marco de trabajo disciplinado y estructurado en donde predomine el trabajo en equipo, para evitar resultados inesperados.

Las herramientas **CASE**, están tomando cada vez más relevancia en la planeación y ejecución de proyectos que involucren sistemas de información, pues suelen inducir a sus usuarios a la correcta utilización de metodologías que le ayudan a llegar con facilidad a los productos de software construidos. Es importante considerar que tipo de herramienta **CASE** necesitamos al momento de llevar adelante una actividad relacionada con la **IS**. Por este motivo es sumamente importante conocer las distintas taxonomías y la usabilidad relacionada a cada fase del ciclo de vida de desarrollo de software.

Es necesario considerar esta afirmación, tomando como actividad el proceso de enseñanza **IS**. Es sumamente importante conocer las limitaciones, ventajas y desventajas de utilizar una herramienta **CASE** durante el proceso de enseñanza en curso de **IS** donde el modelado es el principal punto a considerar, tanto durante el análisis como el diseño o la implementación.

En este capítulo presentamos el marco teórico para poder contextualizar las herramientas **CASE** y una metodología de evaluación para poder conocer en qué grado una herramienta se adapta con la funcionalidad requerida, Por último, realizamos un estudio de algunas de las herramientas de modelado **UML** vigentes en el mercado para poder instaurar un punto de comparación. Presentamos en base a esto las características deseables de una herramienta **CASE** destinada al a enseñanza de la **IS**.

4.2. Concepto de herramientas CASE

Las últimas décadas fueron de constante crecimiento para la **IS**. Este tiempo su evolución fue constante hasta convertirse en una disciplina legítima de ingeniería. La constante evolución de los sistemas hardware y los impactos que acompañaron a las crecientes tecnologías antes de los 70, dieron origen al movimiento conocido como crisis del software. La introducción de nuevos componentes y circuitos integrados, permitieron que las aplicaciones que hasta ese entonces no era factible, comenzaran a ser propuestas válidas y a materializarse en productos mucho más amplios y complejos, dejando como experiencia que el enfoque informal en la construcción de software no era muy bueno, generaban productos mucho más caros de lo presupuestado, difíciles de mantener, con desempeños muy pobres y mala calidad. Los costos de software subían, mientras los de hardware iban por el camino opuesto: surgió la necesidad de nuevas técnicas y métodos que permita controlar la creciente complejidad y así también poder controlar los costos [SOM05].

Desde la crisis del software hasta la actualidad la creciente demanda de calidad en el desarrollo de software incremento la necesidad de establecer un proceso de desarrollo que permita desarrollar software de manera confiable y que sea económico. Se hizo necesario obtener productos más fáciles de mantener, con más capacidad de adaptación y con costos más accesibles. Se hizo necesario, también, una ayuda o automatización para cada una de las actividades que propone la **IS**

En base a esto la **IEEE** desarrollo diferentes estándares y una definición completa: “La Ingeniería de Software es: 1) la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo., operación y mantenimiento de software; es decir, la aplicación de la Ingeniería de Software. 2) El estudio de enfoques según el punto 1.”

Pressman asegura que las herramientas **CASE** ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto [PRE10]. Propone entender la **IS** como una tecnología con diferentes capas (ver figura 4.1), apoyándose en el compromiso con la calidad. Luego el proceso, como base de control; los métodos, que brindan la experiencia necesaria para elaborar software por medio de un conjunto de tareas (como el análisis, diseño, implementación y pruebas). Estos métodos están presentes en cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Por último, se encuentran las Herramientas que proporcionan un apoyo automatizado para el proceso y los métodos. Cuando se integran herramientas que permiten automatizar los procesos y los métodos en diferentes áreas de la IS, se crean herramientas **CASE** (*Computer-Aided Software Engineering*).



Figura 4.1. Capas de la IS (Extraído de [PRE10])

Las primitivas herramientas **CASE**, se dirigieron principalmente a la automatización de la documentación y la comunicación como una mejora. Clave de la productividad del software [MCC93]. Las primeras herramientas para apoyar el proceso de desarrollo de software fueron los editores y procesadores de texto, usados para escribir programas y su documentación. A través de las metodologías antes descritas, se automatizaron tareas de amplio espectro.

Por su parte, el ciclo de vida del software es entendido como la secuencia de fases por las cuales atraviesa un proyecto de desarrollo desde su concepción hasta el fin del uso del producto software obtenido, atravesando su construcción y mantenimiento. Así se aplica entonces una mitología basada en la planificación, gestión, análisis, diseño, codificación, pruebas, documentación, mantenimiento, validación y verificación del proyecto.

4.2.1. Características

La utilización de modelos para crear software utilizando lenguajes, técnicas y herramientas dentro de la **IS** es una práctica común durante las etapas de análisis y diseño. Todas las herramientas de modelado prestan soporte a un lenguaje específico

para acompañar la metodología y es lógico suponer, que un alto porcentaje de ellas soportan **UML**. Teniendo en cuenta la amplia aceptación de este lenguaje y el valor conceptual y visual que proporciona, y su facilidad para extender el lenguaje para representar elementos particulares a determinados tipos de aplicaciones

Si la herramienta soporta **UML**, la funcionalidad de intercambio de modelos con otras herramientas se convierte en una meta alcanzable, teniendo en cuenta la estandarización sintáctica y semántica que persigue dicho lenguaje. En esta línea de estandarización alrededor de **UML**, la **OMG** definió un estándar de intercambio llamado **XMI**, que basado en el meta modelo de **UML**, sirve para el intercambio de metadatos utilizando **XML**.

De acuerdo al trabajo de Mc Clure [MCC93] los objetivos principales de las herramientas CASE son:

- Aumentar la productividad en el desarrollo.
- Dar calidad a los productos desarrollados.
- Reducir el costo del software.
- Automatizar los chequeos de errores.
- Acelerar el desarrollo de las aplicaciones.
- Automatizar tareas de desarrollo.
- Automatizar la generación de documentación.
- Dar posibilidad al software.
- Implantar metodologías de desarrollo.
- Datos reutilizables y compartidos.
- Administrar el proyecto.
- Ingeniería hacia atrás (reingeniería)
- Estándares aplicables al Modelado de Software

Cabe destacar que la carencia funcionalidades destinadas a la enseñanza de la **IS** como característica es recurrente, por lo que, en un entorno académico, los alumnos deben utilizar herramientas que estén pensadas para utilizar en el ámbito profesional, sin

disponer de aquellas funcionalidades requeridas en el ámbito académico [NEI16] [BAT17c].

4.3. Modelado UML

Un modelo es una representación de algo en un medio determinado. Los modelos capturan los aspectos de lo que se desea modelar con un punto de vista en particular y provee un nivel de abstracción sobre los aspectos menos importantes [RUM05].

Un modelo posee dos aspectos importantes, información semántica y presentación visual por un lado y el contexto por otro. Los *aspectos semánticos* capturan el significado de las construcciones que son descriptos de forma separada, pero están relacionadas como parte de un modelo único y coherente. La presentación visual muestra la información semántica para que pueda ser usada por humanos. El *contexto* incluye información interna para que los modelos no carezcan de sentido. Es decir, que un modelo tiene un sentido distinto según el contexto en el que se utilice.

Un modelo de software este hecho en un lenguaje de modelado como, por ejemplo, **UML**. En este contexto los modelos se utilizan para diversos propósitos, entre ellos destacamos los más importantes:

- Para que un equipo interdisciplinario pueda entenderse. Es muy importante que un equipo de trabajo, muchas veces compuesto por participantes que no se relacionan directamente con el desarrollo de software, es decir, interdisciplinario; pueda mantener un estándar comunicacional que permita romper la ambigüedad inherente al proceso comunicacional humano.
- Para capturar y precisar requerimientos funcionales, Como destacamos anteriormente, permitir la ambigüedad durante el proceso de desarrollo de software es una de las grandes fallas. Es importante destacar que el relevamiento y elicitación de requisitos y su documentación forma parte de una de las fases más importante del proceso de desarrollo de software. Si no logramos una documentación concreta, con modelos legibles y estandarizados, es probable que se obtengan resultados no esperados en las etapas posteriores.

- Para Diseñar un sistema. De la misma forma que se utilizan los modelos para documentar requisitos funcionales durante la etapa de análisis, los utilizamos durante la etapa de diseño. De la forma que el software debería comportarse ante un estímulo externo, ya sea una interacción de un usuario o bien de otro sistema deberá tener su correspondiente diseño desde el punto de vista estructural y de comportamiento. Es decir, todo aquello que no se documente, quedará sujeto a la interpretación de quien lo implemente.
- Para crear productos de trabajo. Es común encontrar herramientas que permitan desarrollar software en base a un modelo determinado. Esta metodología de trabajo ayuda a la creación de prototipos, o bien permite crear bloques funcionales que sirvan para determinada actividad, fomentando el reuso y favoreciendo a la escalabilidad de los sistemas.
- Para organizar grandes sistemas. Los sistemas de información deberían poder adaptarse a cambios en el contexto organizacional y de negocio. Esta característica hace que un sistema pueda escalarse y transformarse en una herramienta enorme, distribuida, con miles de bloques funcionales que, ante una mala organización, hace que estos sistemas sean propensos a bajar su nivel de mantenibilidad. Es decir, cuanto más grande es el sistema mejor debería organizarse, no solo para ayudar a su crecimiento, sino también a su mantenimiento.

4.3.1. UML

UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado) no es un lenguaje de programación sino un estándar que se publicó a mediados de los 90', como un lenguaje visual de modelado para visualizar, especificar, construir y documentar los artefactos de un sistema software.

Se utiliza para entender, diseñar, navegar, configurar y mantener toda la información relativa a tales sistemas. No está pensado como un método completo de desarrollo ni un marco metodológico de trabajo, sino como una herramienta que en concurrencia con una metodología brinde soporte al proceso de desarrollo en cada una de sus etapas.

Hasta ese entonces (mediados de los 90) el modelado de software estaba repleto de notaciones completamente diferentes e incompatibles que daba como resultado una gran fragmentación del mercado. Por esto es que los autores del estándar describen en su libro que **UML** se crea con el objetivo de unificar experiencias anteriores acerca de técnicas de modelado y para incorporar las mejores prácticas en un estándar calificable. Para finales de la primera década del siglo 21, se estimaba que más del 70% de las empresas de desarrollo habían adoptado el estándar. Al día de hoy UML es soportado por la mayoría de las herramientas comerciales que se utilizan para modelar software [WAT08].

Luego de varios años de experiencia, **OMG** [OMG17] presento una propuesta para actualizar, corregir y extender el lenguaje. Para mediados de 2003 fue presentado como la *versión 2.0* de la especificación, la cual fue probada y corregida para 2005. Las características más destacadas de esta versión se basan en la aparición de los diagramas de secuencia y el uso de los diagramas de actividad para modelar procesos de negocio.

UML Captura información acerca de las estructuras estáticas y el comportamiento de los sistemas, que son modelados con una colección de objetos que interactúan con el objetivo de cumplir las expectativas y necesidades del usuario externo que lo utilice. Las estructuras estáticas definen los objetos software y sus relaciones y la forma en que se implementan mientras que el comportamiento define la historia de los objetos en el tiempo y como colaboran y se comunican para cumplir un objetivo. Esta vista se diseñó con influencias muy notorias del modelo **ERD** (Diagrama Entidad Relación) propuesto por Peter Chen para 1976 [CHE76]. También posee estructuras físicas relacionadas al despliegue de las aplicaciones y también provee un sistema organizacional que permite mantener modelos grandes dentro de paquetes con el objetivo de que equipos de trabajo puedan mantener los sistemas de forma organizada y para manejar diferentes versiones de modelos y unidades de modelado en entornos complejos de desarrollo.

La tabla 4.1 muestra la taxonomía indicada [RUM05].

Área	Vista	Diagrama
Estructural	Vista Estática	Diagrama de Clases

	Vista de Diseño	Estructura Interna
		Diagrama de Colaboración
		Diagramas de Componentes
	Vista de Casos de Uso	Diagramas de Casos de Uso
Física	Vista de Despliegue	Diagrama de Despliegue
	Vista de Actividad	Diagrama de Actividad
	Vista de Interacción	Diagrama de Secuencia
		Diagrama de Comunicación
Organizacional	Vista de Organización	Diagrama de paquetes

Tabla 4.1. Taxonomía del estándar UML

4.4. Caracterización de una Herramienta CASE para Modelado

Toda herramienta **CASE** de Modelado, debería soportar algún lenguaje de modelado estándar que permita simplificar tanto la comunicación como la documentación del sistema objeto de estudio. [LAR99].

Las tendencias metodológicas de los últimos años están altamente influenciadas por el enfoque de Orientación a Objetos (**OO**), permitiendo identificar las abstracciones necesarias para proponer una solución consistente con los elementos del dominio del problema. Para lograr esto, es necesario disponer de una herramienta de modelado que brinde soporte a la comunicación entre participantes y representar estos conceptos a lo largo proceso propuesto por la **IS**, y que brinde una semántica común [PRE10] [QUI12].

Para poder comprender el tipo de solución que vamos a necesitar, es sumamente importante conocer el ámbito de aplicación. En este sentido, determinamos que la herramienta que vamos a describir está relacionada con la enseñanza de modelado **UML** en el marco de la Ingeniería de Software, y los usuarios objetivos son alumnos de nivel universitario con, posiblemente, poca o nula experiencia en herramientas **CASE**.

Para poder conseguir una mirada objetiva sobre las herramientas **CASE** de modelado y poder realizar una comparación, es necesario obtener un set de características deseables. El objetivo de obtener esta comparación es conocer las ventajas y desventajas que tiene cada herramienta analizada con el fin de encontrar las características deseables de una herramienta objetivo.

La caracterización de una herramienta **CASE** es una tarea compleja. Durante los últimos años han aparecido en el mercado muchos productos (en su mayoría comerciales) que poseen diversas características lo que hace que su selección no sea nada fácil. Desde que **UML** se comenzó a utilizar como estándar para el modelado orientado a objetos, muchas herramientas comenzaron a adoptarlo para no perderle pisada al mercado. Dos de las características más importantes (descriptas a continuación) se basan en el soporte completo de modelado **UML** y su validación semántica y en la posibilidad de utilizar las herramientas en entornos virtuales para permitir el trabajo online de múltiples individuos.

El primer elemento que vamos a considerar como característica es la posibilidad de brindar soporte completo a **UML** y sus características notacionales, según propone González Génova [GEN06]. En el mismo trabajo, los autores proponen 3 tipos de herramientas **CASE** para modelado según su notación.

- Las herramientas *sintácticas*, que solo permiten dibujar diagramas correctos en base a las reglas Notacionales de **UML**, beneficiando a los usuarios con asistentes que evitan errores de diseño y,
- Las herramientas *Semánticas* que proveen mecanismos de validación para garantizar construcciones con sentido y coherencia. En este punto, los autores proponen ubicar las herramientas **CASE** de Modelado.

4.4.1. Método de Evaluación

Si bien es sencillo encontrar información cualitativa sobre las herramientas objeto de análisis, muchas veces resultan insuficientes debido a que están atadas a promesas de sus fabricantes y hasta campañas publicitarias que no caracterizan la herramienta acorde a la necesidad del cliente. Es por esto que es necesario definir una metodología que

permita valorar de forma cuantificable con el objeto de bajar la ambigüedad inherente a la interpretación humana [BAT17c].

Existen numerosos trabajos que proponen métodos para evaluar una herramienta **CASE** acorde a un proyecto o empresa. Es necesario considerar aspectos técnicos y organizacionales (cómo la empresa se prepara para asimilar el nuevo producto). Para considerar aspectos técnicos se utilizan un conjunto de métricas definidas que refieren a la cobertura de la herramienta respecto a etapas del ciclo de vida, metodologías soportadas, trabajo concurrente, plataformas, manejo de bases de datos, generación de código, ingeniería de reverso, horario de atención del soporte técnico, material de apoyo, y temas relativo al costo [MEN01] [ROJ00].

Todos estos puntos sugieren un ranking para poder comenzar con la selección de la herramienta. Consideramos que este punto propone una división general sin tener en cuenta aspectos determinantes como, por ejemplo, tipos de bases de datos que utiliza o si posee soporte para generar código a un lenguaje en particular. Creemos que la primera división es interesante para luego especificar según criterios a definir en base al uso en particular.

En otro trabajo [QUI12], los autores definen un conjunto de características deseables agrupados en cuatro enfoques: Enfoque procedimental, definiendo como las herramientas utilizan las metodologías para guiar al usuario a través de un proceso de **IS**, el segundo aspecto es sobre el soporte al modelado que proveen las herramientas, evaluando la capacidad de modelado **UML** para apoyar la definición de un sistema. El tercer enfoque lo denominan Apoyo al Repositorio, refiriéndose a la capacidad de la herramienta de Mantener trabajo colaborativo permitiendo niveles de robustez y consistencia de los datos y, por último, un enfoque funcional en base a como las herramientas ayudan a los usuarios a desarrollar un sistema de información con mayor o menor facilidad, utiliza criterios como versionamiento, navegación de diagramas, etc.

Para poder evaluar con objetividad una solución informática es importante centrarse en aquellas características deseables que describen el tipo de herramienta, por ejemplo, para un Herramienta **CASE** de modelado es de suma importancia que posea soporte para **UML** no así para una herramienta **CASE** para análisis de riesgos o de gestión de proyectos.

En este sentido, Mendoza afirma que la selección de una herramienta **CASE** puede verse afectada por la cantidad de aspectos a considerar [MEN01].

Otro trabajo relacionado a la selección de Herramientas **CASE** de Modelado [GEN06] utiliza criterios para definir el soporte **UML** desde el punto de vista notacional, es decir, que la herramienta provea de todos los métodos gráficos, sintácticos y semánticos para poder dibujar diagramas, darles significado a los componentes y darles sentido a las relaciones existentes, utilizando como base el estándar **UML**. Los autores proponen definir un conjunto de características y contabilizar los números de respuesta afirmativas y negativas de cada herramienta. Si bien plantean la dificultad de poder catalogar todas las posibles características, proponen un método dinámico para ir confeccionando las preguntas solo cuando alguna herramienta objeto de análisis no la cubre.

Un último punto a considerar para definir los criterios de evaluación, se basa en la selección de un enfoque metodológico y que soporte propone **UML** para tal. Para esto tomamos el trabajo de Kruchten quien propone utilizar la metodología denominada **RUP** (*Rational Unified Process*) en diversos trabajos y libros según describe en [KRU04].

RUP es un enfoque metodológico como proceso para apoyar a la IS, por medio de un Framework. Según los autores, dos de las características más destacadas son: ser iterativo y estar guiado por casos de uso. Como se puede observar en la figura 4.2, el framework posee dos dimensiones. La dimensión de Tiempo está organizada en fases, iteraciones y *Milestones* y la dimensión de contenido presenta las disciplinas de software y los artefactos que aplican a cada una de ellas

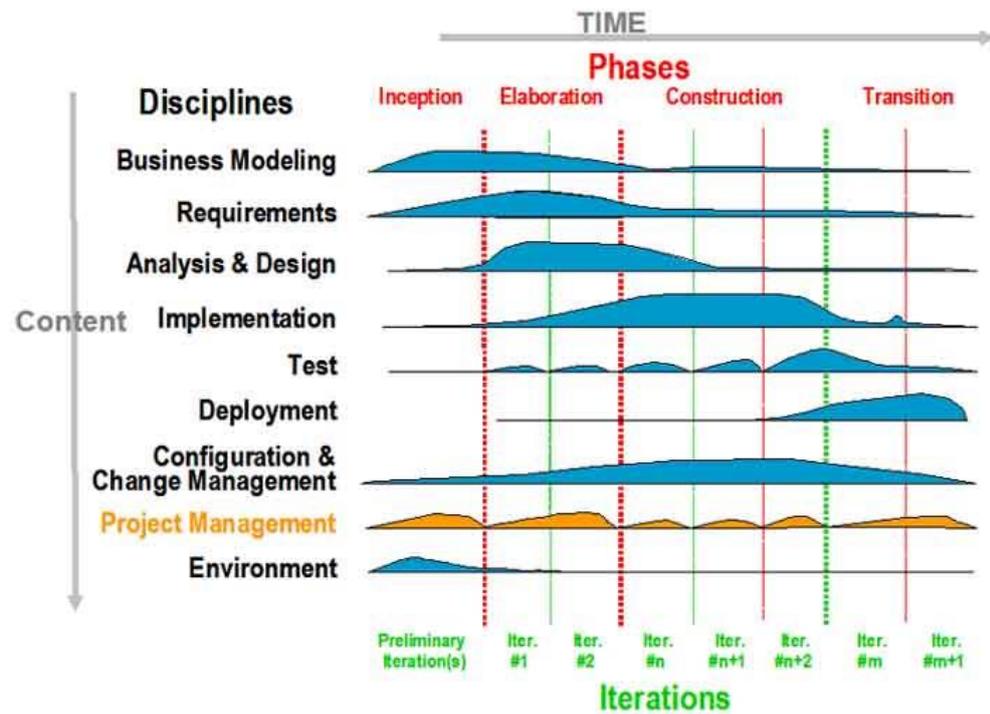


Figura 4.2. Framework RUP (Extraída de [IBM17])

La primera característica fundamental del proceso unificado se basa en definir el proceso como iterativo. Tal como se observa en la figura 4.3, está compuesto por una secuencia de pasos como se puede observar en la siguiente ilustración. Estos pasos serán repetidos en cada iteración representando diferentes fases del proceso de desarrollo.

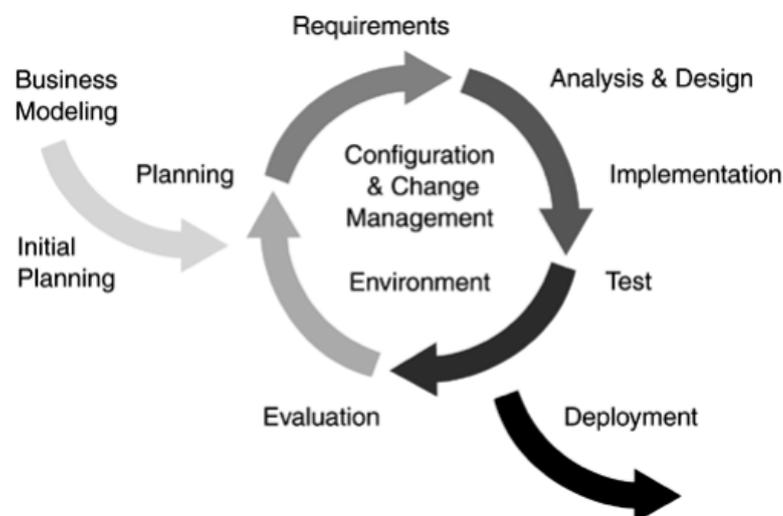


Figura 4.3. Desarrollo Iterativo e Incremental con RUP (Extraída de [KRU04])

Generar valor para el cliente es una meta obvia en todo desarrollo de software. Por este motivo, **RUP** propone como característica que el desarrollo esté guiado por casos de uso (**CU**). Un caso de uso describe requisitos funcionales, en termino de como los Actores interactúan con el sistema. Este enfoque “*use-case driven*” permite describir como eje como los usuarios la interacción que va a existir para conseguir un objetivo funcional, de forma temporal y mostrando todas las características del Sistema para una funcionalidad determinada. En **RUP** esta actividad es la más importante para formalizar la elicitación resultante en requisitos funcionales. En este punto, el autor propone detallar los casos de por medio de escenarios, para describir distintas variantes de interacción dentro de una funcionalidad.

4.4.2. Nuestra Propuesta de Evaluación

En base al estudio anterior hemos definido el siguiente método de evaluación de herramientas **CASE** centrado en estas características deseables para acotar el espectro de análisis según el tipo de herramienta [BAT17c] [BAT17e]. El objetivo principal es enmarcar y valorar la herramienta ideal como punto de comparación, definiendo como criterio de evaluación a un conjunto de características que a su vez dispondrán de ítems denominados *funcionalidades* o *especialidades*, según si son aspectos funcionales a la **IS** o funcionales al usuario.

Cada ítem podrá valorarse como *verdadero* o *falso* lo que dará un número finito que identifique en qué medida cumple las expectativas de las funcionalidades.

El total de puntos posibles estará dado por el total de funcionalidades positivas que tiene la herramienta ideal para el trabajo en cuestión, pudiendo utilizar un porcentual para identificar en qué grado cumple con las funcionalidades esperadas de una herramienta Ideal necesaria para determinada actividad.

Por ejemplo, si la herramienta ideal pose 200 funcionalidades aceptadas y la herramienta objeto de estudio arroja un total de 150, nos indicará de forma macro el acercamiento hacia el ideal en un 75%.

La puntuación total posible, denominado *Nivel de Aceptación*, está dada por la cantidad de funcionalidades o especialidades requeridas para la evaluación. En nuestro caso es 21.

4.4.2.1. Definición de Criterios de Evaluación

La herramienta a evaluar será una herramienta **CASE** para modelado **UML** de tipo semántica. Para desarrollar el correcto análisis de las herramientas propuestas a continuación, hemos definido criterios que consideramos acordes en base a las necesidades y dificultades detectadas con el objetivo de calificar un conjunto de Herramientas **CASE** que sirvan de forma referencial para poder obtener una puntuación objetiva.

Escoger una herramienta **CASE** acorde a la aplicación que se requiere, supone mejoras durante el proceso de la **IS**. Para esto es necesario conocer ciertas características fundamentales. Proponemos dos enfoques, uno funcional a la ingeniería de software y otro funcional a la experiencia del usuario.

En base a los trabajos estudiados en el apartado anterior [QUI12] [KRU04] [MEN01] y desde el enfoque funcional a la **IS** proponemos las siguientes características a considerar para la definición de los criterios de evaluación:

- Enfoque procedimental
- Apoyo metodológico
- Soporte completo **UML**
- Especificación de casos de uso
- Facilidad de extensión del lenguaje
- Modelado de Datos
- Autogeneración de código
- Ingeniería Inversa
- Métricas
- Apoyo a lenguajes formales

- Soporte al modelado arquitectónico
- Apoyo al modelado por capas

Desde el punto de vista de la experiencia del usuario, las características a considerar deberán ser las siguientes:

- Control de concurrencia
- Versionado
- Navegación
- Manejo y compatibilización de diagramas
- Visualización
- Trabajo Online
- Colaboración entre usuarios

En base a las características enumeradas anteriormente se definieron los siguientes conjuntos de *Criterios de Evaluación*. A cada Criterio de Evaluación se le definió un conjunto de Funcionalidades o Especialidades esperadas en base al tema tratado.

- Extensivo al proceso completo de desarrollo
- Adaptabilidad y flexibilidad
- Amigable para el usuario
- Producir documentación sólida
- Trabajo online y colaborativo
- Interpretar y validar Invariantes **OCL**
- Ingeniería Inversa y generación de código
- Apoyo metodológico

Como se observa, se excluyeron las dimensiones relativas a la facilidad de uso y aprendizaje, amigabilidad, y otros criterios relacionados a la experiencia del usuario. Comprendemos que arrojar un valor objetivo para esta evaluación es una tarea específica que a fines prácticos podrá ser reemplazado por un comentario.

4.4.2.1.1. Extensivo al Proceso Completo de Desarrollo

Muchas de las herramientas actualmente vigentes en el mercado no contemplan el proceso de ingeniería de software de forma integral, sino que se especializan en una parte específica del proceso.

Los autores Rojas, Perez y Griman realizan la siguiente mención: “La herramienta de desarrollo deberá soportar todas las etapas del ciclo de vida del desarrollo de un Sistema de Información, a saber: Análisis, Diseño, Implementación y Prueba” [ROJ00].

El Proceso Unificado de Rational propone un conjunto de fases y disciplinas que deberán recibir soporte **UML** en consecuencia. También es importante que la herramienta pueda dar un soporte completo de **UML**, adaptador a la norma, es decir la capacidad de las herramientas de construir todos los diagramas que propone este modelo, o por lo menos los más relevantes: el diagrama de casos de uso, que representa la funcionalidad o alcance del sistema; el diagrama de clases, que escribe la estructura de objetos y sus relaciones; el diagrama de interacción (secuencia o colaboración) , diagramas de actividad para describir un proceso de negocio y en algunos casos, el diagrama de estados, que visualiza el ciclo de vida de un objeto.

Las características a evaluar son las siguientes:

- Diagramas de casos de uso. Según el Proceso Unificado de Rational (**RUP**), centrar el desarrollo de software en casos de uso, permite acercarse a la validez funcional propuesta por los requisitos descriptos durante el análisis.
- Especificación de casos de uso, Describir la interacción entre el actor y el sistema de forma detallada promueve m corrección funcional.
- Diagrama de Dominio. Permite describir por medio de un diagrama de clases UML simplificado la estructura abstracta del sistema, conociendo solo entidades y relaciones más importantes. Es un modelo de Análisis que será tratado independiente al diagrama de clases.
- Diagrama de Actividades. Permite modelar procesos de negocio

- Diagrama de Clases. De refinar el diagrama de dominio obtenemos un diagrama de clases **UML** que permite especificar relaciones, estado y comportamiento de las entidades que se transformaran en objetos software. Es un modelo de diseño.
- Diagrama de Secuencia. Es un modelo de diseño que permite relacionar las clases software con la arquitectura y el comportamiento con el objetivo de darle una vista dinámica al sistema que se está diseñando.
- Aspectos Notacionales UML. Según [GEN06], es importante que la herramienta analizada “utilice fielmente la notación **UML**”. Es decir, que se adapte de forma correcta al estándar propuesto con el fin de mantener correctos los criterios de sintaxis y semántica dentro de los modelos. Este punto también considera los metadatos necesarios como, por ejemplo, métodos y atributos dentro de una clase, tipos de datos, etc.
- Validación de modelo UML. Para mantener la semántica de los modelos, es necesario que se puedan realizar validaciones. Por ejemplo, que un actor no se conecte con una clase.

4.4.2.1.2. Adaptabilidad y Flexibilidad

Se valora la posibilidad de que la información de distintos proyectos, guardados por una herramienta en un formato específico, pueda ser leída y compatibilizada por otras herramientas **CASE**, que soporten estos formatos, con el fin de ser reutilizados en otros desarrollos posteriores. Si la herramienta **CASE** soporta **UML**, la funcionalidad de intercambio de modelos con otras herramientas **CASE** se convierte en una meta alcanzable, teniendo en cuenta la estandarización sintáctica y semántica que persigue dicho lenguaje.

Las características a evaluar son las siguientes:

- Compatibilidad **XMI** (*Metadata Interchanger*). Es una especificación que permite compatibilizar entre Herramientas **CASE** que interpreten modelos **UML**.

- Exportación de imágenes. Permite portar los diagramas a otras plataformas como por ejemplo editores de texto.

4.4.2.1.3. Amigable para el Usuario

Según los estándares **ISO** dedicadas a la calidad del Software (**ISO 9241** e **ISO 14598**) la usabilidad es “grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso”. Es por este motivo que resulta sumamente subjetiva la evaluación de usabilidad.

Como ya sabemos, las herramientas **CASE** tienen un impacto positivo en la calidad y productividad de los desarrollos efectuados. Sin embargo, se requiere un excesivo esfuerzo y tiempo en comprender y utilizar las mismas. Por lo tanto, se deberían tener en cuenta los factores que influyen favorablemente en la tasa de aprendizaje.

Las características a evaluar son las siguientes:

- Ayuda en Línea. El usuario puede acceder a una ayuda indexada, actualizada e idioma propio.
- Manual de Usuario. La herramienta tiene un manual de uso disponible desde el momento de su instalación
- Soporte Técnico Flexible. Disponibilidad de acceso al soporte técnico de forma

4.4.2.1.4. Producir Documentación

Se deberá considerar un repositorio que permita centralizar, administrar y gestionar las versiones o estados de un proyecto en el que se requieren revisiones frecuentes. Deberá ser consistente con los requisitos del mercado y la comunidad, como ser la adopción de **UML** como lenguaje de modelado y otros estándares definidos en base a los criterios anteriormente mencionados. La documentación deberá estar basada en Diagramas, Modelos y Patrones y deberá ser personalizable.

4.4.2.1.5. Trabajo Online y Colaborativo

Consideramos para nuestro análisis se basa en utilización de la herramienta de *Google* denominada *Trends* [GOO17b] que muestra un resumen de tendencias sobre un determinado criterio de búsqueda en base a un plazo de tiempo determinado.

Google Trends es una herramienta que proporciona información sobre las búsquedas que los usuarios ingresan durante un periodo de tiempo y en un área geográfica determinada, es decir el volumen total de consultas para el término de búsqueda. Los datos de consulta se analizan con un método de muestreo ya que considera solo aquellas consultas realizadas un número significativo de veces.

Los autores del trabajo presentado en 2011 [CHO12] afirman que *Google Trends* no sirve para predecir el futuro, pero se puede utilizar como herramienta para “Predecir el Presente”. En estos términos, se puede lograr una predicción inmediata sobre temas de particular interés.

En base a esto utilizamos como criterio de búsqueda la frase “*UML tool*” y tomamos tres franjas de tiempo: entre enero de 2004 y enero de 2017 (Figura 4.4), entre enero de 2012 y enero de 2017 (Figura 4.5), y durante los últimos 12 meses (desde enero 2016 hasta enero 2017) (Figura 4.6). Notamos como a medida que se extiende en el tiempo la tendencia a la baja es aún mayor. Esto se debe a que los resultados son analizados en base a muestreos.

Podemos deducir entonces que durante los últimos años hubo una merma significativa en la cantidad de veces que se realizaron las consultas relacionadas, por este motivo y al no alcanzar los datos para el muestreo, descartamos este último período para el estudio en cuestión.

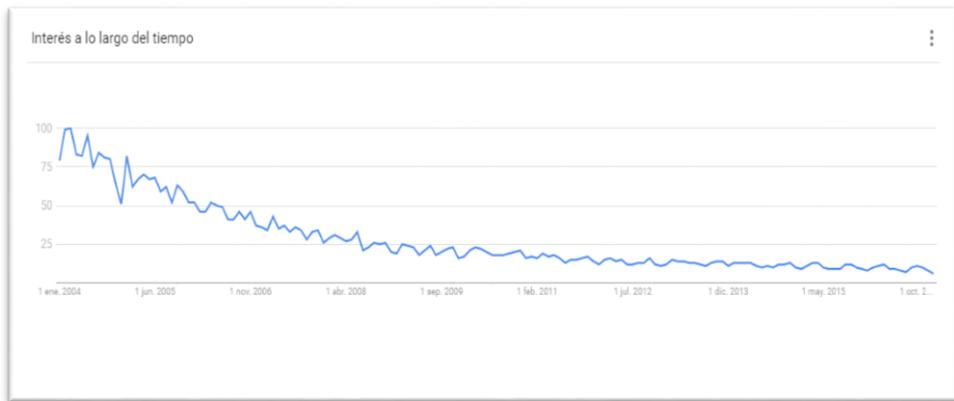


Figura 4.4. Tendencia de búsqueda entre 01/2004 y 01/2017

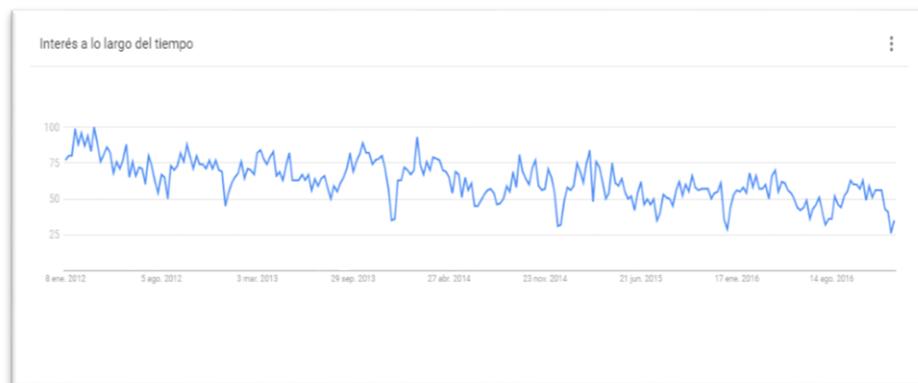


Figura 4.5. Tendencia de búsqueda entre 01/2012 y 01/2017

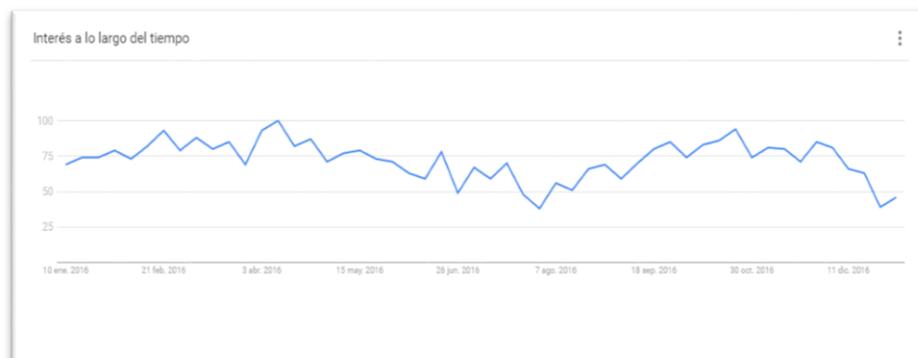


Figura 4.6. Tendencia de búsqueda entre 01/2016 y 01/2017

El concepto de Modelado Online aparece en cada una de las consultas que realizamos, incluso destacamos que muchas de las herramientas que aparecen dentro de las

consultas relacionadas son herramientas que se utilizan de forma Online, por eso planteamos la premisa de que la tendencia de modelado online sigue en camino ascendente.

Por este motivo, volvimos a realizar una consulta en *Google Trends* sobre "**ONLINE UML TOOL**" y descubrimos que, según los resultados, la tendencia en alza continúa siendo está una característica deseable para el modelado en el presente (Figura 4.7).

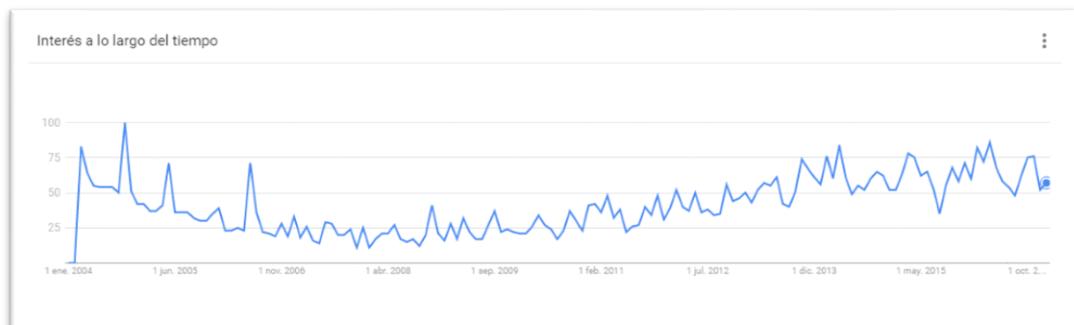


Figura 4.7. Tendencia en Google sobre "ONLINE UML TOOL"

Las características a evaluar son las siguientes:

- Modelado Online. Permite trabajar de forma online, es decir, por medio de una aplicación **WEB** que permita independizarse de la plataforma
- Trabajo Colaborativo. Permite realizar trabajo concurrente y colaborativo, diversas personas pueden operar sobre el mismo diagrama, en diferente tiempo y lugar. Posee control de concurrencia.

4.4.2.1.6. Manejo de OCL

OCL es el lenguaje formal adoptado alrededor de **UML**, es utilizado para especificar las invariantes, precondiciones y post condiciones y otras restricciones de modelos orientados a objetos con el fin de otorgar mayor formalidad a las especificaciones y aspectos visuales de **UML**. La integración de un lenguaje formal, es una de las características sobresalientes que distinguen una herramienta **CASE** robusta. El mismo

dota los diagramas con una semántica precisa y consistente, lo cual permite que la generación de código se pueda hacer de manera completa y eficiente.

Las características a evaluar son las siguientes:

- Interpretación de **OCL**. Provee un mecanismo de interpretación de lenguaje de restricción de objetos para definir reglas de negocio y validaciones que solo con **UML** no pueden definirse.
- Validación de Invariantes **OCL**. Provee un intérprete que permita validar las invariantes propuestas por el lenguaje.

4.4.2.1.7. Ingeniería Inversa y Generación de Código

Las herramientas que asisten al proceso de implementación, pueden generar código fuente. Si bien **UML** no es un lenguaje de programación, se puede usar para escribir programas, pero no posee la sintaxis y ni la semántica necesaria. Diferentes herramientas pueden generar código de tal manera que puedan importarse en un lenguaje de programación para luego utilizarse durante el proceso de implementación,

Generar e interpretar código existente hace que una herramienta **CASE** pueda participar activamente del proceso de Implementación. Para esto la herramienta debe poder analizar diferentes lenguajes y debe ser escalable en cuanto a la diversidad de lenguajes existente. Para nuestro estudio es necesario que la herramienta interprete lenguajes .NET

Las características a evaluar son las siguientes:

- Generación de Código .NET. Permite generar código a lenguajes .NET
- Ingeniería Inversa. Permite importar el código fuente de una solución .NET para realizar el diagrama de clases correspondiente.

4.4.2.1.8. Apoyo Metodológico

Es importante que la herramienta provea soporte para alguna metodología tradicional de desarrollo de software para poder realizar trabajos de modelado en todas las fases principales: Análisis, Diseño e Implementación. De esta manera, el usuario deberá conocer en todo momento que rol está cumpliendo al momento de operar la herramienta.

Las características a evaluar son las siguientes:

- Soporte para identificación de Roles. deberá identificar en todo momento que rol está cumpliendo el usuario. Por ejemplo, si está haciendo un diagrama de clases, podría indicarse que está trabajando en el Diseño.

4.4.2.2. Ficha de Evaluación

A Continuación, se describe la ficha de evaluación resultante para para poder realizar el estudio en cuestión. De cada herramienta nos interesa conocer Nombre, Empresa, Descripción, Versión analizada, Año, Plataforma y Precio en primera instancia y luego utilizamos los criterios del apartado anterior con el sistema de puntuación definido (Tabla 4.2).

1. Principal		
Nombre		
Empresa		
Descripción		
Versión		
Plataforma		
Precio		
2. Experiencia del Usuario		
Ayuda en Línea		
Manual de Usuario		
Soporte Técnico Flexible		
Trabajo Online		
Trabajo Colaborativo		
Compatibilidad XMI		
Exportar Imágenes		
Producir Documentación		
TOTAL PUNTUACIÓN 2		
3. Ingeniería de Software		
Diagrama CU		
Especificación CU		
Diagrama de Dominio		
Diagrama de Actividades		
Diagrama de Clases		
Diagrama de Secuencia		
Aspectos Notacionales UML		

Validación de Modelo UML		
Interpretación OCL		
Validación Inv. OCL		
Generación Código .NET		
Ingeniería Inversa		
Identificación de Roles		
TOTAL PUNTUACIÓN 3		
PUNTIACIÓN TOTAL		

Tabla 4.2. Ficha de Evaluación para Herramientas CASE

4.4.3. Proceso de Evaluación de Herramientas CASE

Para comparar diferentes herramientas **CASE** se utilizó el proceso de selección propuesto por Topper, Et al TOP94]. Este proceso consta de 4 pasos bien definidos, de los cuales, y a fines prácticos, sólo usaremos solo 3 ya que el último paso corresponde a la realización de un desarrollo con la herramienta de puntuación más alta para poder medir resultados asociados a la relación costo-beneficio. Creemos que este último punto no es necesario para el tipo de análisis que estamos llevando adelante y por tal motivo lo excluimos del proceso.

- (1) Realizar una revisión exhaustiva de las herramientas disponibles,
- (2) probar un pequeño grupo de una selección del paso anterior, probarlas a través de un proyecto piloto o de una evaluación más detallada,
- (3) presentar una puntuación de las herramientas y seleccionar la de puntuación más alta,

4.4.3.1. Puntuación de las Herramientas

A continuación, se resume el nivel de aceptación de las herramientas de modelado que estudiamos (**Ver Anexo I**). Si bien existe una gran diferencia entre los extremos en el resultado, destacamos un aspecto muy importante: Ninguna de estas herramientas posee de forma explícita, información contextual relativo al proceso de desarrollo. En nuestra metodología de evaluación lo denominamos “Identificación de roles”, alineado al concepto de “Apoyo Metodológico”. Esto es, la capacidad de los sistemas para brindar

información contextual relativa al proceso de desarrollo de software, a que fase corresponde y que rol (dentro de un posible equipo de desarrollo) está llevando a cabo la actividad; por ejemplo: El modelo de casos de uso es parte del análisis y es llevado a cabo por analista funcionales luego de la identificación de los requisitos funcionales.

Otro aspecto a considerar, es el relacionado con la Interfaz de usuario y la relación directa que tiene con la cantidad de funcionalidades que posee la herramienta. A mayor nivel de aceptación, mayor es la complejidad de uso por lo tanto más difícil de aprender.

Una consideración que arrojó el estudio es que no todas las herramientas hacen foco en el trabajo colaborativo. Si bien las tendencias actuales sobre la forma de trabajo son crecientes sobre la necesidad de desarrollar software en equipo, muchas herramientas tienen la posibilidad de hacer trabajo compartido, pero no poseen las herramientas necesarias para que la colaboración sea determinante en el proceso de desarrollo de software.

A continuación, en la tabla 4.3, se muestra el resumen de aceptación de cada herramienta analizadas e indicando ventajas y desventajas más destacadas.

Herramienta	Aceptación	Ventajas	Desventajas
Enterprise Architect	86.40 %	Herramienta muy completa con funcionalidades CASE de extremo a extremo del proceso de desarrollo. Soporte especificación completa de elementos UML como ser Clases, Casos de Uso, etc. Documentación muy completa y detallada.	Interfaz muy compleja, difícil de aprender, con muchas funcionalidades.
ArgoUML	77.27 %	Herramienta completa. La Interfaz de usuario no es muy compleja, por lo	No contempla trabajo colaborativo y no posee soporte completo para la

		tanto, es accesible para su aprendizaje.	información inherente a los elementos UML . No permite especificar casos de uso
Visual Paradigm	77.27 %	Contempla trabajo colaborativo y posee interfaz fácil de aprender y accesible para usuarios de baja experiencia (o nula)	No permite especificaciones detalladas de componentes UML . Interfaz de usuario. Es una herramienta muy cara para un usuario que requiera las funcionalidades básicas de modelado.
StarUML	68.28 %	Es Opensource. No presenta complejidad para su uso y es fácil de aprender.	No posee soporte para especificación de componentes UML .
Gliffy	47.6 %	Herramienta sencilla y rápida. Fácil de aprender.	Incompleta. No soporta UML de forma integral y no permite especificar componentes.
yEd	45 %	Herramienta sencilla para aprender.	Muy incompleta. No soporta UML de forma integral.

Tabla 4.3. Resumen comparativo sobre Herramientas de Modelado

4.4.3.2. Características Deseables

El objetivo del trabajo es proponer una solución a los problemas en la enseñanza de la ingeniería de software. Los potenciales usuarios serán alumnos de los primeros años de

la carrera de **IS** en general, con poca o nula experiencia tanto en el modelado como en el uso de herramientas **CASE**.

Luego del estudio realizado sobre las principales herramientas de modelado, determinamos un conjunto de características deseables que sirven como base para el desarrollo de una solución acorde a nuestras necesidades.

- La herramienta debe ser muy amigable y sencilla de aprender ya que va a estar destinada a alumnos universitarios de los primeros años.
- Debe mantener constantemente información contextual sobre el proceso de desarrollo, indicando roles e información inherente a cada actividad desarrollada.
- Debe soportar **UML** de forma integral, desde aspectos de diseño y notacionales, hasta validación de modelos y especificación detallada de clases, casos de uso, objetos, etc.
- Las validaciones de los modelos UML deben estar acompañados con mensajes personalizados en cada caso, mostrando el error y la forma correcta de hacerlo.
- Debe permitir interacción *online* y colaborativa para que múltiples usuarios trabajen juntos o separados, en el mismo tiempo o en momentos diferentes, sin que esta característica afecte negativamente el proceso de enseñanza y aprendizaje. Por tal razón, la herramienta debe disponer de información de *awareness* acorde a esto. Además, debe permitir mantener versiones y revisiones del trabajo realizado, con la información de quién lo hizo, qué hizo, cuándo y qué observaciones tuvo el grupo sobre esto.
- Debe facilitar una completa integración con otras herramientas del mercado y permitir compatibilidad con otros formatos estándares tales como imágenes o PDF.
- Debe tener documentación de uso y manuales de usuario accesible y fácil de leer con videos y ejemplos prácticos.
- Debe permitir la generación de documentación de forma personalizada y compatible con los formatos más comunes del mercado.

4.5. Resumen

En este capítulo definimos el concepto de herramienta **CASE** y sus principales características. Contextualizamos **UML** y lo relacionamos con el proceso unificado de Rational con el objetivo de entender más en profundidad las herramientas de modelado y el impacto en el proceso de desarrollo de software. Definimos en base a esto, un conjunto de criterios para poder evaluar y comparar diferentes herramientas. Diseñamos un método para hacer esta evaluación con el objetivo de obtener un índice de aceptación de cada una y poder tener una comparación objetiva. Definimos las características más importantes para una herramienta **CASE** de modelado destinada a la enseñanza y el aprendizaje de la ingeniería de software.

Capítulo 5

uCASE-CL: Enseñanza y Aprendizaje Colaborativo de la IS en Entornos Virtuales Ubicuos

5.1. Introducción

El impacto de las **TIC** en la educación genera nuevas oportunidades para el aprendizaje colaborativo, creando los entornos de aprendizaje colaborativos asistidos por computadora (**CSCL**) para generar conocimiento.

Sin embargo, existen áreas de conocimiento específicos que, aparte de la psicología y la pedagogía propias de los entornos **CSCL**, requieren técnicas y herramientas específicas para poder completar efectivamente el proceso de enseñanza y aprendizaje. No solo son herramientas propias de la temática que está estudiando, sino que el proceso, además, requiere actividades de evaluación y seguimiento.

La ingeniería de software (**IS**) es un área de conocimiento que requiere de herramientas específicas, como son las herramientas **CASE**. En este sentido, comprendemos que para que el proceso de enseñanza y aprendizaje dictaminado en un ámbito **CSCL** cumpla su objetivo, deberá disponer de forma integrada una herramienta con las características necesarias para esto, considerando el concepto de **CASE** como cualquier herramienta que brinde soporte a alguna actividad enmarcada en la **IS**.

Enseñar **IS** no es solo disponer de una herramienta acorde, también requiere un proceso de evaluación que permita medir el nivel de aprendizaje de un alumno o grupo de alumnos que participan en este proceso de aprendizaje específico.

Es importante destacar la complejidad que representa la realización de modelos para principiantes o novatos (independientemente del lenguaje utilizado). Para esto, suelen utilizarse las herramientas presentes en el mercado (de uso profesional) o algunas herramientas de modelado que no son tan comunes, pero en su mayoría o son muy complejas o son incompletas para el proceso de enseñanza, aprendizaje y evaluación de la temática específica.

En este capítulo presentamos un modelo que especifica y vincula las plataformas **CSCL** con la enseñanza de la ingeniería de software, en particular con el modelado por medio de **UML**. La idea principal de esta propuesta es definir y especificar los bloques funcionales que son necesarios para el proceso específico de la enseñanza y aprendizaje de la **IS**.

5.2. Enseñanza del Modelado UML Utilizando Herramientas CASE

En un mundo rico de información, no es suficiente con que ésta esté disponible en cualquier momento, lugar y forma. La información que llega al educando de forma ubicua debe ser pertinente con el grado de necesidad y acorde al contexto en que se encuentre el individuo, en nuestro caso, el estudiante. Las actuales plataformas de formación mediadas por la tecnología están destinadas a facilitar las tareas del docente y del alumno, pero carecen de herramientas que permitan la interacción entre ambos en tiempo real [FIL10].

La planificación y ejecución de procesos de enseñanza y aprendizaje para cursos de **IS**, plantean un gran desafío a los docentes universitarios involucrados. La necesidad de una actualización dinámica de los contenidos no debe provocar el descuido de conceptos básicos vinculados a los principios fundamentales del desarrollo de sistemas de software [DAN10].

Las actuales plataformas de formación mediadas por las nuevas **TIC** están orientadas principalmente a facilitar la tarea del docente para que los alumnos puedan acceder al material educativo en cualquier momento y lugar [FIL10]. Sin embargo, hay áreas de conocimiento que requieren de recursos externos para que ciertos temas puedan ser enseñados y evaluados, como es el caso de las herramientas **CASE**. Para cursos específicos sobre **IS**, requieren de esfuerzos extra no solo para mejorar la colaboración entre docentes y alumnos, sino también para poder realizar evaluaciones y seguimiento de trabajos [NEI16]. En este sentido W. Chen [CHE06] afirma que las herramientas comúnmente no involucran ningún modelo pedagógico que contemple actividades de aprendizaje.

Es importante que los alumnos no pierdan el foco y no solo puedan aprender de experiencias propias y de otros alumnos, sino que también el docente pueda brindarles el conocimiento necesario. Esto sucede también, en aquellas áreas de conocimiento dentro de la **IS** que requiere el uso de herramientas como, por ejemplo, el modelado de software, en donde las realizaciones de las actividades académicas logran mejores resultados por medio de la utilización de una herramienta acorde y con tutores que conocen a la perfección el dominio [CON00].

En este sentido, y dentro del proceso de enseñanza y aprendizaje de la **IS**, proponemos un enfoque destinado al modelado de software, en particular al relacionado con **UML** como lenguaje sintáctico y semántico para tal fin.

5.2.1. Enseñanza de UML

Kendra Cooper [COO05] indica que la notación gráfica es normalmente más fácil de comprender que las descripciones textuales, especialmente a fines de enseñanza. En este sentido, comprendemos que el estándar **UML** provee una notación que permite enseñar a modelar software pudiendo obtener el foco de atención en el problema y no en el medio.

En base a resultados obtenidos en diferentes estudios realizados [FLI05] [COO05], [CHE06], deducimos que **UML** presenta diferentes problemas durante el proceso de enseñanza y aprendizaje:

- El modelado **UML** es muy complejo en especial para alumnos como poca experiencia.
- **UML** posee rango amplio de aplicabilidad, generando un lenguaje grande y ambiguo en algunos casos. El tamaño de especificidad de la norma es muy grande y complejo para lo que debe ser enseñado a los alumnos.
- Los enfoques populares para el desarrollo de software están en desacuerdo con algunos principios importantes de la **IS**, en particular a aquellos que son parte del proceso de enseñanza y aprendizaje. Estos métodos suelen fomentar la confusión de las actividades del análisis y diseño, y entra en conflicto la capacidad de enseñar y demostrar la importancia de separar en conceptos un software (**SOC**, *Separation of Concerns*). Por último, los autores proponen que los métodos que

fomentan los modelos parciales, entran en conflicto con las buenas prácticas de la industria.

- Los estudiantes suelen utilizar las últimas tecnologías de implementación, pero estas no suelen tener soporte (o es muy pobre) para los principios de la ingeniería que se enseñan durante el modelado **UML**. En particular, los alumnos suelen estar influenciados por las tecnologías, generando complicaciones al momento de enseñar **IS**.
- No hay manera de saber si los casos de uso (**CU**) son realmente los **CU** correctos ni tampoco de asegurar la trazabilidad entre **CU** funcionales. Esta ambigüedad impacta directamente en el proceso de su enseñanza generando confusiones y alentando a la omisión de funcionalidades.
- La falta de conocimiento detallada de la norma, fomenta confusiones relativas a las relaciones entre diagramas o elementos dentro de un diagrama y las validaciones semánticas.
- Al ser un lenguaje de notación, no provee mecanismos de validación explícitos y automáticos que permita conocer las validaciones semánticas.
- El uso del mismo modelo **UML** para distintos niveles de abstracción genera confusión en los alumnos. Por ejemplo, el uso del diagrama de clases para modelar el dominio en la fase de Análisis y luego el diseño detallado en la fase siguiente (ver figuras 5.1 y 5.2).

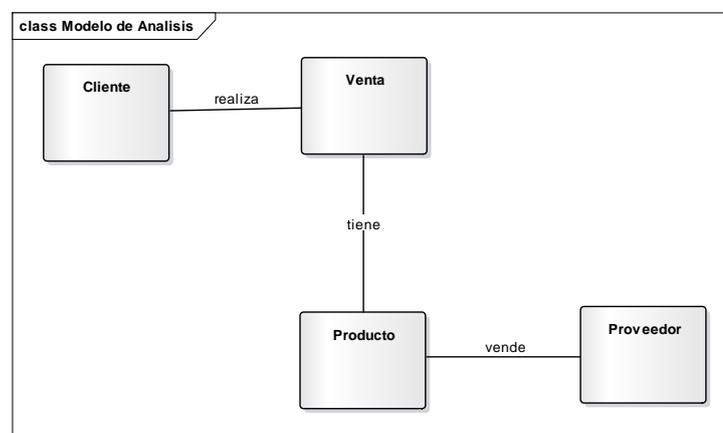


Figura 5.1. Diagrama de clases en la vista de análisis

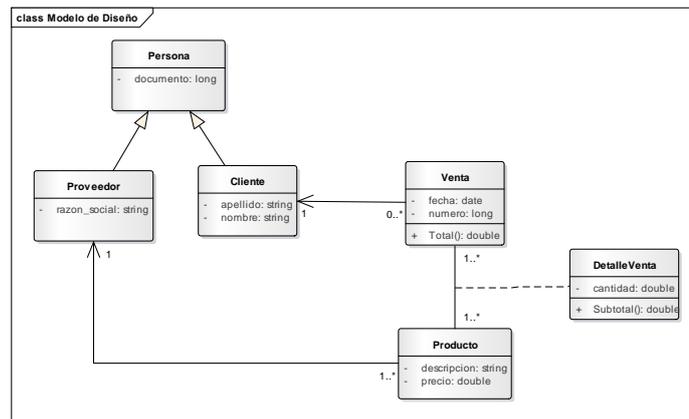


Figura 5.2. Diagrama de clases en la vista de diseño

5.2.2. Ventajas y Desventajas de Utilizar Herramientas CASE

Utilizar herramientas de modelado durante el proceso de enseñanza y aprendizaje de la IS sugiere un conjunto de ventajas y desventajas que deberán ser tratadas cuidadosamente [NEI16] [BAT16] [DEL11a] [ENG05] [TEN08].

Las ventajas destacadas son

- Utilizar herramientas **CASE** permite manejar desarrollo de sistemas grandes y complejos, permitiendo su escalabilidad de forma sencilla.
- Este mismo concepto aplica a la mantenibilidad, ya que permite realizar los cambios con el menor esfuerzo posible y de forma sencilla.
- Utilizar herramientas **CASE** permite achicar la brecha entre fases del proceso de desarrollo. Un modelo de clases **UML** puede ser convertido en código para utilizar en algún lenguaje de programación.
- El resultado no solo supera ampliamente a los papeles “garabateados”, sino que también permite un mejor entendimiento por parte del alumno, que puede concentrarse en la notación por sobre la forma.
- Otra ventaja es el acercamiento a la práctica profesional. Utilizar este tipo de herramientas permiten al alumno comenzar con experiencia previa a su graduación

En contrapartida a las ventajas, el uso de herramientas **CASE** para la enseñanza promueve un conjunto crítico de desventajas, a saber

- Las herramientas más comunes en la industria poseen muchas funcionalidades y maneja muchos aspectos que no son necesarios a fines académicos. Esto hace que los alumnos no puedan concentrarse en el proceso de aprendizaje. El uso de herramientas **CASE** tienen impacto negativo en la curva de aprendizaje. Los alumnos deberán comprender como usar la herramienta antes de centrarse en el aprendizaje, por ejemplo, de modelado **UML**.
- En muchas ocasiones los costos de instalación, mantenimiento y operación resultan excesivos, sobre todo si se considera la cantidad de licencias necesarias para un entorno académico.
- Las interfaces resultan muy complejas para los alumnos poco experimentados, lo que puede fomentar la pérdida de interés o de concentración.
- En muchas oportunidades, las herramientas no soportan de forma correcta el estándar **UML**, generando posible inconsistencia sobre la formación de conocimientos.
- Las herramientas carecen de funcionalidades destinadas al proceso de enseñanza y aprendizaje. No disponen de herramientas colaborativas para la evaluación y autoevaluación de modelos software.
 - El alumno no dispone de la información o *awareness* inherente al proceso colaborativo de aprendizaje que se está llevando adelante. Esta información es importante ya que facilita el trabajo teniendo percepción de lo que está sucediendo con él y con los otros usuarios del ambiente.
- El docente tampoco cuenta con el entorno necesario para poder hacer un seguimiento y evaluación de los proyectos llevados a cabo por un alumno o grupo de alumnos. Esto implica tener que utilizar herramientas externas que permitan poder trazar las correcciones realizadas e inclusive, herramientas que coordinen la comunicación

5.4. Nuestra propuesta: uCASE-CL

En el capítulo 3 estudiamos el modelo **CSCL**. Es un enfoque del aprendizaje basado en la psicología de las interacciones sociales, la pedagogía de la enseñanza, y la informática.

Estudiamos también, que existen áreas de conocimiento específico (como por ejemplo el modelado de software) que requieren herramientas específicas para utilizar durante los procesos de enseñanza y aprendizaje inherentes.

Esta afirmación da la pauta que es necesaria una especificación del modelo **CSCL** que permita utilizar una herramienta de dominio específico para utilizar durante la enseñanza y aprendizaje incluyendo los procesos específicos de evaluación y seguimiento de trabajos

En esta línea, durante el proceso de modelado de software, se utilizan herramientas **CASE** de para tal fin no integradas con entornos de enseñanza y aprendizaje. Esto se opone al crecimiento acelerado de las **TIC** y su aplicación constante en ese tipo de entornos, los cuales están cada vez más integrados con el aula generando entornos mixtos que permiten aprovechar al máximo todo el potencial de la tecnología aplicada a la educación.

Comprendemos que es sumamente necesario que los modelos de aprendizaje colaborativo introduzcan en su especificación las herramientas necesarias para que, además de aprender y evaluar de forma colaborativa, permitan también mayor acercamiento a las prácticas profesionales impuestas en el mercado.

En la actualidad, los modelos de aprendizaje colaborativo asistido por computadora poseen sus procesos formativos impactados por un paradigma educativo desarrollado en cualquier espacio, lugar, tiempo y dispositivo. Estos se ven inmersos en un universo tecnológico ubicuo, basado en las tecnologías de la comunicación. Este modelo fue denominado **uCSCL** por Coto, Collazos y Rivera [CCR16] [COL15] y por Yandar & Moreno [YAN15], entre otros autores.

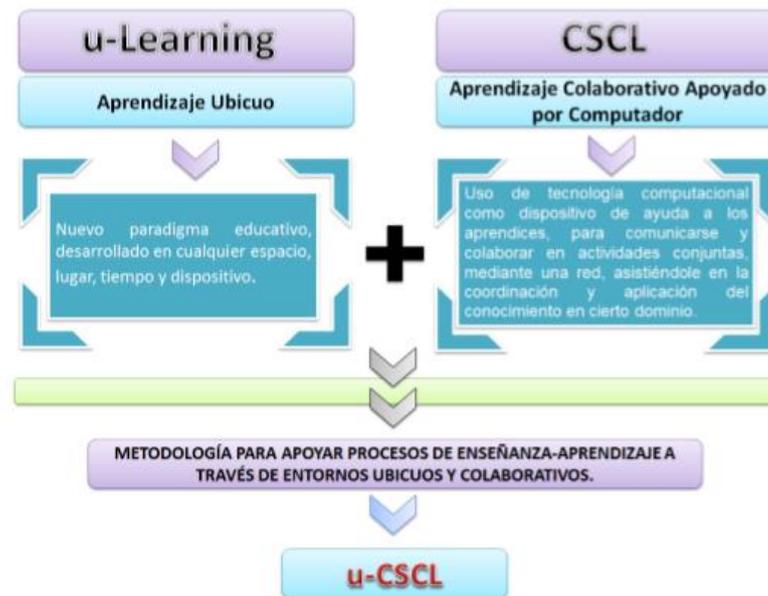


Figura 5.3. Modelo uCSCL de Coto y Collazos [CCR16]

En el modelo se observa un bloque que refiere a una “Metodología para apoyar procesos de enseñanza-aprendizaje a través de entornos ubicuos y colaborativos”. Esto es, apoyado por la ingeniería de colaboración, una forma de modelar todos los procesos colaborativos necesarios por el modelo (actividades y contenidos) incluyendo actividades formativas como la evaluación.

Comprendemos que el modelo **CSCL** planteado con sus tres dimensiones (Ciencias de la Computación, Psicología y Pedagogía) resulta afectado por una cuarta dimensión a la que denominaremos *ciencias de la comunicación*, en representación de la tecnología subyacente que permite materializar el concepto de ubicuidad. La transformación planteada se ve reflejada en el siguiente diagrama (figura 5.4).



Figura 5.4. Modelo uCSCL

Los autores del modelo **uCSCL** aseveran que todos los modelos educativos poseen 5 componentes principales y se ve representado en la siguiente ilustración. 1) Docentes, 2) Estudiantes, 3) Materiales de Estudio, 4) Plataforma Tecnológica y 5) Servicios de Acceso. Esta distribución de recursos la plantean en la siguiente ilustración (figura 5.5).

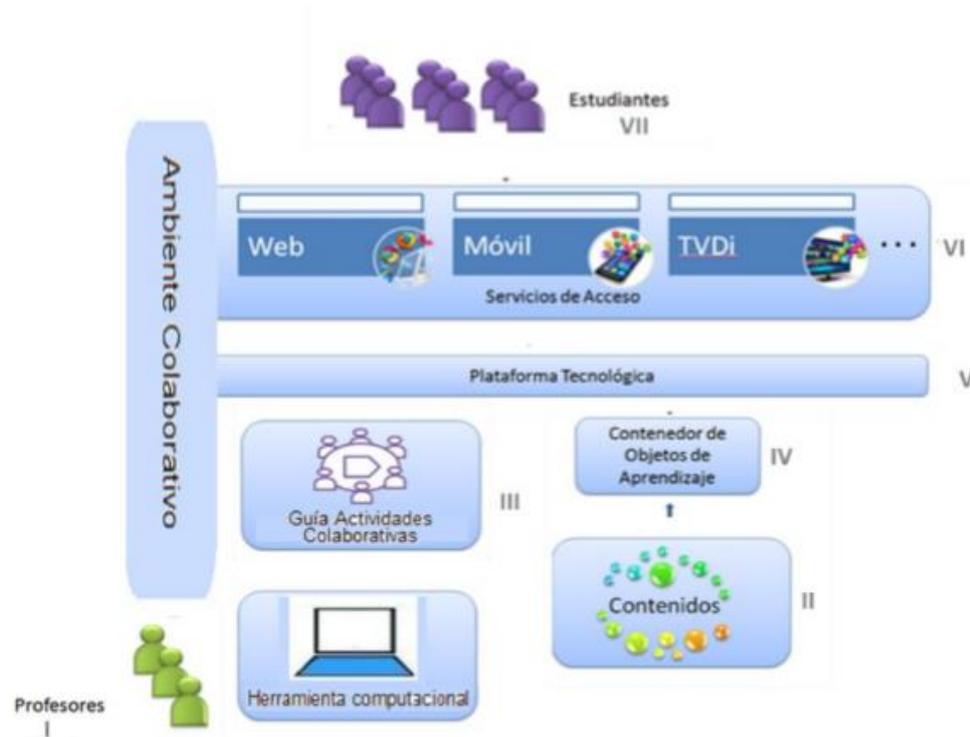


Figura 5.5. Modelo U-CSCL propuesto en [CCR16]

En ese trabajo [CCR16], los autores explican que los contenidos y las actividades deben ser incorporados utilizando estándares definidos por el modelo, estableciendo un contenedor de objetos de aprendizaje. Proponen también una herramienta computacional que será utilizada por los docentes para la elaboración de cursos. Por último, la plataforma tecnológica, basada en el diseño de una herramienta que integre componentes de *e-learning* para prestar servicios por medio de un modelo de aprendizaje colaborativo de forma ubicua, lo que permitirá a los alumnos acceder a la información desde cualquier parte y lugar.

Comprendemos que es necesario ampliar este modelo, primero para brindar un bloque funcional de evaluación y por último para poder integrar nuevas herramientas de enseñanza y aprendizaje para aquellas áreas de conocimiento específico, como ser la ingeniería de software.

En primer lugar, creemos importante extender los componentes necesarios que conforman los modelos **CSCL** ya que los entornos colaborativos de enseñanza comienzan

a tener injerencia en el proceso de evaluación. Esto transforma al modelo **CSCL** tradicional, y le abre nuevas puertas a la evaluación en el entorno colaborativo.

En segundo lugar, es necesario integrar el entorno de aprendizaje con aquellas herramientas propias de la práctica profesional. Estas herramientas deberán contar con recursos colaborativos no solo para la actividad principal de la temática, sino también para actividades pedagógicas que deben ser utilizadas durante el aprendizaje, como ser la evaluación colaborativa y autoevaluación de equipos de trabajo. Esto da lugar a la participación activa del docente en el entorno y la posibilidad de explotar al máximo el aprendizaje utilizando la ubicuidad no solo para aprender, sino también para evaluar.

El modelo propuesto integra el modelo **uCSCL** con una herramienta **CASE** (en nuestro caso, de modelado **UML**) con todas las funcionalidades requeridas por la industria y aquellas funcionalidades específicas de la enseñanza, aprendizaje y evaluación relativa a la **IS**. Denominamos a este modelo como **uCASE-CL** (*Ubicuos CASE Collaborative Learning*) [BAT17b] [BAT17d]. En la figura 5.6 se muestra el enfoque general.

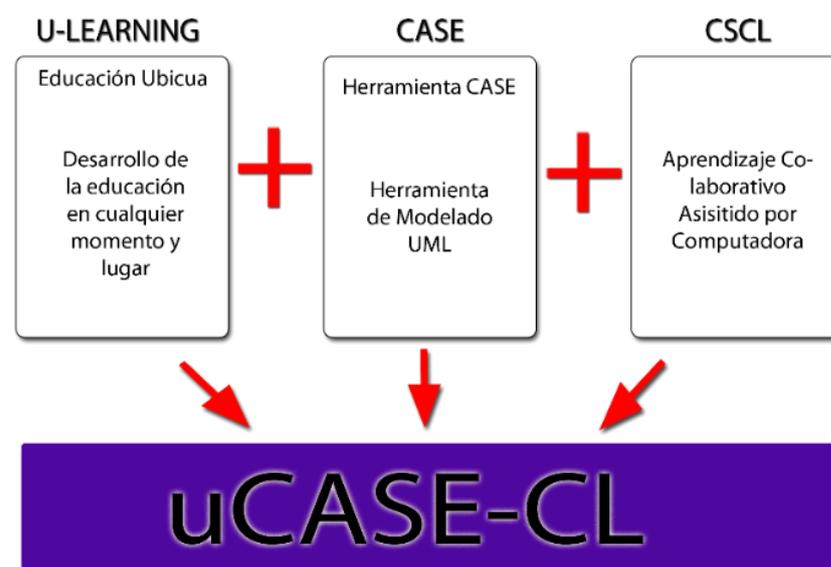


Figura 5.6. Enfoque general uCASE-CL

Planteamos que además de los 5 componentes propuestos en [CCR16], existe una convergencia con los elementos propuestos en [ROS08]. En este punto comprendemos que son necesarios, además, las herramientas y las actividades de evaluación específicas de la **IS**.

El modelo **uCASE-CL** queda propuesto con los siguientes componentes

- Roles (Docentes y Estudiante)
- Plataforma de acceso
- Servicios de acceso
- Plataforma tecnológica Colaborativa
- Herramienta **CASE**
- Objetos de aprendizaje
- Contenidos
- Evaluación colaborativa
- Actividades colaborativas
- Guía para definir actividades colaborativas.

En la siguiente figura se observa la integración explícita de una Herramienta **CASE** en un entorno **CSCL**, denominado **uCASE-CL**.

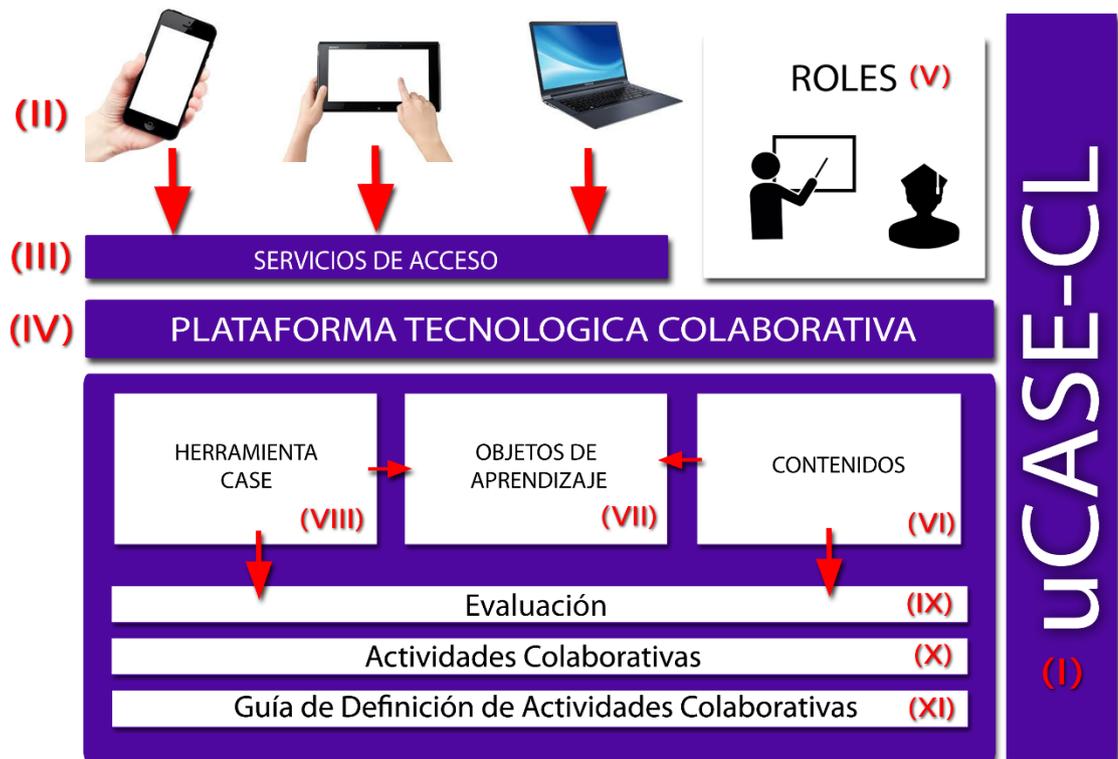


Figura 5.7. Modelo UCASE-CL detallado

A continuación, explicamos las referencias indicadas en la figura 5.7 (en números romanos).

(II) En la capa superior, el acceso ubicuo desde diferentes plataformas. Este punto conforma uno de los pilares del modelo, ya que el mismo se centra en la enseñanza ubicua. (III) Los Servicios de Acceso representan al paradigma tecnológico en el cual se apoya la Enseñanza Ubicua. Esto está directamente relacionado con la tecnología utilizada para poder brindar un servicio ubicuo.

(IV) La Plataforma Tecnológica Colaborativa es un nivel de abstracción que permite encapsular la tecnología necesaria para brindar servicios colaborativos. En este punto se apoyan los docentes y los alumnos para poder realizar actividades de colaboración de forma ubicua.

La Herramienta **CASE** (VIII) y los Contenidos (IV) representan todos elementos fundamentales del entorno de enseñanza. Es decir, conforman los objetos de aprendizaje (VII), en donde . Este es el caso, por ejemplo, una actividad teórica específica que requiera colaboración grupal para su resolución. Sin duda que esto puede representar a los alumnos o a los docentes, incluso los docentes podrían crear de forma colaborativa

los contenidos de un curso en particular. Esto último, representa la dimensión pedagógica de la enseñanza en entornos colaborativos.

Tanto las herramientas **CASE** como los contenidos, poseen actividades de evaluación. En nuestro modelo, la evaluación a definir está ligada con el trabajo colaborativo del modelado **UML**, por tal motivo es muy importante definir actividades de evaluación (IX): Evaluación colaborativa y autoevaluación (entre otros) tal como se planteó en capítulos anteriores.

Por último, se encuentra un nivel de abstracción que representa el conjunto de actividades colaborativas (X), donde se incluyen, entre otras, las de evaluación y las propias de la **IS**. Este nivel corresponde a los bloques afectados por la dimisión psicológica del modelo CSCL, ya que es aquí donde se trabajan con todas las actividades colaborativas, utilizando una técnica predefinida para su representación (XI). Esta abstracción representa y da soporte a la psicología de las actividades sociales, en este caso entre docentes y alumnos (V).

5.5. Implementación del modelo

Un entorno de aprendizaje apoyado en la tecnología como el propuesto (**uCASE-CL**) requiere de cuatro dimensiones interrelacionadas entre sí. Las ciencias de la computación, las ciencias de la comunicación, la pedagogía y la psicología. Para poder realizar el diseño y una posterior implementación de una plataforma **uCASE-CL** será necesario incluir todos estos factores.

5.5.1. Dimensión Tecnológica

De estas cuatro dimensiones, dos están estrechamente vinculadas en forma de una dimensión tecnológica: La computación y las comunicaciones, representando las **TIC**. Si bien las ciencias de la comunicación se apoyan en la tecnología, este aspecto representa la infraestructura necesaria para brindar un servicio a la educación ubicua. La computación o ciencias de la computación representan a las herramientas informáticas que se utilizan para construir el entorno de aprendizaje en su totalidad. Son sumamente necesarias las tecnologías de integración entre estos conceptos, ya que es a través de la

computación y la comunicación que la plataforma tecnológica será accesible de forma ubicua.

Esta dimensión representa los puntos (II), (III) y (IV) del modelo **uCASE-CL** planteado anteriormente, incluyendo toda la plataforma de aprendizaje (con la herramienta **CASE** integrada)

5.5.2. Dimensión Pedagógica

La perspectiva pedagógica permite comprender los procesos de interacción social y el impacto pedagógico en la formación. En esta dimensión deberán verse reflejadas todos los procesos donde habrá interacción social entre las personas involucradas. Estos procesos representan las actividades de formación específicas incluidas en las herramientas de enseñanza propias de la **IS (CASE)**. Esto significa que las herramientas deberán contar con las utilidades de colaboración necesarias para consolidar la creación de conocimiento de forma colaborativa. Esto último deberá contemplar e incluir también temas relacionados a la evaluación y seguimiento de trabajos específicos.

Esta dimensión representa los objetos de aprendizaje (VII) junto con los contenidos (VI) y la herramienta **CASE** (VIII). Las actividades de colaboración dentro de la plataforma (IX), (X) y la interacción entre los docentes y alumnos (V). Deberá considerar de forma explícita diferentes medios de evaluación general sobre la plataforma de aprendizaje y específica sobre el proceso de **IS** correspondiente.

5.5.3. Dimensión Psicológica

El modelo propuesto debe considerar aspectos colaborativos entre docentes y alumnos, en particular, los que derivan de la evaluación, seguimiento del aprendizaje y la aplicación de modelos para el análisis y diseño de software. Identificar y describir esta actividad no es un proceso sencillo. Contar con un método de especificación y el uso de modelos para formalizar los roles, las interacciones y las transformaciones de forma colaborativa, representa acercarse a un proceso de calidad.

La interacción social es importante para establecer un espacio en el que se pueda encontrar una estructura que abarque las relaciones sociales, la cohesión grupal, la

confianza y la pertenencia, todo lo cual contribuye a la comunicación abierta, el pensamiento crítico.

Esta es la razón por la cual los educadores han encontrado una relación positiva entre la interacción social en la dimensión socio-psicológica y los resultados en términos resultados de aprendizaje y satisfacción del estudiante. Por este motivo consideramos necesario representar las interacciones sociales apoyadas en el concepto psicológico del estudio de la actividad humana.

La teoría de la actividad (**TA**) identifica los elementos de la actividad e indica cómo los aspectos sociales influyen en la forma de actuar de los grupos. Para pasar de este plano sociológico (teórico) a uno tecnológico (práctico) es necesario hacer un modelo abstracto con los elementos de la teoría y posteriormente implementarlos en un sistema informático, representando los escenarios de colaboración.

La **TA** es útil como método de análisis para el desarrollo de un sistema, ya que ayuda a identificar e interrelacionar los elementos que intervienen en su definición atendiendo, fundamentalmente, a la organización y estructuración social. En este sentido, comprendemos que es útil diseñar una metodología que permitiese aplicar la **TA** comprendiendo que una actividad puede descomponerse en otras más simples, que se detallan como partes de la misma. Esta división del trabajo se descubre como una manera de realizar una adecuada planificación del proyecto de desarrollo que se asocia a la actividad. Cuando se implementa el contenido teórico de la **TA** en un modelo informático, es necesario decidir cómo se va a representar cada elemento que lo conforma.

El concepto de actividad en el proceso de enseñanza y aprendizaje está asociado a todas las acciones que lleven a facilitar el proceso. Entre ellas destacamos observación, escucha, trabajo en equipo. Las realizadas por medios virtuales poseen las mismas características y funcionalidades que las presenciales. Estos entornos permitirán que los estudiantes y docentes puedan realizar además de las actividades individuales, las de carácter grupal y colaborativas. Entre estas últimas podemos destacar actividades de evaluación, de autoevaluación, en grupo y con el docente [CAB06b].

Alcanzar un objetivo implica realizar un conjunto de tareas relacionadas. Si bien Hay tareas que son de realización individual, como por ejemplo la corrección de un examen en un curso presencial, es muy común encontrar que estas tareas requieren de trabajo

colaborativo para poder conseguir los resultados esperados. Este es el caso, por ejemplo, de un conjunto de docentes colaborando para cumplir el objetivo de definir el plan de estudio de un curso en particular. Esta tarea, de realización íntegramente colaborativa, requiere de un conjunto de procesos y actividades que son importantes para el objetivo [SOL13].

Para esto es necesario asumir la necesidad de aplicar una metodología que permita materializar el modelo para poder implementarlo en la plataforma tecnológica. Esta dimensión psicológica representa la guía de definición de actividades colaborativas (XI) planteada en el modelo **uCASE-CL**. Para poder mantener un enfoque adecuado en el uso de la **TA**, es necesario identificar en cada actividad los siguientes elementos constitutivos [BAR04]:

- La *comunidad* es el conjunto de individuos que conforman los grupos colaborativos
- La *división del trabajo* responde a la descripción del conjunto de tareas que se van a llevar a cabo. En este punto se describen los roles y las tareas que lleva acabo cada uno.
- Las *normas de la comunidad* definen la forma que se tiene que trabajar en grupo.
- Las *herramientas* constituyen la manera de expresar todo el material que se proporciona al grupo, tanto textual, como los medios que permiten llevar a cabo las tareas, la comunicación, la compartición de información.
- El *objetivo* de la experiencia colaborativa se indica en términos de objetivos instruccionales y/o en términos del tipo de destrezas o habilidades que se pretenden conseguir junto con una descripción clara de qué se pretende con la actividad.
- El *resultado* esperado de la actividad.

5.6. Resumen

En este capítulo realizamos un estudio sobre el estado del arte de las herramientas **CASE** para la enseñanza y de las plataformas **CSCL** para la **IS**, específicamente para modelado

UML. Se investigó sobre el modelo **uCSCL** para aprendizaje ubicuo y en base a esto se planteó un modelo conceptual denominado **uCASE-CL** para describir el área de investigación relacionada al aprendizaje colaborativo de la **IS** en entornos virtuales ubicuos.

Capítulo 6

UAI Case: Implementación del modelo uCASE-CL

6.1. Introducción

UAI Case es la denominación de una herramienta que se plantea como implementación del modelo **uCASE-CL** [BAT17] [BAT17b]. Integra un entorno ubicuo y colaborativo de enseñanza asistido por computadora con una herramienta **CASE** para modelado **UML**.

Esta integración abre nuevas puertas a propuestas de enseñanza de la **IS** mediadas por tecnología en las que el aprendizaje colaborativo cobra un rol fundamental, permitiendo generar nuevas soluciones que brinden mayor potencial a las propuestas formativas [NEI16].

6.3. UAI Case: Nuestra Propuesta

UAI Case se plantea como propuesta a la necesidad de contar con herramientas que puedan ser utilizadas en la enseñanza de la **IS**, y que permitan asistir al alumno a adquirir y afianzar los contenidos académicos desarrollados. Además, brindar un espacio que facilite tanto a sus pares como al docente colaborar entre sí, sin que se aleje de la forma de trabajo de las herramientas del mercado y con las características necesarias para cumplir los objetivos de enseñanza del modelado **UML**.

En este sentido, comprendemos que el modelo **uCASE-CL** anteriormente planteado responde a esta necesidad. La propuesta está pensada dentro de un entorno académico mixto, en donde la herramienta no reemplace al modelo presencial, sino que brinde un espacio de coordinación y colaboración que permita optimizar las relaciones entre

docentes y alumnos, así como también la evaluación y seguimiento de trabajos desarrollados y obtener los resultados de forma ubicua.

6.3.1. Contexto

Este proyecto enmarca y potencia el trabajo de integración curricular desarrollado en la facultad de Tecnología Informática de la Universidad Abierta Interamericana (**UAI**). A partir del uso de la herramienta colaborativa (*UAI Case*) los alumnos podrán adquirir y utilizar los conceptos de modelado e ingeniería de software en un proyecto áulico colaborativo que evolucionará, iterativa e incrementalmente, desde 2° año, en las asignaturas que introducen los conceptos de modelado, hasta 3° año y 5° año donde desarrollan trabajos de integración de contenidos que incluyen las asignaturas de cierre del ciclo intermedio (Analistas de Sistemas) y del ciclo final (Ingeniero en Sistemas Informáticos) [NEI16].

En este marco, las materias involucradas son las siguientes:

- Metodologías de Desarrollo de Sistemas 1 y 2 (2° año).
- Bases de Datos, Trabajo de Diploma y Trabajo de Campo 1 (3° año)
- Seminario de Aplicación Profesional y Trabajo Final de Ingeniería (5° año)

6.3.2. Características Principales

UAI Case es un entorno de enseñanza y aprendizaje, en donde los alumnos podrán adquirir y utilizar los conceptos de modelado e ingeniería de software en distintos proyectos colaborativos donde desarrollan trabajos de Integración de contenidos.

Tiene dos áreas funcionales bien definidas.

- Por un lado, la plataforma de educación a distancia con diversas herramientas de colaboración entre alumnos y docentes.
- Por el otro *la herramienta CASE colaborativa*, integrada en el proceso de enseñanza y aprendizaje y con los mecanismos requeridos para que los docentes puedan realizar correcciones y evaluaciones de forma colaborativa y que permita obtener información de forma ubicua.

El relevamiento de las principales funcionalidades y características se llevó a cabo por medio de foros y reuniones entre todos los docentes participantes de las materias indicadas anteriormente. Estas reuniones se dieron en el marco de la reunión de claustro docente con el fin de formalizar la documentación funcional principal del proyecto.

A continuación, describimos las dimensiones que confirman la implementación de modelo **uCASE-CL**.

6.3.3. Dimensión Tecnológica

La dimensión tecnológica está dada por la especificación del modelo de despliegue de la solución planteada en *UAI Case* (Ver figura 6.1). En este modelo especificamos la tecnología requerida para lograr un ambiente colaborativo ubicuo que cumpla con los requisitos planteados en la especificación de **uCASE-CL**. Este diagrama está explicado con más detalle en el **Anexo III** donde se documenta el desarrollo del prototipo.

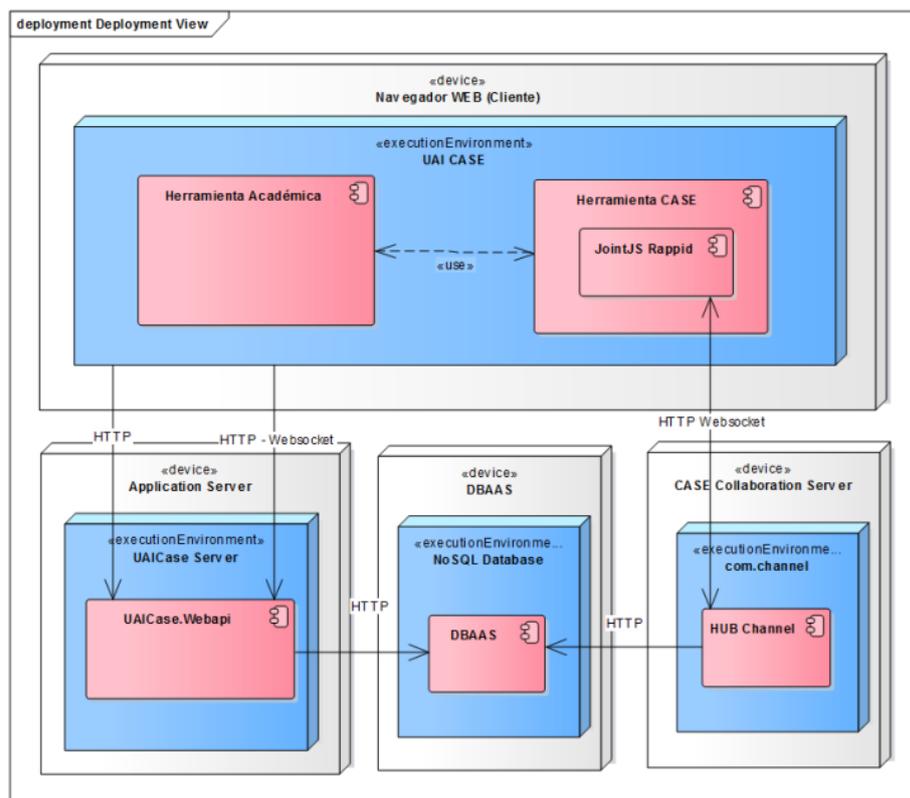


Figura 6.1. Modelo de despliegue de UAI Case

6.3.4. Dimensión Pedagógica

En esta dimensión planteamos las actividades formativas y las propias del modelo de enseñanza de la **IS**, en relación a los procesos colaborativos que las definen. Además de actividades colaborativas como un canal de *chat*, una lista de tareas o una mensajería interna, planteamos las siguientes actividades como actividades recurrentes e importantes, las cuales debemos especificar, como parte de la documentación del modelo **uCASE-CL**:

6.3.4.1. Modelado UML colaborativo libre.

Esta actividad se realiza en la herramienta de modelado, en un proyecto compartido entre un grupo de alumnos. El objetivo que persigue es realizar un modelo **UML** entre todos los participantes de forma libre, es decir, que todos pueden interactuar de forma no deliberada, sin ninguna coordinación. En este proceso, se destaca la libertad de cada usuario para participar en el modelado con la información de quien está participando y que componente se encuentra seleccionado por cada uno.

6.3.4.2. Modelado UML colaborativo coordinado.

Esta actividad corresponde a la realización de un modelo **UML** basado en la coordinación del grupo de trabajo.

6.3.4.3. Evaluación colaborativa de modelos UML.

Esta actividad corresponde a una técnica de evaluación basada en la colaboración entre docentes y alumnos utilizando como herramienta de comunicación colaborativa un foro. Este mecanismo se apoya en el feedback del docente la interpretación colaborativa basada en la discusión.

6.3.4.4. Autoevaluación colaborativa de grupos de trabajo utilizando rubricas.

Esta actividad se apoya en el uso de rúbricas para la autoevaluación de un equipo de trabajo utilizando la discusión y el consenso para la consecución del objetivo.

6.3.5. Dimensión Psicológica

Para poder desarrollar las actividades colaborativas involucradas en los procesos de enseñanza y aprendizaje mencionados, es necesario realizar una especificación formal que permita la identificación de las actividades recurrentes o destacadas.

Para esto utilizamos a continuación la técnica propuesta en el *Anexo II, Metodología para Diseñar Actividades Colaborativas*, con el objetivo de describir las actividades más importantes y recurrentes durante el proceso puntual de enseñanza de la **IS**.

Las actividades colaborativas que requieren su documentación son:

- Modelado **UML** colaborativo coordinado
- Evaluación colaborativa de modelos **UML**
- Autoevaluación colaborativa de grupos de trabajo utilizando rubricas

6.3.5.1. Modelado UML colaborativo coordinado

6.3.5.1.1. Descripción

Esta actividad permite trabajar sobre un ejercicio propuesto por el docente, de forma coordinada con los participantes de un grupo de trabajo. El modelado será llevado a cabo por medio de una solicitud que cada alumno podrá realizar. Esto será muy similar a pedir turno, el cual será otorgado por un lapso de tiempo determinado, mientras el resto del equipo observa lo que se está realizando. Este turno o *token* se asigna de forma secuencial en base al orden de solicitudes y en base al estado del solicitante (si el alumno no está conectado, el *token* se reasigna automáticamente). La solicitud dura un tiempo determinado por inactividad, esto quiere decir que, si al pasar un tiempo predefinido (podrían ser, por ejemplo, 30 segundos) no hubo ninguna interacción del participante, el *token* también se reasigna. Al final cada sesión de modelado, el alumno que la realiza escribe un Resumen de Cambios (**RC**) el cual será aceptado o rechazador por el resto de los alumnos participantes. Para que el *token* pueda ser reasignado, es necesario que todos los participantes estén de acuerdo. El **RC** está representado por un *foro*, donde cada participante podrá aceptar o rechazar el cambio, utilizando esta herramienta para exponer sus justificaciones en caso de un rechazo, e incurrir en la discusión para lograr el

acuerdo. En caso de que no exista acuerdo, algún alumno podrá pedir la mediación del docente para aceptar o rechazar el cambio propuesto y continuar con el proceso.

Al completar el proceso de modelado (cuando todos los participantes acepten el trabajo terminado) se obtendrá como resultado un modelo **UML** con todos los aportes realizados por cada alumno y estará en condiciones de ser enviado a corrección.

Al igual que la actividad libre, también dispone de un canal de *chat* constante, en donde podrá participar el docente en caso de estar conectado.

6.3.5.1.2. Evaluación

Concepto	Categoría	Definición
ALUMNO	ACTOR	Estudiante cursando una materia de la carrera de IS .
DOCENTE	ACTOR	Profesional que cumpla el rol de docente y sea el responsable de guiar el aprendizaje y posea los conocimientos técnicos necesarios de análisis de sistemas y modelado UML .
DIAGRAMA UML	OBJETO	Representa el modelo UML objeto de estudio. Este diagrama es el resultado de la colaboración entre alumnos y docentes al finalizar el ejercicio.
RESUMEN CAMBIOS (RC)	OBJETO	Es el resultado de una sesión de modelado realizada por un alumno. Esto sucede luego de que un alumno realice las actividades correspondientes durante su turno. Al finalizar, de forma literal intentará describir lo que hizo y por qué lo hizo, utilizando un foro para tal fin.
PEDIR TURNO	INTERACCIÓN	El alumno que no disponga un turno podrá solicitarlo e ingresará en una cola de turnos. El turno es la solicitud de modelado, que una vez se utilice podrá solicitarse otro.
MODELAR	INTERACCIÓN	El alumno que esté utilizando su turno realizará los cambios en el modelo UML que considere necesarios durante el tiempo asignado por el turno. Esto es denominado como sesión de modelado.
CANCELAR TURNO	INTERACCIÓN	El alumno al que le toca el turno podrá rechazarlo y continuar el proceso sin actividad.
FINALIZAR SESION DE MODELADO	INTERACCIÓN	El alumno que está modelando, termina la sesión con la creación del RC .
ACEPTAR CAMBIOS	INTERACCIÓN	Los alumnos que no participan de la sesión de modelado deberán estar de acuerdo con los cambios realizados por quien utilizó el turno. Esta acción corresponde a "Aceptar" los cambios propuestos e indicados en el RC .
RECHAZAR CAMBIOS	INTERACCIÓN	Los alumnos que no participan de la sesión de modelado deberán estar de acuerdo con los cambios realizados por quien utilizó el turno. Esta acción corresponde a "Rechazar" los cambios propuestos e indicados en el RC .
FUNDAMENTAR RECHAZO	INTERACCIÓN	Si un cambio es rechazado, deberá fundamentar el rechazo por medio de una entrada en el foro dispuesto en el RC correspondiente. Esto dará lugar a un debate para buscar la unanimidad en la decisión. También tendrán disponible un canal de <i>chat</i> para tal fin.

Concepto	Categoría	Definición
SOLICITUD REVISIÓN DE SESION	INTERACCIÓN	Cuando los alumnos no logran unanimidad en una decisión sobre algún cambio, podrán solicitar la participación del docente en el debate, el cual podrá aceptar o rechazar el cambio.
ACEPTAR DIAGRAMA	INTERACCIÓN	Al finalizar el proceso de modelado, todos los alumnos tendrán que estar de acuerdo con el resultado, para esto deberán aceptar el diagrama.
RECHAZAR DIAGRAMA	INTERACCIÓN	Al finalizar el proceso de modelado, los alumnos podrán rechazar el modelo terminado, fundamentando su decisión y dando lugar a una nueva discusión
SOLICITUD DE REVISION DE DIAGRAMA	INTERACCIÓN	En caso de no haber unanimidad en la aceptación de un diagrama, se solicitará revisión de diagrama al docente para que tome la decisión en base al modelo presentado.
SOLICITUD DE REVISION FINAL	INTERACCIÓN	En caso de que haya unanimidad, se envía el diagrama para su revisión final.
REVISAR DIAGRAMA	INTERACCIÓN	El docente dispondrá de un espacio para tomar la decisión de aceptar o rechazar el diagrama terminado si luego de la sesión completa de modelado los alumnos no logran unanimidad en el criterio de aceptación de la resolución planteada.
CORREGIR DIAGRAMA	INTERACCIÓN	Luego de que un diagrama es aceptado por todos los alumnos, se solicita corrección al docente. Esta corrección puede dar lugar a una discusión en caso de que esté incorrecta la resolución, dando lugar a nuevas sesiones de modelado.

6.3.5.1.3. Interacción y Transformación

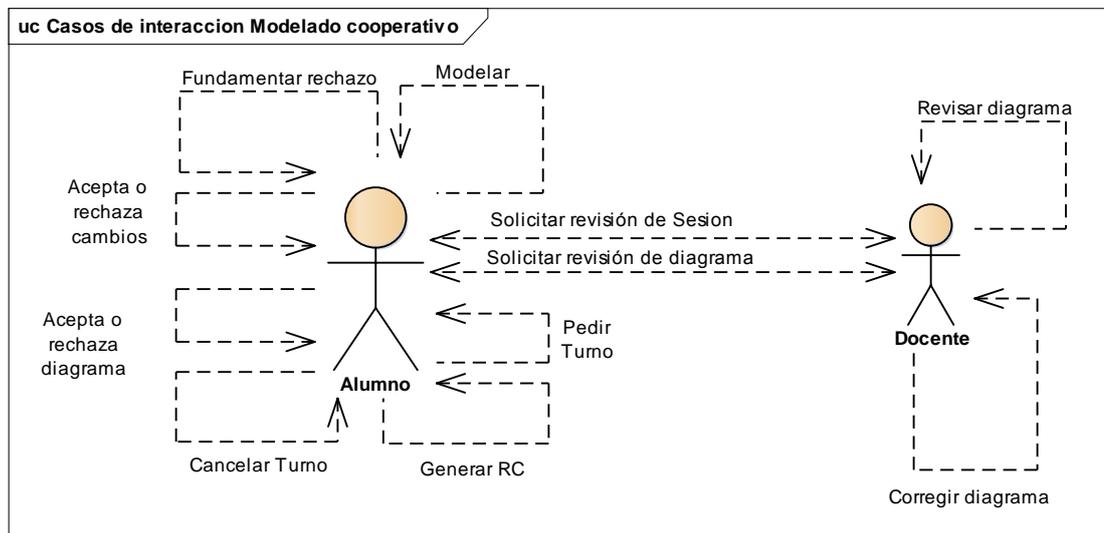


Figura 6.2. Casos de interacción para el proceso de modelado colaborativo

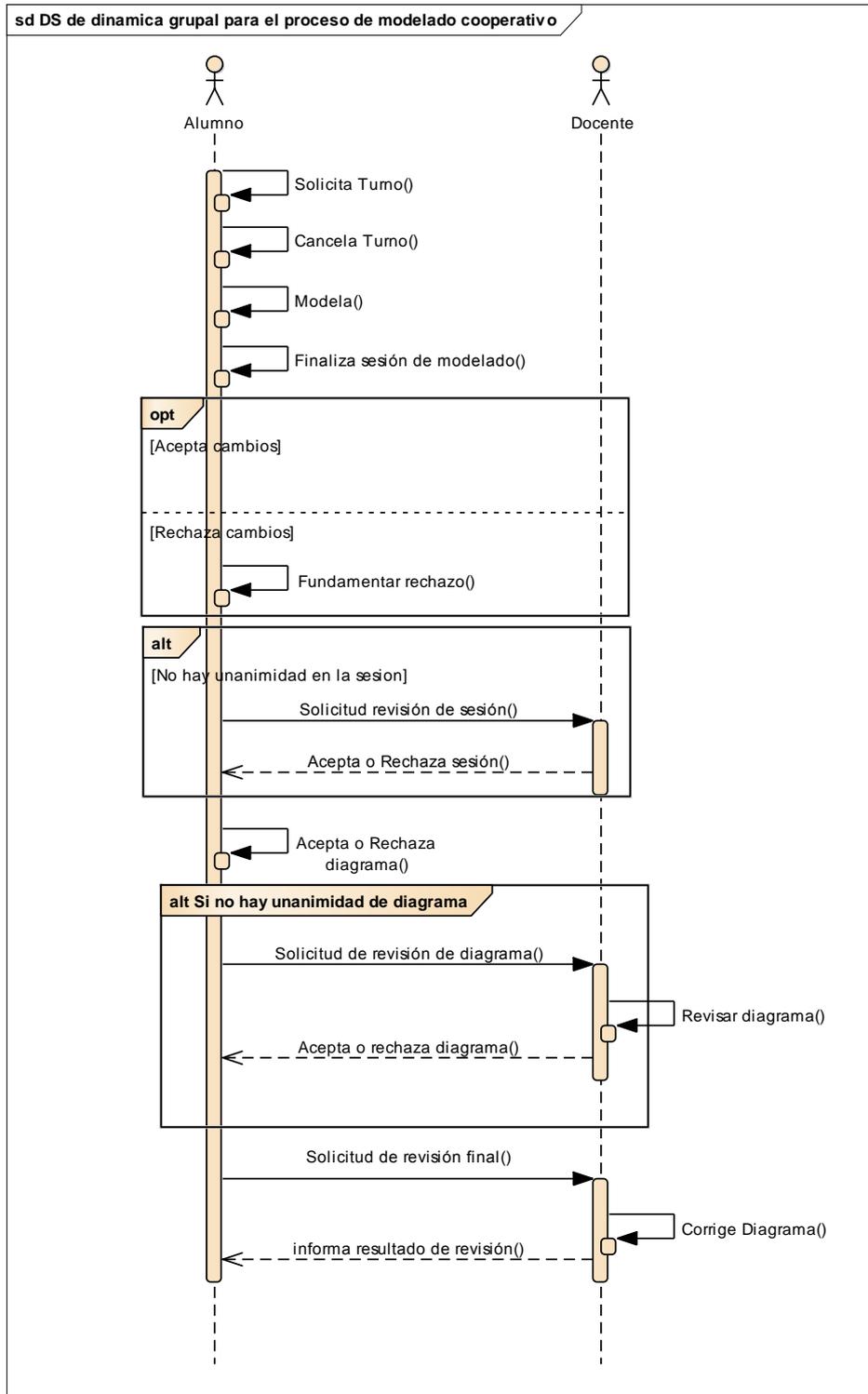


Figura 6.3. DS de dinámica grupal para el proceso de modelado cooperativo.

6.3.5.2. Evaluación colaborativa de modelos UML

6.3.5.2.1. Descripción

La evaluación colaborativa [BAT16b] se propone por medio de un espacio de colaboración asincrónico, utilizando como herramienta un foro, el cual denominaremos **PC** (Punto de Crítica). Durante el proceso de evaluación el docente puede considerar necesario abrir un **PC** utilizando un elemento propio de la herramienta, el cual pueda arrastrarse al diagrama en cuestión. Este elemento simboliza que, en ese lugar o componente del diagrama, el docente tiene algo para corregir, informar o criticar el cual podrá ser accedido por los alumnos que colaboren con ese modelo. En ese **PC** el docente va a abrir un hilo de un foro con el estado “en proceso” el cual va a poder ser comentado por cualquier participante durante el proceso de evaluación. Cuando algún alumno participante solicite revisión, el estado cambiará a “solicitud de revisión” y quedará a la espera de la misma para que luego el docente (en ese mismo momento, o en algún otro) pueda reevaluar ese punto y dar por terminado o reabrir el mismo según crea correspondiente para iniciar nuevamente el proceso. Todos los mensajes realizados por los alumnos dentro de un **PC** podrán ser aceptados o rechazados por el docente. Siempre que exista un comentario rechazado, deberá estar acompañado con otro comentario justificando el rechazo y las acciones a tomar.

6.3.5.2.2. Evaluación

Concepto	Categoría	Definición
DOCENTE	ACTOR	Profesional que cumpla el rol de docente y sea el responsable de guiar el aprendizaje y posea los conocimientos técnicos necesarios de análisis de sistemas y modelado UML .
ALUMNO	ACTOR	Estudiante cursando una materia de la carrera de IS .
PUNTO DE CRÍTICA	OBJETO	Componente de un modelo que refiere a una situación a analizar dentro de un modelo UML. Este componente se simbolizará con un ícono predefinido y contendrá un foro como herramienta de colaboración.

MODELO UML	OBJETO	Representa el modelo UML utilizado como herramienta de análisis funcional.
MODELA	INTERACCIÓN	El actor que representa un alumno dentro de un grupo de alumnos realiza una actividad de modelado utilizando la herramienta CASE provista.
SOLICITA REVISION	INTERACCION	El actor que representa un alumno dentro de un grupo de alumnos realiza una solicitud de revisión a quien sea el docente responsable de ese grupo, participante de la actividad de modelado.
SOLICITA CORRECCION	INTERACCIÓN	El docente utiliza el componente "Punto de crítica" para abrir un hilo de un foro que represente la solicitud de corrección.
REvisa	INTERACCION	El docente realiza una revisión del modelo sin realizar ninguna.
CORRIGE	INTERACCION	Un alumno realiza las correcciones correspondientes en base al punto de crítica iniciado por el docente. Luego podrá solicitar nuevamente revisión.
SOLICITA APROBACION	INTERACCION	El alumno envía correcciones realizadas al docente para que sean aprobadas.
APRUEBA CORRECCIONES	INTERACCION	El docente da por finalizada la evaluación realizada y la da por aprobada.
ENVIA APROBACIÓN	INTERACCION	El docente envía la aprobación del modelo.
REALIZA COMENTARIO	INTERACCIÓN	Cualquier involucrado en el proceso de evaluación podrá realizar comentarios sobre el foro abierto por el docente. Estos comentarios deberán estar relacionados con el tema a tratar.
ACEPTA COMENTARIO	INTERACCION	El docente acepta un comentario realizado por un alumno. La aceptación implica una respuesta.
RECHAZA DE COMENTARIO	INTERACCIÓN	El docente rechaza los comentarios realizados por los alumnos. El rechazo implica una respuesta a través de un comentario.

Tabla 6.1. Tabla CCD para el proceso de evaluación colaborativa.

6.3.5.2.3. Interacción y Transformación

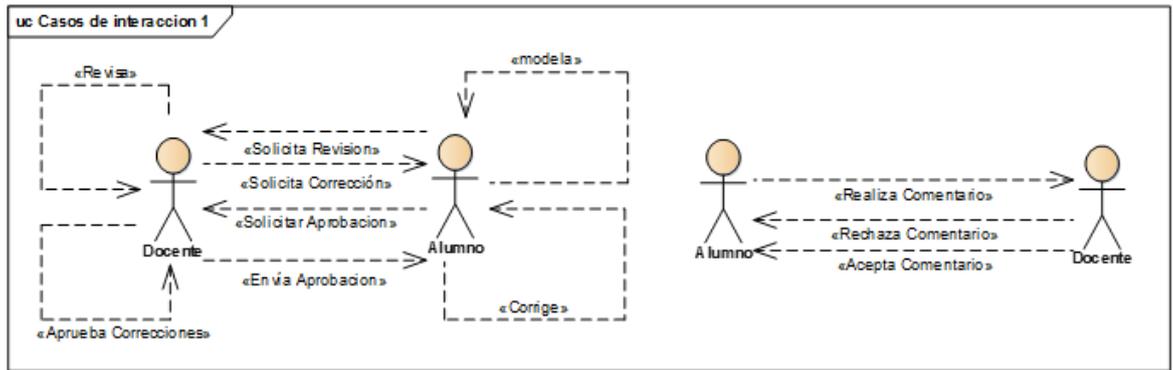


Figura 6.4. Casos de interacción para el proceso de evaluación colaborativa.

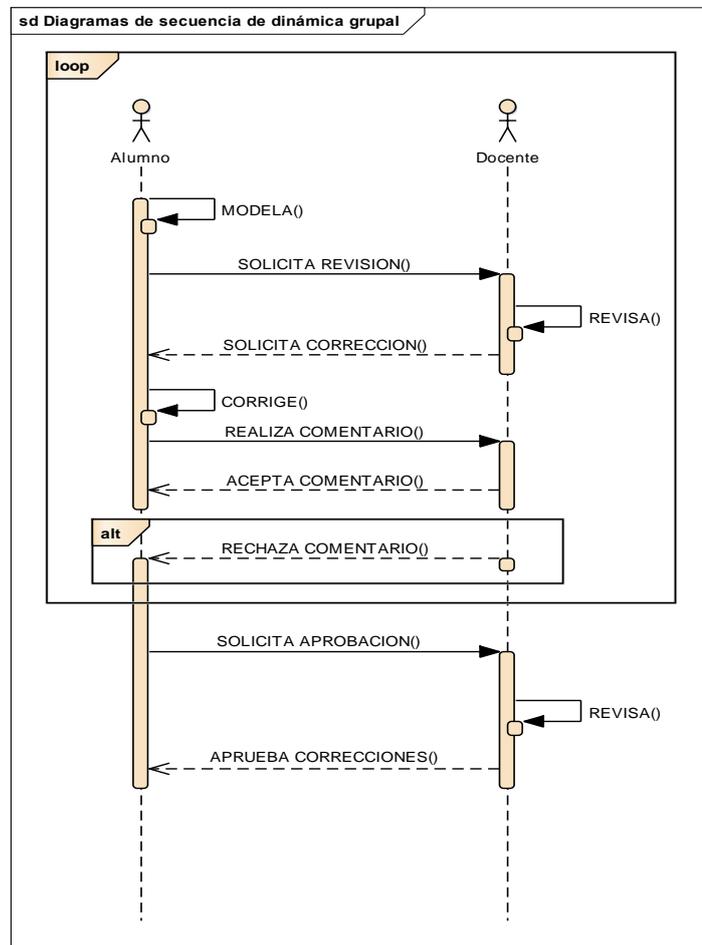


Figura 6.5. DS de dinámica grupal para el proceso de evaluación colaborativa.

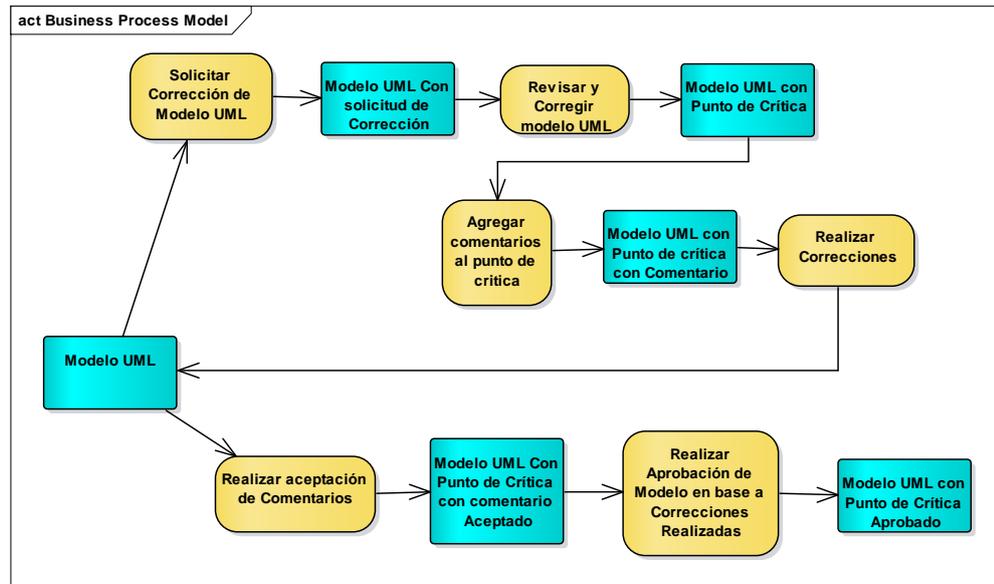


Figura 6.6. DOC para el proceso de evaluación colaborativa

6.3.5.3. Autoevaluación colaborativa de grupos de trabajo utilizando rubricas

6.3.5.3.1. Descripción

El docente, el alumno y el objeto de autoevaluación (**OA**) se integran en entornos de trabajo común denominados cursos. Cuando el docente propone un trabajo a realizar, también define los criterios y, cuando los alumnos den por terminado el trabajo (deberá estar confirmado por todos los integrantes del grupo) tendrán a su disposición la rúbrica para dar comienzo al proceso.

La autoevaluación colaborativa por medio de rubricas se propone en dos etapas y se utilizarán herramientas sincrónicas (*chats*) y asincrónicas (*foros*) para realizar tareas de comunicación y cooperación.

La primera etapa se centra en la construcción individual de una rúbrica propuesta previamente por el tutor del curso. De esta manera uno alumno de un grupo de trabajo interactuará con el entorno para proponer su evaluación sobre cada uno de los criterios de evaluación (**CE**) en los objetivos de aprendizaje que propone el docente por medio de cada rúbrica.

La segunda etapa es en donde se buscará la aceptación mayoritaria de la evaluación propuesta por cada alumno en cada uno de los criterios que conforman la rúbrica. De

esta manera y luego de que todos los alumnos completen la primera etapa, el sistema pondrá en conocimiento todos los criterios de la rúbrica en donde hubo un resultado equivalente para cada uno de los participantes y expondrá aquellos criterios en donde no hubo mayoría en la decisión. En este caso se creará el Punto de Evaluación (**PE**). El **PE** tiene como objetivo crear un espacio de discusión para que todos los participantes puedan justificar su autoevaluación y debatir con el grupo al respecto para lograr homogeneizar los criterios y conseguir que el resultado sea el obtenido por la mayoría. Este espacio de discusión es en base a un foro propio de la herramienta. Y tiene como característica la opción de aceptar o rechazar un comentario de un alumno junto con el espacio necesario para poder justificar la respuesta.

Durante la elaboración de la rúbrica, en esta segunda etapa, los alumnos tendrán también, un canal de chat disponible en todo momento para que puedan intercambiar ideas en tiempo real.

El objetivo de la segunda etapa del proceso de autoevaluación es encontrar mayoría en la valoración de cada uno de los **CE** de una rúbrica. De esta forma cada alumno realiza las valoraciones de forma individual y utiliza un sistema colaborativo de estimación para conciliar entre todos los participantes del proceso y obtener una aceptación mayoritaria para dar por cerrado ese criterio.

Una vez finalizado el **PE** (los alumnos deberán indicar todos que la actividad concluyó), es el docente quien provee su evaluación en base a la misma rúbrica (sin conocer el resultado de la evaluación de los alumnos) y al finalizar, propone otro **PE** por cada uno de los criterios donde la diferencia entre la evaluación del docente es muy distinta a la de la propuesta por los alumnos de forma colaborativa [BAT16c].

6.3.5.3.2. Evaluación

Concepto	Categoría	Definición
DOCENTE	ACTOR	Profesional que cumpla el rol de docente y sea el responsable de guiar el aprendizaje y definir CE
ALUMNO	ACTOR	Estudiante que realiza OA
RUBRICA	OBJETO	Elemento utilizado para realizar la autoevaluación.
OBJETO DE AUTOEVALUACIÓN (OA)	OBJETO	Trabajo propuesto por docente para realizar en equipo
PUNTO DE EVALUACIÓN (PE)	OBJETO	Foro creado para discutir cuando la autoevaluación no tuvo mayoría en cada criterio de la rúbrica.

Concepto	Categoría	Definición
FINALIZACION OE	INTERACCIÓN	Cuando se finalice el OE deberá ser indicado por todos los integrantes del equipo.
VALORACIÓN DE CE	INTERACCIÓN	Cada alumno realizará la valoración de todos PE propuestos en la rúbrica.
COMENTAR PE	INTERACCIÓN	Los alumnos discuten los criterios por medio de un foro cuando estos no son iguales en su mayoría.
ACEPTAR COMENTARIO	INTERACCIÓN	Un alumno acepta un comentario dentro del PE
RECHAZAR COMENTARIO	INTERACCIÓN	Un alumno rechaza un comentario dentro de un PE
FINALIZAR PE	INTERACCIÓN	Cuando las revaloraciones de los CE generan mayoría

Tabla 6.2. Tabla CCD para el proceso de autoevaluación colaborativa

6.3.5.3.3. Interacción y Transformación

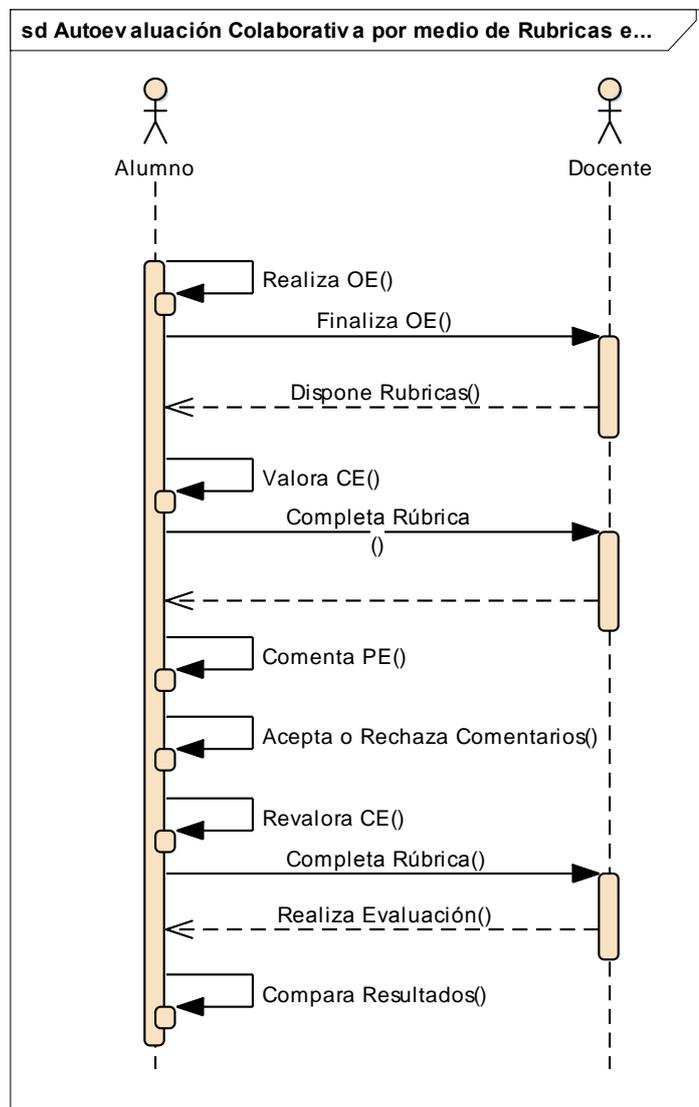


Figura 6.7. DS de dinámica grupal para el proceso de autoevaluación colaborativa.

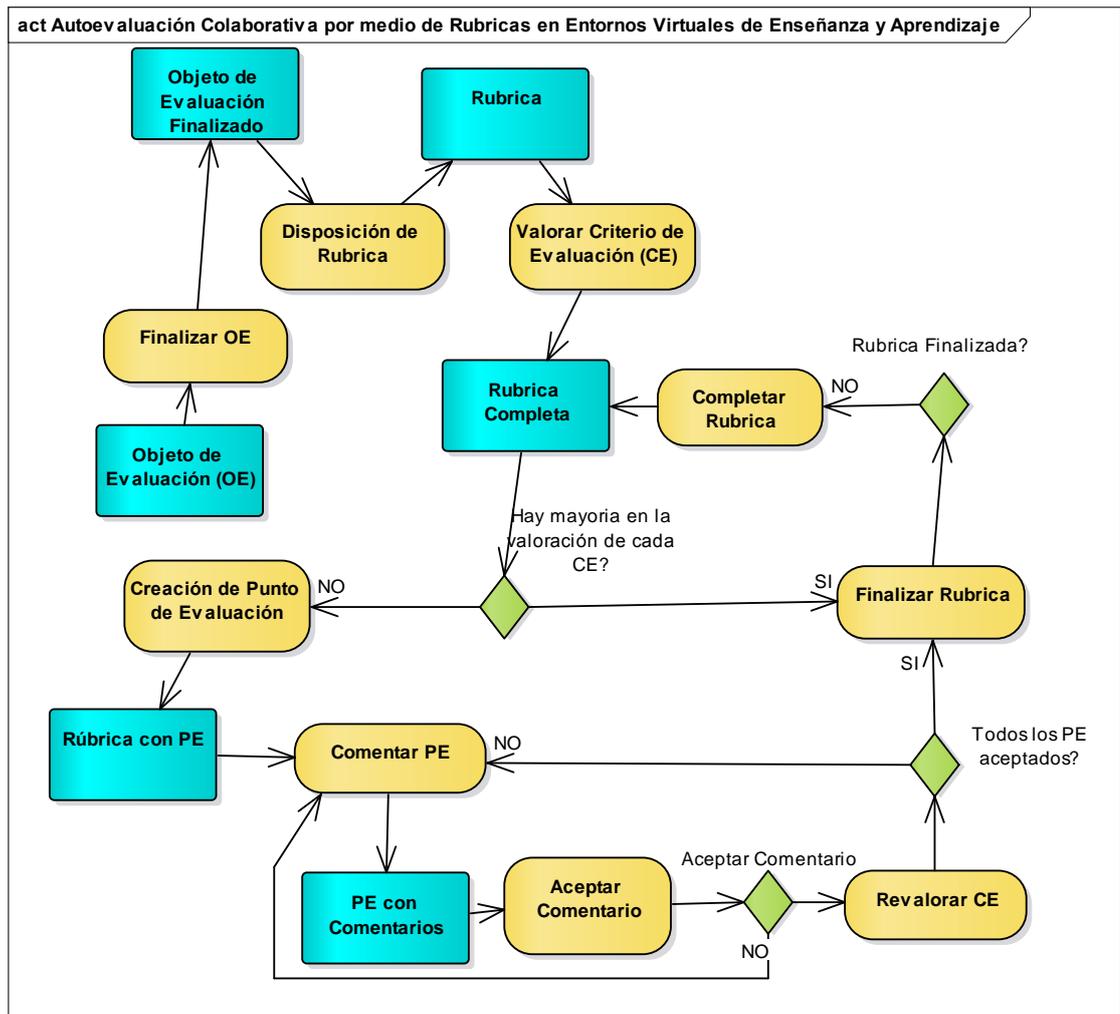


Figura 6.8. DOC para autoevaluación colaborativa

6.4. Resumen

En este capítulo presentamos *UAI Case* como la implementación del modelo **uCASE-CL** planteado en el **capítulo 5**. Contextualizamos el proyecto, definimos sus características principales y lo referenciamos a las dimensiones propuestas por nuestro modelo detallando las características de cada actividad que son impactadas por la tecnología, la pedagogía y la psicología.

Capítulo 7

Trabajos Relacionados

7.1. Introducción

Existe un número significativo de trabajos que estudian la evolución de las herramientas **CASE** como soporte para la ingeniería de software, en base a sus características desde el punto de vista técnico, funcional, y desde la mirada objetiva de su integración con las organizaciones que diseñan software. En general, estos estudios no contemplan temas relacionados a las herramientas **CASE** en la enseñanza de la IS en entornos virtuales colaborativos [ROJ00] [QUI12] [GEN06] [MEN01].

La problemática expuesta a continuación por Fowler, Allen, Armarego y Mackenzie [FOW00], refleja lo comentado anteriormente: "Mientras que los estudios muestran que las herramientas **CASE** tienen un impacto positivo en la calidad y la productividad, estos también muestran que han sido lentos para ser adoptados por la industria. Este fenómeno se explica en parte por el esfuerzo en aprender a usar la función".

Otro tema importante, es que la construcción de controles adicionales en las herramientas **CASE** sería un beneficio significativo [COO05]. Agregamos también, la necesidad de que esas herramientas cuenten con la información inherente a la validación que se está haciendo. De nada sirve para la enseñanza ver un error, pero no comprenderlo.

Presentamos a continuación, los principales trabajos de investigación vinculados a la enseñanza de IS asistida por computadora.

7.2. Resumen de Trabajos Relacionados

- En [SOL10] los autores proponen una herramienta web que tiene como objetivo principal brindar soporte para la enseñanza y aprendizaje del diagrama de clases

UML. Esta herramienta es un módulo que contiene un entorno de enseñanza distancia y como característica destacada, brinda la posibilidad de corregir automáticamente un ejercicio provisto. Para esto, el sistema propone un ejercicio que será resuelto por el alumno quien recibirá de forma instantánea el feedback con información detallada sobre los errores. Durante el proceso, el sistema acepta que el alumno ingrese más de una solución hasta que sea alcanzada la correcta o se llegue al plazo máximo indicado en el ejercicio. El sistema va a registrar todo el trabajo realizado por el alumno. Esta información será utilizada por el docente para propósitos de evaluación continua. La forma de corrección de la herramienta es en base a un proceso de comparación. En primer lugar, el docente propone un ejercicio y escribe diferentes soluciones posibles en un editor que dispone la herramienta. Esto es el principal punto a considerar, ya que es posible que se encuentren dos soluciones al mismo problema, y sean las dos válidas. Una vez que el alumno resuelve el problema, el módulo de evaluación se encarga de hacer las comparaciones y análisis pertinentes, para luego emitir el feedback correspondiente, de tal manera que sirva para guiar el proceso de aprendizaje. La herramienta no propone trabajo colaborativo.

- Otro trabajo [CHE06], proponen una herramienta para facilitar y mejorar el proceso de aprendizaje del modelado **UML** de forma colaborativa. Para este proyecto, los autores se apoyaron en un entorno **CSCL** y su integración con agentes pedagógicos representados por tutores que tienen como responsabilidad guiar a los alumnos desde el punto de vista técnico del proceso y desde el punto de vista de la colaboración. El trabajo presentado combina diferentes tecnologías y métodos para diseñar un sistema *groupware* con la información *awareness* correspondiente para coordinar la colaboración. Este entorno **CSCL** provee mecanismos de información adicional, que informan sobre la actividad que está llevando adelante el alumno. Esta propuesta no incluye definiciones sobre métodos de evaluación ni semántica **UML**.
- En [BOU05] los autores trabajaron en el desarrollo un entorno de aprendizaje uniforme basado en la plataforma ECLIPSE. La solución permite a los estudiantes distribuidos físicamente comunicarse, coordinarse y colaborar en equipos para producir grandes sistemas y tener un espacio de práctica necesario para el

aprendizaje de la Ingeniería de Software. Para lograr este objetivo, utilizaron *plugins* existentes y crearon el código necesario para que puedan coexistir. De esta forma los estudiantes acceden a un conjunto de herramientas que le permita disponer de un marco de trabajo con herramientas de comunicación sincrónica y asincrónica como estructura colaborativa, de comunicación y cooperación. Por otro lado, dispondrá de las herramientas necesarias para que el docente pueda supervisar, evaluar, identificar problemas y guiar a los estudiantes a través del proceso.

- Una de las herramientas colaborativas educativas, más importantes en el mercado actual es *Moodle (Modular Object Oriented Dynamic Learning Environment, Entorno virtual de formación de código abierto)* [MOO17]. Esta se basa en un modelo pedagógico de constructivismo social, que trabaja con la colaboración de todos los participantes siendo estos, estudiantes y cuerpo docente, de forma que éstos permitan contribuir a la experiencia educativa. Se permite a los participantes generar conocimiento y reflexionar de forma crítica sobre una determinada temática, como así construir conocimiento de manera colaborativa y realizar un seguimiento y evaluación del alumnado. Es importante destacar que, analizando la capacidad de gestión docente del entorno *Moodle*, las rúbricas han tenido una muy buena valoración y son consideradas un valioso aporte para facilitar la tutoría, en especial cuando un equipo de profesores trabaja en una misma actividad formativa. También se ha destacado el beneficio que pueden suponer para el proceso de aprendizaje del alumnado [MAR15]. Esta herramienta permite plantear actividades de varios enfoques didácticos, tales como en forma individual y grupal. Finalmente, esta plataforma permite utilizarse como una herramienta de evaluación no sólo a nivel alumnado, sino también a nivel docente en forma exitosa [SAL15].
- Los trabajos [MAR07] y [DEL11a], proponen la necesidad de contar con un conjunto de herramientas automatizadas que permitan al alumno adquirir y afianzar los conocimientos en el área de las bases de datos, abarcando todos los pasos para la creación de una base de datos eficiente. Su investigación se centra en el desarrollo y aplicación de nuevas tecnologías en la enseñanza de la teoría de bases de datos. Desarrollaron una herramienta **CASE** denominada *Powermodeller*

que contempla las primeras etapas de modelado de datos utilizando modelos de entidad relación extendido, utilizando los objetos del modelo entidad-relación extendido de Teorey, Yang y Fry. Esta herramienta permite que el alumno pueda comprobar los resultados obtenidos luego de efectuar las prácticas de las asignaturas, ejecutando los algoritmos de análisis que han sido automatizados.

- Otro trabajo relacionado es [CON00] en el cual los autores presentan una herramienta **WEB** que tiene como objetivos, mejorar el rendimiento de los alumnos en el diseño de bases de datos utilizando el modelo entidad relación y ayudar al alumno a desarrollar pensamiento crítico y capacidades para trabajar en equipo y de forma colaborativa. La herramienta denominada *COLER* propone un entorno donde se presenta un problema a resolver. Está diseñada para que los estudiantes puedan resolver problemas de forma individual y luego unirse a un pequeño grupo. Para esto, el alumno dispone de un espacio de trabajo privado y uno compartido. El privado lo utiliza para construir la solución de forma individual y el espacio compartido es donde podrá hacer la construcción de forma grupal. Para esto se propone un canal de chat y un botón de ayuda con información contextual sobre la notación. Como actividad colaborativa, dispone de una herramienta de coordinación donde figuran todos los alumnos conectados y una opción para solicitar el control (Tomar o Liberar) y un espacio para comentar sobre el problema en cuestión. Cada alumno tendrá la opción de poner si está de acuerdo de forma total, parcial o no está de acuerdo con lo que se está modelando. La herramienta posee un espacio para un tutor automatizado que podrá monitorear tanto el espacio privado como el espacio compartido del grupo de trabajo. De este monitoreo mostrará información relacionada con la colaboración.
- En el trabajo [HER16], los autores presentan el desarrollo de una extensión para la herramienta *Visual Paradigm* con el objetivo de evaluar y corregir diagramas de casos de uso. La evaluación debe determinar si los elementos del diagrama están bien representados y relacionados. El proceso de evaluación propuesto detecta errores cometidos en el modelado del diagrama y genera como resultado, una lista de errores cometidos. Esta lista será la entrada al proceso de corrección. LA

corrección se hace para cada uno de los errores listados, en donde el sistema arroja las posibles opciones y sugerencias.

- En [TEN08] presentan una herramienta denominada *SketchUML*. Fue desarrollada para utilizar en una *Tablet-PC* para guiar a los estudiantes durante el proceso de aprendizaje del modelado **UML** y para ayudar a los docentes durante el proceso de crítica y evaluación. La herramienta se diseñó completamente centrada en el alumno, con el objetivo de facilitar el aprendizaje del modelado **UML** asistido por computadora. Para esto proponen utilizar una interfaz que permita convertir los gestos de dibujo natural en componentes UML y de esa forma el alumno pueda obtener feedback necesario para comprender la sintaxis.
- *Educause Learning Initiative* [ELI17] es una comunidad de instituciones y organizaciones de educación superior comprometidas con el avance y la promoción del aprendizaje a través de la aplicación innovadora de la tecnología. Esta comunidad lidera una iniciativa en relación con el desarrollo de una nueva generación de ambientes digitales de aprendizaje, en inglés: “*Next generation digital learning environment (NGDLE)*” definidos, fundamentalmente, por las siguientes funcionalidades: interoperabilidad, personalización, analíticas, asesoramiento, evaluación del aprendizaje, colaboración, accesibilidad y diseño universal. En un artículo de la revista *Campus Technology* [CAM17], David Rathes analiza la enorme diversidad de necesidades que presenta la educación superior. Hay una población estudiantil cada vez más diversa en los colegios y universidades, así como una amplia variedad de modalidades y modelos de cursos. “Traten de imaginar una simple aplicación como un **LMS** para dar respuesta a tan amplio espectro de necesidades...”. ELI describe entonces una arquitectura basada en componentes en la que ciertas aplicaciones individuales (incluido un **LMS**) pueden estar “nadando” alrededor de este espacio conectado, a través de estándares abiertos, capaces de intercambiar datos e información. (...) Múltiples aplicaciones en una arquitectura distribuida. Este abordaje permite pensar en la posibilidad de customización y personalización de los entornos o ambientes de aprendizaje, que las instituciones han necesitado por años. En este sentido, y ante la posibilidad de este tipo de adaptación/personalización, los responsables de las instituciones consideran de enorme valor, por ejemplo, poder

adicionar a estos entornos aplicaciones específicas para diferentes disciplinas, ajustando el ambiente de aprendizaje e incluyendo herramientas que posibiliten el trabajo de los estudiantes y de los tutores. Este refinamiento requiere el uso de herramientas específicas de dominio, normalmente distribuidas. En relación con esto existe una iniciativa del *IMS Global Learning Consortium* [IMS17a] denominada *Learning Tools Interoperability* (LTI) [IMS17b]. El concepto principal de LTI es establecer una forma estándar de integrar aplicaciones de aprendizaje (a menudo alojadas de forma remota y proporcionada a través de servicios de terceros) con plataformas **LMS**, portales, repositorios de objetos de aprendizaje y otros entornos educativos. De acuerdo con **LTI** estas aplicaciones de aprendizaje se llaman “Herramientas” (suministradas por Proveedores de Herramientas) y el **LMS** o plataformas se denominan “Consumidores de Herramientas”. El caso de uso básico, detrás del desarrollo de la especificación **LTI**, es posibilitar la perfecta conexión de aplicaciones y contenidos basados en web, alojados externamente, o de herramientas (desde simples aplicaciones de comunicación como chat, hasta entornos de aprendizaje específicos de dominio para temas complejos como Matemáticas o Ciencia) a las plataformas que los presentan a los usuarios. En otras palabras, una aplicación de evaluación interactiva o un laboratorio de química virtual podrían conectarse de forma segura a una plataforma educativa, de manera estándar, sin tener que desarrollar y mantener integraciones personalizadas para cada plataforma.

7.3. Comparación de nuestra Propuesta vs Trabajos Relacionados

Los trabajos incluidos y estudiados en la sección anterior, tienen como temática principal la enseñanza en entornos colaborativos centrada en áreas de conocimiento específica que requieren de herramientas propias de la temática, en particular, la ingeniería de software. A continuación, agrupamos los trabajos en dos grandes grupos que permiten comparar de forma más clara y específica con nuestra propuesta

7.3.1. Uso de Herramientas CASE en la Educación

A diferencia de los temas estudiados en cada uno de los trabajos que hacen referencia al uso de herramientas **CASE** en la educación, nuestra propuesta se basa en la integración de un entorno colaborativo específico para la enseñanza de la **IS** durante el modelado de software, dispone de una herramienta **CASE** con las características acordes al uso relacionado al aprendizaje del modelado **UML**, centrando el foco en el aprendizaje ubicuo y colaborativo.

La concepción inicial de nuestra propuesta es contar con una herramienta que acerque al alumno a las prácticas profesionales, pero sin sobrecargarlo con funcionalidades e interfaces de usuario complejas y difíciles de utilizar. Además de esto, propone brindar información contextual inherente al proceso de aprendizaje, de integrar al docente como parte del equipo de trabajo y provee actividades de evaluación y autoevaluación y, además, una forma para realizar el seguimiento de trabajos académicos. Esto se distingue del resto de los trabajos, ya que ninguno propone una herramienta **CASE** para la enseñanza y aprendizaje, que además incluya funcionalidades para la evaluación y seguimiento de trabajos.

7.3.2. Enseñanza y Aprendizaje de la IS en entornos Virtuales Colaborativos

Muchas herramientas **CASE** proponen espacios para que dos o más colaboradores puedan realizar el trabajo de forma simultánea. En base al análisis previo, en trabajos como [CON00] [MAR07] [DEL11a] [BOU05] (entre otros), identificamos que, ni las herramientas contemplan ni los colaboradores consideran, al docente como parte del equipo de trabajo. En consecuencia, no considerar al docente como parte del equipo significa no contar con capacidades relacionadas a su rol, como ser tutorización, seguimiento y evaluación.

Nuestra propuesta prescribe un modelo que integra junto con la plataforma **CSCL** una herramienta **CASE** colaborativa, con las herramientas y actividades definidas para ser utilizada durante la enseñanza y aprendizaje de la **IS**. Este modelo, posee un bloque que es funcional al proceso de evaluación.

7.4. Publicaciones Vinculadas a la Tesis

A continuación, se listan los trabajos publicados cuya temática está vinculada al de los objetivos planteados en esta tesis:

- Neil C., De Vincenzi M., Battaglia N., Martínez R. (2016). Herramientas Colaborativas Multiplataforma en la Enseñanza de la Ingeniería de Software. In XVIII Workshop de Investigadores en Ciencias de la Computación
- Battaglia, N., Neil, C., De Vincenzi, M., & Martínez, R. (2016). UAI Case: integración de un entorno académico con una herramienta CASE en una plataforma virtual colaborativa. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).
- Battaglia, N., Neil C., Cardacci, D., De Vincenzi M., Martínez R. (2016). Evaluación y Seguimiento Durante el Proceso de Enseñanza y Aprendizaje del Modelado UML en Entornos Colaborativos. In V Workshop de Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).
- Battaglia, N. Martínez, R. Otero, M., Neil, C., De Vincenzi M., (2016). Autoevaluación Colaborativa por medio de Rubricas en Entornos Virtuales de Enseñanza y Aprendizaje. In I Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), II Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC)
- Battaglia, N., Neil, C., De Vincenzi, M., Martínez, R., González Dana (2017). uCASE-CL: Aprendizaje Colaborativo de la Ingeniería de Software en Entornos Virtuales Ubicuos. In XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2017).
- Battaglia, N. Martínez, M., Neil, C., De Vincenzi M., (2017). UAI Case: Implementación del modelo uCASE-CL. In II Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), III Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC).
- Battaglia, N., Martínez R., Neil C., De Vincenzi M. (2017). Una Propuesta de Evaluación de Herramientas CASE para la Enseñanza. In VI Workshop de

Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).

- Battaglia, N., Neil C., Martínez R., González, D., De Vincenzi M. (2017). Learning of Software Engineering on Collaborative Virtual Environments. In 7th World Engineering Education Forum (WEEF).
- Battaglia, N., Martínez R. (2017). Propuesta de Metodología de Evaluación para el Análisis de Modelado en Herramientas CASE. CoNaII SI.

7.5. Resumen

En este capítulo analizamos trabajos de investigación vinculados al uso de herramientas **CASE** en la educación de la **IS** en entornos virtuales. En base a esto realizamos una comparación entre nuestra propuesta y los trabajos relacionados. Por último, presentamos los diferentes trabajos que hemos desarrollado vinculados a la temática de esta tesis.

Capítulo 8

Conclusiones y Trabajos Futuros

8.1. Introducción

En este último capítulo, presentaremos primero un resumen de la tesis, luego, las contribuciones principales y, por último, las líneas futuras de investigación. En la sección resumen e interpretación final, sintetizaremos la propuesta de la tesis respecto del modelo **uCASE-CL** y mostraremos las diferencias entre nuestro planteo respecto de los trabajos vinculados; en la siguiente sección, contribuciones principales, detallaremos los aportes más significativos de nuestra propuesta; finalmente, en la sección trabajos futuros, presentaremos distintas líneas de investigación que permitirán continuar con el trabajo desarrollado en la tesis.

8.2. Resumen e Interpretación Final

Hoy en día es común encontrar plataformas de educación en línea, muchas de ellas con características colaborativas y métodos de comunicación sincrónicos y asincrónicos, como por ejemplo canales de *chat* o foros. Estas plataformas también poseen espacios dedicados a la evaluación, en donde el docente tiene la posibilidad de dar una devolución a un trabajo determinado, incluso obtener, por medio de dispositivos móviles, los resultados de la evaluación de forma ubicua.

No sucede lo mismo en cursos de ingeniería de software y en el uso de herramientas específicas a esta temática. El fuerte avance en materia de herramientas **CASE** generó un impacto en el mercado de estas aplicaciones, pero en mayor escala esto se dio en el

ámbito profesional. Actualmente, existen muchas herramientas que brindan asistencia en todas las fases del proceso de desarrollo de software. Permiten incluso trabajo colaborativo, muchas de ellas con gran número de funciones. Normalmente, estas herramientas están destinadas al uso profesional pero no suelen tener espacios destinados a la enseñanza y aprendizaje, ni aspectos pedagógicos centrados en la educación.

En el presente trabajo se abordó el uso de las nuevas **TIC** aplicadas a la educación a distancia con el objetivo de desarrollar un entorno académico, en donde el docente y los estudiantes cuenten con un espacio de trabajo común que coexista con el espacio tradicional del curso presencial y que puedan desarrollar prácticas relacionadas a la **IS** de forma colaborativa. La propuesta cuenta con herramientas de coordinación, cooperación y con espacios disponibles para optimizar la comunicación entre alumnos en un grupo de trabajo o bien brindar un espacio para que el docente pueda, incluso también en tiempo real, realizar las correcciones correspondientes; para que de este modo el alumno pueda disponer de esa información y realizar el seguimiento correspondiente, en cualquier momento y lugar, por medio de una aplicación que sea independiente del dispositivo con la que se desee acceder, es decir, de forma ubicua.

El resultado de nuestra investigación se resume en un modelo multidisciplinario enmarcado en una herramienta **CSCL** en donde la pedagogía, la psicología, la informática y el área de conocimiento al que se le desea dar soporte durante el proceso de enseñanza y aprendizaje, deben adaptarse a los cambios propuestos por la sociedad del conocimiento y la información. A partir de lo anteriormente descrito, se propuso la integración de éstas áreas por medio del modelo **uCASE-CL** con el objetivo de asistir al proceso planteado mediante la colaboración, en donde un grupo de alumnos, también multidisciplinario y con diferentes *skills*, pueda modelar un software de forma coordinada, con la participación activa de un tutor o docente (en caso de aplicar a un curso mixto de enseñanza) y con características que permitan fomentar la discusión y el aprendizaje colaborativo.

Este modelo (**uCASE-CL**) se planteó inicialmente para integrar entornos de aprendizaje colaborativo con herramientas **CASE**. A pesar de esto, identificamos que podría ser fácilmente extensible a otras áreas de conocimiento e, incluso, adoptar algún modelo estándar para que esta integración sea más sencilla de implementar. Por otro lado,

tampoco consideramos el concepto innovador que plantea la ludificación en la enseñanza, el cual debería integrarse de forma explícita mediante actividades y objetos de enseñanza.

8.3. Contribuciones Principales

La principal contribución de esta tesis es la especificación de un modelo conceptual para representar la integración de un entorno de aprendizaje colaborativo con una herramienta **CASE** [BAT17b] [BAT17]. En segundo lugar, definimos y especificamos dos procesos colaborativos, una técnica para evaluar herramientas de modelado basada en las funcionalidades esperadas [BAT17c] [BAT17e] y un proceso de evaluación colaborativa para aplicar durante la enseñanza de las técnicas de modelado durante el proceso de desarrollo de Software [BAT16b] [BAT16c]. Para finalizar, utilizamos el modelo planteado junto con los procesos especificados para desarrollar un entorno colaborativo que permita a los alumnos y docentes utilizar una herramienta basada en el modelo anteriormente indicado; se propone utilizar este modelo durante el proceso de enseñanza y aprendizaje de la ingeniería de software [NE16] [BAT16] [BAT17d].

8.4. Trabajos Futuros

- **Diseño de una técnica de evaluación automática de modelos UML** por medio del diseño de algoritmos que permitan obtener resultados por aproximación e identificación de imágenes. De esta forma, se podrá brindar nuevas soluciones de evaluación en tiempo real.
- **Implementación y validación los procesos colaborativos** planteados en el modelo **uCASE-CL** por medio de casos de estudio reales. Esto permitirá medir los resultados de las actividades colaborativas con el objetivo de tener mejores resultados durante la aplicación de las mismas.
- **Corroboración empírica que permita validar las conclusiones planteadas en esta tesis.** Para ello propondremos un análisis cuantitativo y cualitativo comparando UAI CASE con un método tradicional de enseñanza, utilizando como muestra un conjunto de alumnos homogéneos de dos cursos de la misma materia. De esta

forma podremos obtener resultados que nos permitirían validar las ventajas del método propuesto respecto de los métodos de enseñanza tradicionales.

- **Adaptación del modelo uCASE-CL al modelo de enseñanza y aprendizaje basado en competencias.** Actualmente existe una corriente que hace hincapié en el aprendizaje basado en competencias, de esta manera, podremos alinear nuestro trabajo con las tendencias actuales.
- **Integración de herramientas cognitivas en la educación** como, por ejemplo, **IBM Watson**. Estas herramientas permitirían asistir al alumno en tiempo real durante el proceso de aprendizaje. Se espera poder introducir al proyecto la capacidad cognitiva de estas herramientas no solo para brindar un canal adicional de comunicación e información para quien utilice la herramienta (alumnos o tutores), sino que, además, se puedan obtener nuevos medios de evaluación en tiempo real.
- **Ludificación durante el proceso de enseñanza y aprendizaje de la IS.** La gamificación en la educación es un enfoque para motivar el aprendizaje de los estudiantes utilizando elementos del diseño de juegos en un ambiente educativo. Hemos propuesto un modelo para integrar una herramienta **CASE** en un entorno **uCSCL**, pero no consideramos aspectos relacionados al uso de juegos.
- **Adoptación la iniciativa propuesta por el IMS Global Learning Consortium** denominada *Learning Tools Interoperability* (LTI) para la estandarización en la interoperabilidad de herramientas específicas de aprendizaje, y así poder extender el modelo planteado en esta tesis a otros ámbitos educativos.

Anexo I

Revisión, Pruebas y Puntuación de Diferentes Herramientas CASE

I.1. Introducción

En el Capítulo 4 planteamos como objetivo evaluar los aspectos más importantes y característicos se realiza un análisis comparativo entre herramientas **CASE** para modelado de software

En base a esto, seleccionamos un conjunto de herramientas sobre las cuales realizaremos el relevamiento y posterior clasificación.

I.2. Análisis de Herramientas

A Continuación, se analizan las siguientes herramientas CASE.

I.2.1. Enterprise Architect



Figura I.1. Logo Enterprise Architect

Enterprise Architect [SPA17] se presenta como una herramienta de diseño **UML** con soporte para todo el equipo de trabajo y las diferentes fases del proceso de desarrollo de

software (**CASE** de extremo a extremo). Algunas de las características que destacan en el sitio web son las siguientes

- Ayuda Comprensiva para UML con soporte completo para todos los diagramas.
- Interfaz de Usuario intuitiva
- Soporte para transformación MDA (*Model Driven Architecture*).
- Documentación flexible y comprensible.
- Generación de código fuente e ingeniería inversa.
- Interconexión entre sistemas como .NET y Eclipse
- Modelado de Bases de Datos
- Portabilidad e importación e Modelos (XML, XMI, Repositorios basados en SQL)
- Soporte de línea base administración de requisitos
- Soporte para el mantenimiento y pruebas
- Soporte para Administración de Proyecto
- Soporte de Trazabilidad de extremo a extremo

En base a todas las características que propone la herramienta, comprendemos que no es solo una herramienta de modelado. Incluye soporte para todas las fases del proceso de desarrollo. La herramienta tiene 3 ediciones con diferentes limitaciones. La edición *Profesional*, la edición *Corporativa* y la edición *Corporativa Flotante*.

Con respecto a la interfaz de usuario, podemos afirmar que es una desventaja notable ya que posee muchas funcionalidades, lo que hace que la herramienta sea muy compleja y no se termine de utilizar siempre. Es una herramienta compleja de aprender, pero muy potente.

1. Principal	
Nombre	Enterprise Architect
Empresa	SPARX

Descripción	Herramienta de modelado con soporte para todo el proceso de desarrollo.	
Versión	12 Corporate	
Plataforma	Windows	
Precio	US\$	239 (1 a 4 licencias)
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online	X	Permite utilizar repositorios sobre bases de datos SQL. Esto permite que los usuarios puedan trabajar en simultáneo, generando una sensación de entornos distribuidos.
Trabajo Colaborativo		
Enseñanza y Aprendizaje		
Compatibilidad XMI	X	
Exportar Imágenes	X	
Producir Documentación	X	
TOTAL PUNTUACIÓN 2		7
3. Ingeniería de Software		
Diagrama CU	X	
Especificación CU	X	
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia	X	
Aspectos Notacionales UML	X	
Validación de Modelo UML	X	
Interpretación OCL	X	
Validación Inv. OCL	X	
Generación Código .NET	X	
Ingeniería Inversa	X	
Identificación de Roles		
TOTAL PUNTUACIÓN 3		12
PUNTIACIÓN TOTAL		19

Tabla I.1. Evaluación de Enterprise Architect

El porcentaje de aceptación de la herramienta es $(19 \cdot 100) / 22$. Lo que da un total de **86.40%**.

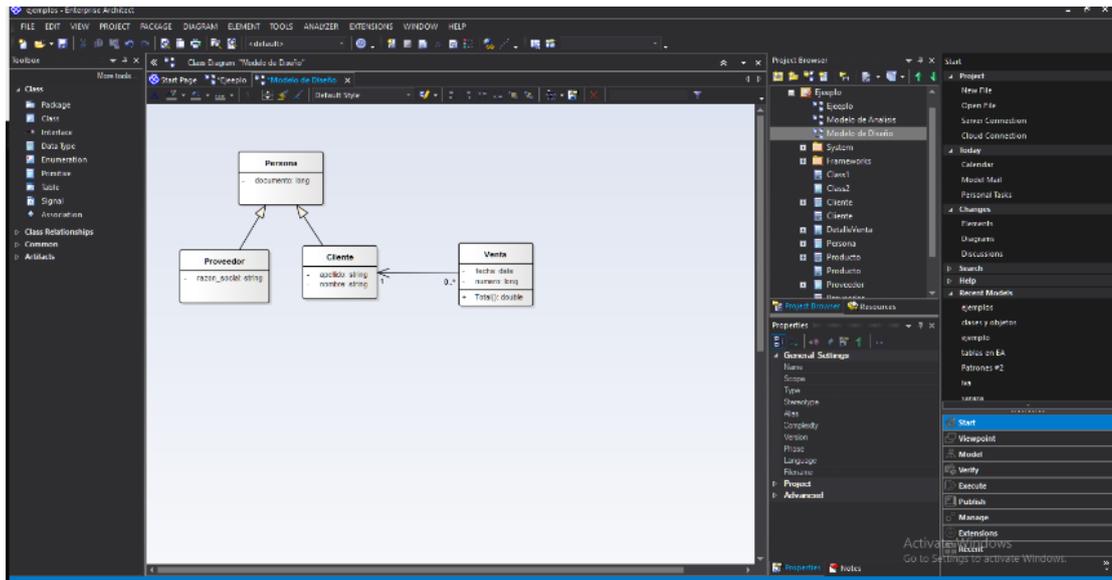


Figura I.2. Interfaz de usuario de Enterprise Architect

I.2.2. Gliffy



Figura I.3. Logo Gliffy

Gliffy [GLI17] se presenta como una herramienta de modelado para trabajo en equipo. Las características destacadas más importantes son las siguientes

- Soporte **UML**
- Fácil de usar y rápido
- Control de cambios y manejo de revisiones
- Colaborativo
- Publicar y compartir
- API para reúso

Al ser una aplicación web la forma de licenciamiento es relativo a la cantidad de usuarios y el plan contratado. Los planes posibles son *Gliffy Standard*, *Gliffy Business* y *Gliffy Business Team*. En cada caso, el precio varía en función de la cantidad de usuarios. Para hacer las pruebas usamos un usuario free que dura quince días.

La evaluación arrojó que la interfaz de usuario denota una herramienta muy sencilla de aprender, fácil de usar y con la ventaja de ser amigable al usuario.

1. Principal		
Nombre	gliffy	
Empresa	Gliffy	
Descripción	Herramienta de modelado	
Versión	Gliffy BUSINESS	
Plataforma	WEB	
Precio	US\$	8 / mes (Usuario único)
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online	X	
Trabajo Colaborativo	X	
Enseñanza y Aprendizaje		
Compatibilidad XMI		
Exportar Imágenes	X	
Producir Documentación		
TOTAL PUNTUACIÓN 2		6
3. Ingeniería de Software		
Diagrama CU		
Especificación CU		
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia	X	
Aspectos Notacionales UML		
Validación de Modelo UML		
Interpretación OCL		
Validación Inv. OCL		
Generación Código .NET		
Ingeniería Inversa		
Identificación de Roles		
TOTAL PUNTUACIÓN 3		4

PUNTIACIÓN TOTAL	10
------------------	----

Tabla I.2. Evaluación Giffy

El porcentaje de aceptación de la herramienta es $(10 \cdot 100) / 22$. Lo que da un total de **45.45%**.

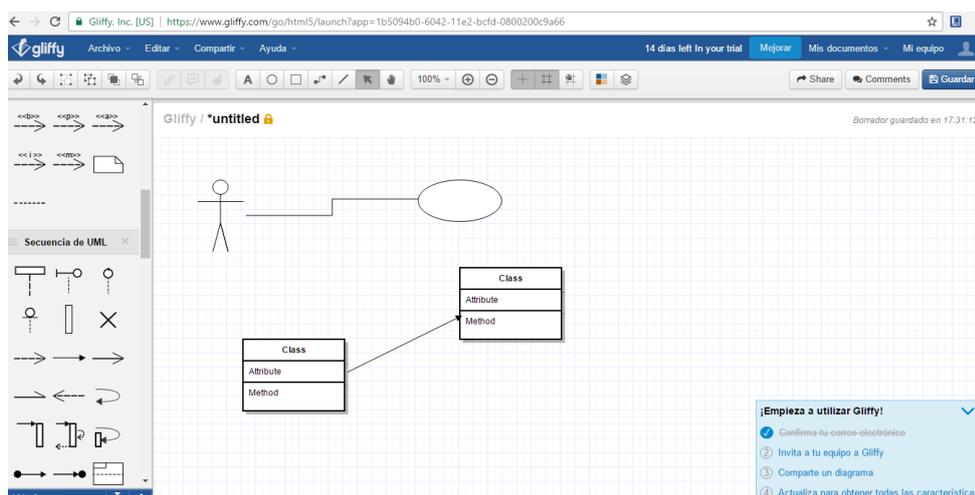


Figura I.4. Interfaz de usuario de Giffy

I.2.3. yEd



Figura I.5. yEd

yEd [YWO17] es un editor gráfico de escritorio, pero la empresa está desarrollando para ambientes web. La versión analizada corresponde a la edición web. Entre las características que destaca el sitio web encontramos la posibilidad de crear diagramas de forma sencilla con la posibilidad de exportar a distintos formatos gráficos (mapas de bits o vectores).

La versión descargada viene muy limitada en cuanto a las posibilidades de diseño, pero es fácilmente extensible bajando librerías adicionales de su catálogo. Esto la hace muy potente ya que posee las Apis disponibles para desarrollar nuevas bibliotecas de gráficos.

Con respecto a la interfaz de usuario, es sencilla y minimalista pero no es muy agradable al uso. Es sencilla de aprender, pero no muy completa.

1. Principal		
Nombre	yEd	
Empresa	yworks	
Descripción	Editor Gráfico	
Versión	3.16.2.1	
Plataforma	Multiplataforma	
Precio		free
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online	X	
Trabajo Colaborativo	X	
Enseñanza y Aprendizaje		
Compatibilidad XMI		
Exportar Imágenes	X	
Producir Documentación		
TOTAL PUNTUACIÓN 2		6
3. Ingeniería de Software		
Diagrama CU		
Especificación CU		
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia		
Aspectos Notacionales UML		La definición de los metadatos UML es muy acotada. No permite definiciones notacionales ni estereotipos.
Validación de Modelo UML		
Interpretación OCL		
Validación Inv. OCL		
Generación Código .NET		
Ingeniería Inversa		
Identificación de Roles		
TOTAL PUNTUACIÓN 3		3
PUNTIACIÓN TOTAL		9

Tabla I.3. Evaluación yED

El porcentaje de aceptación de la herramienta es $(9 \cdot 100) / 22$. Lo que da un total de **40.90%**.

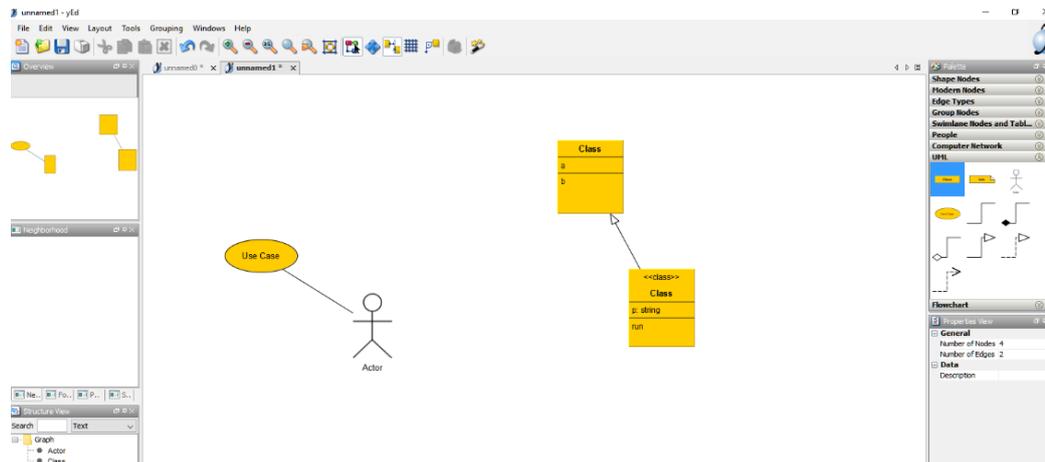


Figura I.6. Interfaz de usuario de yED

I.2.4. ArgoUML



Figura I.7. ArgoUML

ArgoUML [TIG17] es una herramienta de modelado *OpenSource* muy completa, disponible desde 1998. Se presenta como una herramienta de análisis y diseño orientado a objetos y se enfoca en “psicología cognitiva para proporcionar nuevas características que incrementen la productividad” según necesidades de diseñadores y arquitectos de software.

La versión descargada es completa, pero no tiene toda la funcionalidad esperada. Esto se soluciona descargando los *plugins* necesarios del sitio web. Entre ellos, se encuentran herramientas para **OCL**, generación de código en distintos lenguajes, etc.

Con respecto a la interfaz de usuario, si bien tiene bastante funcionalidad, la distribución y organización es muy efectiva. Es una herramienta fácil de aprender y muy amigable.

1. Principal		
Nombre	Argo UML	
Empresa	Tigris	
Descripción	Herramienta de Modelado UML para Análisis y Diseño de Software	
Versión	0.34	
Plataforma	Multiplataforma (Java)	
Precio		OpenSource
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online		
Trabajo Colaborativo		
Enseñanza y Aprendizaje		
Compatibilidad XMI	X	
Exportar Imágenes	X	
Producir Documentación	X	
TOTAL PUNTUACIÓN 2		6
3. Ingeniería de Software		
Diagrama CU	X	
Especificación CU		
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia	X	
Aspectos Notacionales UML	X	
Validación de Modelo UML	X	
Interpretación OCL	X	
Validación Inv. OCL	X	
Generación Código .NET	X	
Ingeniería Inversa	X	
Identificación de Roles		
TOTAL PUNTUACIÓN 3		11

PUNTIACIÓN TOTAL

17

Tabla I.4. Evaluación ArgoUML

El porcentaje de aceptación de la herramienta es $(17 \cdot 100) / 22$. Lo que da un total de **77.27%**.

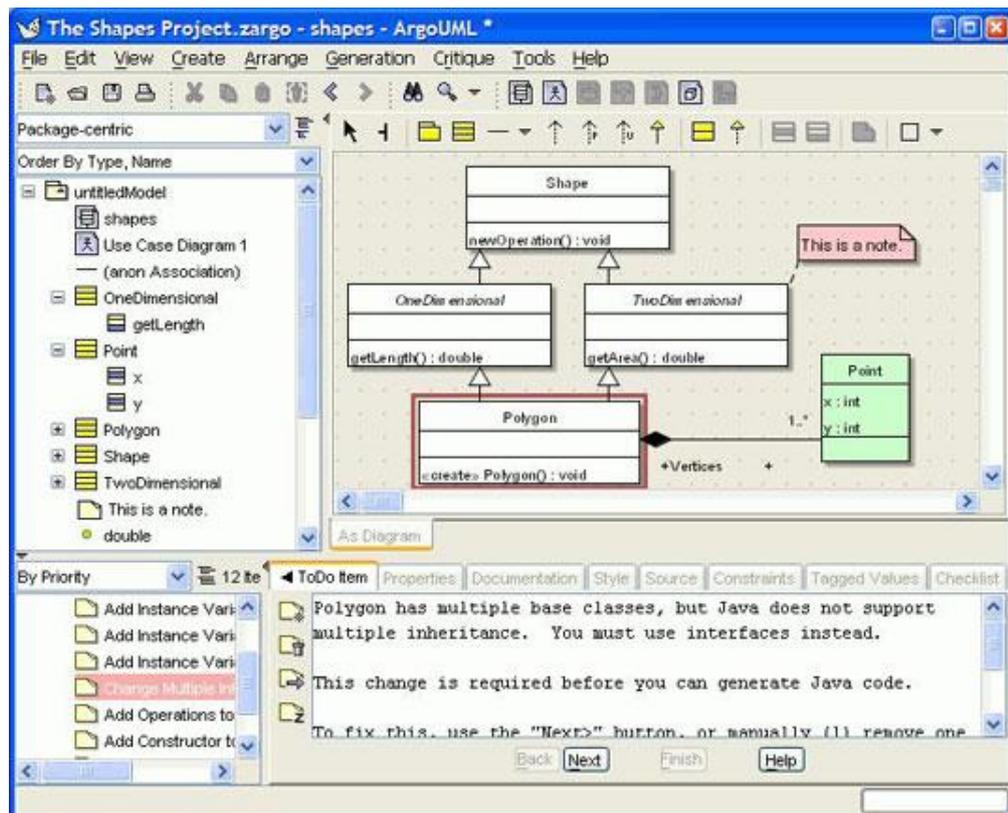


Figura I.8. Interfaz de usuario de ArgoUML

I.2.5. StarUML



Figura I.9. Logo StarUML

StarUML [STA17] es una herramienta de modelado con soporte para **UML 2.x**. Entre sus principales características, su sitio web publica:

- Modelado Rápido
- Soporte para ERD (Diagrama Entidad-Relación).
- Extensible por medio de *plugins*.
- Ingeniería de código incluyendo Java, C#, C++
- Validación de Modelos UML
- Ingeniería Inversa, Generación de código y Documentación.

Posee diferentes licencias, Personal, Comercial, Educacional y de Laboratorio. La diferencia entre estas últimas es que La Educacional es individual y la otra por cantidad.

Con respecto a la interfaz de Usuario, es sencilla de usar y aprender. Está bien organizada.

1. Principal		
Nombre	StarUML	
Empresa	MKLab	
Descripción	Herramienta avanzada de Modelado UML	
Versión	2.8	
Plataforma	Multiplataforma	
Precio		OpenSource
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online		
Trabajo Colaborativo		
Enseñanza y Aprendizaje		
Compatibilidad XMI	X	
Exportar Imágenes	X	
Producir Documentación	X	
TOTAL PUNTUACIÓN 2		6
3. Ingeniería de Software		
Diagrama CU	X	
Especificación CU		
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia	X	

Aspectos Notacionales UML	X	
Validación de Modelo UML	X	
Interpretación OCL		
Validación Inv. OCL		
Generación Código .NET	X	
Ingeniería Inversa	X	
Identificación de Roles		
TOTAL PUNTUACIÓN 3		9

PUNTIACIÓN TOTAL	15
-------------------------	-----------

Tabla I.5. Evaluación StarUML

El porcentaje de aceptación de la herramienta es $(15 \cdot 100) / 22$. Lo que da un total de **68.18%**.

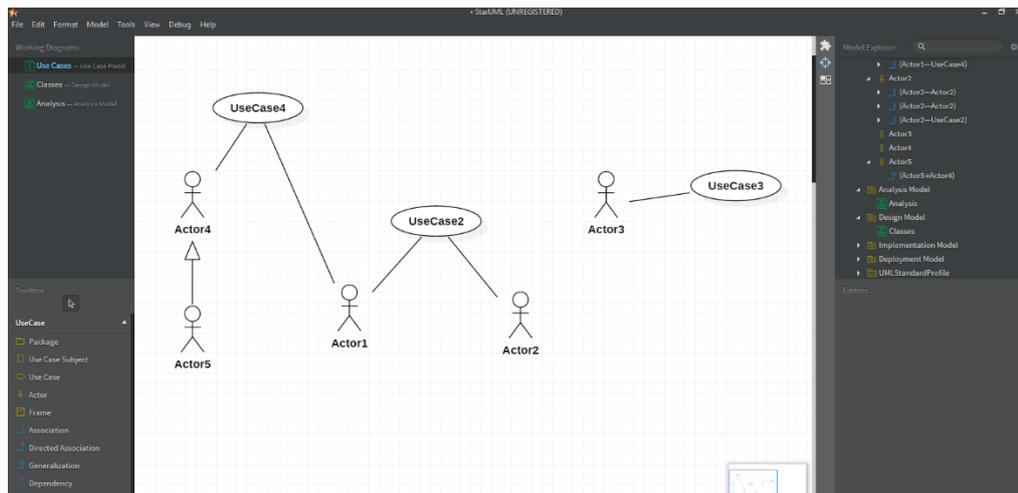


Figura I.10. Interfaz de usuario de StarUML

I.2.6. VisualParadigm



Figura I.11. Logo Visual Paradigm

Visual Paradigm [VIS17] es una herramienta de modelado y diseño de sistemas con soporte para UML. Entre sus características detalladas en el sitio web, posee las siguientes:

- Soporte DER y diseño de bases de datos
- Herramientas de gestión de proyectos como historias de usuario, kanban, glosario.
- Soporte para **BPMN**
- Soporte para *wireframing*
- Diseño de **APIS**, mapas mentales

En base al estudio realizado, comprendemos que es una herramienta avanzada la cual se distribuye en cuatro ediciones. *Modeler, Standard, Profesional y Enterprise*. Como características adicionales, ofrecen una versión *Community FREE* para usuarios no comerciales y, además, promocionan un programa académico.

Con respecto a la Interfaz de usuario, presenta muchas funcionalidades en pantalla, pero permite personalizar su uso. Es fácil de aprender y amigable a usuarios con poca experiencia.

1. Principal		
Nombre	Visual Paradigm	
Empresa	Visual Paradigm	
Descripción	Herramienta avanzada de Modelado UML	
Versión	14 Standard	
Plataforma	Multiplataforma	
Precio	U\$S	349
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online		
Trabajo Colaborativo	X	
Enseñanza y Aprendizaje		
Compatibilidad XMI	X	
Exportar Imágenes	X	
Producir Documentación	X	

TOTAL PUNTUACIÓN 2	7
3. Ingeniería de Software	
Diagrama CU	X
Especificación CU	X
Diagrama de Dominio	X
Diagrama de Actividades	X
Diagrama de Clases	X
Diagrama de Secuencia	X
Aspectos Notacionales UML	X
Validación de Modelo UML	X
Interpretación OCL	
Validación Inv. OCL	
Generación Código .NET	X
Ingeniería Inversa	X
Identificación de Roles	
TOTAL PUNTUACIÓN 3	10
PUNTIACIÓN TOTAL	17

Tabla I.6. Evaluación Visual Paradigm

El porcentaje de aceptación de la herramienta es $(17 \cdot 100) / 22$. Lo que da un total de **77.27%**.

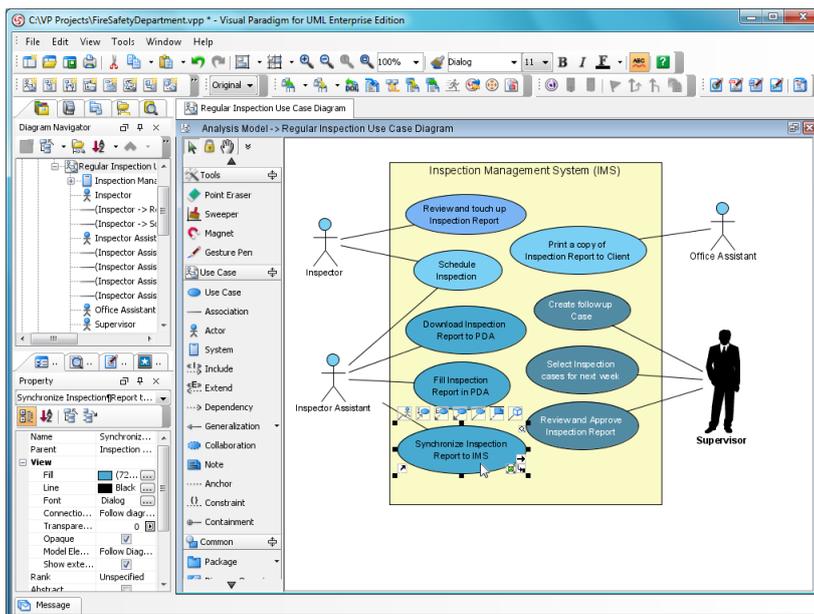


Figura I.12. Interfaz de usuario Visual Paradigm

I.3. Resumen

En este anexo presentamos un análisis detallado de diferentes herramientas **CASE** de modelado. Este estudio se hizo en base a la grilla de evaluación propuesta en el capítulo 4 con el objetivo de obtener el porcentaje de aceptación de cada una de las herramientas en base a los criterios previamente definidos.

Anexo II

Metodología para Diseñar Actividades Colaborativas

II.1. Introducción

Diseñar software que encuadre con el paradigma **CSCW** requiere abordar de forma metodológica su descripción a partir de la identificación de las actividades recurrentes o destacadas de una aplicación.

Para esto utilizaremos una metodología basada en la propuesta por Coto en [CCR16] y la propuesta por Rodríguez en [ROD12].

II.2. Propuesta

A continuación, se muestra la secuencia de fases propuesta por Coto en su trabajo. Fue utilizada para describir la metodología propuesta.

La metodología de diseño de actividades colaborativas (**MDAC**) propone un conjunto de fases de las cuales, a fines prácticos, solo usaremos las siguientes:

- Fase 1 *Diagnóstico de la Tarea*: Se realizar una descripción detallada de la tarea objeto de estudio.
- Fase 2 *Evaluación de la Actividad*: Deben identificarse las actividades generales que la componen y determinar la secuencia entre ellas
- Fase 3 *Descomposición de la Actividad*: Descripción de las sub actividades que componen cada una de las actividades generales descriptas en la fase 2.
- Fase 4 *Validación del diseño*: Esta metodología ofrece 4 formas de validación. (1) Prueba piloto, que consiste en ejecutar el proceso para evaluar la efectividad. (2)

Recorrido con algunos participantes en la ejecución del proceso, para luego realizar una evaluación para identificar falencias y debilidades. (3) Simulación, para que los diseñadores del proceso den respuestas a un conjunto de preguntas definidas. Esto se usa para probar la lógica del diseño y si cada paso genera el entregable requerido. (4) Revisión. En donde se genera una discusión entre el equipo de trabajo y los usuarios del proceso, para evaluar el proceso diseñado.

Comprendemos que son fases muy importantes a considerar en la documentación de actividades colaborativas. Para complementar estas actividades, coincidimos con Rodríguez quien asegura que no existen artefactos de especificación conceptual para el modelado de tareas colaborativas, y la dificultad de modelar aspectos de interacción y trabajo en grupo. Es por esto que se destaca la importancia de modelar aspecto de dinámica grupal relacionados a las interacciones sociales y a las responsabilidades inter-individuales.

En este sentido, propone un conjunto de formalismos, de los cuales utilizaremos los descriptos a continuación:

- **Tabla Concepto-Categoría-Definición (CCD).** Esta tabla se utiliza para representar los conocimientos relativos al modelo de la dinámica grupal. Un concepto puede ser alguna de las siguientes categorías: actor, objeto o interacción. Los actores son sujetos de la dinámica, un objeto es el asunto que recibe la facultad de interactuar con actores y las interacciones definen los procesos que los actores acuerdan sobre los objetos.
- **Casos de Interacción y Diagramas de Interacción Grupal.** son dos formalismos para modelar interacción. Los casos de interacción modelan la interacción entre dos actores y los diagramas de interacción grupal muestra de forma integrada las interacciones de todos los actores considerados en el proceso de modelado. Estos diagramas se apoyan en los diagramas de casos de uso
- **Diagramas de Secuencia de Dinámica Grupal.** Este diagrama expresa la dinámica grupal dentro de un contexto temporal. Utiliza un diagrama de secuencia **UML**.
- **Diagrama de Desarrollo de Objetos Conceptuales (DOC).** Este diagrama representa el objeto conceptual que grupo de trabajo debe desarrollar. El espacio

colaborativo debe satisfacer las actividades que soporten las distintas transformaciones que marquen la evolución de dicho objeto.

En base a estos dos trabajos, proponemos un modelo que utilizaremos durante la definición de las actividades colaborativas. Nuestro modelo propone 4 etapas y en cada etapa utiliza un formalismo. La etapa 2 genera un conjunto de artefactos que serán tratados en las etapas que continúan y dependen de esta.

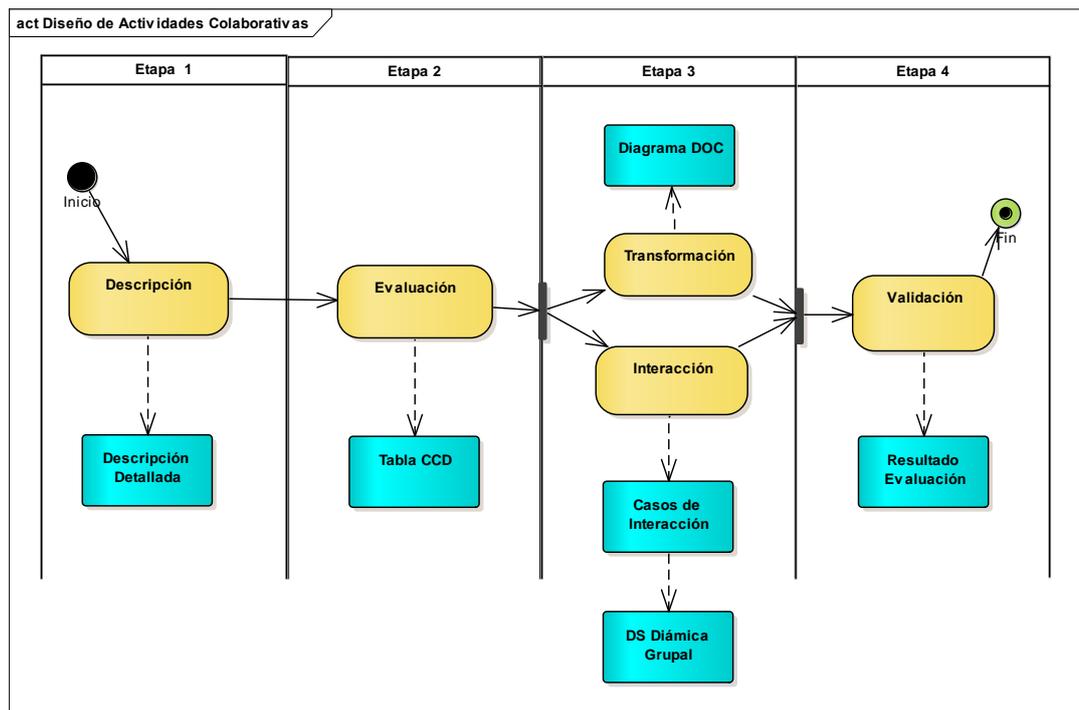


Figura II.1. Modelo para el diseño de actividades colaborativas

A Continuación, describimos las etapas del modelo planteado (**MDAC**)

- **Etapa 1 Descripción.** En esta etapa se realiza una descripción detallada de la tarea, con todas las personas o roles intervinientes y las actividades que llevan a cabo.
- **Etapa 2 Evaluación.** SE describe todo el conocimiento sobre de la tarea por medio de la "Tabla **CCD**". En este punto se formalizan los actores, los objetos y las interacciones identificadas en la etapa 2.

- **Etapa 3.1 Interacción.** Se modelan las interacciones descritas en la etapa 2. Se utilizan los formalismos “Casos de Interacción” y “Diagrama de secuencia de Dinámica Grupal” para tal fin.
- **Etapa 3.2 Transformación.** Se modela por medio de un diagrama de actividad UML las interacciones en cada etapa de construcción de los objetos conceptuales identificados. Utilizamos en esta etapa el “Diagrama de Desarrollo de Objetos Conceptuales (Diagrama **DOC**)”.
- **Etapa 4 Validación.** Se analiza el proceso y se realiza una evaluación final en base al resultado esperado.

La última instancia es la presentación de relación entre las distintas fases de la **MDAC** y la **TA**, como se observa en la tabla II.1:

Componentes TA	MDAC
Comunidad	Etapa 1
Objetivos	Etapa 1
División del trabajo	Etapa 2
Herramientas	Etapa 3
Resultados	Etapa 4

Tabla II.1. Relación entre MDAC Y TA

II.2. Resumen

En este anexo se presentó una metodología para especificar actividades colaborativas.

Anexo III

UAI Case: Prototipo

III.1. Introducción

En este anexo abordamos de forma detallada el diseño del prototipo. El desarrollo se dividió físicamente en dos módulos definidos por el modelo **uCASE-CL**. Por un lado, la herramienta académica y por otro la herramienta **CASE** integradas en una plataforma tecnológica ubicua. El prototipo se encuentra disponible para acceder por medio de la siguiente dirección <http://case.uai.edu.ar>

III.2. Plataforma Tecnológica Ubicua

El diseño de la plataforma tecnológica se centró en dos aspectos principales. Por un lado, el aspecto ubicuo, y por el otro el colaborativo. Para esto fue necesario diseñar la arquitectura del software pensando en estos dos factores. Por tal motivo se diseñó una aplicación web utilizando una arquitectura basada en tecnologías **REST**, con acceso a los servicios de colaboración por medio de *Websockets*.

Los ambientes ubicuos tienen su representación en Arquitecturas Orientadas a Servicios (**SOA**), En este sentido, Montejano [MON16] y Liu [LIU08] aseguran que esta arquitectura posee la potencialidad necesaria para responder a los requisitos planteados por este tipo de ambientes. La arquitectura **SOA** puede ser implementada con la arquitectura propuesta por **REST**, que al igual que los entornos ubicuos, posee de forma natural acceso orientado a recursos (**URI**). Esto permite crear de forma sencilla servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda la semántica propuesta por **HTTP**.

La aplicación utiliza un modelo distribuido con una arquitectura cliente-servidor. De esta manera desarrollamos una aplicación **WEB** que consume servicios alojados en un servidor en el que se implementó una arquitectura

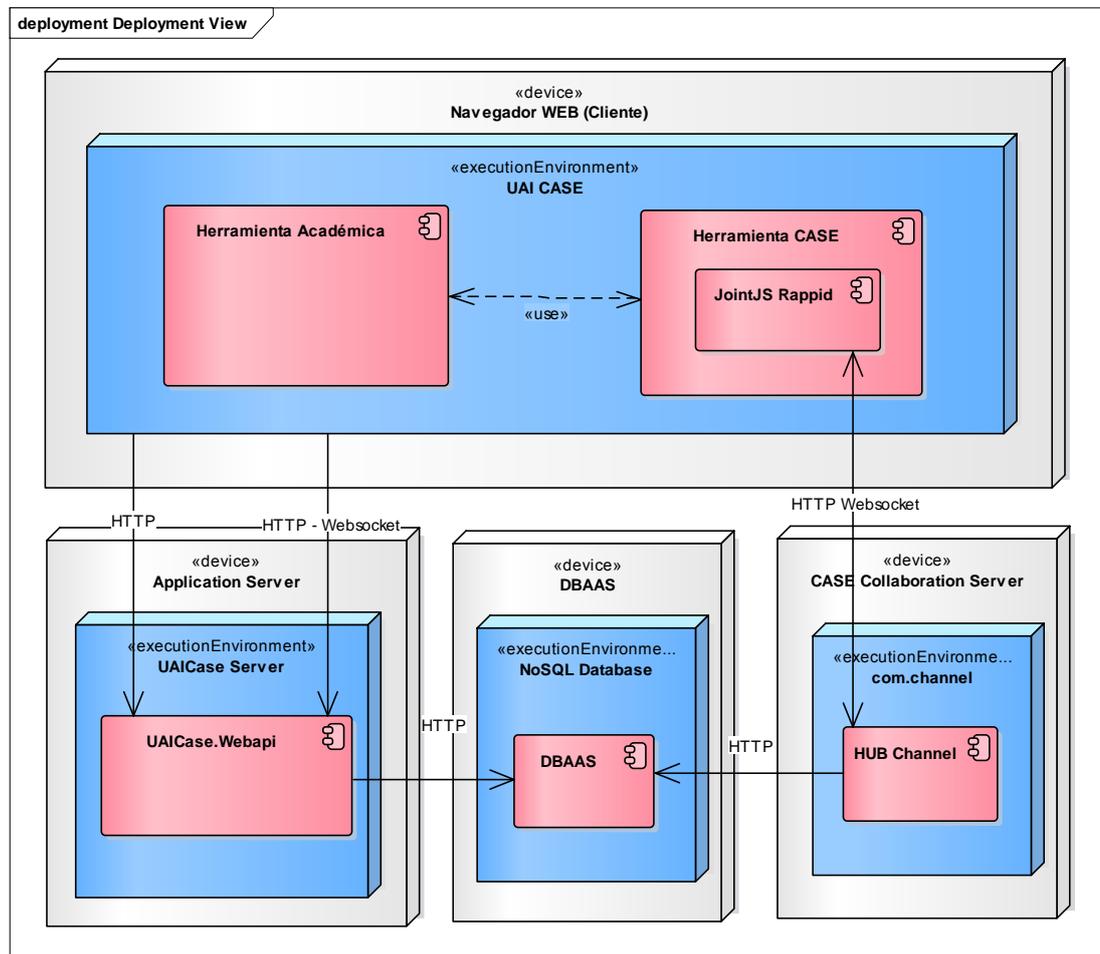


Figura III.1. Despliegue de UAI Case

Los distintos módulos del sistema serán implementados utilizando *Bluemix* [IBM17a] y la capa subyacente para computación en la nube (*Cloud Computing*) denominado *Softlayer* [IBM17b] provista por **IBM** en el marco del acuerdo con **UAI**, obtenido para implementar el proyecto **UAI Case** utilizando esta infraestructura.

III.2.1. Cliente (*Frontend*)

La herramienta está diseñada como una aplicación web compatible con la mayoría de los navegadores del mercado. Esta es la parte que el usuario ejecuta, es la que se encarga de mostrar la información y de comunicarse con el servidor (*Backend*). Esta plataforma

permite que un conjunto de usuarios geográficamente dispersos pueda trabajar de forma colaborativa en tiempo real.

El cliente se diseñó utilizando **HTML** y *Javascript*. Se utilizaron las siguientes tecnologías.

- **SPA** (*Single Page Application*), es una aplicación de una sola página que no se recarga en cada llamada que se haga al servidor, sino que se comunica de manera asíncrona. El contenido de la página se actualizará de forma dinámica según la interacción del usuario.
- *AngularJS* [ANG17] como framework **MVC** de *JavaScript* para el Desarrollo web *Frontend* que permite crear aplicaciones **SPA**. Este *framework* permite conectarse de forma asíncrona al servidor *Backend* por medio de llamadas *Ajax* a servicios **REST**.
- *Bootstrap* [BOO17] fue utilizado para desarrollar el proyecto web con los conceptos “*Responsive*” y “*Mobile First*” es decir, adaptable a distintas plataformas móviles que acceden a la aplicación.
- *JointJS* [JOI17] para diagramación **UML** y colaboración entre usuarios en tiempo real.
- La comunicación entre el *Frontend* y el *Backend* se realiza por medio de llamadas a la **API REST** expuesta por el servidor de aplicaciones. Estas llamadas se hacen sobre el protocolo **HTTP** utilizando el formato **JSON** para transferencia de datos. Los aspectos colaborativos se resolvieron utilizando el *framework SignalR* [SIG17]. Esta herramienta se utiliza para interconectar distintas plataformas por medio de *Websockets*. Un *Websocket* es una tecnología de comunicación que establece un canal full dúplex sobre un *socket TCP* entre un navegador y un servidor web, aunque esta misma tecnología puede utilizarse en cualquier tipo de aplicación que sea de tipo cliente servidor. Una **API** representa la capacidad de comunicación que tienen diferentes componentes software. De esta manera, una aplicación puede brindar diferentes funciones que podrán ser utilizadas por otras aplicaciones que las requieran, por ejemplo, una aplicación web o una aplicación nativa de algún dispositivo móvil.

III.2.2. Servidor (*Backend*)

A continuación, detallamos los aspectos técnicos relativos al servidor y su arquitectura. Utilizamos como plataforma de desarrollo *Visual Studio 2017 Community* con el *framework* ASP.NET CORE 2.0 [NET17]. Esta plataforma *open source* de Microsoft apuesta al uso de múltiples sistemas operativos (**SO**) para el desarrollo ya que por un lado posee implementación para *Windows, Linux* y *Mac* permitiendo así que usuarios de diversos **SO** puedan participar en el desarrollo

III.2.2.1. Arquitectura

Para la arquitectura planteada se utilizaron 6 capas físicas como se ve en la Figura III.2, detallando una capa transversal de seguridad y otra de entidades de negocio. Luego la capa de aplicación con la lógica de negocio y la capa como repositorio de datos. En la misma se utilizó el patrón *Repository* propuesto por Fowler [FOW17]. Por último, una de acceso a datos que permita escalar y portar el modelo a diferentes motores de bases de datos. En nuestro caso utilizamos una base de datos como servicio (**DBAAS**) provista por **IBM Bluemix**.

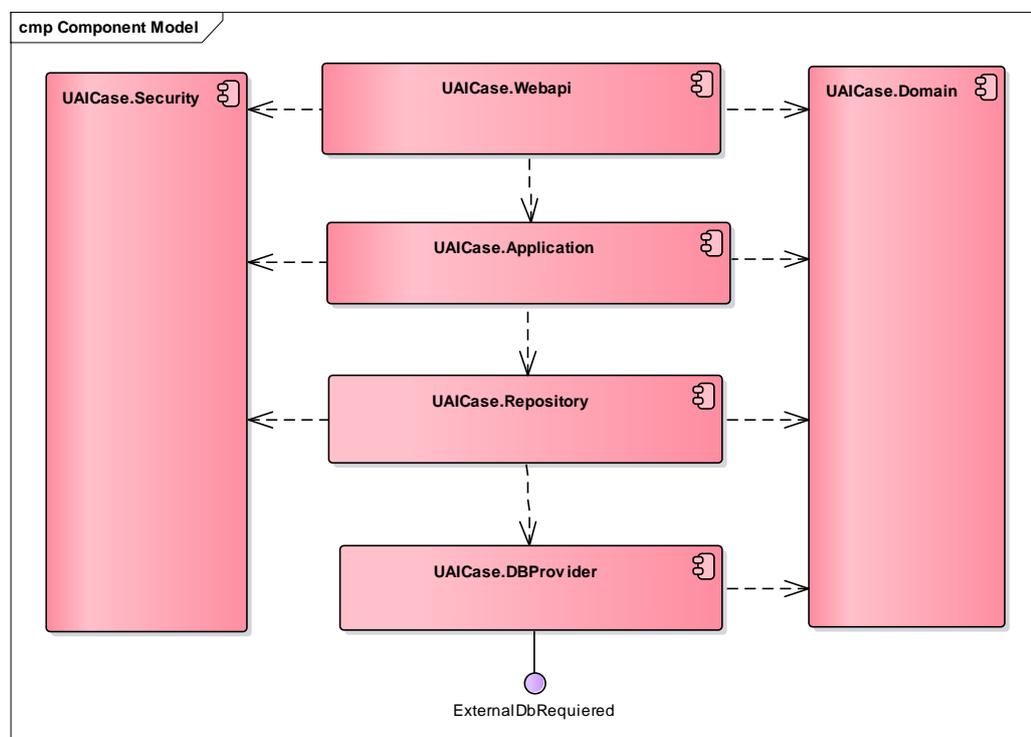


Figura III.2. Arquitectura Servidor UAI Case

III.2.2.2. Seguridad

Dos aspectos muy importantes a considerar desde el punto de vista de la seguridad inherente al acceso a las plataformas ubicuas, son los relacionados con *autenticación* y *autorización*. Una de las características más importantes de las arquitecturas **REST**, es que es libre de estado. Esto significa que el servidor no mantiene la sesión del usuario conectado, sino que la información viene en cada solicitud o *request* al mismo. De esta forma cada vez que un usuario del sistema quiera acceder a algún recurso (**URI**) va a hacerlo indicando dentro del mensaje **HTTP** (*HTTP request*) su información de acceso.

En primer lugar, la *autenticación* responde a la validación de las credenciales del usuario, es decir, si el nombre de usuario y contraseña se corresponden y son válidos. Una vez que el recurso de autenticación valida esta información genera un *token* o un componente léxico que tiene encriptada con una clave que genera el servidor, aquella información sensible del usuario que será requerida en futuros accesos a los recursos del sistema. En segundo lugar, la autorización. Esta acción es llevada a cabo en cada solicitud que el servidor recibe. Es decir, ante cada petición a un recurso en particular, el servidor analiza esta información y valida si el usuario tiene o no acceso al recurso solicitado.

Para implementar esto anteriormente mencionado, utilizamos el estándar abierto denominado **JWT** (*JSON Web Token*) [JWT17] identificado con la RFC 7519. Este estándar define una forma *compacta* (por su tamaño pequeño) y *auto contenida* (contiene toda la información requerida acerca del usuario sin necesidad de acceder a una base de datos en cada petición) para transmitir información de forma segura entre partes de una red en forma de objetos **JSON**.

A continuación, mostramos por medio de un diagrama de secuencia los dos escenarios descritos anteriormente. La autenticación para obtener el **JWT** y cada una de las solicitudes subsiguientes utilizando este dato obtenido en el encabezado (*header*) de cada mensaje enviado por **HTTP**. En caso de que la solicitud sea aceptada se informa por medio de los códigos de estado de **HTTP**, utilizando el *200 OK* o bien el código *401 UNAUTHORIZED* en caso contrario.

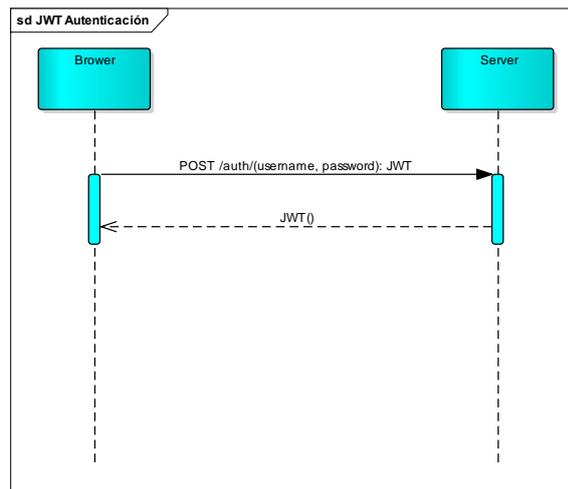


Figura III.3. Autenticación

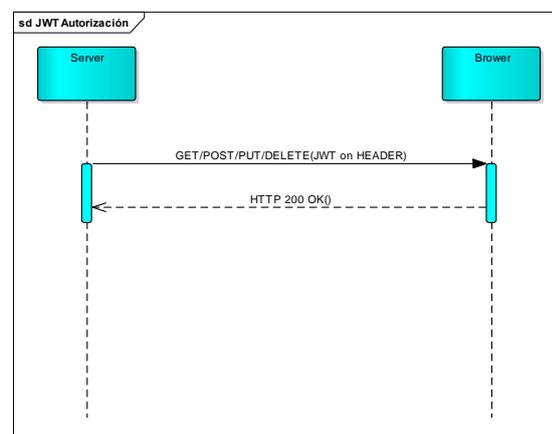


Figura III.4. Autorización

III.2.2.3 Trabajo en Tiempo Real

Para el trabajo en tiempo real utilizamos *SignalR [SIG17]* en su versión para **.NET CORE**. Se creó un punto de acceso (denominado **HUB**) por el cual los clientes (*Browsers*) establecen una conexión con el servidor, por donde recibirán información relacionada al entorno de trabajo, *awareness* y mensajes directo del servidor.

III.2.3. Servidor CASE

Este bloque corresponde a un servidor que se utiliza para la sincronización de un modelo **UML** entre todos los usuarios que están conectados al mismo. Esta plataforma utiliza

tecnología *Javascript* tanto en el servidor como en el cliente, por medio de una librería denominada *JointJS* [JOI17] que brinda un canal directo de comunicación entre el browser y el servidor creado por un *websocket*.

Esta librería posee un plugin denominado *Channel* que utiliza un algoritmo denominado transformación operacional (**TO**) para mantener la consistencia entre los usuarios concurrentes. Esto quiere decir que, cuando exista algún cambio en el modelo, todos los usuarios que lo están observando, recibirán las *modificaciones* y el plugin las reflejara en su pantalla.

El algoritmo **TO** tiene como objetivo transformar (o ajustar) los parámetros de una operación de edición de acuerdo con los efectos de operaciones simultáneas ejecutadas previamente para que la operación transformada pueda lograr el efecto correcto y mantener la consistencia del documento. La implementación propuesta por *JointJS* está basada en [NIC95].

III.2.4. DBAAS

La persistencia tanto de los datos como de los modelos **UML** se realiza en un motor de base de datos **NOSQL**, en particular orientado a documentos. La utilización de *MongoDB* [MON17] como motor, permite además de mantener una escalabilidad alta, realizar consultas mucho más rápido y simplificar la complejidad de la persistencia de los modelos **UML**.

En nuestro caso utilizamos una base de datos como servicio (**DBAAS**) provista por **IBM Bluemix**.

III.3. Plataforma CSCL

Esta sección describe la plataforma **CSCL**, denominada *Herramienta Académica*, con todas sus características y funcionalidades destacadas.

III.3.1. Trabajo Colaborativo

En esta sección incluiremos las principales actividades colaborativas de la herramienta CSCL.

III.3.1.1. Espacio de Trabajo para Grupos de Alumnos

El espacio de trabajo destinado a los grupos de alumnos, tiene como finalidad integrar aquellas actividades que deberán llevarse a cabo de esta manera. La herramienta dispone de una lista de “Grupos Activos” que representan todos los grupos en los que el usuario activo en el sistema.

El espacio de trabajo corresponde a un grupo en particular. Esto significa que, si el alumno participa en diferentes grupos, de diferentes materias, va a tener un entorno para cada uno. Como información contextual, el entorno dispone del estado del docente (contacto o no conectado) y un registro de eventos (*logs*) del grupo, por ejemplo, quien agrego una nueva actividad, quien se unió al grupo, quien modificó un documento (figura III.5).

UAI Case / MATERIA DE PRUEBA / G1

Grupo
Aquí encontrara toda la información sobre el grupo

Docente docente, docente

Principal | Salón de Chat (1) | Pendientes (1) | Proyectos CASE | **Logs** | Mensajes

Alumnos del Grupo			
LEGAJO	NOMBRE	APELLIDO	ACCIONES
0	alumno2	alumno2	
0	alumno	alumno	📧 ●

TODO Reactivado
terminar ejercicio 1
🕒 hace 13 minutos

TODO Finalizado
terminar ejercicio 1
🕒 hace 22 minutos

TODO Finalizado
agregar caso de uso
🕒 hace 22 minutos

Figura III.5. Espacio de Trabajo del Grupo de Alumnos

Cada entorno dispone de diferentes solapas que representan las siguientes características. (1) Listado de alumnos del grupo, con información de su estado en el sistema indicando con verde si está conectado o con rojo si no lo está. (2) Salón de *chat* con acceso total para los integrantes del grupo, al cual el docente no tiene acceso. (3) Listado de tareas Pendientes (**ToDo**) del grupo. (4) Proyectos **CASE** que están siendo

trabajados por el grupo de forma colaborativa (figura III.6). (5) Documentos que está trabajando el grupo con su correspondiente control de versionado (Esta característica aún se está desarrollando en el prototipo)



Gestión de Proyectos CASE del Grupo
Aquí encontrara todos sus proyectos compartidos para el grupo

1 Proyectos

<input type="checkbox"/>	NUMERO	AUTOR	CREADO	NOMBRE	CURSO	DESCRIPCION	ESTADO	ACCIONES
<input type="checkbox"/>	0	alumno alumno	hace 8 meses	PRUEBA	MATERIA DE PRUEBA	PROYECTO DE PRUEBA	EnProceso	   

Figura III.6. Proyectos Colaborativos del Grupo de Trabajo

III.3.1.2. Espacio de Trabajo para Grupos de Docentes

Los docentes también disponen de un entorno para que puedan interactuar y realizar actividades relacionadas al control y seguimiento de cada una de las materias en la cuales un docente puede ser, titular o adjunto (Esta diferenciación es en base al rol que cumple con la materia y se define de forma automática). Si bien la vista muy similar, cambiaran las acciones. Este entorno dispone de (1) Información sobre los docentes que dictan la materia, en que curso y en qué sede. (2) Un canal de chat. (3) Gestión de Contenido, editable para el titular y visible para el resto. En esta actividad el docente titular podrá subir el contenido definitivo para cada materia, que luego el adjunto deberá utilizar en el dictado del curso. (4) Rendimiento Académico. Esta característica muestra el rendimiento académico de los diferentes cursos, separando por tipo de examen (primer parcial, segundo parcial, recuperatorio, final, etc.).

Posee como información contextual, el estado del docente titular (contactado o desconectado) y un visor de eventos (*logs*) del grupo colaborativo (Ver figura III.7).



Figura III.7. Espacio de Trabajo para un Grupo de Docentes

III.3.1.3. Espacio de Trabajo para un Curso

Este entorno es el que vincula Alumnos y Docentes en lo que denominamos Curso. Para acceder al mismo, utilizan la opción “*Mis Cursos Activos*” en donde el alumno verá los cursos en lo que está inscripto y el docente de los cuales es adjunto. El entorno dispone de la lista de los alumnos de curso con el estado de conexión (conectado y desconectado), la planificación de la materia, referencia del docente adjunto responsable del curso, un canal de *chat* y un listado de eventos (*logs*) que están compartidos por ambos roles. En adición, desde la perspectiva del docente, aparecen actividades de planificación (con la posibilidad de editarla), evaluación y colaboración en proyectos **CASE**.

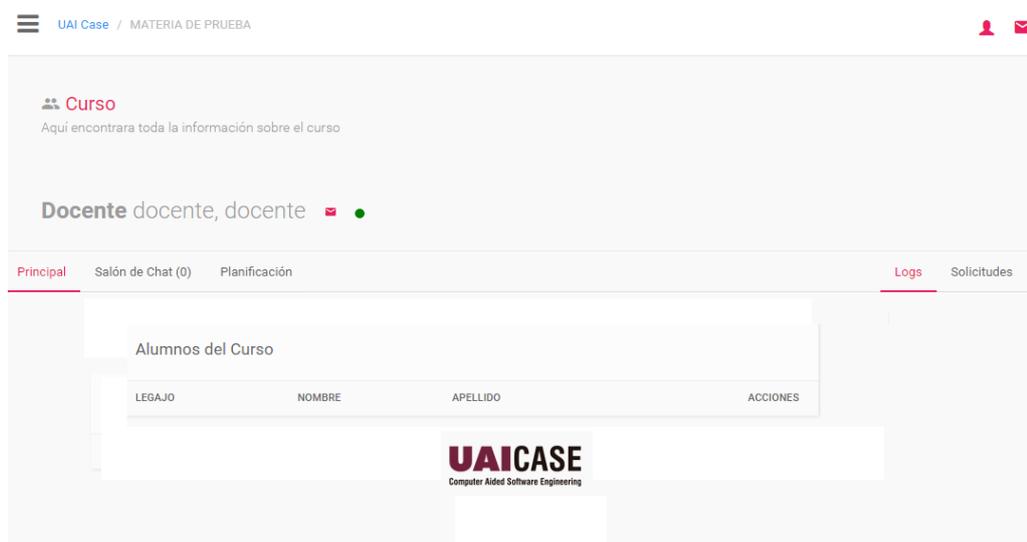


Figura III.8. Espacio de Trabajo del Curso desde la Perspectiva el Alumno

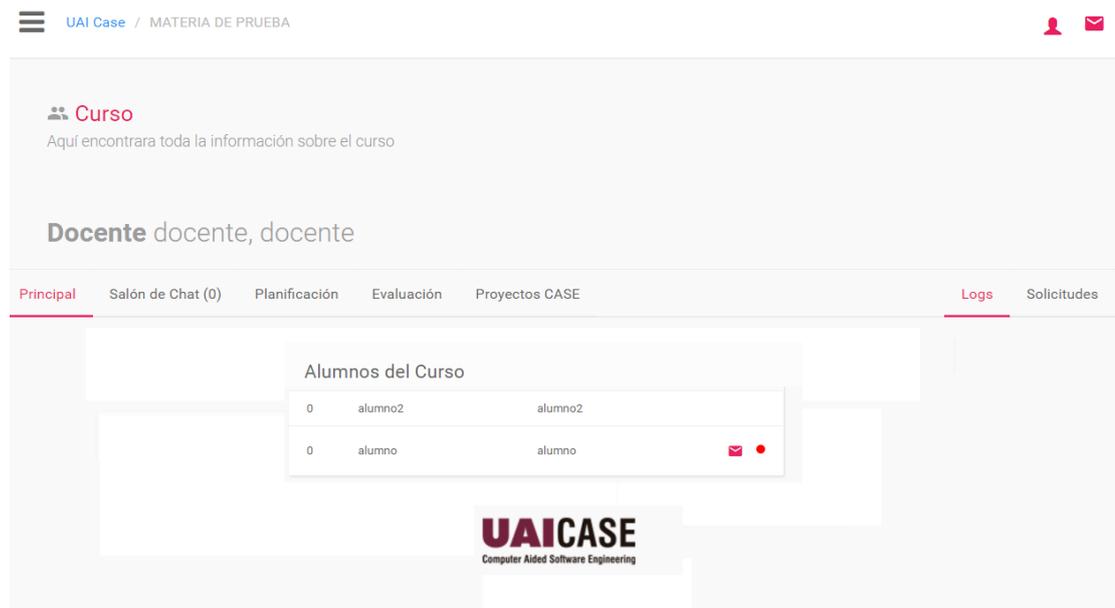


Figura III.9. Espacio de Trabajo del Curso desde la Perspectiva de Docente

III.3.1.4. Sesiones de Chat.

La herramienta cuenta con el desarrollo de un *plugin* que se puede instanciar para distintos grupos colaborativos, y de esta forma crear distintos canales de chat, en base a las necesidades. El salón de chat creado, mantiene todo el historial de las conversaciones realizadas con fecha y hora, y dispone la información sobre los mensajes no leídos.

Este *plugin* está disponible en los entornos de trabajo grupales tanto de docentes, como de alumnos, en el entorno de trabajo de curso y en las sesiones de modelado a nivel diagrama. El control de acceso al canal de chat está dado por una lista de usuarios habilitados.

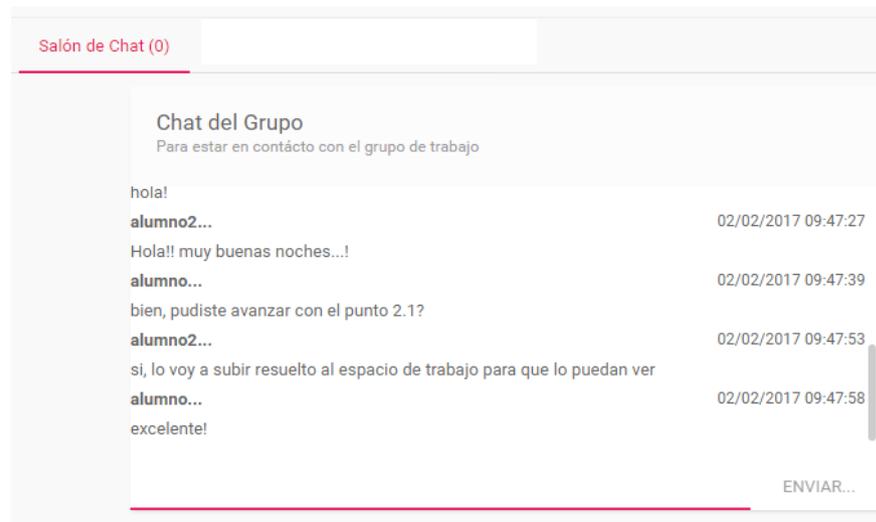


Figura III.10. Salón de chat en UAI Case.

III.3.1.3. Tareas (ToDo)

UAI Case cuenta con un módulo diseñado para control de tareas pendientes en un grupo de trabajo. Este módulo, diseñado como un *plugin*, permite crear distintas instancias que representan listad de tareas pendientes denominado “*Todo*” en la herramienta. Esta solución permite que los diferentes participantes del grupo colaborativo que utiliza la lista, puedan cargar ítems para resolver, indicando fecha de carga y usuario. Una vez cargados, permite que cualquier usuario los pueda dar por terminados o reactivar, indicado con información en cada caso. Cada vez que sucede algún evento, es informado al resto de los participantes, los cuales se cargaran en una lista relacionada. En este modelo, solo los usuarios creadores podrán editar o eliminar una tarea de la lista.

Este *plugin* ofrece como información contextual, la cantidad de ítems pendientes, y los eventos que dispara cada interacción con la lista. El control de acceso a la lista, está dado por un conjunto de usuarios definidos. Está disponible en todas las áreas de trabajo. La individual, la grupal (grupos de alumnos o docentes), a nivel curso, y en la herramienta de modelado a nivel proyecto.



Figura III.11. Lista de Tareas Pendientes.

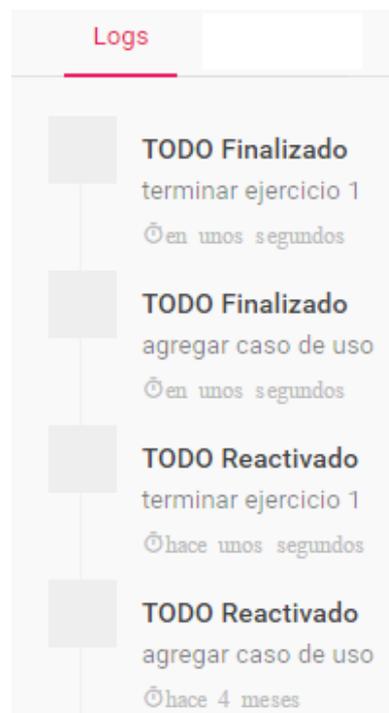


Figura III.12. Registro de Eventos sobre una Lista de Tareas

III.3.1.4. Mensajería Interna

La herramienta dispone de un centro de mensajería interna para que se puedan enviar mensajes entre alumnos o con los docentes. Estos mensajes representan un sistema asincrónico de colaboración. Los mensajes se crean en formato **HTML**. El centro de mensajería tiene una bandeja de entrada y una bandeja de elementos enviados. Como

información contextual informa la cantidad de mensajes no leídos, y muestra en tiempo real cuando se recibe un mensaje y cuando otro usuario lee un mensaje enviado.



Figura III.13. Redacción de un Mensaje Interno

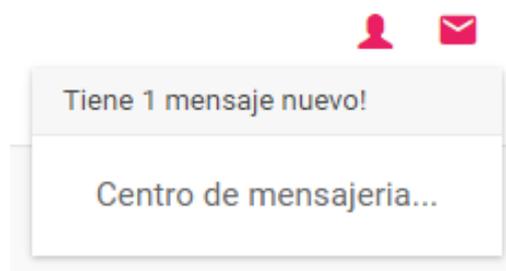


Figura III.14. Información Contextual sobre Mensaje Recibido.

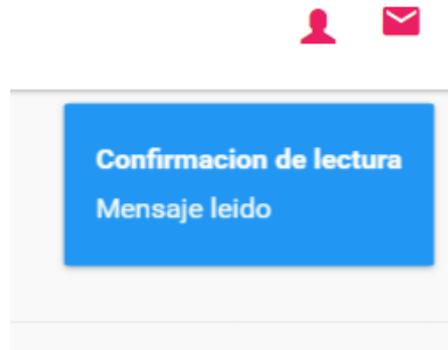


Figura III.15. Información Contextual sobre Mensaje Leído.

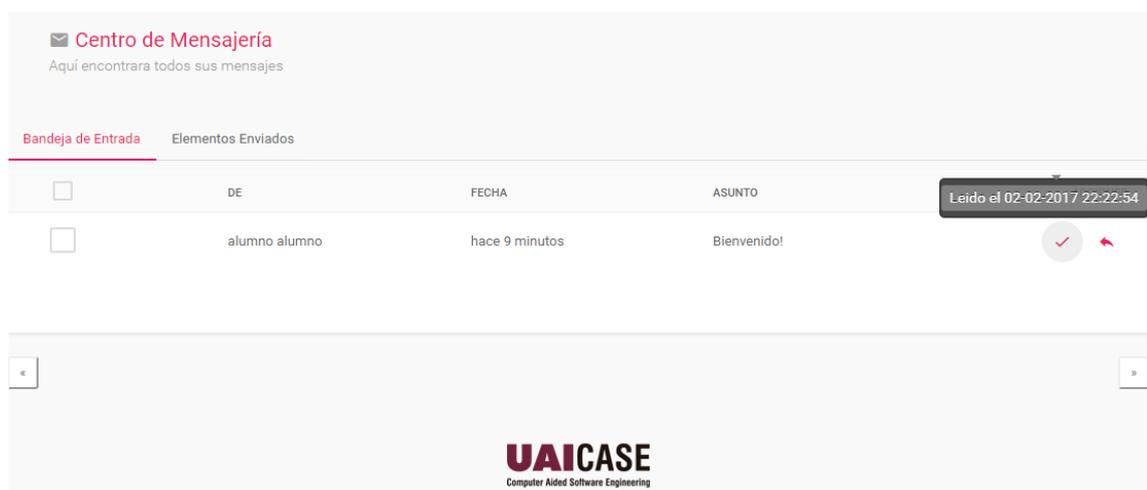


Figura III.16. Centro de Mensajería

III.4. Herramienta CASE

En esta sección describiremos la herramienta CASE desde las características principales y actividades colaborativas.

III.4.1. Características Principales

La herramienta **CASE** que diseñamos cumple con las características deseables de una herramienta de modelado **UML** según lo estudiado en el capítulo 4.

A continuación, enumeramos las características principales de **UAI CASE**.

- Herramienta **CASE** diseñada en un ambiente online **WEB**.
- Permitirá compatibilizar con otras herramientas CASE por medio del estándar **XMI**.
- Modelado **UML** adaptado al estándar, con la posibilidad de realizar diagramas de clases, de secuencia, de casos de uso, de actividad, de estado.
- Permite la especificación de casos e uso utilizando una plantilla predefinida.
- Permite modelado de datos utilizando **DER**.
- Validación de modelos conforme al estándar.
- Permitirá generación de código a diferentes lenguajes de programación y permitirá la generación de lenguaje de definición de datos para utilizar en motores **SQL**.
- Permite trabajo colaborativo y coordinado
- Posee un espacio de interacción para que los docentes y los alumnos puedan interactuar entre sí.
- Posee un espacio de evaluación como herramienta pedagógica.
- Posee identificación de roles dentro del proceso de desarrollo de software.
- Permitirá interpretar invariantes **OCL**.
- Posee diferentes herramientas para utilizar diagramas de forma externa, como generación de archivos en formato *mapa de bits* o impresión.

III.4.3. Actividades Colaborativas

UAI Case contará con tres actividades colaborativas recurrentes planteadas en la dimensión pedagógica, planteado en el capítulo 6. Las mismas se resumen en

- Modelado UML colaborativo libre.
- Modelado UML colaborativo coordinado.
- Evaluación colaborativa de modelos UML.

III.4.4. Evaluación de UAI Case

Utilizamos la metodología propuesta en el Capítulo 4 para analizar la herramienta CASE propuesta. En este sentido, y comprendiendo que aún está en proceso de desarrollo, al momento no cuenta con todas las funcionalidades indicadas.

1. Principal		
Nombre	UAI CASE	
Empresa		
Descripción	Herramienta CASE académica de modelado	
Versión	Prototipo	
Plataforma	Multiplataforma	
Precio		N/D
2. Experiencia del Usuario		
Ayuda en Línea	X	
Manual de Usuario	X	
Soporte Técnico Flexible	X	
Trabajo Online	X	
Trabajo Colaborativo	X	
Enseñanza y Aprendizaje	X	
Compatibilidad XMI	X	
Exportar Imágenes	X	
Producir Documentación	X	
TOTAL PUNTUACIÓN 2		9
3. Ingeniería de Software		
Diagrama CU	X	
Especificación CU	X	
Diagrama de Dominio	X	
Diagrama de Actividades	X	
Diagrama de Clases	X	
Diagrama de Secuencia	X	
Aspectos Notacionales UML	X	
Validación de Modelo UML	X	
Interpretación OCL	X	
Validación Inv. OCL	X	
Generación Código .NET	X	
Ingeniería Inversa	X	
Identificación de Roles	X	
TOTAL PUNTUACIÓN 3		13
PUNTIACIÓN TOTAL		22 (100%)

Tabla III.1. Tabla de evaluación UAI Case

Basado en nuestro modelo de análisis, el porcentaje de aceptación de la herramienta será **(23 * 100) / 23**. Lo que da un total de **100 %**.

Acrónimos

API	Application Programming Interface Interfaz de programación de aplicaciones. Conjunto de subrutinas, funciones y métodos que ofrece cierta biblioteca para ser utilizado por otro software.
CASE	Computer Aided Software Engineering Ingeniería de Software Asistida por Computadora. Conjunto de herramientas utilizadas para asistir al proceso de IS.
CCD	Concepto-Categoría-Definición Tabla utilizada en la descripción de actividades colaborativas
CSCL	Computer Supported Collaborative Learning Aprendizaje Colaborativo Asistido por Computadora
CSCW	Computer Supported Collaborative Work Trabajo Colaborativo Asistido por Computadora
DBAAS	Data Base As A Service Es un enfoque basado en la nube para el almacenamiento y la gestión de datos estructurados accesibles por medio de servicios.
DOC	Diseño de Objetos Conceptuales Modelo utilizado en la descripción de actividades colaborativas
FTP	File Transfer Protocol Protocolo de Transferencia de Archivos. Protocolo de red utilizado para transferir archivos entre sistemas conectados por TCP.
GDSS	Group Decision Support Systems
GDSS	Group Decision Support System
HTML	HyperText Markup Language Lenguaje de marcas de hipertexto. Usado para elaborar páginas web.
HTTP	Hypert Text Transfer Protocol Protocolo de Transferencia de Hipertexto. Permite la transferencia de información en la web.
IRC	Internet Relay Chat Protocolo de comunicación en tiempo real basado en texto
IS	Ingeniería de Software Ingeniería de Software
JSON	JavaScrip Object Notation Es un formato de texto ligero para el intercambio de datos
JWT	JSON Web Token Estándar abierto para la creación de tokens de seguridad en la autenticación
LCMS	Learning Content Management System Es un sistema de gestión de contenidos (CMS) que se utiliza para el aprendizaje.
LMS	Learning Management System Sistema para la gestión del aprendizaje
MDAC	Metodología para el Diseño de Actividades Colaborativas

	Propuesta en el Anexo II para la definición de las actividades de UAI Case.
MVC	Model View Controller Es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario
OCL	Object Constraint Language Lenguaje para la descripción formal de expresiones en los modelos UML.
OO	Orientado a Objetos Paradigma de análisis, diseño y desarrollo de software
REST	Representational State Transfer Estilo de arquitectura software para sistemas distribuidos
RFC	Request for Comments Serie de publicaciones del grupo de trabajo de ingeniería de internet que describen diversos aspectos del funcionamiento de Internet y otras redes de computadoras, como protocolos, etc.
RUP	Rational Unified Process Proceso Unificado de Rational propuesto por IBM para enmarcar a la IS
SOA	Service Oriented Architecture Es un marco de trabajo conceptual que establece una estructura de diseño para la integración de aplicaciones
SOC	Separation of Concerns Separación de Conceptos. Principio de diseño para separar un programa en distintas secciones
SPA	Single Page Application Aplicación de una Sola Página. Se enfoca en la comunicación asíncrona
UML	Unified Modeling Language Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software
URI	Uniform Resource Identifier Cadena de caracteres que identifica los recursos de una red de forma unívoca
XMI	XML Metadata Interchange Formato de intercambio de modelos UML

Referencias

- [ANG17] Angularjs.org. (2017). AngularJS — Superheroic JavaScript MVW Framework. [online] Available at: <https://angularjs.org/> [Accessed 1 Feb. 2017].
- [ARE09] Area, M., & Adell, J. (2009). E-learning: enseñar y aprender en espacios virtuales. En J. De Pablos (Coord.): Tecnología Educativa. La formación del profesorado en la era de Internet. Aljibe, Málaga, pags. 391-424.
- [ART05] Arteaga, J. M., Rodríguez, F. Á., González, L. A. G., & Delgado, F. J. P. (2005). Modelo para el aprendizaje colaborativo del análisis y diseño orientados a objetos. Apertura impresa.
- [BAI89] Bair, J. H. (1989). Supporting cooperative work with computers: Addressing meeting mania. In COMPCON Spring'89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers. (pp. 208-217). IEEE.
- [BAN88] Bannon, L., Bjørn-Andersen, N., & Due-Thomsen, B. (1988). Computer support for cooperative work: An appraisal and critique. *Eurinfo*, 88, 297-303.
- [BAN89] Bannon, L. J., & Schmidt, K. (1989). CSCW-four characters in search of a context. *DAIMI Report Series*, 18(289).
- [BAR04] Barros, B., Vélez, J., & Verdejo, F. (2004). Aplicaciones de la Teoría de la Actividad en el desarrollo de Sistemas Colaborativos de Enseñanza y Aprendizaje. *Experiencias y Resultados. Revista Iberoamericana de Inteligencia Artificial*, 24, 67-76.
- [BAR06] Barberá, E. (2006). Los fundamentos teóricos de la tutoría presencial y en línea: una perspectiva socio-constructivista. *Educación en red y tutoría en línea*, 161-180.
- [BAR06b] Barberá, E. (2006). Aportaciones de la tecnología a la e-Evaluación. *Revista de Educación a Distancia*.
- [BAR08] Barberá, E. (2008). *Aprender e-learning* (Vol. 25). Grupo Planeta (GBS).
- [BAT16] Battaglia, N., Neil, C., De Vincenzi, M., & Martínez, R. (2016). UAICase: integración de un entorno académico con una herramienta CASE en una plataforma virtual colaborativa. In XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016).
- [BAT16b] Battaglia, N., Neil C., Cardacci, D., De Vincenzi M., Martínez R. (2016). Evaluación y Seguimiento Durante el Proceso de Enseñanza y Aprendizaje del Modelado UML en Entornos Colaborativos. In V Workshop de Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).
- [BAT16c] Battaglia, N. Martínez, R. Otero, M., Neil, C., De Vincenzi M., (2016). Autoevaluación Colaborativa por medio de Rubricas en Entornos Virtuales de Enseñanza y Aprendizaje. In I Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), II Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC)
- [BAT17] Battaglia, N., Neil, C., De Vincenzi, M., Martínez, R., González, Dana. (2017). uCASE-CL: Aprendizaje Colaborativo de la Ingeniería de Software en Entornos Virtuales Ubicuos. In XII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2017).

- [BAT17b] Battaglia, N., Neil C., Martínez R., González, D., De Vincenzi M. (2017). Learning of Software Engineering on Collaborative Virtual Environments. In 7th World Engineering Education Forum (WEEF).
- [BAT17c] Battaglia, N., Martínez R., Neil C., De Vincenzi M. (2017). Una Propuesta de Evaluación de Herramientas CASE para la Enseñanza. In VI Workshop de Innovación en Educación en Informática (WIEI), Congreso Argentino de Ciencias de la Computación (CACIC).
- [BAT17d] Battaglia, N. Martínez, M., Neil, C., De Vincenzi M., (2017). UAI Case: Implementación del modelo uCASE-CL. In II Workshop sobre Innovación en Centros Educativos y de Investigación (I WICEI), III Jornadas Argentinas de Tecnología, Creatividad e Innovación (JATIC).
- [BAT17e] Battaglia, N., Martínez R. (2017, noviembre). Propuesta de Metodología de Evaluación para el Análisis de Modelado en Herramientas CASE. CoNaISI.
- [BAU06] Bauman, Zygmunt. (2006) Modernidad líquida. Buenos Aires: Fondo de Cultura Económica.
- [BIB09] Bibbó, L. M. (2009). Modelado de sistemas colaborativos (Master dissertation, Facultad de Informática UNLP).
- [BON07] Boneu, J. M. (2007). Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. RUSC. Universities and Knowledge Society Journal, 4(1).
- [BOO17] Bootstrap. (2017). Bootstrap · The world's most popular mobile-first and responsive front-end framework.. [online] Getbootstrap.com. Available at: <http://getbootstrap.com/> [Accessed 1 Feb. 2017].
- [BOR00] Borghoff, U. M., & Schlichter, J. H. (2000). Computer-Supported Cooperative Work - Introduction to Distributed Applications. Heidelberg: Springer Berlin / Heidelberg.
- [BOU05] Bouillon, P., Krinke, J., & Lukosch, S. (2005, April). Software engineering projects in distant teaching. In 18th Conference on Software Engineering Education & Training (CSEET'05) (pp. 147-154). IEEE.
- [BRI01] Briggs, R. O., De Vreede, G. J., Nunamaker, J. F., & Tobey, D. (2001, January). ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS). In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on* (pp. 9-pp). IEEE.
- [BRI03] Briggs, R. O., De Vreede, G. J., & Nunamaker Jr, J. F. (2003). Collaboration engineering with ThinkLets to pursue sustained success with group support systems. *Journal of Management Information Systems*, 19(4), 31-64.
- [BRI06] Briggs, R., Kolschoten, G., Gert-Jan, V., & Douglas, D. (2006). Defining key concepts for collaboration engineering. AMCIS 2006 Proceedings, 17.
- [CAB06] Cabero, J., & Róman, P. (2006). de las e-actividades. E-actividades: un referente básico para la formación en Internet.
- [CAB06b] Cabero-Almenara, J. (2006). Bases pedagógicas del e-learning. RUSC. Universities and knowledge society journal, 3(1), 1.
- [CAL02] Calzadilla, M. E. (2002). Aprendizaje colaborativo y tecnologías de la información y la comunicación. *Revista Iberoamericana de educación*, 1(10), 1-10.

- [CAM17] Campus Technology. (2017). Looking Beyond the LMS: Why a Single App Won't Work -- Campus Technology. [online] Available at: <https://campustechnology.com/articles/2017/01/23/looking-beyond-the-lms-why-a-single-app-wont-work.aspx?m=1> [Accessed 24 Jan. 2017].
- [CCR16] Coto, M., Collazos, C. A., & Rivera, S. M. (2016). Modelo Colaborativo y Ubicuo para apoyar los procesos de enseñanza-aprendizaje a nivel Iberoamericano. *Revista de Educación a Distancia*, (48)
- [CHE06] Chen, W., Pedersen, R. H., & Pettersen, Ø. (2006). CoLeMo: A collaborative learning environment for UML modelling. *Interactive Learning Environments*, 14(3), 233-249.
- [CHE76] Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1), 9-36.
- [CHO12] Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. *Economic Record*, 88(s1), 2-9.
- [COB16] Cobos Sanchiz, D., Gómez Galán, J., & López-Meneses, E. (2016). La Educación Superior en el Siglo XXI. Nuevas características profesionales y científicas.
- [COB16b] Cobo, Cristóbal (2016) La Innovación Pendiente. Reflexiones (y Provocaciones) sobre educación, tecnología y conocimiento. Colección Fundación Ceibal/ Debate: Montevideo. P. 83
- [COL13] Collazos, C., et al. (2013). "Propuesta metodológica de apoyo a los procesos de enseñanza-aprendizaje a través de entornos ubicuos y colaborativos: u-CSCL". Presentado en VIII Congreso Colombiano de Computación-8CCC.
- [COL14] Collazos, C. A. (2014). DISEÑANDO ACTIVIDADES DE APRENDIZAJE COLABORATIVO ASISTIDAS POR COMPUTADOR. *Revista educación en ingeniería*, 9(17), 143-149.
- [CON00] Constantino-González, M., & Suthers, D. D. (2000, June). A coached collaborative learning environment for entity-relationship modeling. In *International Conference on Intelligent Tutoring Systems* (pp. 324-333). Springer Berlin Heidelberg.
- [COO05] Cooper, K., Dong, J., Zhang, K., & Chung, L. (2005, October). Teaching experiences with UML at the University of Texas at Dallas. In *Proceedings of the Educators' Symposium of the ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, Montego Bay, Jamaica (pp. 1-8).
- [COR08] Corcuff, P. (2008). sociologías contemporáneas. *Recherche*, 2(No. 4), 9–41.
- [COZ14] Cózar Gutiérrez, R., & Roblizo Colmenero, M. J. (2014). La competencia digital en la formación de los futuros maestros: percepciones de los alumnos de los Grados de Maestro de la Facultad de Educación de Albacete. *RELATEC*.
- [DAN09] Dans, E. (2009). Educación online: plataformas educativas y el dilema de la apertura. *RUSC. Revista de Universidad y Sociedad del Conocimiento*, 6(1).
- [DAN10] Daniele, M., Uva, M., Martelloto, P., & Picco, G. (2010). Aplicación de herramientas CASE a la enseñanza de Ingeniería de Software: Gestión de

- la Configuración de Software y testing Funcional. In V Congreso de Tecnología en Educación y Educación en Tecnología.
- [DEL11a] Del Ben, E., & Vinjoy, M. (2011). Tecnología para la enseñanza de bases de datos: Un entorno de desarrollo enfocado a la enseñanza del diseño de bases de datos. In XIII Workshop de Investigadores en Ciencias de la Computación.
- [DEL11b] Del Dujo, Á. G., & Guerrero, C. S. (2011). Interacción virtual y aprendizaje cooperativo. Un estudio cualitativo Virtual Interaction and cooperative learning. A qualitative analysis. *Revista de educación*, 354, 473-498.
- [DEL96] Delors, J. (1996). *La Educación Encierra un Tesoro* (Libro).
- [DOR06] Dorrego, E. (2006). Educación a Distancia y Evaluación del Aprendizaje. UCV. RED. *Revista de Educación a Distancia*
- [DOU92] Dourish, P., & Bellotti, V. (1992, December). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work* (pp. 107-114). ACM.
- [ELI17] ELI, (2017). EDUCAUSE Learning Initiative. [online] Available at: <https://www.educause.edu/eli> [Accessed 24 Jan. 2017].
- [ELL91] Ellis, C. A., Gibbs, S. J., & Rein, G. (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1), 39-58.
- [ENG05] Engels, G., Hausmann, J. H., Lohmann, M., & Sauer, S. (2005, October). Teaching UML is teaching software engineering is teaching abstraction. In *Proceedings of the Educators' Symposium of the ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, Montego Bay, Jamaica (pp. 27-32).
- [FIL10] Filippi, J. L., Lafuente, G. J., & Bertone, R. A. (2010). Diseño de un ambiente de aprendizaje colaborativo. In V Congreso de Tecnología en Educación y Educación en Tecnología
- [FLI05] Flint, S., & Boughton, C. (2005, October). Three years experience teaching Executable/Translatable UML. In *Proceedings of the Educators' Symposium of the ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, Montego Bay, Jamaica (pp. 17-24).
- [FOW00] Fowler, L., Allen, M., Armarego, J., & Mackenzie, J. (2000). Learning styles and CASE tools in Software Engineering.
- [FOW17] P of EAA: Repository. (2017). *Martinfowler.com*. Retrieved 8 February 2017, from <https://martinfowler.com/eaCatalog/repository.html>
- [FUR01] Furco, A. (2001). Rúbrica de autoevaluación para la institucionalización del aprendizaje-servicio en la educación superior (revisión 2003). *Educación Global Research*, 77-88.
- [GAR06] García-Beltrán, Á., Martínez, R., Jaén, J. A., & Tapia, S. (2006). La autoevaluación como actividad docente en entornos virtuales de aprendizaje/enseñanza. *Revista de Educación a Distancia*.
- [GAR90] García Aretio, L. (1990). Objetivos y funciones de la educación a distancia. Publicado en *Actas del Congreso Internacional de Filosofía de la Educación*, UNED, Mari, pp. 44-48.
- [GAR99] Garcia Aretio, L. (1999). Historia de la educación a distancia. RIED. *Revista Iberoamericana de Educación a Distancia*, 2(1).

- [GEN06] Génova, G., Fuentes, J., & Valiente, M. (2006). Evaluación comparativa de herramientas CASE para UML desde el punto de vista notacional. *Novática*, 181, 59-64.
- [GIT17] Build software better, together. (2017). GitHub. Retrieved 9 February 2017, from <https://github.com/>
- [GLI17] Gliffy.com. (2017). Gliffy | Online Diagram and Flowchart Software. [online] Available at: <https://www.gliffy.com/> [Accessed 16 Jan. 2017].
- [GOD96] Goodnow, J. J. (1996). Acceptable ignorance, negotiable disagreement: Alternative views of learning. En D. R. Olson & N. Torrance (Eds.), *The handbook of education and human development*. Oxford: Blackwell Publishers.d
- [GOO17] Google Suite (2017). G Suite: Gmail, Drive, Documentos y mucho más. [online] Available at: <https://gsuite.google.com/> [Accessed 15 Jan. 2017].
- [GOO17b] Google Trends. (2017). Tendencias de Google. [online] Available at: <https://www.google.com.ar/trends> [Accessed 6 Jan. 2017].
- [GRA00] Graells, P. M. (2000). Las TIC y sus aportaciones a la sociedad. Departamento de pedagogía aplicada, facultad.
- [GRE87] Greif, I., & Sarin, S. (1987). Data sharing in group work. *ACM Transactions on Information Systems (TOIS)*, 5(2), 187-211.
- [GRU94] Grudin, J. (1994). Computer-supported cooperative work: history and focus. *Computer*, 27(5), 19-26.
- [GUE04] Guerrero, C. S. (2004). LA ZONA DE DESARROLLO PRÓXIMO, CATEGORÍA PEDAGÓGICA PARA EL ANÁLISIS DE LA INTERACCIÓN EN CONTEXTOS DE VIRTUALIDAD. *Pixel-Bit. Revista de Medios y Educación*, (24), 5-10.
- [GUE08] Guerra, L. (2008). Estrategias de aprendizaje colaborativo utilizando las Nuevas Tecnologías de Información y Comunicación. (Evaluación por grupos). *Docencia Universitaria*, 9(2), 11-34.
- [GUT02] Gutwin, C., & Greenberg, S. (2002). A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3-4), 411-446.
- [HAN17] Hangouts.google.com. (2017). Google Hangouts. [online] Available at: <https://hangouts.google.com/?hl=es> [Accessed 15 Jan. 2017].
- [HAS04] Hassan, Y., Martín Fernández, F. J., & Iazza, G. (2004). Diseño web centrado en el usuario: usabilidad y arquitectura de la información. *Hipertext. net*, (2).
- [HER13] Herrera, A., Rodríguez, D., & García Martínez, R. (2013). Taxonomía de mecanismos de awareness. In XVIII Congreso Argentino de Ciencias de la Computación.
- [HER16] Hernández, L. R. B., Peña, D. M., Valdés, O. R., & Cornelio, O. M. (2016). Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso Plugin of Visual Paradigm for UML tool for evaluation and correction of Use Case Diagram.
- [IBA04] Ibáñez, J. S. (2004). Innovación docente y uso de las TIC en la enseñanza universitaria. *RUSC. Universities and Knowledge Society Journal*, 1(1), 3.
- [IBM17] Ibm.com. (2017). Standards, compliance, and Rational Unified Process. [online] Available at: <http://www.ibm.com/developerworks/rational/library/4763.html> [Accessed 12 Jan. 2017].

- [IBM17a] IBM Bluemix - Infraestructura de cloud, servicios de plataforma, Watson y más soluciones PaaS. (2017). Ibm.com. Retrieved 8 February 2017, from <https://www.ibm.com/cloud-computing/bluemix/es>
- [IBM17b] SoftLayer | Cloud Servers, Storage, Big Data, & More IAAS Solutions. (2017). Softlayer.com. Retrieved 9 February 2017, from <http://www.softlayer.com/es>
- [ILL01] Illera, J. L. R. (2001). Aprendizaje colaborativo en entornos virtuales. *Anuario de psicología/The UB Journal of psychology*, 32(2), 63-76.
- [IMS17a] Imsglobal.org. (2017). IMS Global Learning Consortium |. [online] Available at: <https://www.imsglobal.org/> [Accessed 24 Jan. 2017].
- [IMS17b] Imsglobal.org. (2017). Learning Tools Interoperability | IMS Global Learning Consortium. [online] Available at: <https://www.imsglobal.org/activity/learning-tools-interoperability> [Accessed 24 Jan. 2017].
- [ISL08] Islas, O. (2008). La sociedad de la ubicuidad, los prosumidores y un modelo de comunicación para comprender la complejidad de las comunicaciones digitales. *Razón y palabra*, 65, 15.
- [JOH88] Johansen, R. (1988). *Groupware: Computer support for business teams*. The Free Press. New York and London, 1988
- [JOH99] Johnson, D. W., Johnson, R. T., & Holubec, E. J. (1999). *El aprendizaje cooperativo en el aula*. Buenos Aires: Paidós.
- [JOI17] Jointjs.com. (2017). Rappid Diagramming Framework. [online] Available at: <http://jointjs.com/> [Accessed 1 Feb. 2017].
- [JON99] Jonassen, D. H. (1999). Designing constructivist learning environments. *Instructional design theories and models: A new paradigm of instructional theory*, 2, 215-239.
- [JWT17] JWT.IO. (2017). Jwt.io, Available at <https://jwt.io/> [Accessed 8 Feb 2017]
- [KOF25] Koffka, K., & Ogden, R. M. (1925). *The growth of the mind*.
- [KOL04] Kolfschoten, G. L., Briggs, R. O., Appelman, J. H., & de Vreede, G. J. (2004, September). ThinkLets as building blocks for collaboration processes: a further conceptualization. In *International Conference on Collaboration and Technology* (pp. 137-152). Springer Berlin Heidelberg.
- [KOL06] Kolfschoten, G. L., Briggs, R. O., De Vreede, G. J., Jacobs, P. H., & Appelman, J. H. (2006). A conceptual foundation of the thinkLet concept for Collaboration Engineering. *International Journal of Human-Computer Studies*, 64(7), 611-621.
- [KOL07] Kolfschoten, G. L., & De Vreede, G. J. (2007, September). The collaboration engineering approach for designing collaboration processes. In *International Conference on Collaboration and Technology* (pp. 95-110). Springer Berlin Heidelberg.
- [KRU04] Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.
- [LAR99] Larman, C. (1999). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Ed. Pearson
- [LAV12] Lavigne, G., Vasconcelos Ovando, M. P., Organista Sandoval, J., & McAnally Salas, L. (2012). Exploración preliminar del aprendizaje colaborativo dentro un entorno virtual. *Revista Electrónica "Actualidades Investigativas en Educación"*, 12(3).

- [LAV91] Lave, J. W., & Wenger, E. E.(1991). Situated learning: Legitimate peripheral participation.
- [LIU08] Liu, Y., & Connelly, K. (2008, September). Realizing an open ubiquitous environment in a restful way. In *Web Services, 2008. ICWS'08. IEEE International Conference on* (pp. 96-103). IEEE.
- [LYN90] Lynch, K. J., Snyder, J. M., Vogel, D. R., & McHenry, W. K. (1990, January). The Arizona Analyst Information System: Supporting collaborative research on international technological trends. In *Proceedings of the IFIP WG 8.4 confernece on Multi-user interfaces and applications* (pp. 159-174). Elsevier North-Holland, Inc.
- [MAR04] Marrou, H. I. (2004). *Historia de la educación en la antigüedad* (Vol. 80). Ediciones AKAL.
- [MAR07] Martín, C. D. S., Franco, M. A., & Vinjoy, M. (2007). Tecnología para la enseñanza de bases de datos. In *IX Workshop de Investigadores en Ciencias de la Computación*.
- [MAR15] Martínez, T. S., & Torres, J. M. T. (2015). Posibilidades didácticas de las herramientas Moodle para producción de cursos y materiales educativos. *Digital Education Review*, (28), 59-76
- [MAS09] Masanet, M. I., Zavalla, E., & Fernández, A. (2009). Un enfoque integrado para las prácticas de laboratorio en la educación a distancia. In *XV Congreso Argentino de Ciencias de la Computación*.
- [MCC93] Mc Clure, C. O., & Ortega, J. M. (1993). Case: la automatización del software.
- [MEN01] Mendoza, L. E., Rojas, T., & Pérez, M. A. (2001). Organizational indicators for CASE tools selection: a case study. *Revista Colombiana de Computación*, 2(2), 41-60.
- [MET17] Meetingsphere.com. (2017). www.meetingsphere.com | Engage. Align. Achieve.. [online] Available at: <https://www.meetingsphere.com/> [Accessed 15 Jan. 2017].
- [MIR17] mIRC Ltd., m. (2017). mIRC: Download mIRC. [online] Mirc.com. Available at: <http://www.mirc.com/get.html> [Accessed 15 Jan. 2017].
- [MON16] Montejano, G. A., Testa, O., Pizarro, R. A., Segovia, D., Dieste, O., & Fonseca, E. R. (2016, May). Aplicación de principios SOA para la composición de servicios en ambientes ubicuos. In *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*.
- [MON17] MongoDB. Reinventando la gestión de datos. (2017). Retrieved 13 February 2017, from <https://www.mongodb.com/es>
- [MOO17] Moodle.org. (2017). Moodle - Open-source learning platform | Moodle.org. [online] Available at: <https://moodle.org/> [Accessed 18 Jan. 2017].
- [NEI16] Neil C., De Vincenzi M., Battaglia N., Martínez R. (2016). Herramientas Colaborativas Multiplataforma en la Enseñanza de la Ingeniería de Software. In *XVIII Workshop de Investigadores en Ciencias de la Computación*
- [NET17] .NET - Powerful Open Source Development. (2017). Microsoft.com. Retrieved 1 October 2017, from <https://www.microsoft.com/net/core>

- [NIC95] Nichols, D.A.; Curtis, P.; Dixon, M.; Lamping, J. (1995). "High-latency, low-bandwidth windowing in the Jupiter collaboration system". Proceedings of the 8th annual ACM symposium on User interface and software technology: 111–120.
- [OBJ17] Objectsbydesign.com. (2017). Objects by Design: Home. [online] Available at: <http://www.objectsbydesign.com> [Accessed 5 Jan. 2017].
- [OCH08] Ochoa, S., Collazos, C., Bravo, G., Neyem, A., Ormeño, E., & Guerrero, L. (2008, June). Una técnica de evaluación colaborativa soportada por computador para escenarios de educación superior. In IX Congreso Internacional Interacción (pp. 71-80).
- [OMG17] Omg.org. (2017). Object Management Group. [online] Available at: <http://www.omg.org/> [Accessed 4 Jan. 2017].
- [OXF97] Oxford, R. L. (1997). Cooperative learning, collaborative learning, and interaction: Three communicative strands in the language classroom. *The Modern Language Journal*, 81(4), 443-456.
- [PEÑ05] Peñalvo, F. J. G. (2005). Estado actual de los sistemas e-learning. *Education in the knowledge society (EKS)*, 6(2), 1.
- [PIA95] Piaget, J. (1995). *El Estructuralismo* (Vol. 34). Publicaciones Cruz O., SA.
- [PON90] De Pablos Pons, J. (2009) Coord. *Tecnología Educativa. La formación del profesorado en la era de Internet*. Málaga: Aljibe
- [PRE10] Pressman, R. S. (2010). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
- [QUI12] Quintero, J. B., de Páez, R. A., Marín, J. C., & López, A. B. (2012). Un estudio comparativo de herramientas para el modelado con UML. *revista universidad eafit*, 41(137), 60-76.
- [RAP11] Raposo, M., & Martínez, E. (2011). La rúbrica en la enseñanza universitaria: un recurso para la tutoría de grupos de estudiantes. *Formación universitaria*, 4(4), 1928.
- [REA17] realtimeboard.com. (2017). [online] Available at: <https://realtimeboard.com/> [Accessed 15 Jan. 2017]
- [ROC16] Rochera, M. J., Engel, A., & Coll, C. (2016) El efecto del feedback del profesor en foros de discusión en línea.
- [ROD12] Rodríguez, D., & García Martínez, R. (2012). Modelado de interacciones en espacios virtuales dedicados a trabajo colaborativo. In *XVIII Congreso Argentino de Ciencias de la Computación*.
- [ROJ00] Rojas, T., Pérez, M., Grimán, A., Ortega, M., & Diaz, A. (2000). Modelo de decisión para soportar la selección de herramientas CASE. *Revista de la Facultad de Ingeniería, UCV*, 15(2), 117-144.
- [ROM10] Romero, D., Molina, A., & Chirino, V. (2010). Aprendizaje móvil: tendencias, cuestiones y retos. *IEEE-Rita*. Vol. 5, 4, 123-124.
- [ROS08] Rosales-Gracia, S., Gómez-López, V. M., Durán-Rodríguez, S., Salinas-Fregoso, M., & Saldaña-Cedillo, S. (2008). Modalidad híbrida y presencial: Comparación de dos modalidades educativas. *Revista de la educación superior*, 37(148), 23-29.
- [ROT06] Rotstein, B., Scassa, A. M., Sáinz, C., & Simesen de Bielke, A. M. (2006). El trabajo colaborativo en entornos virtuales de aprendizaje. *Cognición*, 1(7), 38-45.
- [RUM05] Rumbaugh, J., Jacobson, I., & Booch, G. (2005). *Unified Modeling Language Reference Manual, Second Edition*, Addison-Wesley.

- [SAL04] SALINAS, Jesús (2004). "Innovación docente y uso de las TIC en la enseñanza universitaria". Revista de Universidad y Sociedad del Conocimiento (RUSC). UOC. Vol. 1, nº 1.
- [SAL15] Salas, M. C. M., Barrera, S. R., & Fuentes, M. G. L. (2015). La plataforma Moodle como herramienta de evaluación docente. Revista Iberoamericana para la Investigación y el Desarrollo Educativo ISSN: 2007-2619, (9)
- [SAL92] Salomon, G. (1992). Las diversas influencias de la tecnología en el desarrollo de la mente. *Infancia y aprendizaje*, 15(58), 143-159.
- [SCA06] Scardamalia, M., & Bereiter, C. (2006). Knowledge building. The Cambridge.
- [SIG17] signalr.net. (2017). Incredibly simple real-time web for .NET. [online] Available at: <http://signalr.net/> [Accessed 1 Feb. 2017].
- [SKY17] Skype. (2017). Skype.com. Available at: <https://www.skype.com/> [Accessed 15 Jan. 2017]
- [SLA99] Slavin, R. E., & Johnson, R. T. (1999). Aprendizaje cooperativo: teoría, investigación y práctica. Buenos Aires: Aique.
- [SOL10] Soler, J., Boada, I., Prados, F., Poch, J., & Fabregat, R. (2010). A web-based e-learning tool for UML class diagrams. *Education Engineering (EDUCON)*, 973-979.
- [SOL13] Solano, A., & Collazos, C. (2013). Modelo para el diseño de actividades colaborativas desde un enfoque práctico. *Revista Universitaria RUTIC*, 1(2).
- [SOM05] Sommerville, I., & Galipienso, M. I. A. (2005). Ingeniería del software. Pearson Educación.
- [SPA17] Sparxsystems.com.ar. (2017). Enterprise Architect - Herramienta de diseño UML y herramienta CASE UML para desarrollo de software. [online] Available at: <http://www.sparxsystems.com.ar/products/ea.html> [Accessed 16 Jan. 2017].
- [STA17] Staruml.io. (2017). StarUML. [online] Available at: <http://staruml.io/> [Accessed 17 Jan. 2017].
- [SUA03] Suárez Guerrero, C. (2003). Los Entornos Virtuales de Aprendizaje como Instrumento de Mediación. En Revista digital Enciclopedia Virtual de Tecnología Educativa. España: Universidad de Salamanca.
- [TEN08] Tenbergen, B., Buck, R., & Lazarro, L. (2008). Sketch UML: A Tablet PC-Based E-Learning Tool for UML Syntax Using a Minimalistic Interface.
- [TIG17] tigris.org. (2017). ArgoUML. [online] Available at: <http://argouml.tigris.org/> [Accessed 17 Jan. 2017].
- [TOP94] Topper, A., Ouellette, D., & Jorgensen, P. (1994). Structure Methods Merging Models. Techniques an CASE Mc Graw Hill.
- [TUD93] Tudge, J. (1993). Vygotsky, la zona de desarrollo próximo y la colaboración entre pares: connotaciones para la práctica del aula. L. Moll (Comp.), Vygotsky y la educación. Connotaciones y aplicaciones de la Psicología Sociohistórica en la educación, 187-207.
- [VAC97] Vacas, F. S. (1997). Innovación tecnológica y reingeniería de los procesos educativos. In La Tecnología Educativa a finales del siglo XX: concepciones, conexiones y límites con otras disciplinas. III Jornadas

- Universitarias de Tecnología Educativa: Barcelona, 21 y 22 de septiembre de 1995 (pp. 13-22). Departament de Didáctica i Organització Educativa.
- [VAR06] Vargas, E. (2006). La situación de enseñanza y aprendizaje como sistema de actividad: el alumno, el espacio de interacción y el profesor. *Revista Iberoamericana de Educación*, 39(4), 1-10.
- [VIG79] Vygotski, L. S., Cole, M., & Luriia, A. R. (1996). El desarrollo de los procesos psicológicos superiores (p. 66). Barcelona: Crítica.
- [VIG87] Vygotsky, L. S. (1987). Historia del desarrollo de las funciones psíquicas superiores. La Habana: Editorial Científico-Técnica, 456.
- [VIG97] Vygotsky, L. S. (1997). The collected works of LS Vygotsky: Problems of the theory and history of psychology (Vol. 3). Springer Science & Business Media.
- [VIS17] Visual-paradigm.com. (2017). Software Design Tools for Agile Teams, with UML, BPMN and More. [online] Available at: <https://www.visual-paradigm.com/> [Accessed 17 Jan. 2017].
- [VIT08] Vitturini, M., Benedetti, L., & Señas, P. (2008). Comunidades virtuales para la educación a distancia. In III Congreso de Tecnología en Educación y Educación en Tecnología.
- [VRE06] De Vreede, G. J., & Briggs, R. O. (2005, January). Collaboration engineering: designing repeatable processes for high-value collaborative tasks. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (pp. 17c-17c). IEEE.
- [WAT08] Watson, A. (2008). Visual Modelling: past, present and future. Online, Object Management Group, <http://www.uml.org>.
- [YAJ10] Yahya, S., Ahmad, E. A., & Jalil, K. A. (2010). The definition and characteristics of ubiquitous learning: A discussion. *International Journal of Education and Development using Information and Communication Technology*, 6(1), 1.
- [YAN15] Yandar, M. F., & Moreno, A. (2015). Definición de un proceso para apoyar ambientes de enseñanza-aprendizaje a través de entornos ubicuos y colaborativos: U-CSCL (Trabajo de Pregrado). Colombia, Popayán: Universidad del Cauca. Facultad de Ingeniería Electrónica y Telecomunicaciones. Departamento de Sistemas
- [YWO17] yWorks, t. (2017). yEd Graph Editor. [online] yWorks, the diagramming company. Available at: <https://www.yworks.com/products/yed> [Accessed 16 Jan. 2017].
- [ZAP07] ZAPATA, C. M., & ARANGO, F. (2007). Un ambiente para la obtención automática de diagramas UML a partir de un lenguaje controlado. *Dyna*, 74(153).